Гомотопическая теория типов:
Унивалентные основания математики

(международный коллектив авторов)

Перевод и редактирование:
Геннадий Чернышев
(https://henrychern.wordpress.com/)

Программа унивалентных оснований

Институт перспективных исследований

Book version: `first-edition-1287-g1ac9408`

MSC 2010 classification: `03-02, 55-02, 03B15`

Acknowledgment

Apart from the generous support from the Institute for Advanced Study, some contributors to the book were partially or fully supported by the following agencies and grants:

- Association of Members of the Institute for Advanced Study: a grant to the Institute for Advanced Study.
- Agencija za raziskovalno dejavnost Republike Slovenije: P1–0294, N1–0011.
- Air Force Office of Scientific Research: FA9550-11-1-0143, and FA9550-12-1-0370.

  This material is based in part upon work supported by the AFOSR under the above awards. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the AFOSR.
- Engineering and Physical Sciences Research Council: EP/G034109/1, EP/G03298X/1.
- European Union's 7th Framework Programme under grant agreement nr. 243847 (ForMath).
- National Science Foundation: DMS-1001191, DMS-1100938, CCF-1116703, and DMS-1128155.

  This material is based in part upon work supported by the National Science Foundation under the above awards. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
- The Simonyi Fund: a grant to the Institute for Advanced Study

Оглавление

Предисловие **1**

Введение **3**

I Основы **19**

1 Теория типов **21**
1.1 Теория типов и теория множеств . . . 21
1.2 Типы функций . . . 25
1.3 Универсумы и семейства . . . 28
1.4 Типы зависимых функций ($\prod$-типы) . . . 29
1.5 Типы произведений . . . 30
1.6 Типы зависимых пар ($\sum$-типы) . . . 34
1.7 Типы копроизведений . . . 38
1.8 Тип булевых значений . . . 39
1.9 Натуральные числа . . . 41
1.10 Сопоставление с образцом и рекурсия . . . 44
1.11 Высказывания как типы . . . 46
1.12 Типы тождественности . . . 52
1.12.1 Индукция пути . . . 53
1.12.2 Эквивалентность индукции и базированной индукции . . . 57
1.12.3 Неэквивалентности . . . 59
Примечания . . . 59
Упражнения . . . 61

2 Гомотопическая теория типов **63**
2.1 Типы — высшие группоиды . . . 66
2.2 Функции являются функторами . . . 75
2.3 Семейства типов являются расслоениями . . . 76
2.4 Гомотопии и эквивалентности . . . 80
2.5 Высшая группоидная структура конструкторов типов . . . 84
2.6 Типы декартова произведения . . . 85
2.7 $\sum$-типы . . . 88
2.8 Единичный тип . . . 90
2.9 $\prod$-типы и аксиома функциональной экстенсиональности . . . 91

2.10 Универсумы и аксиома унивалентности . . . 93
2.11 Тип тождественности . . . 95
2.12 Копроизведения . . . 97
2.13 Натуральные числа . . . 100
2.14 Пример: равенство структур . . . 101
2.14.1 Поднятие эквивалентностей . . . 102
2.14.2 Равенство полугрупп . . . 104
2.15 Универсальные свойства . . . 104
Примечания . . . 107
Упражнения . . . 109

3 Множества и логика **113**
3.1 Множества и n-типы . . . 113
3.2 Высказывания как типы? . . . 116
3.3 Простые высказывания . . . 117
3.4 Классическая и интуиционистская логики . . . 119
3.5 Подмножества и пропозициональное изменение размера . . . 121
3.6 Логика простых высказываний . . . 123
3.7 Пропозициональное усечение . . . 124
3.8 Аксиома выбора . . . 125
3.9 Принцип единственности выбора . . . 127
3.10 Когда высказывания усекаются? . . . 128
3.11 Стягиваемость . . . 131
Примечания . . . 133
Упражнения . . . 134

4 Эквивалентность **137**
4.1 Квазиобратные . . . 138
4.2 Полусопряженные эквивалентности . . . 140
4.3 Би-обратимые отображения . . . 144
4.4 Стягиваемые слои . . . 145
4.5 Об определении эквивалентностей . . . 146
4.6 Сюръекции и вложения . . . 146
4.7 Свойства замкнутости эквивалентностей . . . 148
4.8 Классификатор объектов . . . 150
4.9 Унивалентность — функциональная экстенсиональность . . . 152
Примечания . . . 155
Упражнения . . . 155

5 Индукция **157**
5.1 Введение в индуктивные типы . . . 157
5.2 Единственность индуктивных типов . . . 160
5.3 W-типы . . . 162
5.4 Индуктивные типы являются инициальными алгебрами . . . 165
5.5 Гомотопически-индуктивные типы . . . 168
5.6 Общий синтаксис индуктивных определений . . . 172
5.7 Обобщения индуктивных типов . . . 176

5.8 Типы тождественности и системы тождественности . . . 179
Примечания . . . 183
Упражнения . . . 184

6 Высшие индуктивные типы **187**
6.1 Введение . . . 187
6.2 Принципы индукции и зависимые пути . . . 189
6.3 Интервал . . . 193
6.4 Окружности и сферы . . . 195
6.5 Надстройки . . . 197
6.6 Клеточные комплексы . . . 200
6.7 Концентраторы и спицы . . . 202
6.8 Амальгамы . . . 203
6.9 Усечения . . . 207
6.10 Частные . . . 210
6.11 Алгебра . . . 215
6.12 Лемма сглаживания . . . 220
6.13 Общий синтаксис высших индуктивных определений . . . 226
Примечания . . . 227
Упражнения . . . 229

7 Гомотопические n-типы **231**
7.1 Определение n-типов . . . 231
7.2 Доказательства единственности тождественности . . . 235
7.3 Усечения . . . 238
7.4 Копределы n-типов . . . 244
7.5 Связность . . . 248
7.6 Ортогональная факторизация . . . 254
7.7 Модальности . . . 258
Примечания . . . 263
Упражнения . . . 263

II Математика **269**

8 Теория гомотопий **271**
8.1 Гомотопическая группа сфер . . . 275
8.1.1 Введение . . . 275
8.1.2 Классическое доказательство . . . 275
8.1.3 Универсальное покрытие в теории типов . . . 276
8.1.4 Доказательство кодированием-декодированием . . . 278
8.1.5 Теоретико-гомотопическое доказательство . . . 281
8.1.6 Универсальное покрытие как система тождественностей . . . 282
8.2 Связность надстроек . . . 283
8.3 $\pi_{k\leqslant n}$ n-связного пространства и $\pi_{k<n}(\mathbb{S}^n)$ . . . 285
8.4 Последовательности слоев . . . 285
8.5 Расслоение Хопфа . . . 290

8.5.1 Расслоения над амальгамами . . . 291
8.5.2 Конструкция Хопфа . . . 292
8.5.3 Расслоение Хопфа . . . 293
8.6 Теорема Фрейденталя о надстройке . . . 296
8.7 Теорема Ван Кампена . . . 302
8.7.1 «Наивная» теорема ван Кампена . . . 302
8.7.2 Теорема ван Кампена с множеством отмеченных точек . . . 307
8.8 Теорема и принцип Уайтхеда . . . 311
8.9 Метод кодирования-декодирования . . . 315
8.10 Дополнительные результаты . . . 316
Примечания . . . 317
Упражнения . . . 319

9 Теория категорий **321**
9.1 Категории и предкатегории . . . 322
9.2 Функторы и преобразования . . . 326
9.3 Сопряжения . . . 329
9.4 Эквивалентности . . . 330
9.5 Лемма Йонеды . . . 336
9.6 Строгие категории . . . 340
9.7 †-категории . . . 340
9.8 Принцип структурной тождественности . . . 342
9.9 Пополнение Резка . . . 345
Примечания . . . 351
Упражнения . . . 352

10 Теория множеств **355**
10.1 Категория множеств . . . 355
10.1.1 Пределы и копределы . . . 356
10.1.2 Образы . . . 356
10.1.3 Частные . . . 360
10.1.4 $\mathcal{S}$et — это **П**W-предтопос . . . 362
10.1.5 Аксиома выбора подразумевает исключение третьего . . . 363
10.2 Кардинальные числа . . . 364
10.3 Ординальные числа . . . 368
10.4 Классические порядки . . . 374
10.5 Кумулятивная иерархия . . . 377
Примечания . . . 383
Упражнения . . . 384

11 Действительные числа **387**
11.1 Поле рациональных чисел . . . 388
11.2 Действительные числа Дедекинда . . . 389
11.2.1 Алгебраическая структура действительных чисел Дедекинда . . . 390
11.2.2 Действительные числа Дедекинда являются полными в смысле Коши . . . 393
11.2.3 Действительные числа Дедекинда являются полными в смысле Дедекинда 394
11.3 Действительные числа Коши . . . 396

11.3.1 Построение действительных чисел Коши . . . 397
11.3.2 Индукция и рекурсия на действительных числах Коши . . . 399
11.3.3 Алгебраическая структура действительных чисел Коши . . . 410
11.3.4 Действительные числа Коши являются полными в смысле Коши . . . 414
11.4 Сравнение действительных чисел Коши и Дедекинда . . . 415
11.5 Компактность интервала . . . 416
11.6 Сюрреалистические числа . . . 423
Примечания . . . 435
Упражнения . . . 436

Приложение **440**

A Формальная теория типов **441**
A.1 Первая нотация . . . 443
A.1.1 Универсумы типов . . . 444
A.1.2 Зависимые функциональные типы (**Π**-типы) . . . 445
A.1.3 Зависимые типы пар (**Σ**-типы) . . . 445
A.1.4 Типы копроизведений . . . 446
A.1.5 Конечные типы . . . 446
A.1.6 Натуральные числа . . . 446
A.1.7 W-типы . . . 447
A.1.8 Типы тождественностей . . . 447
A.2 Вторая нотация . . . 447
A.2.1 Контексты . . . 448
A.2.2 Структурные правила . . . 448
A.2.3 Универсумы типов . . . 449
A.2.4 Зависимые функциональные типы (**Π**-типы) . . . 450
A.2.5 Зависимые типы пар (**Σ**-типы) . . . 450
A.2.6 Типы копроизведений . . . 451
A.2.7 Пустой тип **0** . . . 451
A.2.8 Единичный тип **1** . . . 452
A.2.9 Тип натуральных чисел . . . 452
A.2.10 Типы тождественностей . . . 452
A.2.11 Определения . . . 453
A.3 Гомотопическая теория типов . . . 453
A.3.1 Функциональная экстенсиональность и унивалентность . . . 453
A.3.2 Окружность . . . 454
A.4 Базовая метатеория . . . 454

Литература **458**

Список обозначений **467**

Предметный указатель **473**

Предисловие

Особый год IAS по унивалентным основаниям

Год унивалентных оснований математики был провозглашен в 2012-13 учебном году в Институте перспективных исследований, организованный Стивом Аводей, Тьерри Коквандом и Владимиром Воеводским. Официальными участниками были следующие люди.

Торстен Альтенкирх (Thorsten Altenkirch)
Бенедикт Аренс (Benedikt Ahrens)
Брюно Баррас (Bruno Barras)
Андрей Бауер (Andrej Bauer)
Марк Безем (Marc Bezem)
Ив Берто (Yves Bertot)
Бенно ван ден Берг (Benno van den Berg)
Владимир Воеводский
Дэниэл Грейсон (Daniel Grayson)
Андре Жуайяль (André Joyal)
Тьерри Кокан (Thierry Coquand)
Дэн Ликата (Dan Licata)
Питер Лумсдейн (Peter Lumsdaine)
Пер Мартин-Лёф (Per Martin-Löf)
Ассайя Махбуби (Assia Mahboubi)
Сергей Мелихов
Альваро Пелайо (Alvaro Pelayo)
Эндрю Полонский (Andrew Polonsky)
Матье Созо (Matthieu Sozeau)
Бас Спиттерс (Bas Spitters)
Майкл Уоррен (Michael Warren)
Эрик Финстер (Eric Finster)
Ноам Цейльбергер (Noam Zeilberger)
Майкл Шульман (Michael Shulman)
Стив Ауди (англ. Steve Awodey)
Питер Эксел (Peter Aczel)
Юго Эрбелен (Hugo Herbelin)

Также, отмечаем следующих студентов, чье участие было не менее ценным.

Карло Ангиули (Carlo Angiuli)
Энтони Бордж (Anthony Bordg)
Гийом Брюнери (Guillaume Brunerie)
Крис Капулкин (Chris Kapulkin)
Эгберт Рийке (Egbert Rijke)
Кристина Соякова (Kristina Sojakova)

В дополнение, благодарим следующих участников, в том числе студентов, за их эпизодическую, либо более продолжительную, помощь, чей вклад в этом году также имел важное значение.

Джереми Авигад (Jeremy Avigad)
Никола Гамбино (Nicola Gambino)
Ричард Гарнер (Richard Garner)
Джорж Гонтье (Georges Gonthier)
Питер Дибьер (Peter Dybjer)
Иоахим Кок (Joachim Kock)
Роберт Констебл (Robert Constable)
Кирил Коэн (Cyril Cohen)
Николай Краус (Nicolai Kraus)
Куэн-Ванг-Хоу (Kuen-Bang Hou)
Пьер-Луи Кюрьен (Pierre-Louis Curien)
Нуо Ли (Nuo Li)
Заохи Лу (Zhaohui Luo)
Майкл Нахас (Michael Nahas)
Эрик Палмгрен (Erik Palmgren)
Эмили Риль (Emily Riehl)
Дана Скотт (Dana Scott)
Филип Скотт (Philip Scott)
Сергей Соловьев (Sergei Soloviev)
Роберт Харпер (Robert Harper)
Томас Хейлз (Thomas Hales)
Мартин Хофманн (Martin Hofmann)
Питер Хофстра (Pieter Hofstra)
Мартин Эскардо (Martín Escardó)

Об этой книге

Мы не собирались писать книгу. Данная работа берет свое начало в коллективных попытках разработать новый стиль «неформальной теории типов», которая может быть рассмотрена и понята человеком, как дополнение к формальному доказательству, которое может быть проверено машиной. Унивалентные основы тесно связаны с идеей основ математики, которую можно реализовать в компьютерных помощниках по доказательствам. Хотя описание такой формализации не является частью этой книги, большая часть материала, представленного здесь, на самом деле была сделана сначала в полностью формализованной обстановке в помощниках по доказательствам, и только позже «неформализована», чтобы получить представление, которое вы видите перед собой — поразительная инверсия обычного положения дел в формализованной математике.

Каждый из вышеназванных личностей внес свой вклад в особом году — а значит, и в эту книгу — в виде идей, слов или дел. Дух сотрудничества, царивший в течение всего года, был поистине экстраординарным.

Особая благодарность Институту перспективных исследований, без которого эта книга, очевидно, никогда бы не появилась. Это оказалось идеальным местом для создания новой области математики: стимулирующей, благоприятной и поддерживающей. Пусть какой-то след этой уникальной атмосферы сохранится на страницах этой книги и в будущем развитии этой новой области исследований.

Программа унивалентных основ

Институт перспективных исследований

Принстон, апрель 2013 г.

Введение

Гомотопическая теория типов — это новая ветвь математики, которая удивительным образом сочетает в себе аспекты нескольких различных областей. Она основана на недавно открытой связи между теорией гомотопий и теорией типов. Гомотопическая теория — это продукт алгебраической топологии и гомологической алгебры, имеющая отношение к теории высших категорий, а теория типов — это раздел математической логики и теоретической информатики. Несмотря на то, что в настоящее время связи между этими двумя направлениями являются предметом интенсивных исследований, становится все более очевидным, что это всего лишь начало новой темы, которая потребует времени и значительных усилий для полного понимания. Она затрагивает вопросы, казалось бы, далекие от гомотопических групп сфер, алгоритмов проверки типов и определения слабых ∞-группоидов.

Гомотопическая теория типов также вносит новые идеи в сам фундамент математики. С одной стороны, существует тонкая и красивая *аксиома унивалентности* Воеводского. Аксиома унивалентности подразумевает, в частности, что изоморфные структуры могут быть идентифицированы, принцип, который математики успешно использовали в своей работе, несмотря на его несовместимость с «официальными» доктринами традиционных устоев. С другой стороны, у нас имеются индуктивные типы высших порядков, которые обеспечивают прямое, логическое описание некоторых основных пространств и конструкций гомотопической теории: сфер, цилиндров, усечений, локализаций и т.д. Обе идеи невозможно выразить непосредственно в классическом множестве, но в сочетании с теорией гомотопического типа они допускают совершенно новый тип «логики гомотопических типов».

Это предполагает новую концепцию основ математики с внутренним гомотопическим содержанием, «инвариантную» концепцию объектов математики и удобные машинные реализации, которые могут служить практической помощью работающему математику. Это программа *Унивалентных Оснований* (*Univalent Foundations*). Настоящая книга задумана как первое систематическое изложение начал унивалентных оснований (математики) и содержит примеры этого нового стиля рассуждений, не требуя от читателя знания или изучения какой-либо формальной логики или использования компьютерного инструментария для доказательств.

Мы подчеркиваем, что гомотопическая теория типов — это молодая область, а унивалентные основания — это очень большая работа, находящаяся в развитии. Эту книгу следует рассматривать как «моментальный снимок» состояния этой области в то время, когда она была написана, а не как отполированное изложение состоявшейся системы понятий. Как мы кратко обсудим далее, существует много аспектов гомотопической теории типов, которые еще не до конца поняты, но на момент написания этой работы ее контуры представляются достаточно ясными. Конечная теория, вероятно, не будет похожа на ту, что описана в этой книге, но она, несомненно, будет по меньшей мере столь же полезной и мощной; поэтому мы считаем, что унивалентные основания в конечном итоге станут жизнеспособной альтернативой теории множеств как «неявной основы» для неформализованной математики, созданной большой армией математиков.

Теория типов

Теория типов была изначально изобретена Бертраном Расселом [Rus08], как средство для устранения парадоксов в логических основах математики, которые исследовались в то время. Позднее она была развита как строгая формальная система (под названием «λ-исчисление») Алонзо Черчем [Chu33, Chu40, Chu41]. Хотя она обычно не рассматривается как основа классической математики, теория множеств является более привычной, теория типов до сих пор имеет множество приложений, особенно в области информатики и теории языков программирования [Pie02]. Мартин-Лёф [ML98, ML75, ML82, ML84], среди прочего, разработал «предикативную» модификацию системы типов Черча, которая теперь обычно называется зависимой, конструктивной, интуиционистской или просто теорией типа Мартина-Лёфа. Она является основой системы, которую мы рассматриваем здесь; изначально она была задумана как строгая основа для формализации конструктивной математики. В дальнейшем мы часто будем использовать «теорию типов», чтобы специально ссылаться на эту систему и подобные ей, хотя теория типов, как субъект, гораздо шире (см. [Som10, KLN04] по истории теории типов).

В теории типов, в отличие от теории множеств, объекты классифицируются с использованием примитивного понятия *типа*, подобного типам данных, используемым в языках программирования. Эти тщательно структурированные типы могут использоваться для выражения подробных спецификаций классифицируемых объектов, что приводит к принципам рассуждения об этих объектах. Взять очень простой пример, объекты типа произведения $A \times B$, как известно, имеют вид (a, b), каждый знает, как их построить и как их разложить. Аналогично, объект функции типа $A \to B$ может быть получен из объекта типа B, параметризованного объектами типа A, и может быть вычислен при аргументе типа A. Это жестко предсказуемое поведение всех объектов (в отличие от более либеральных принципов теории множеств, допускающих неоднородные множества) является одним из аспектов теории типов, который привел к его широкому использованию в проверке правильности компьютерных программ. Четкие принципы рассуждений, связанные с конструированием типов, также составляют основу современных компьютерных помощников — программ, которые используются для формализации математики и проверки правильности формализованных доказательств. Мы вернемся к этому аспекту теории типов ниже.

Однако, одной из проблем в понимании теории типов с математической точки зрения всегда было то, что базовое понятие *типа* отличается от *множества*, и его трудно было точно сформулировать. Мы считаем, что новая идея рассматривать типы не как странные множества (возможно, построенные без использования классической логики), а как пространства, рассматриваемые с точки зрения гомотопической теории, является важным шагом вперед. В частности, это решает проблему понимания того, как понятие равенства элементов типа отличается от понятия элементов множества.

В гомотопической теории речь идет о пространствах и непрерывных отображениях между ними, с точностью до гомотопии. *Гомотопия* между парой непрерывных отображений $f : X \to Y$ и $g : X \to Y$ — это непрерывное отображение $H : X \times [0, 1] \to Y$, для которого $H(x, 0) = f(x)$ и $H(x, 1) = g(x)$. Гомотопию H можно рассматривать как «непрерывную деформацию» f в g. Пространства X и Y называются *гомотопически эквивалентными*, $X \simeq Y$, если существуют взаимо обратные непрерывные отображения, композиции которых гомотопны соответствующим тождественным отображениям, т.е. если они изоморфны «с точностью до гомотопии». Гомотопически эквивалентные пространства имеют одни и те же алгебраические инварианты (например, гомологии или фундаментальную группу) и, как говорят, имеют один и тот же *гомотопический тип*.

Гомотопическая теория типов

Гомотопическая теория типов (HoTT) интерпретирует теорию типов с гомотопической точки зрения. В гомотопической теории типов мы рассматриваем типы как «пространства» (как изучается в теории гомотопий) или высшие группоиды, а логические конструкции (такие как произведение $A \times B$) — как гомотопически-инвариантные конструкции на этих пространствах. Таким образом, мы можем непосредственно манипулировать пространствами без предварительной разработки точечной топологии (или любой ее комбинаторной замены, например, теории симплициальных множеств). Чтобы кратко объяснить эту перспективу, рассмотрим сначала основное понятие теории типов, а именно то, что *терм* a имеет *тип* A, которое записывается в виде:

$$a : A$$

Это выражение традиционно понимается как:

«a есть элемент множества A»

Однако в гомотопической теории типов мы представляем себе это как:

«a есть точка пространства A»

Аналогично, каждая функция $f : A \to B$ в теории типов рассматривается как непрерывное отображение из пространства A в пространство B.

Следует подчеркнуть, что эти «пространства» трактуются чисто гомотопически, а не топологически. Например, не существует понятия «открытого подмножества» типа или «сходимости» последовательности элементов типа. У нас есть только «гомотопические» понятия, такие как пути между точками и гомотопии между путями, что также имеет смысл в других моделях гомотопической теории (таких, как симплициальные множества). Таким образом, было бы правильнее сказать, что мы рассматриваем типы как ∞-*группоиды*; это — название для «инвариантных объектов» гомотопической теории, которые могут быть представлены топологическими пространствами, симплициальными множествами или любой другой моделью для гомотопической теории. Однако, иногда бывает удобно использовать топологические термины, такие как «пространство» и «путь», при условии, что другие топологические названия неприменимы.

(Заманчиво также использовать для этих объектов словосочетание *гомотопический тип*, предлагая двойственную интерпретацию «типа (как в теории типов) рассматриваемого гомотопически» и «пространства, рассматриваемого с точки зрения гомотопической теории». Последнее немного отличается от классического смысла «гомотопического типа» как *класса эквивалентности* пространств по модулю гомотопической эквивалентности, хотя сохраняется смысл таких фраз, как «эти два пространства имеют один и тот же гомотопический тип»).

Идея интерпретировать типы как структурированные объекты, а не множества, имеет длинную родословную и, как известно, разъясняет различные таинственные аспекты теории типов. Например, интерпретация типов как пучков помогает объяснить интуиционистский характер теоретико-типологической логики, а интерпретация их как отношений частичной эквивалентности или «областей» помогает объяснить ее вычислительные аспекты. Это также подразумевает, что мы можем использовать теоретико-типологические рассуждения для изучения структурированных объектов, приводящие к богатому полю категорной логики. Гомотопическая интерпретация соответствует этой же схеме: она уточняет природу *тождества* (или равенства) в теории типов и позволяет нам использовать теоретико-типологические рассуждения при изучении гомотопической теории.

Ключевой новой идеей гомотопической интерпретации является то, что логическое понятие тождества $a = b$ двух объектов $a, b : A$ одного и того же типа A можно понимать как существование пути $p : a \rightsquigarrow b$ от точки a до точки b в пространстве A. Это также означает, что две функции $f, g : A \to B$ можно отождествить, если они гомотопны, поскольку гомотопия является просто (непрерывным) семейством путей $p_x : f(x) \rightsquigarrow g(x)$ в B, для любого $x : A$. В теории типов для каждого типа A существует (ранее несколько таинственный) тип Id_A идентифицирующий два одинаковые объекта A; в гомотопической теории типов это просто пространство путей A^I всех непрерывных отображений $I \to A$ из единичного интервала. Таким образом, терм $p : \mathsf{Id}_A(a, b)$ представляет собой путь $p : a \rightsquigarrow b$ в A.

Идея гомотопической теории типов возникла примерно в 2006 году в самостоятельной работе Аводей и Уоррена [AW09] и Воеводского [Voe06], но она была основана на более ранней группоидной интерпретации Хофмана и Стрейчера [HS98]. Действительно, теперь известно, что теория больше-размерных категорий (в частности, теория слабых ∞-группоидов) тесно связана с теорией гомотопий, как это предлагал Гротендик, и в настоящее время интенсивно изучается математиками обоих направлений. Исходные семантические модели Аводей-Уоррена и Воеводского используют хорошо известные понятия и приемы из теории гомотопий, которые в настоящее время также используются в теории высших категорий, такие как модельные категории Квиллена и симплициальные множества Кана.

В частности, Воеводский определил, что симплициальная интерпретация теории типов удовлетворяет еще одному важнейшему свойству, названному *унивалентностью*, которое ранее не рассматривалось в теории типов (хотя принцип экстенсиональности Черча для высказываний оказывается очень частным случаем этого). Добавление унивалентности к теории типов в форме новой аксиомы имеет далеко идущие последствия, многие из которых являются естественными, упрощающими и безусловными. Аксиома унивалентности также дополнительно усиливает гомотопический взгляд на теорию типов, так как она сохраняется в симплициальной модели и других связанных моделях, в то же время не опираясь на представление типов в виде множеств.

Унивалентные основания математики

Вкратце, основная идея аксиомы унивалентности может быть объяснена следующим образом. В теории типов можно иметь тип *универсума* $\mathcal{U}$, члены которого являются собственно типами, $A : \mathcal{U}$ и т.д. Те типы, которые являются членами $\mathcal{U}$, обычно называют *малыми* типами. Как и любой тип, $\mathcal{U}$ содержит тип тождества $\mathsf{Id}_\mathcal{U}$, который выражает тождество $A = B$ между малыми типами. Рассуждая о типах как о пространствах, $\mathcal{U}$ — это пространство, точки которого — пространства; чтобы понять тип тождественности, мы должны задаться вопросом, что представляет собой путь $p : A \rightsquigarrow B$ между пространствами в $\mathcal{U}$? В аксиоме унивалентности говорится, что такие пути соответствуют гомотопическим эквивалентностям $A \simeq B$ (грубо говоря), как объяснено выше. Несколько более точно, для любых (малых) типов A и B, в дополнение к примитивному типу $\mathsf{Id}_\mathcal{U}(A, B)$ отождествлений A с B, существует определенный тип $\mathsf{Equiv}(A, B)$ эквивалентностей от A к B. Так как тождественное отображение на любом объекте является эквивалентностью, то существует каноническое отображение

$$\mathsf{Id}_\mathcal{U}(A, B) \to \mathsf{Equiv}(A, B)\,.$$

Аксиома унивалентности утверждает, что это отображение само по себе является эквивалентностью. Рискуя слишком упростить, мы можем констатировать это кратко следующим образом:

Аксиома унивалентности: $(A = B) \simeq (A \simeq B)$.

Другими словами, тождественность равносильна эквивалентности. В частности, можно сказать, что «эквивалентные типы тождественны». Однако эта фраза несколько вводит в заблуждение, так как это может звучать как своего рода условие «скелетальности», которое *сводит* понятие эквивалентности к совпадению с тождеством, тогда как на самом деле унивалентность заключается в *расширении* понятия тождественности, чтобы соответствовать (неизменному) понятию эквивалентности.

С гомотопической точки зрения унивалентность подразумевает, что пространства одного и того же гомотопического типа соединены путем в универсуме $\mathcal{U}$ в соответствии с интуицией классифицирующего пространства для (малых) пространств. С логической точки зрения, однако, это радикально новая идея: она говорит, что изоморфные вещи могут быть идентифицированы! Математики, конечно, используют для идентификации изоморфные структуры на практике, но обычно они делают это путем «злоупотребления нотацией» или каким-либо другим неформальным приемом, зная, что задействованные объекты не «действительно» идентичны. Но в этой новой основополагающей схеме такие структуры могут быть формально идентифицированы, в логическом смысле, что каждое свойство или конструкция, включающее одно, также применяется к другому. Действительно, теперь идентификация становится явной, и свойства и конструкции могут систематически транспортироваться вдоль нее. Более того, различные способы, с помощью которых такие идентификации могут быть сделаны сами по себе, образуют структуру, которую можно (и следует!) принимать во внимание.

Таким образом, для точек A и B универсума $\mathcal{U}$ (т.е. малых типов) аксиома унивалентности отождествляет следующие три понятия:

- (логическая) идентификация $p : A = B$ для A и B,
- (топологический) путь $p : A \rightsquigarrow B$ от A к B в $\mathcal{U}$,
- (гомотопическая) эквивалентность $p : A \simeq B$ между A и B.

Высшие индуктивные типы

Одним из классических преимуществ теории типов являются простые и эффективные методы работы с индуктивно определенными структурами. Простейшей нетривиальной индуктивно определенной структурой являются натуральные числа, которые индуктивно генерируются нулем и функцией следования. Из этого утверждения можно алгоритмически извлечь принцип математической индукции, который характеризует натуральные числа. Более общие индуктивные определения включают списки и хорошо определенные деревья всех видов, каждый из которых характеризуется соответствующим «принципом индукции». Сюда включаются большинство структур данных, используемых в ряде языков программирования; отсюда и полезность теории типов в формальных рассуждениях о последних. Если понимать в самом общем смысле, индуктивные определения включают также такие примеры, как несвязное объединение $A + B$, которое можно рассматривать как «индуктивно» порожденное двумя инъекциями $A \to A + B$ и $B \to A + B$. «Принцип индукции» в данном случае — это «доказательство посредством анализа случая», которое характеризует несвязное объединение.

В теории гомотопий естественно рассматривать также «индуктивно определенные пространства», которые порождаются не просто набором точек, но также наборами путей и путей высших порядков. Классически, такие пространства названы *CW-комплексами*. Например, окружность S^1 порождается одной точкой и одним путем от этой точки к самой себе. Аналогично, 2-сфера S^2 порождается одной точкой b и единственным двумерным путем от постоянного пути при b

к себе, а тор T^2 порождается одной точкой, двумя путями p и q от этой точки к самой себе, и двумерным путем от $p \cdot q$ к $q \cdot p$.

Используя идентификацию путей с тождествами в гомотопической теории типов, такие «пространства с индуктивно определенными свойствами» можно охарактеризовать в теории типов «принципами индукции», полностью аналогичных классическим примерам, таким как натуральные числа и дизъюнктное объединение. Получающиеся в результате *высшие индуктивные типы* дают прямой «логический» способ рассуждения о знакомых пространствах, таких как сферы, которые (в сочетании с унивалентнотью) могут быть использованы для выполнения роли привычных аргументов из гомотопической теории, таких как вычисление гомотопических групп сфер, чисто формальным способом. Полученные доказательства являются союзом классических теоретико-гомотопических идей с классическими теоретико-типовыми, что дает новое понимание обеих областей.

Более того, это лишь верхушка айсберга: многие абстрактные конструкции из теории гомотопий, такие как гомотопические копределы, надстройки, башни Постникова, локальность, пополнение и спектрофикация, также могут быть выражены в виде высших индуктивных типов. Многие из них построены классически с использованием «аргумента малых объектов» Квиллена, который можно рассматривать как способ конечного алгоритмического описания бесконечного представления CW пространства, так же как «нуль и последующий» — это конечное алгоритмическое описание бесконечного множества натуральных чисел. Пространства, создаваемые аргументом малого объекта, являются извращенными и сложными для понимания; теоретико-типовой подход потенциально намного проще, не требующий какой-либо явной конструкции, предоставляя прямой доступ к соответствующему «принципу индукции». Таким образом, сочетание унивалентности и высших индуктивных типов наводит на мысль о возможности своего рода революции в практике гомотопической теории.

Множества в унивалентных основаниях

Мы заявляли, что унивалентные основания могут в конечном итоге служить фундаментом для «всей» математики, но до сих пор обсуждали только теорию гомотопий. Конечно, существует масса конкретных примеров использования теории типов без новых возможностей гомотопической теории типов для формализации математики, таких как недавняя формализация теоремы о нечетном порядке Фейта-Томпсона в Coq [GAA+13].

Но традиционная точка зрения заключается в том, что математика основана на теории множеств, в том смысле, что все математические объекты и конструкции могут быть закодированы в теории, такой как теория множеств Цермело-Френкеля (ZF). Тем не менее, к настоящему времени уже установлено, что для большинства математиков, не работающих в основах собственно теории множеств, сложная иерархическая структура членства множеств в ZF действительно не нужна: достаточно более «структурной» теории, такой как элементарная теория категории множеств (ETCS) Ловера [Law05].

В унивалентных основаниях базовыми объектами являются «гомотопические типы», а не множества, но мы можем определить класс типов, которые ведут себя как множества. Гомотопически они могут рассматриваться как пространства, в которых каждая связная компонента стягиваема, т.е. гомотопически эквивалентна дискретному пространству. Существует теорема о том, что категория таких «множеств» удовлетворяет аксиомам Ловера (или связана с ними, в зависимости от деталей теории). Таким образом, любая область математики, которая может быть представлена в теории, подобной ETCS (что, является, по существу, всей математикой), может одинаково хорошо быть представлена в унивалентных основаниях.

Это подтверждает то, что унивалентные основания, по меньшей мере, так же хороши, как и другие основания математики. Математик, работающий в унивалентных основаниях, может строить структуры из множеств знакомым способом, используя более общие гомотопические типы, ожидающими, пока они не понадобятся. По этой причине большинство приложений в этой книге выбраны для областей, где унивалентные основания могут внести что-то новое, что отличает их от существующих основополагающих систем.

Неудивительно, что гомотопическая теория и теория категорий — две из таких областей, но, возможно, менее очевидно, что унивалентные основания предлагают нечто новое и интересное даже в таких предметах, как теория множеств и вещественный анализ. Например, аксиома унивалентности позволяет идентифицировать изоморфные структуры, тогда как высшие индуктивные типы позволяют непосредственно описывать объекты по их универсальным свойствам. Таким образом, мы можем вообще избежать обращения к произвольно выбранным представителям или трансфинитным итерационным конструкциям. Фактически, даже объекты исследования в теории множеств ZF могут быть охарактеризованы внутри множеств унивалентных оснований таким индуктивным универсальным свойством.

Неформальная теория типов

Одна из трудностей, с которыми часто сталкивается классический математик, когда берется за изучение теории типов, заключается в том, что она обычно представляется как полностью или частично формализованная дедуктивная система. Этот стиль, который очень полезен для теоретико-доказательных исследований, не особенно удобен для использования в прикладных, неформальных рассуждениях. Он даже не знаком большинству работающих математиков, даже тех, кто может интересоваться основами математики. Одной из задач настоящей работы является разработка неформального стиля математической деятельности в унивалентных основаниях, который одновременно строг и точен, но также близок к языку и стилю представления повседневной математики.

В современной математике обычно строят и рассуждают о математических объектах таким образом, который в принципе, как можно предположить, формализуется в системе элементарной теории множеств, такой как ZFC, — по крайней мере, с достаточной изобретательностью и терпением. По большей части, даже не нужно осознавать эту возможность, поскольку она во многом совпадает с условием, что доказательство является «вполне точным» (в том смысле, что все математики поняли интуитивно через образование и опыт). Но нужно научиться осторожно относиться к нескольким аспектам «неформальной теории множеств»: использованию слишком больших или рудиментарных множеств, аксиоме выбора и ее эквивалентам, даже (для студентов) к методу доказательства от противного и т.д. Принятие новой основополагающей системы, такой как гомотопическая теория типов, как неявной формальной основы неформальных рассуждений, требует корректировки некоторых интуиций и практик. Настоящий текст призван служить примером такого «нового вида математики», который по-прежнему неформален, но теперь может быть в принципе формализован в гомотопической теории типов, а не в ZFC, опять же с достаточной изобретательностью и терпением.

Следует подчеркнуть, что в этой новой системе такая формализация может иметь реальные практические выгоды. Формальная система теории типов подходит для компьютерных систем и реализована в существующем инструментарии доказательств — это компьютерная программа, которая помогает пользователю строить полностью формальные доказательства, и допускающая только обоснованные шаги рассуждений. Этот инструмент также обеспечивает некоторую степень автоматизации, может искать библиотеки для существующих теорем и даже извлекать численные алгоритмы из полученных (конструктивных) доказательств.

Мы считаем, что этот аспект программы унивалентных оснований отличает его от других аналогичных подходов, потенциально предоставляя новую практическую полезность для работающего математика. Действительно, средства поддержки доказательств, основанные на теориях более старого типа, уже были использованы для формализации существенных математических доказательств, таких как теорема о четырех красках и теорема Фейта-Томпсона. В настоящее время выполняются компьютерные реализации унивалентных оснований (как и самой теории). Тем не менее, даже его доступные в настоящее время реализации (которые в основном являются небольшими модификациями существующих вспомогательных средств доказательства, таких как Coq и Agda), уже продемонстрировали свою ценность не только в формализации известных доказательств, но и в открытии новых. Действительно, многие из доказательств, описанных в этой книге, фактически были *впервые* выполнены в полностью формализованной форме с помощью таких средств. Это существенным образом изменяет традиционные отношения между формальной и неформальной математикой.

Можно представить себе не слишком отдаленное будущее, когда математики смогут проверять правильность собственных результатов, работая в рамках системы унивалентных оснований, формализованной в помощнике по доказательствам, и что это станет таким же естественным, как верстка своих документов в TeX. В принципе, это может быть одинаково справедливо для любой другой основополагающей системы, но мы считаем такую цель более практически достижимой с использованием унивалентных оснований, о чем свидетельствует настоящая работа и ее формальный аналог.

Конструктивность

Одним из самых ярких различий между классическими основами и теорией типов является идея *релевантности доказательства*, согласно которой математические высказывания и даже их доказательства становятся первоклассными математическими объектами. В теории типов мы представляем математические утверждения с использованием типов, которые могут рассматриваться одновременно, как математические построения, так и математические высказывания, — концепция, известная под названием *высказывания как типы*. Соответственно, мы можем рассматривать терм $a : A$ и как элемент типа A (или в теории гомотопического типа, точку пространства A), и как доказательство высказывания A. В качестве примера, предположим, что имеются множества A и B (дискретные пространства), и рассмотрим высказывание «A изоморфно B». В теории типов это можно представить как:

$$\mathsf{Iso}(A,B) :\equiv \sum_{(f:A\to B)} \sum_{(g:B\to A)} \left(\left(\prod_{x:A} g\left(f(x)\right) = x\right) \times \left(\prod_{y:B} f\left(g(y)\right) = y\right)\right).$$

Интерпретируя конструкторы типов $\sum$, $\prod$, $\times$ как «существует», «для всех» и «и» соответственно, получаем обычную формулировку «A и B изоморфны»; с другой стороны, интерпретация их как сумм и произведений дает *тип всех изоморфизмов* между A и B! Чтобы доказать, что A и B изоморфны, строится доказательство $p : \mathsf{Iso}(A,B)$, которое поэтому является тем же самым, что и построение изоморфизма между A и B, т.е. экспонирование пары функций f, g вместе с *доказательством*, что их сочетания являются соответствующими тождественными отображениями. Последнее доказательство, в свою очередь, не что иное, как гомотопии соответствующих видов. Таким образом, *доказательство высказывания такое же, как при построении элемента определенного типа*. В частности, для доказательства высказывания вида «A и B» нужно просто доказать A и доказать B, т.е. предоставить элемент типа $A \times B$. А чтобы доказать, что A подразумевает B, надо просто найти элемент из $A \to B$, то есть функцию от A к B (определяя отображение доказательства A на доказательство B).

Логика высказываний-как-типов является гибкой и поддерживает множество вариантов, таких как использование только подкласса типов для представления высказываний. В гомотопической теории типов существуют, естественно, такие подклассы, возникающие из факта, что система всех типов, таких как пространства в классической гомотопической теории, являются «расслоенными» в соответствии с размерностями, в которых их высшая гомотопическая структура существует или рушится. В частности, Воеводский нашел чисто теоретико-типовое определение *гомотопических n-типов*, соответствующих пространствам без нетривиальной гомотопической информации над размерностью n (0-типы — это «множества», упомянутые ранее как удовлетворяющие аксиомам Ловера). Более того, с высшими индуктивными типами можно универсально «усекать» тип в n-тип; в классической теории гомотопий это было бы n-ое сечение Постникова. Особенно важен для логики случай гомотопических (-1)-типов, которые мы называем *простыми высказываниями*. Классически, каждый (-1)-тип является пустым или стягиваемым; мы интерпретируем эти возможности как истинностные значения «ложь» и «истина», соответственно.

Использование всех типов в качестве высказываний дает весьма «конструктивную» концепцию логики; подробнее об этом см. [Kol32, TvD88a, TvD88b]. Например, каждое доказательство существования чего-либо несет в себе достаточную информацию, чтобы фактически найти такой объект, а из доказательства того, что «A или B» имеет место, можно извлечь либо доказательство того, что A имеет место, либо то, что B имеет место. Таким образом, из каждого доказательства мы можем автоматически извлечь алгоритм; это может быть очень полезно в приложениях к компьютерному программированию.

Однако, эта логика расходится с традиционным пониманием доказательств существования в математике. В частности, она не точно отражает некоторые важные классические принципы рассуждения, такие как аксиома выбора (AC) и закон исключенного третьего (LEM). Для этого нам нужно использовать «(-1)-усеченную» логику, в которой только гомотопические (-1)-типы представляют высказывания.

Конкретно, рассмотрим, с одной стороны, *аксиому выбора*: «если для каждого $x : A$ существует $y : B$ такое, что $R(x, y)$, то существует функция $f : A \to B$ такая, что для всех $x : A$ имеем $R(x, f(x))$». Чистое понятие высказывания-как-типы «существует» достаточно сильное, чтобы сделать утверждение просто доказуемым, но оно не имеет всех следствий обычной аксиомы выбора. Однако, в (-1)-усеченной логике это утверждение не является автоматически верным, но является сильным предположением с теми же последствиями, что и его аналог в классической теории множеств.

С другой стороны, рассмотрим *закон исключения третьего*: «для всех A, либо A, либо не A». Интерпретация этого в чистой логике высказываний-как-типов дает утверждение, несовместимое с аксиомой унивалентности. Поскольку доказательство «A» означает демонстрацию его элемента, это предположение дало бы единообразный способ выбора элемента из каждого непустого типа — своего рода гильбертова оператора выбора. Унивалентность означает, что элемент A, выбранный таким оператором выбора, должен быть инвариантным относительно всех само-эквивалентностей A, поскольку они отождествляются с само-тождественностью, и каждая операция не должна нарушать тождественность; но очевидно, что некоторые типы имеют автоморфизмы без неподвижных точек, например можно поменять местами элементы двухэлементного типа. Однако «(-1)-усеченный закон исключенного третьего», хотя и не выполняется автоматически, может последовательно предполагаться с большинством тех же последствий, что и в классической математике.

Другими словами, в то время как чистая логика предложений-как-типов является «конструктивной» в строгом алгоритмическом смысле, упомянутом выше, по умолчанию (-1)-усеченная логика является «конструктивной» в другом смысле (а именно, логика, формализованная Гейтингом под названием «интуиционистская»); и к последней мы можем свободно добавлять аксиомы выбора и исключения третьего, чтобы получить логику, которую можно назвать «классической». Таким образом, гомотопическая теория типов совместима, как с конструктивными, так и с классическими концепциями логики, и многими другими. В самом деле, гомотопическая перспектива показывает, что классическая и конструктивная логики могут сосуществовать как конечные точки спектра различных систем с бесконечным количеством промежуточных возможностей (гомотопические n-типы для $-1 < n < \infty$). Мы можем говорить о «LEM_n» и «AC_n», при этом AC_∞ доказуема, а LEM_∞ несовместим с унивалентностью, тогда как AC_{-1} и LEM_{-1} — версии, известные классическим математикам (поэтому в большинстве случаев уместно предполагать (-1), если никакой индекс не указан). В самом деле, могут быть полезными даже системы, в которых только *определенные* типы удовлетворяют таким дополнительным «классическим» принципам, а типы в целом остаются «конструктивными».

Следует подчеркнуть, что унивалентные основания *не требуют* использования конструктивной или интуиционистской логик. Большая часть классической математики, которая зависит от закона исключенного третьего и аксиомы выбора, может быть реализована на унивалентных основаниях, просто предполагая, что эти два принципа выполняются (в их собственном, (-1)-усеченном, виде). Однако, теория типов рекомендует избегать этих принципов, когда они не нужны, по нескольким причинам.

Во-первых, каждый математик знает, что теорема является более мощной, если доказанное использует меньшее количество допущений, поскольку тогда оно относится к большему числу примеров. Ситуация с AC и LEM ничем не отличается: теория типов допускает множество интересных «нестандартных» моделей, например, в пучках топосов, где класические принципы, такие как AC и LEM, как правило, терпят неудачу. Гомотопическая теория типов допускает сходные модели в высших топосах, например, изучаемые в [TV02, Rez05, Lur09]. Таким образом, если мы будем избегать использования этих принципов, то доказанные теоремы будут справедливы внутри всех таких моделей.

Во-вторых, одним из дополнительных достоинств теории типов является ее вычислительный характер. Помимо того, что она входит в основания математики, теория типов является формальной теорией вычислений и может рассматриваться как мощный язык программирования. С этой точки зрения, правила системы не могут быть выбраны произвольно так, как это допускают теоретико-множественные аксиомы: должна быть гармония между ними, которая позволяла бы всем доказательствам «исполняться» как программы. С этой точки зрения мы еще не полностью понимаем новые принципы, введенные гомотопической теорий типов, такие как унивалентность и высшие индуктивные типы, но уже появляются основные контуры (см., например, [LH12]). Однако, уже давно известно, что такие принципы, как AC и LEM, принципиально противоположны вычислимости, поскольку они утверждают, что существуют определенные сущности, не давая при этом возможности вычислить их. Таким образом, избегая их, необходимо поддерживать образ теории типов как теории вычислений.

К счастью, конструктивное рассуждение не так сложно, как может показаться. В некоторых случаях, просто перефразируя некоторые определения, теорему можно сделать конструктивной, а ее доказательство — более элегантным. Более того, в унивалентных основаниях это, по-видимому, происходит чаще. Например:

1. В теоретико-множественных основаниях, в различных областях теории гомотопий и теории категорий требуется аксиома выбора для выполнения трансфинитных конструкций. Но с высшими индуктивными типами мы можем кодировать эти конструкции непосредственно и конструктивно. В частности, ни одна из «синтетических» гомотопных теорий в главе 8 не требует LEM или AC.

2. В теоретико-множественных основаниях высказывание «каждый вполне точный и, по существу, сюръективный функтор является эквивалентностью категорий» равнозначен аксиоме выбора. Но с аксиомой унивалентности это непосредственно *истинно*; см. главу 9.

3. В теории множеств для получения понятий «кардинальное число» и «порядковый номер» требуются различные перефразирования, которые канонически представляют классы изоморфизмов множеств и упорядоченных множеств соответственно, возможно, с использованием аксиомы выбора или аксиомы основания. Но с унивалентностью и высшими индуктивными типами мы можем получить таких представителей непосредственно, сокращая универсум рассмотрения; см. главу 10.

4. В теоретико-множественных основаниях для определения действительных чисел как классов эквивалентности последовательностей Коши требуется либо закон исключения третьего, либо аксиома (счетного) выбора, для корректности. Но с высшими индуктивными типами мы можем дать вариант этого определения, который корректен и не требует каких-либо принципов выбора; см. главу 11.

Разумеется, эти упрощения можно было бы также принять за доказательство того, что новые методы в конечном итоге не окажутся действительно конструктивными. Однако мы вновь подчеркиваем, что читателю не нужно заботиться или волноваться о конструктивности, чтобы читать эту книгу. Дело в том, что во всех приведенных примерах версия теории, которую мы приводим, имеет самостоятельные преимущества, независимо от того, считаются ли LEM и AC достижимыми. Конструктивность, если она будет достигнута, будет дополнительным преимуществом.

Основываясь на обсуждении добавления новых принципов, таких как унивалентность, высшие индуктивные типы, AC и LEM, можно задаться вопросом, остается ли результирующая система последовательной (одно из первоначальных достоинств теории типов, по отношению к теории множеств, состояло в том, что, как можно видеть, она согласуется с теоретико-теоретическими средствами). Как и в случае с любой основополагающей системой, согласованность — это относительный вопрос: «согласованно по отношению к чему?». Короткий ответ заключается в том, что все конструкции и аксиомы, рассмотренные в этой книге, имеют модель в категории комплексов Кана благодаря Воеводскому [KLV12] (см. [LS17] для высших индуктивных типов). Таким образом, они известны как последовательные относительно системы ZFC (с таким количеством недостижимых кардиналов, сколько нам требуется вложенных унивалентных универсумов). Более традиционный теоретико-типовый подсчет этой согласованности находится в стадии разработки (см., например, [LH12, BCH13]).

Мы суммируем разные точки зрения на теоретико-типовые операции в таблице 1.

Типы	Логика	Множества	Гомотопия
A	высказывание	множество	пространство
$a : A$	доказательство	элемент	точка
$B(x)$	предикат	семейство множеств	расслоение
$b(x) : B(x)$	условие доказательство	семейство элементов	сечение
$\mathbf{0}$, $\mathbf{1}$	$\bot$, $\top$	$\varnothing$, $\{\varnothing\}$	$\varnothing$, $*$
$A + B$	$A \vee B$	несвязное объединение	копроизведение
$A \times B$	$A \wedge B$	множество пар	пространство произведений
$A \to B$	$A \Rightarrow B$	множество функций	функциональное пространство
$\sum_{(x:A)} B(x)$	$\exists_{x:A} B(x)$	несвязанная сумма	пространство расслоений
$\prod_{(x:A)} B(x)$	$\forall_{x:A} B(x)$	произведение	пространство сечений
Id_A	равенство $=$	$\{(x,x) \mid x \in A\}$	пространство путей A^I

Таблица 1. Сравнение терминологии по теоретико-типовым обозначениям

Открытые проблемы

Для тех, кто заинтересован в том, чтобы внести свой вклад в эту новую область математики, имеются обнадеживающие известия о наличии множества интересных открытых вопросов.

Пожалуй, наиболее насущным из них касается «конструктивности» аксиомы унивалентности, поставленный Воеводским в [Voe12]. Основная система теории типов следует структуре естественного вывода Генцена. Логические связки определяются их правилами введения и имеют правила исключения, оправдываемые правилами вычисления. Следуя этой схеме и используя метод вычислимости Тейта, первоначально разработанный для анализа интерпретации Диалектики Гёделя, можно показать свойство *нормализации* для теории типов. Это, в свою очередь, подразумевает важные свойства, такие как разрешимость проверки типов (важнейшее свойство, поскольку проверка типов соответствует проверке доказательств, и можно утверждать, что мы должны быть способны «распознавать доказательство, когда мы его видим»), а также так называемое «свойство каноничности», заключающееся в том, что любой замкнутый член типа натуральных чисел приводится к числу. Это последнее свойство и единообразная структура правил введения/исключения теряются, когда расширяется теория типов с аксиомой, например, аксиомой расширения функций или аксиомой унивалентности. Воеводский сформулировал точную математическую гипотезу, связанную с этим вопросом о каноничности для теории типов, расширенной аксиомой унивалентности: если задан замкнутый терм типа натуральных чисел, всегда можно найти цифру и доказать, что этот терм равен этой цифре, где это доказательство равенства может само использовать аксиому унивалентности? В более общем плане важной проблемой является возможность конструктивного обоснования аксиомы унивалентности. А если добавить другие гомотопически мотивированные конструкции, например, высшие индуктивные типы? Эти вопросы остаются открытыми в настоящее время, хотя разрабатываются методы в надежде найти ответы.

Другой основной проблемой является трудность работы с типами, такими как натуральные числа, которые по существу являются множествами (т.е. дискретными пространствами), содержащими только тривиальные пути. В настоящее время гомотопическая теория типов действительно может характеризовать пространства только с точностью до гомотопической эквивалентности, а это означает, что эти «дискретные пространства» могут быть только *гомо-*

топически эквивалентны дискретным пространствам. С теоретико-типовой точки зрения это означает, что существует много путей, которые равносильны рефлексивности, но не являются *суждениями*, эквивалентными им (см. §1.1 для значения «субъективно»). Хотя эта гомотопическая инвариантность имеет свои преимущества, эти «бессмысленные» тождества, связывающие тождественности, вносят ненужные усложнения в аргументы и конструкции, поэтому было бы удобно иметь систематический способ их устранения или свертывания.

Более специализированная, но не менее важная проблема — это связь между гомотопической теории типов и исследованием *высших топосов*, которая в настоящее время попадает на пересечение теории высших категорий и теории гомотопий. Среди тех, кто знаком с обеими областями, растет уверенность в том, что они тесно связаны. Например, понятие унивалентного универсума должно совпадать с понятием классификатора объектов, в то время как высшие индуктивные типы должны быть «элементарным» отражением локальной презентабельности. В более общем плане теория гомотопического типа должна быть «внутренним языком» $(\infty, 1)$-топосов, точно так же, как интуиционистская логика высшего порядка является внутренним языком обычных 1-топосов. Тем не менее, несмотря на это общее согласие, детали еще предстоит проработать, в частности, остаются нерешенными вопросы согласованности и строгости — и это, несомненно, приведет к дальнейшему пониманию обеих концепций.

Но самое большое поле для работы — это постоянная формализация повседневной математики в этой новой системе. Недавние успехи в формализации некоторых фактов из базовой теории гомотопий и теории категорий были обнадеживающими; некоторые из них описаны в главах 8 и 9. Очевидно, однако, многое еще предстоит сделать.

Сообщество гомотопической теории типов поддерживает веб-сайт и групповой блог по адресу http://www.homotopytypetheory.org, а также список рассылки для обсуждения. Новичкам всегда рады!

Как читать эту книгу

Эта книга состоит из двух частей. Часть I, «Основы», развивает фундаментальные понятия гомотопической теории типов. Это математическая основа, на которой строится разработка конкретных тематик, и которая необходима для понимания подхода, основанного на использовании унивалентных оснований. Для программиста это «библиотечный код». Поскольку унивалентные основания — это новая и разноплановая математика, ее основные понятия требуют некоторого усвоения; таким образом, Часть I довольно обширна.

Часть II, «Математика», состоит из четырех глав, которые основываются на основных понятиях Части I, чтобы раскрыть некоторые из новых возможностей, которые мы можем реализовать с унивалентными основаниями, в четырех различных областях математики: гомотопической теории (глава 8), теории категорий (глава 9), теории множеств (глава 10) и вещественном анализе (глава 11). Главы в Части II более или менее независимы друг от друга, хотя изредка может использоваться лемма, доказанная в другой главе.

Читатель, который хочет серьезно разобраться в унивалентных основаниях и иметь возможность работать в этой среде, должен будет в конце концов прочитать и понять почти весь материал Части I. Однако читателю, которому просто хочется почувствовать стиль унивалентных оснований, будет тяжело работать над более чем 200 страницами, прежде чем перейти к «мясу» в Части II. К счастью, не все из первой части необходимо для того, чтобы читать главы в Части II. Каждая глава в Части II начинается с краткого обзора предмета, того, что унивалентные основания должны внести в него, и необходимого фона из Части I, чтобы мужественный читатель мог немедленно обратиться к соответствующей главе. Для тех, кто хочет понять одну или несколько глав в Части II более глубоко, но не готовы читать всю часть I, мы предлагаем здесь краткое резюме Части I с замечаниями о том, какие ее главы необходимы для того, чтобы увереннее себя чувствовать в главах Части II.

Глава 1 посвящена основным понятиям теории типов, предшествующей любой гомотопической интерпретации. Читатель, знакомый с теорией типов Мартина-Лёфа, может бегло просмотреть ее, чтобы понять, какие сведения из нее мы используем. Тем не менее, читателям, не имеющим опыта работы с теорией типов, необходимо будет прочитать главу 1, поскольку существует много тонких различий между теорией типов и другими основаниями, такими как теория множеств.

Глава 2 знакомит с гомотопической точкой зрения на теорию типов, а также вводит базовые понятия, подтверждающие эту точку зрения, и описывает гомотопическое поведение каждого составного элемента теории типов из главы 1. Она также вводит аксиому унивалентности (§ 2.10) — первое из двух основных нововведений гомотопической теории типов. Таким образом, эта глава достаточно проста и мы рекомендуем всем ее прочитать, особенно §§ 2.1–2.4.

Глава 3 описывает, как мы представляем логику в гомотопической теории типов, ее связь с классической логикой, а также с конструктивной и интуиционистской логикой. Здесь мы определяем закон исключения третьего, аксиому выбора и аксиому пропозиционального изменения размера (хотя, по большей части, нам не нужно принимать ни одно из них для остальной части книги), а также пропозициональное усечение, которое необходимо для представления традиционной логики. Эта глава является существенным справочным материалом для глав 10 и 11, менее важной для главы 9 и не столь необходимой для главы 8.

В главах 4 и 5 подробно рассматриваются две специальные темы: эквивалентности (и связанные с ними понятия) и обобщенные индуктивные определения. Хотя они являются важными сами по себе и обеспечивают более глубокое понимание теории гомотопического типа, по большей части они не являются необходимыми для Части II. Лишь несколько лемм из главы 4 используются здесь и там, в то время как общие обсуждения в §§ 5.1, 5.6 и 5.7 полезны для обеспечения интуиции, требуемой для главы 6. Также используются в нескольких местах в главах 10 и 11 обобщенные виды индуктивного определения, обсуждаемые в § 5.7.

Глава 6 вводит второе базовое новшество гомотопической теории типов — высшие индуктивные типы, со многими примерами. Высшие индуктивные типы являются первичным объектом изучения в главе 8, а некоторые из них играют важную роль в главах 10 и 11. Они не так необходимы для главы 9, хотя один пример используется в § 9.9.

Наконец, в главе 7 обсуждаются гомотопические n-типы и связанные с ними понятия n-связанных типов. Они важны для главы 8, но не особо востребованы в остальных местах Части II, хотя случай $n = 1$ некоторых лемм используется в § 10.1.

На этом завершается Часть I. Как уже упоминалось, Часть II состоит из четырех, в основном, не связанных между собой глав, в каждой из которых описываются то, что унивалентные основания предлагают по конкретной теме.

В Части II глава 8 (теория гомотопий) является, пожалуй, самой радикальной. Унивалентные основания имеют совершенно особенный «синтетический» подход к гомотопической теории, в которой гомотопические типы являются основными объектами (а именно, типами), а не конструируются с использованием топологических пространств или какой-либо другой теоретико-множественной модели. Это порождает новые стили доказательства для классических теорем в алгебраической топологии, из которых мы представляем выборку от $\prod_1(S^1) = \mathbb{Z}$ до теоремы о приостановке Фройденталя.

В главе 9 (теория категорий) мы развиваем некоторую базовую теорию (1)-категорий, придерживаясь принципа аксиомы унивалентности, что равенство есть изоморфизм. Это имеет приятный эффект, гарантирующий, что все определения и конструкции автоматически инварианты относительно эквивалентности категорий: действительно, эквивалентные категории равны, равно как и эквивалентные типы равны (что также связано с теорией высшиих категорий и теорией высших топосов).

Глава 10 (теория множеств) изучает множество в унивалентных основаниях. Категория множеств имеет обычные свойства, следовательно, является основой любой математики, которая не нуждается в гомотопической структуре или структуре высших категорий. Мы также замечаем, что унивалентность делает кардинальные и порядковые числа немного более привлекательными, а высшие индуктивные типы производят кумулятивную иерархию, удовлетворяющую обычным аксиомам теории множеств Цермело-Френкеля.

В главе 11 (вещественные числа) мы резюмируем построение вещественных чисел Дедекинда, а затем замечаем, что высшие индуктивные типы позволяют определить вещественные числа Коши, что позволяет избежать некоторых связанных с этим проблем в конструктивной математике. Затем мы обозначим аналогичный подход к сюрреальным числам Конвэйя.

Каждая глава этой книги заканчивается разделом «Примечания», в котором, насколько это возможно, собраны исторические комментарии, ссылки на литературу и атрибуты результатов. Мы также включили упражнения в конце каждой главы, чтобы помочь читателю ближе познакомиться с математикой в унивалентных основаниях.

Наконец, напомним, что эта книга была написана в результате совместных усилий большого числа людей. Мы сделали все возможное, чтобы добиться согласованности в терминологии и нотации и выстроить математику в логически-линейной последовательности, но весьма вероятно, что остались некоторые недостатки. Мы просим у читателя прощения за любые подобные ошибки и приветствуем предложения по улучшению следующей редакции книги.

Часть I

Основы

Глава 1

Теория типов

1.1 Теория типов и теория множеств

Гомотопическая теория типов является (помимо прочего) основополагающим языком математики, т.е. альтернативой теории множеств Цермело-Френкеля. Однако, в отличии от теории множеств, она ведет себя несколько иначе в использовании ряда важных подходов, к чему необходимо привыкнуть. Тщательное объяснение этих различий требует от нас быть здесь более формальными, чем в остальной части книги. Как сказано во введении, наша цель — *неформально* описать теорию типов, но для математика, привыкшего к теории множеств, скрупулезная точность в начале может помочь избежать некоторых распространенных заблуждений и ошибок.

Заметим, что теоретико-множественная основа имеет два «уровня»: дедуктивную систему логики первого порядка и сформулированные внутри этой системы аксиомы конкретной теории, такой как ZFC. Таким образом, теория множеств связана не только с множествами, но и с взаимодействием между множествами (объекты второго уровня) и высказываниями (объекты первого уровня).

Напротив, теория типов — это самостоятельная дедуктивная система, ее не нужно формулировать внутри какой-либо надстройки, такой как логика первого порядка. Вместо двух основных понятий теории множеств, множеств и высказываний, теория типов имеет одно основное понятие — *типы*. Высказывания (утверждения, которые мы можем доказать, опровергнуть, предположить, отрицать и т.д.[1]) отождествляются с конкретными типами посредством соответствия, представленного в таблице 1. Таким образом, математическая деятельность по *доказательству теоремы* отождествляется со специальным случаем *конструирования объекта* — в данном случае — представителя типа, который представляет собой высказывание.

Это приводит нас к другому различию между теорией типов и теорией множеств, для объяснения которого мы должны немного рассказать о дедуктивных системах в целом. Неформально, дедуктивная система представляет собой набор **правил** для вывода сущностей, называемых **суждениями**. Если мы думаем о дедуктивной системе как о формальной игре, то суждения – это «позиции» в игре, которые мы достигаем, следуя правилам игры. Мы также можем представить себе дедуктивную систему как своего рода алгебраическую теорию, и в этом случае

[1]Приводит в замешательство также то, что обычная практика (восходящая к Евклиду) заключается в том, чтобы использовать слово «высказывание» синонимично с «теоремой». Мы ограничимся использованием логики, согласно которой *высказывание* является выражением, *восприимчивым* к доказательству, в то время как *теорема* (или «лемма», или «следствие») является таким выражением, которое *было* доказано. Таким образом, «$0 = 1$» и его отрицание «$\neg(0 = 1)$» — являются высказываниями, но только последнее является теоремой.

суждения являются элементами (как элементы группы), а дедуктивные правила — операциями (например, групповым умножением). С логической точки зрения суждения можно рассматривать как «внешние» высказывания, обитающие в метатеории, в отличие от «внутренних» высказываний самой теории.

В дедуктивной системе логики первого порядка (на которой основана теория множеств) существует только один вид суждений: у данного высказывания есть доказательство. То есть каждое высказывание A порождает суждение «A имеет доказательство», и все суждения имеют эту форму. Правило логики первого порядка, такое как «из A и B вывести $A \wedge B$», фактически является правилом «доказательной конструкции», которое гласит, что, учитывая суждения «A имеет доказательство» и «B имеет доказательство», мы можем вывести, что «$A \wedge B$ имеет доказательство». Заметим, что суждение «A имеет доказательство» существует на другом уровне от самого высказывания A, которое является внутренним высказыванием теории.

Основное суждение теории типов, аналогичное «A имеет доказательство», записывается «$a : A$» и произносится как «терм a имеет тип A», или более свободно «a является элементом A» (или, в гомотопической теории типов, «a — точка A»). Если A — тип, представляющий высказывание, то a может быть назван *свидетелем* доказуемости A или доказательства истинности A (или даже *доказательства* A, но мы попытаемся избежать этой запутанной терминологии). В этом случае суждение $a : A$ выводимо в теории типов (для некоторого a) именно тогда, когда аналогичное суждение «A имеет доказательство» выводимо в логике первого порядка (по модулю различий в предполагаемых аксиомах и в кодировке математики, как мы обсудим в этой книге).

С другой стороны, если тип A трактуется скорее как множество, чем как высказывание (хотя, как мы увидим, различие может стать размытым), то «$a : A$» можно считать аналогичным теоретико-множественному высказыванию «$a \in A$». Однако есть существенное различие в том, что «$a : A$» является *суждением*, тогда как «$a \in A$» является *высказыванием*. В частности, когда мы внутренне работаем в теории типов, мы не можем делать такие высказывания, как «если $a : A$, то не верно, что $b : B$», и мы не можем «опровергнуть» суждение «$a : A$».

Хорошим способом понять это является то, что в теории множеств «членство» — это отношение, которое может быть или не быть между двумя ранее существовавшими объектами «a» и «A», в то время как в теории типов мы не можем говорить об элементе «a» изолированно: каждый элемент по самой своей природе является элементом некоторого типа, и этот тип (вообще говоря) однозначно определяется. Таким образом, когда мы неформально говорим «пусть x — натуральное число», в теории множеств это сокращение означает «пусть x вещь и предполагается, что $x \in N$», тогда как в теории типов «пусть $x : N$» является атомарным высказыванием: мы не можем ввести переменную без указания ее типа.

На первый взгляд это может показаться неудобным ограничением, но оно, пожалуй, ближе к интуитивному математическому значению «пусть x — натуральное число». На практике кажется, что всякий раз, когда нам действительно нужно, чтобы «$a \in A$» было высказыванием, а не суждением, всегда существовало бы объемлющее множество B и было известно, что a является его элементом, а A является его подмножеством. Эту ситуацию легко представить в теории типов, рассматривая a как элемент типа B, а A в качестве предиката на B; см. §3.5.

Последнее различие между теорией типов и теорией множеств — это трактовка равенства. Знакомое понятие равенства в математике является высказыванием: например, мы можем опровергнуть равенство или принять равенство как гипотезу. Поскольку в теории типов высказывания являются типами, это означает, что равенство является типом: для элементов $a, b : A$ (то есть, $a : A$ и $b : A$) мы имеем тип «$a =_A b$». (В теории гомотопической теории типов, конечно, это высказывание равенства может вести себя по-другому: см. §1.12 и главу 2, а также остальную часть книги). Когда $a =_A b$ обитаемо (т.е. найдется хотя бы одна такая пара элементов), мы говорим, что a и b (**пропозиционально**) **равны**.

Однако в теории типов существует также потребность в суждении о равенстве, существующем на том же уровне, что и суждение «$x : A$». Оно называется **субъективным равенством** или **дефинициальным равенством**, и мы записываем его как $a \equiv b : A$ или просто $a \equiv b$. Полезно думать о нем как о значении «равного по определению». Например, если мы определяем функцию $f : \mathbb{N} \to \mathbb{N}$ по уравнению $f(x) = x^2$, то выражение $f(3)$ по определению равно 3^2. Не имеет смысла отрицать или допускать равенство по определению; мы не можем сказать «если x по определению равно y, то z по определению не равно w». Независимо от того, равны или нет по определению два выражения, речь идет только о расширении определений; в частности, такое равенство алгоритмически разрешимо (хотя алгоритм обязательно является метатеоретическим, а не внутренним по отношению к теории).

Поскольку теория типов становится более сложной, субъективное равенство может стать более тонким, чем представлено, но это хорошая интуиция для начала. В качестве альтернативы, если мы рассматриваем дедуктивную систему как алгебраическую теорию, то субъективное равенство является простым равенством в этой теории, аналогичным равенству между элементами группы — единственная возможная путаница в том, что в дедуктивной системе теории типов существует объект (то есть, тип «$a = b$»), который ведет себя внутренне как понятие «равенства».

Причиной, по которой мы хотим, чтобы субъективное понятие равенства было таким, является требование, чтобы оно могло контролировать другую форму суждения «$a : A$». Например, предположим, что мы првели доказательство того, что $3^2 = 9$, т.е. мы вывели решение р : $(3^2 = 9)$ для некоторого p. Тогда то же свидетельство p следует считать доказательством того, что $f(3) = 9$, так как $f(3)$ равно 3^2 по определению. Лучший способ представить это — с помощью правила, в котором говорится, что с учетом суждений $a : A$ и $A \equiv B$, мы можем получить суждение $a : B$.

Таким образом, для нас теория типов будет дедуктивной системой, основанной на двух формах суждения:

Суждение	Смысл
$a : A$	«a является объектом типа A»
$a \equiv_A b$	«a и b являются по определению равными объектами типа A»

При введении дефинициального равенства, то есть, определяя одно равным другому, мы будем использовать символ «$:\equiv$». Таким образом, приведенное выше определение функции f было бы записано в виде $f(x) :\equiv x^2$.

Поскольку суждения не могут быть объединены в более сложные утверждения, символы «:» и «$\equiv$» связываются более свободно, чем все остальное[2]. Например, «$p : x = y$» следует понимать как «$p : (x = y)$», что имеет смысл, поскольку «$x = y$» является типом, а не как «$(p : x) = y$», что бессмысленно, поскольку «$p : x$» — это суждение и не может быть чему-либо равно. Аналогичным образом, «$A \equiv x = y$» может быть проанализировано только как «$A \equiv (x = y)$», хотя в крайних случаях, таких как этот, нужно добавлять круглые скобки, чтобы помочь пониманию прочитанного. Более того, в дальнейшем мы будем применять общую нотацию цепочных равенств — например, записывая $a = b = c = d$ для обозначения «$a = b$, $b = c$

[2] В формализованной теории типов запятые и турникетные знаки («$\vdash$», «$\dashv$», и т.п.) могут связываться еще более свободно. Например, $x : A, y : B \vdash c : C$ разбирается как $((x : A), (y : B)) \vdash (c : C)$. Однако в этой книге мы будем воздерживаться от таких обозначений вплоть до приложения A.

и $c = d$, следовательно, $a = d$» — и мы также будем включать дефинициальные равенства в такие цепочки. Контекста обычно будет достаточно, чтобы сделать такое намерение понятным.

Здесь, пожалуй, подходящее место, чтобы упомянуть также, что общая математическая нотация «$f : A \to B$», выражающее тот факт, что f является функцией от A к B, можно рассматривать как типизирующее суждение, так как мы используем «$A \to B$» в качестве обозначения типа функций от A к B (как это принято в теории типов, см. §1.4).

Суждения могут зависеть от предположений вида $x : A$, где x — переменная, а A — тип. Например, мы можем построить объект $m + n : \mathbb{N}$ в предположениях, что $m, n : \mathbb{N}$. Другой пример: если предположить, что A является типом, $x, y : A$ и $p : x =_A y$, мы можем сконструировать элемент $p^{-1} : y =_A x$. Совокупность всех таких допущений называется **контекстом**; с топологической точки зрения его можно рассматривать как «пространство параметров». На самом деле, технически контекст должен быть упорядоченным списком предположений, поскольку более поздние предположения могут зависеть от предыдущих: предположение $x : A$ может быть сделано только после предположений о любых переменных, появляющихся в типе A.

Если тип A в предположении $x : A$ представляет собой высказывание, то предположение является теоретико-типовым вариантом *гипотезы*: мы предполагаем, что выполнено высказывание A. Когда типы рассматриваются как высказывания, мы можем опускать имена их доказательств. Таким образом, во втором примере, приведенном выше, мы можем вместо этого сказать, что в предположении $x =_A y$ мы можем доказать, что $y =_A x$. Однако, так как мы занимаемся математикой «доказательной важности», мы часто будем ссылаться на доказательства как на объекты. Так, в приведенном выше примере мы можем установить, что p^{-1}, вместе с доказательствами транзитивности и рефлексивности, ведет себя как группоид; см. главу 2.

Обратите внимание, что в этом смысле слова «предположение», мы можем предположить пропозициональное равенство (предполагая переменную $p : x = y$), но мы не можем принять оценочное равенство $x \equiv y$, поскольку оно не является типом, который может иметь элемент. Однако, мы можем сделать кое-что, что выглядит как принятие условного равенства: если у нас есть тип или элемент, который включает в себя переменную $x : A$, то мы можем подставить любой конкретный элемент $a : A$ вместо x для получения более конкретного типа или элемента. Мы будем иногда использовать такой стиль речи, как «теперь предположим, что x $\equiv a$», чтобы сослаться на этот процесс замещения, хотя это не является предположением в техническом смысле, введенном выше.

Точно так же мы не можем доказать и субъективное равенство, поскольку оно не является типом, для которого мы можем привести пример. Тем не менее, мы иногда будем приводить оценочные равенства в качестве части теоремы, например, «существует $f : A \to B$ такое, что $f(x) \equiv$ y». Это следует рассматривать как создание двух отдельных суждений: сначала мы принимаем решение $f : A \to B$ для некоторого элемента f, затем делаем дополнительное суждение, что $f(x) \equiv$ y.

В оставшейся части этой главы мы попытаемся дать неформальное представление теории типов, достаточное для целей этой книги; мы приведем более формальное описание в Приложении А. Помимо некоторых довольно очевидных правил (таких как тот факт, что субъективно равные вещи всегда могут быть заменены друг другом), правила теории типов могут быть сгруппированы в типы формировщиков. Каждый тип состоит из способа конструирования типов (возможно, используя ранее созданные типы), а также правил построения и поведения элементов этого типа. В большинстве случаев эти правила следуют довольно предсказуемой схеме, но мы не будем пытаться делать это именно здесь; см., однако, начало §1.5, а также главу 5.

Важный аспект теории типов, представленный в этой главе, заключается в том, что она полностью состоит из правил, без каких-либо аксиом. В описании дедуктивных систем в терминах суждений, правила — это то, что позволяет нам вывести одно суждение из совокупности других, в то время как аксиомы являются суждениями, которые мы даем в самом начале. Если мы думаем о дедуктивной системе как о формальной игре, то правила — это правила игры, а аксиомы — начальная позиция. И если мы думаем о дедуктивной системе как об алгебраической теории, то правила являются операциями теории, в то время как аксиомы являются образующими для некоторой конкретной свободной модели этой теории.

В теории множеств единственными правилами являются правила логики первого порядка (такие как правило, позволяющее нам вывести «$A \wedge B$ имеет доказательство» из «A имеет доказательство» и «B имеет доказательство»): вся информация о поведении множеств содержится в аксиомах. В противоположность этому, в теории типов это обычно правила, которые содержат всю информацию, без каких-либо аксиом. Например, в §1.5 мы увидим, что существует правило, позволяющее нам вынести суждение «$(a, b) : A \times B$» из «$a : A$» и «$b : B$», тогда как в теории множеств аналогичное высказывание было бы (следствием) аксиомы спаривания.

Преимущество формулирования теории типов с использованием только правил состоит в том, что правила являются «процедурными». В частности, это свойство делает возможным (хотя автоматически и не обеспечивает) хорошие вычислительные свойства теории типов, такие как «каноничность». Однако, хотя этот стиль и работает для теорий традиционного типа, мы пока не знаем, как сформулировать все это в гомотопической теории типов. В частности, в §§ 2.9 и 2.10 и в главе 6 нам придется дополнить правила теории типов, представленные в этой главе, введя дополнительные аксиомы, особенно аксиому унивалентности. Однако в этой главе мы ограничиваемся традиционной теорией типов, основанной на правилах.

1.2 Типы функций

Для типов A и B мы можем построить тип $A \to B$ **функций** с областью A и кообластью B. Мы также иногда рассматриваем функции как **отображения**. В отличие от теории множеств, функции не определяются как функциональные отношения, скорее они являются примитивной концепцией теории типов. Мы объясняем тип функции, указывая, что можно делать с функциями, как их строить и какие равенства они порождают.

Для функции $f : A \to B$ и элемента области $a : A$, мы можем **применить** эту функцию для получения элемента кообласти B, обозначаемого $f(a)$ и называемого **значением** f в точке a. В теории типов принято опускать круглые скобки и выражать $f(a)$ просто как $f\,a$, и мы иногда будем это делать.

Но как мы можем построить элементы $A \to B$? Существует два эквивалентных способа: либо путем прямого определения, либо с помощью λ-абстракции. Введение функции по определению означает, что мы вводим функцию, давая ей имя — скажем, f — и определяя $f : A \to B$ уравнением

$$f(x) :\equiv \Phi \tag{1.2.1}$$

где x — переменная, а Φ — выражение, которое может содержать x. Чтобы это имело место, мы должны убедиться, что $\Phi : B$ в предположении $x : A$.

Теперь мы можем вычислить $f(a)$, заменив переменную x в Φ на a. В качестве примера рассмотрим функцию $f : \mathbb{N} \to \mathbb{N}$, которая определена как $f(x) :\equiv x + x$ (мы определим $\mathbb{N}$ и $+$ в §1.9). Тогда $f(2)$ собственно равно $2 + 2$.

Если мы не хотим вводить имя для функции, то можем использовать **λ-абстракцию**. Для выражения Φ типа B, которое может содержать $x : A$, мы пишем $\lambda(x : A).\,\Phi$, чтобы указать ту же функцию, что и в (1.2.1). Таким образом, имеем

$$(\lambda(x : A).\, \Phi) : A \to B\,.$$

Для примера из предыдущего абзаца соответствующая функция запишется в виде

$$(\lambda(x : \mathbb{N}).\, x + x) : \mathbb{N} \to \mathbb{N}\,.$$

В качестве еще одного примера, для любых типов A и B и любого элемента $y : B$, имеем **постоянную функцию** $(\lambda(x : A).\, y) : A \to B$.

Обычно мы опускаем тип переменной x в λ-абстракции и записываем $\lambda x.\, \Phi$, так как указание $x : A$ выводимо из суждения, что функция $\lambda x.\, \Phi$ имеет тип $A \to B$. По соглашению, «область действия» переменной привязки «$\lambda x.$» — это весь остаток выражения, если он не ограничен круглыми скобками. Так, например, $\lambda x.\, x + x$ следует анализировать как $\lambda x.\, (x + x)$, а не как $(\lambda x.\, x) + x$ (что в этом случае будет неверно).

Другим эквивалентным обозначением является

$$(x \mapsto \Phi) : A \to B\,.$$

Иногда мы будем использовать символ «$-$» в выражении Φ вместо переменной, чтобы обозначить неявную λ-абстракцию. Например, $g(x, -)$ является другим способом записи для $\lambda y.\, g(x, y)$.

Теперь λ-абстракция является функцией, поэтому мы можем применить ее к аргументу $a : A$. Тогда мы получаем следующее **правило вычисления**[3], которое является дефинициальным равенством:

$$(\lambda x.\, \Phi)(a) \equiv \Phi'$$

где Φ' — выражение Φ, в котором все вхождения x заменены на a. Продолжая приведенный выше пример, имеем

$$(\lambda x.\, x + x)(2) \equiv 2 + 2\,.$$

Заметим, что из любой функции $f : A \to B$ можно построить функцию лямбда-абстракции $\lambda x.\, f(x)$. Так как это, по определению, «функция, которая применяет f к своему аргументу», мы считаем ее дефинициально равной f:[4]

$$f \equiv (\lambda x.\, f(x))\,.$$

Это равенство является **принципом уникальности для типов функций**, поскольку оно показывает, что f однозначно определяется своими значениями.

Введение функций по определениям с явными параметрами можно свести к простым определениям с помощью λ-абстракции: т.е. мы можем прочитать определение $f : A \to B$ посредством

$$f(x) :\equiv \Phi$$

как

$$f :\equiv \lambda x.\, \Phi\,.$$

При выполнении вычислений с использованием переменных мы должны быть осторожны при замене переменной выражением, которое также включает переменные, потому что мы хотим сохранить связующую структуру выражений. Под *связующей структурой* понимается

[3] Использование этого равенства часто называют β-**конверсией** или β-**редукцией**.

[4] Использование этого равенства часто называют η-**конверсией** или η-**расширением**.

воображаемая связь, генерируемая связующими, такими как λ, $\prod$ и $\sum$ (с последними мы вскоре встретимся) между местом, где вводится переменная, и где она используется. В качестве примера рассмотрим $f : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})$ с определением

$$f(x) :\equiv \lambda y.\, x + y\,.$$

Теперь, если мы предположили, что $y : \mathbb{N}$, то что такое $f(y)$? Было бы неправильно, просто наивно, заменить x на y всюду в выражении $\lambda y.\, x + y$, определяющем $f(x)$, получив $\lambda y.\, y + y$, потому что это означает, что y станет **связанным**. Без этого y соответствует нашему предположению, но после такой замены он будет относиться к аргументу λ-абстракции. Следовательно, эта наивная замена приведет к разрушению связующей структуры, что позволит нам выполнять вычисления, которые семантически несостоятельны.

Так что же такое $f(y)$ в этом примере? Обратите внимание, что связанные (или «фиктивные») переменные, такие как y в выражении $\lambda y.\, x + y$, имеют только локальное значение и могут быть согласованно заменены любой другой переменной, сохраняющей структуру привязки. Действительно, $\lambda y.\, x + y$ объявляется дефинициально равным[5] $\lambda z.\, x + z$. Отсюда следует, что $f(y)$ дефинициально равно $\lambda z.\, y + z$, и это отвечает на наш вопрос (вместо z может быть использована любая переменная, отличная от y, что приводит к тому же результату).

Конечно, все это должно быть знакомо любому математику: это тот же феномен, что и факт, что если $f(x) :\equiv \int_1^2 \frac{dt}{t-s}$, то $f(t)$ скорее $\int_1^2 \frac{ds}{x-t}$, но никак не $\int_1^2 \frac{dt}{t-t}$. λ-абстракция связывает фиктивную переменную точно так же, как и интеграл.

Мы видели, как определить функции одной переменной. Одним из способов определения функций нескольких переменных было бы использование декартова произведения, которое будет введено позже; функция с параметрами A и B и результатом C может быть задана типом $f : A \times B \to C$. Однако есть еще один выбор, который позволяет избежать использования произведения типов, который называется **каррированием** (по имени математика Хаскелла Карри).

Идея каррирования состоит в представлении функции двух входных данных $a : A$ и $b : B$ как функции, которая принимает *одно* данное $a : A$ и возвращает *другую функцию*, которая затем принимает второе данное $b : B$ и возвращает результат. То есть мы считаем, что функции с двумя переменными принадлежат к типу итерационных функций, $f : A \to (B \to C)$. Мы можем также записать этот тип функции без круглых скобок, как $f : A \to B \to C$, с правой ассоциативностью по умолчанию. Затем, учитывая $a : A$ и $b : B$, мы можем применить f к a, а затем применить результат к b, получив $f(a)(b) : C$. Чтобы избежать обилия круглых скобок, мы позволим себе писать $f(a)(b)$ в виде $f(a, b)$, даже если в нем нет произведений. При полном опускании круглых скобок вокруг аргументов функции мы будем писать $f\,a\,b$ вместо $(f\,a)\,b$, теперь уже с левой ассоциативностью по умолчанию, так что f будет применяться к своим аргументам в правильном порядке.

Наше обозначение для определений с явными параметрами распространяется на такую ситуацию: мы можем определить именованную функцию $f : A \to B \to C$, задаваемую уравнением

$$f(x, y) :\equiv \Phi$$

где $f : C$, при условии $x : A$ и $y : B$. С использованием λ-абстракцию, это соответствует

$$f :\equiv \lambda x.\, \lambda y.\, \Phi\,,$$

[5] Использование этого равенства часто называют α**-конверсией**.

которое также может быть заптисано как

$$f :\equiv x \mapsto y \mapsto \Phi \,.$$

Мы также можем неявно абстрагироваться от нескольких переменных, используя требуемое количество символа «$-$», например. $g(-,-)$ означает $\lambda x.\, \lambda y.\, g(x,y)$. Каррирование функции от трех или более аргументов является прямым расширением того, что мы только что описали.

1.3 Универсумы и семейства

До сих пор мы использовали выражение «A — тип» неформально. Мы сделаем его более точным, вводя **универсумы**. Универсум — это тип, элементы которого являются типами. Как и в теории наивных множеств, мы могли бы пожелать для универсума всех типов $\mathcal{U}_\infty$ включать и самого себя (т.е. $\mathcal{U}_\infty : \mathcal{U}_\infty$). Однако, как и в теории множеств, это необоснованно, т.е. из этого можно вывести, что каждый тип, включая пустой тип, представляющий высказывание $False$ (см. §1.7), заселен. Например, используя представление множеств деревьями, сможно напрямую кодировать парадокс Рассела [Coq92].

Чтобы избежать парадокса, мы вводим иерархию универсумов

$$\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \cdots$$

где каждый универсум $\mathcal{U}_i$ является элементом следующего универсума $\mathcal{U}_{i+1}$. Более того, мы предполагаем, что наши универсумы являются **кумулятивными**, то есть все элементы i-го универсума также являются элементами $(i+1)$-го универсума, то есть если $A : \mathcal{U}_i$, то также $A : \mathcal{U}_{i+1}$. Это удобно, но имеет слегка неприятное следствие, что элементы больше не имеют уникальных типов, и немного сложны в других отношениях, которые нас здесь не интересуют; см. Примечания.

Когда мы говорим, что A — тип, то имеем в виду, что он обитает в некотором универсуме $\mathcal{U}_i$. Обычно мы хотим избегать явного упоминания уровня i и просто предпологать, что уровни могут быть назначены согласованным образом; поэтому, мы можем записать $A : \mathcal{U}$, опуская уровень. Таким образом, мы можем даже писать $\mathcal{U} : \mathcal{U}$ (полагая индексы неявными), что можно понять как $\mathcal{U}_i : \mathcal{U}_{i+1}$. Указание универсумов в этом стиле относится к **типовой неоднозначности**. Это удобно, но немного опасно, так как позволяет нам писать достоверные доказательства, которые воспроизводят парадоксы ссылок на себя. Если имеются какие-либо сомнения относительно правильности аргумента, способ проверить его — попытаться последовательно присвоить уровни всем универсумам, появляющимся в нем. Когда предполагается некоторый универсум $\mathcal{U}$, мы можем ссылаться на типы, принадлежащие $\mathcal{U}$, как на **малые типы**.

Для моделирования набора типов, порожденных заданным типом A, мы используем функции $B : A \to \mathcal{U}$, кодоменом которых является некоторый универсум. Эти функции называются **семействами типов** (или иногда *зависимыми типами*); они соответствуют семействам множеств, которые используются в теории множеств.

Примером семейства типов является семейство конечных множеств $\mathsf{Fin} : \mathbb{N} \to \mathcal{U}$, где $\mathsf{Fin}(n)$ — тип с ровно n элементами (мы еще не можем *определить* семейство Fin — действительно, мы даже не представили его область $\mathbb{N}$, но вскоре сделаем это, см. упражнение 1.9). Можно обозначить элементы $\mathsf{Fin}(n)$ как $0_n, 1_n, \ldots, (n-1)_n$, где индексы акцентируют то, что элементы $\mathsf{Fin}(n)$ отличаются от элементов $\mathsf{Fin}(m)$, если n отлично от m, и все они отличаются от обычных натуральных чисел (которые мы введем в §1.9).

Более тривиальным (но очень важным) примером семейства типов является семейство **постоянных** типов в типе $B : \mathcal{U}$, которое, конечно, является постоянной функцией $(\lambda(x : A).\, B) : A \to \mathcal{U}$.

В качестве *контр-примера*, в нашей версии теории типов нет семейства типов «$\lambda(i : \mathbb{N}).\, \mathcal{U}_i$». Действительно, нет ни одного универсума, достаточно большого, чтобы быть кообластью такого семейства. Более того, мы даже не отождествляем индексы i универсумов $\mathcal{U}_i$ с натуральными числами $\mathbb{N}$ теории типов (последний вводится в §1.9).

1.4 Типы зависимых функций ($\prod$-типы)

В теории типов часто используется более общая версия типов функций, называемая **$\prod$-типом** или **типом зависимых функций**. Элементами $\prod$-типа являются функции, называемые **зависимыми функциями**, тип кообласти которых может меняться в зависимости от элемента области, к которому применяется функция. Используется имя «$\prod$-тип», потому что этот тип также можно рассматривать как декартово произведение над заданным типом.

Для типа $A : \mathcal{U}$ и семейства $B : A \to \mathcal{U}$, можно построить тип зависимых функций $\prod_{(x:A)} B(x) : \mathcal{U}$. Существуют альтернативные обозначения этого типа, такие как

$$\textstyle\prod_{(x:A)} B(x) \qquad \displaystyle\prod_{(x:A)} B(x) \qquad \prod(x : A), B(x)\,.$$

Если B — постоянное семейство, то $\prod$-типом является обычный тип функции:

$$\prod_{(x:A)} B \equiv (A \to B)\,.$$

В действительности, все конструкции $\prod$-типов являются обобщениями соответствующих конструкций на типах обычных функций.

Мы можем ввести зависимые функции с помощью явных определений: чтобы определить $f : \prod_{(x:A)} B(x)$, где f — имя определяемой зависимой функции, требуется выражение $\Phi : B(x)$, возможно, связанное с переменной $x : A$, и мы пишем

$$f(x) :\equiv \Phi \qquad \text{для } x : A\,.$$

В качестве альтернативы мы можем использовать **λ-абстракцию**

$$\lambda x.\, \Phi : \prod_{(x:A)} B(x)\,. \tag{1.4.1}$$

Как и в случае с обычными функциями, мы можем **применить** зависимую функцию $f : \prod_{(x:A)} B(x)$ к аргументу $a : A$, чтобы получить элемент $f(a) : B(a)$. Равенства такие же, как и для обычного типа функции, т.е. для $a : A$ мы имеем правило вычисления $f(a) \equiv \Phi'$ и $(\lambda x.\, F)(a) \equiv \Phi'$, где Φ' получается заменой всех вхождения x из Φ на a (как обычно, избегая захвата переменной). Аналогично, имеем принцип уникальности $f = (\lambda x.\, f(x))$ для любой $f : \prod_{(x:A)} B(x)$.

В качестве примера, напомним из §1.3, что существует семейство типов $\mathsf{Fin} : \mathbb{N} \to \mathcal{U}$, чьи значения — обычные конечные множества с элементами $0_n, 1_n, \ldots, (n-1)_n : \mathsf{Fin}(n)$. Тогда существует зависимая функция $\mathsf{fmax} : \prod_{(n:\mathbb{N})} \mathsf{Fin}(n+1)$, которая возвращает «самый большой» элемент каждого непустого конечного типа, $\mathsf{fmax}(n) :\equiv n_{n+1}$. Как и в случае с Fin, мы пока не можем определить fmax, но вскоре это будет сделано; см. упражнение 1.9.

Другим важным классом типов зависимых функций являются функции, которые являются **полиморфными** над данным универсумом. Полиморфная функция — это функция, которая принимает тип в качестве одного из своих аргументов, а затем действует на элементы этого типа (или на другие, построенные из него, типы). Примером является полиморфная тождественная функция $\mathsf{id} : \prod_{(A:\mathcal{U})} A \to A$, которую мы определяем как $\mathsf{id} :\equiv \lambda(A:\mathcal{U}).\, \lambda(x:A).\, x$ (подобно λ-абстракциям, типы функций Πs автоматически охватывают оставшуюся часть выражения, если не указано иное; таким образом, $\mathsf{id} : \prod_{(A:\mathcal{U})} A \to A$ означает $\mathsf{id} : \prod_{A:\mathcal{U}}(A \to A)$; это соглашение, хотя и необычное для традиционной математики, распространено в теории типов).

Иногда мы записываем некоторые аргументы зависимой функции как индексы. Например, мы можем эквивалентно определить полиморфную функцию тождества как $\mathsf{id}_A(x) :\equiv x$. Более того, если аргумент может быть выведен из контекста, мы можем вообще его опустить. Например, если $a : A$, то запись $\mathsf{id}(a)$ однозначна, так как id должен означать id_A, чтобы он был применим к a.

Другим, менее тривиальным примером полиморфной функции является операция «обмена», которая переключает порядок аргументов (каррированной) функции с двумя аргументами:

$$\mathsf{swap} : \prod_{(A:\mathcal{U})} \prod_{(B:\mathcal{U})} \prod_{(C:\mathcal{U})} (A \to B \to C) \to (B \to A \to C)\,.$$

Можно определить ее как

$$\mathsf{swap}(A,B,C,g) :\equiv \lambda b.\, \lambda a.\, g(a)(b)\,.$$

Мы можем также эквивалентно записать аргументы типа как индексы:

$$\mathsf{swap}_{A,B,C}(g)(b,a) :\equiv g(a,b)\,.$$

Заметим, что как и для обычных функций, мы используем карринг для определения зависимых функций с несколькими аргументами. Однако, в последнем случае вторая область может зависеть от первой, а кодомен может зависеть от обоих областей. То есть, для заданных $A : \mathcal{U}$, семейства типов $B : A \to \mathcal{U}$ и $C : \prod_{(x:A)} B(x) \to \mathcal{U}$, мы можем построить тип $\prod_{(x:A)} \prod_{(y:B(x))} C(x,y)$ функций с двумя аргументами (подобно λ-абстракциям, $\prod s$ автоматически распространяется на остальную часть выражения, если нет ограничителя; поэтому $C : \prod_{(x:A)} B(x) \to \mathcal{U}$ означает $C : \prod_{(x:A)}(B(x) \to \mathcal{U})$). В случае, когда B постоянна и равна A, можно сократить обозначение и писать $\prod_{(\mathrm{x,y:A})}$; например, тип swap также может быть записан как

$$\mathsf{swap} : \prod_{(A,B,C:\mathcal{U})} (A \to B \to C) \to (B \to A \to C)\,.$$

Наконец, для $f : \prod_{(x:A)} \prod_{(y:B(x))} C(x,y)$ и аргументов$a : A$ и $b : B(a)$, имеем $f(a)(b) : C(a,b)$, что, как и ранее, записываем как $f(a,b) : C(a,b)$.

1.5 Типы произведений

Для заданных типов $A, B : \mathcal{U}$ мы вводим тип $A \times B : \mathcal{U}$, который называем их **декартовым произведением**. Мы также вводим нульарный тип произведения, называемый **типом единицы**

$\mathbf{1} : \mathcal{U}$. Мы предполагаем, что элементы $A \times B$ являются **парами** $(a, b) : A \times B$, где $a : A$ и $b : B$, а единственным элементом из $\mathbf{1}$ является некоторый особый объект $\star : \mathbf{1}$. Однако, в отличие от теории множеств, где упорядоченные пары определяются как отдельные множества, а затем собираются вместе в декартово произведение, в теории типов упорядоченные пары являются примитивным понятием, как и функции.

Замечание 1.5.1. Существует общая схема введения новой разновидности типа в теории типов. Мы уже встречались с этой схемой в §§ 1.2 и 1.4[6], поэтому стоит привести ее общий формат. Для определения типа, мы указываем:

1. как формировать новые типы из исходных с помощью **правил формирования** (например: мы можем сформировать тип функции $A \to B$, когда A является типом и когда B является типом; мы можем сформировать тип зависимой функции $\prod_{(x:A)} B(x)$, когда A — тип, а $B(x)$ — тип для $x : A$);

2. как построить элементы этого типа; они называются **конструкторами типа** или **правилами введения** (например, тип функции имеет один конструктор, λ-абстракцию. Напомним, что прямое определение, подобное $f(x) :\equiv 2x$, может быть эквивалентно выражено как λ-абстракция $f :\equiv \lambda x.\, 2x$);

3. как использовать элементы этого типа; они называются **правилами выделения** или **выделителями типов** (например, тип функции имеет один выделитель, именуемый применением функции);

4. **правило вычисления**[7], которое выражает действие выделителя на конструкторе (например, для функций, в правиле вычисления указано, что $(\lambda x.\, \Phi)(a)$ дефинициально эквивалентно замене x на a в Φ);

5. необязательный **принцип уникальности**[8], который выражает уникальность отображений в этот тип или из него. Для некоторых типов принцип уникальности характеризует отображения в тип, заявляя, что каждый элемент типа однозначно определяется результатами применения к нему выделителей и может быть восстановлен из этих результатов путем применения конструктора — таким образом выражая, как действуют конструкторы на выделителях, дважды в правиле вычисления (например, для функций принцип уникальности утверждает, что любая функция f субъективно равна «расширенной» функции $\lambda x.\, f(x)$ и, следовательно, однозначно определяется ее значениями). Для других типов принцип уникальности гласит, что каждое отображение (функция) из этого типа однозначно определяется некоторыми данными (например, тип копроизведения, введенный в §1.7, принцип уникальности которого упомянут в §2.15).
   Когда принцип уникальности не принимается как правило дефинициального равенства, он часто, тем не менее, доказывается как пропозициональное равенство из других правил для данного типа. В этом случае мы называем это **пропозициональным принципом уникальности** (в последующих главах мы также иногда будем иметь дело с *пропозициональными правилами вычислений*).

[6]Исключение составляет описание универсумов.
[7]также называемое β-**редукцией**
[8]также называемый η-**расширением**

Правила вывода в Приложении A.2 организованы и названы соответствующим образом; см., например, Приложение A.2.4, где реализована каждая возможность.

Способ построения пар очевиден: для $a : A$ и $b : B$, мы можем сформировать $(a, b) : A \times B$. Аналогично, существует единственный способ построения элементов из $\mathbf{1}$, а именно: $\star : \mathbf{1}$. Мы ожидаем, что «каждый элемент из $A \times B$ — пара», что является принципом уникальности для произведений; мы не заявляем его как правило теории типов, но позже мы докажем это как пропозициональное равенство.

Теперь, как мы можем *использовать* пары, т.е. как мы можем определить функции от типа произведения? Рассмотрим сначала определение обычной функции $f : A \times B \to C$. Поскольку мы подразумеваем в качестве пар только элементы из $A \times B$, то, чтобы должным образом определить такую функцию, будет достаточно указать результат применения f к паре (a, b). Мы предписываем этот результат посредством предоставления функции $g : A \to B \to C$. Таким образом, мы вводим новое правило (правило исключения для произведений), в котором говорится, что для любого такого g мы можем определить функцию $f : A \times B \to C$:

$$f((a, b)) :\equiv g\,(a)\,(b)\,.$$

Мы не будем записывать здесь g как $g(a, b)$, чтобы подчеркнуть, что g не является функцией на произведении (однако далее мы будем часто писать $g(a, b)$, как для функций на произведении, так и для каррированных функций двух переменных). Это определяющее уравнение является правилом вычисления для типов произведений.

Заметим, что в теории множеств мы бы объяснили приведенное выше определение f тем, что каждый элемент из $A \times B$ является упорядоченной парой, так что достаточно определить f на таких парах. Напротив, теория типов меняет ситуацию: мы предполагаем, что функция на $A \times B$ хорошо определена, как только мы укажем ее значения на парах, и из этого (или, точнее, из более общей версии для зависимых функций, вводимой ниже) мы сможем доказать, что каждый элемент из $A \times B$ — пара. С точки зрения теории категорий мы можем сказать, что мы определяем произведение $A \times B$ как левый сопряженный к «экспоненциалу» $B \to C$, который мы уже ввели.

В качестве примера, мы можем вывести функции **проекции**

$$\begin{aligned} \mathsf{pr}_1 &: A \times B \to A \\ \mathsf{pr}_2 &: A \times B \to B \end{aligned}$$

с определяющими уравнениями

$$\begin{aligned} \mathsf{pr}_1((a, b)) &:\equiv a \\ \mathsf{pr}_2((a, b)) &:\equiv b\,. \end{aligned}$$

Вместо того, чтобы ссылаться на этот принцип определения функции каждый раз, когда мы хотим определить функцию, альтернативный подход заключается в том, чтобы использовать его один раз в универсальном случае, а затем просто применить полученную функцию во всех других случаях. То есть мы можем определить функцию типа

$$\mathsf{rec}_{A \times B} : \prod_{C:\mathcal{U}} (A \to B \to C) \to A \times B \to C \tag{1.5.2}$$

с определяющим уравнением

$$\mathsf{rec}_{A \times B}(C, g, (a, b)) :\equiv g\,(a)\,(b)\,.$$

Тогда вместо определения функций, таких как pr_1 и pr_2, непосредственно определяющими уравнениями, мы могли бы определить

$$\begin{aligned} \mathsf{pr}_1 &:\equiv \mathsf{rec}_{A\times B}(A, \lambda a.\, \lambda b.\, a) \\ \mathsf{pr}_2 &:\equiv \mathsf{rec}_{A\times B}(B, \lambda a.\, \lambda b.\, b)\,. \end{aligned}$$

Мы ссылаемся на функцию $\mathsf{rec}_{A\times B}$ как на **рекурсор** для типов произведений. Название «рекурсор» не совсем удачно, так как рекурсия здесь не имеет места. Это происходит из-за того, что типы произведений является вырожденным примером общей структуры для индуктивных типов, а для таких типов, как натуральные числа, рекурсор будет фактически рекурсивным. Мы можем также говорить о **принципе рекурсии** для декартовых произведений, означающем, что можно определить функцию $f : A \times B \to C$, как указано выше, задавая ее значение на парах.

Мы оставляем в качестве простого упражнения доказательство того, что рекурсор может быть получен из проекций и наоборот.

Также можно привести рекурсор для единичного типа:

$$\mathsf{rec}_{\mathbf{1}} : \prod_{C:\mathcal{U}} C \to \mathbf{1} \to C$$

с определяющим уравнением

$$\mathsf{rec}_{\mathbf{1}}(C, \mathsf{c}, \star) :\equiv c\,.$$

Хотя мы включаем его в сответствии с шаблоном определения типов, рекурсор для $\mathbf{1}$ абсолютно бесполезен, потому что мы могли бы определить такую функцию непосредственно, просто игнорируя аргумент типа $\mathbf{1}$.

Чтобы иметь возможность определять *зависимые* функции над типом произведения, мы должны обобщить рекурсор. Для $C : A{\times}B \to \mathcal{U}$ мы можем определить функцию $f : \prod_{(x:AxB)} C(x)$, предоставляя функцию $g : \prod_{(x:A)} \prod_{(y:B)} C((x,y))$ с определяющим уравнением

$$f((x,y)) :\equiv g\,(x)\,(y)\,.$$

Например, таким образом мы можем доказать принцип пропозициональной уникальности, который гласит, что каждый элемент из $A \times B$ равен некоторой паре. В частности, мы можем построить функцию

$$\mathsf{uniq}_{A\times B} : \prod_{x:A\times B} ((\mathsf{pr}_1(x), \mathsf{pr}_2(x)) =_{A\times B} x)\,.$$

Здесь мы используем тождественный тип, который мы собираемся ввести ниже в §1.12. Тем не менее, все, что нам нужно знать сейчас, это то, что для любого $x : A$ есть элемент рефлексивности $\mathsf{refl}_x : x =_A x$ для любого $x : A$. Учитывая это, мы можем определить

$$\mathsf{uniq}_{A\times B}((a,b)) :\equiv \mathsf{refl}_{(a,b)}\,.$$

Эта конструкция работает, потому что в случае, когда $x :\equiv (a,b)$, мы можем вычислить

$$(\mathsf{pr}_1((a,b)), \mathsf{pr}_2((a,b))) \equiv (a,b)$$

используя определяющие уравнения для проекций. Поэтому

$$\mathsf{refl}_{(a,b)} : (\mathsf{pr}_1((a,b)), \mathsf{pr}_2((a,b))) = (a,b)$$

хорошо типизирован, так как обе стороны равенства дефинициально равны.

В более общем смысле, способность определять зависимые функции подобным образом означает, что для доказательства свойства для всех элементов произведения достаточно доказать его для своих канонических элементов — упорядоченных пар. Когда мы переходим к индуктивным типам, таким как натуральные числа, аналогичным свойством будет возможность записывать доказательства по индукции. Таким образом, если мы делаем так, как было описано, и применяем этот принцип один раз в универсальном случае, мы называем результирующую функцию **индукцией** для типов произведений: для заданных $A, B : \mathcal{U}$ имеем

$$\mathsf{ind}_{A\times B} : \prod_{C:A\times B\to\mathcal{U}} \left(\prod_{x:A}\ \prod_{y:B} C((x,y))\right) \to \prod_{x:A\times B} C(x)$$

с определяющим уравнением

$$\mathsf{ind}_{A\times B}(C, g, (a,b)) :\equiv g\,(a)\,(b)\,.$$

Аналогично, мы можем говорить о зависимой функции, определяемой на парах, получаемых из **принципа индукции** декартова произведения. Легко видеть, что рекурсор является лишь частным случаем индукции в случае, когда семейство C постоянное. Поскольку индукция описывает, как использовать элемент типа произведения, индукция также называется (**зависимым**) **выделителем**, а рекурсия — **независимым выделителем**.

Индукция для единичного типа оказывается более полезной, чем рекурсор:

$$\mathsf{ind}_{\mathbf{1}} : \prod_{C:\mathbf{1}\to\mathcal{U}} C(\star) \to \prod_{x:\mathbf{1}} C(x)$$

с определяющим уравнением

$$\mathsf{ind}_{\mathbf{1}}(C, c, \star) :\equiv c\,.$$

Индукция позволяет доказать принцип пропозициональной уникальности для $\mathbf{1}$, который утверждает, что его единственным обитателем является $\star$. То есть мы можем построить

$$\mathsf{uniq}_{\mathbf{1}} : \prod_{x:\mathbf{1}} x = \star$$

используя определяющее уравнение

$$\mathsf{uniq}_{\mathbf{1}}(\star) :\equiv \mathsf{refl}_{\star}$$

или, что эквивалентно, используя индукцию:

$$\mathsf{uniq}_{\mathbf{1}} :\equiv \mathsf{ind}_{\mathbf{1}}(\lambda x.\, x = \star, \mathsf{refl}_{\star})\,.$$

1.6 Типы зависимых пар ($\sum$-типы)

Подобно тому, как мы обобщили типы функций (§1.2) на типы зависимых функций (§1.4), часто бывает полезно обобщить типы произведений из §1.5 так, чтобы тип второго компонента пары изменялся в зависимости от выбора первого компонента. Это называется **типом зависимой пары** или **$\sum$-типом**, потому что в теории множеств он соответствует индексированной сумме (в смысле копроизведения или несвязного объединения) над данным типом.

Для типа $A : \mathcal{U}$ и семейства $B : A \to \mathcal{U}$ тип зависимой пары записывается как $\sum_{(x:A)} B(x) : \mathcal{U}$. Альтернативные обозначения:

$$\sum_{(x:A)} B(x) \qquad \sum_{(x:A)} B(x) \qquad \sum(x : A), B(x)\,.$$

Как и другие связующие конструкции, такие как λ-абстракции и $\prod$, $\sum$ автоматически распространяются на остальную часть выражения, если нет ограничителя; т.е. $\sum_{(x:A)} B(x) \times C(x)$ означает $\sum_{(x:A)}(B(x) \times C(x))$.

Способ построения элементов типа зависимой пары заключается в спаривании: мы имеем $(a, b) : \sum_{(x:A)} B(x)$ при заданных $a : A$ и $b : B(a)$. Если B постоянно, то тип зависимой пары является обычным типом декартова произведения:

$$\left(\sum_{x:A} B\right) \equiv (A \times B)\,.$$

Все конструкции на $\sum$-типах возникают как простые обобщения для типов произведений, причем зависимые функции часто заменяют обычные.

Например, принцип рекурсии говорит о том, что для определения обычной функции из $\sum$-типа $f : \left(\sum_{(x:A)} B(x)\right) \to C$ мы предоставляем функцию $g : \prod_{(x:A)} B(x) \to C$, а затем используем определяющее уравнение

$$f((a, b)) :\equiv g\,(a)\,(b)\,.$$

Например, мы можем получить первую проекцию из $\sum$-типа:

$$\mathsf{pr}_1 : \left(\sum_{x:A} B(x)\right) \to A$$

используя определяющее уравнение

$$\mathsf{pr}_1((a, b)) :\equiv a\,.$$

Однако, поскольку типом второго компонента пары $(a, b) : \sum_{(x:A)} B(x)$ является $B(a)$, вторая проекция должна быть *зависимой* функцией, тип которой включает в себя первую проекцию:

$$\mathsf{pr}_2 : \prod_{p:\sum_{(x:A)} B(x)} B(\mathsf{pr}_1(p))\,.$$

Таким образом, нам нужен принцип *индукции* для $\sum$-типов («зависимый выделитель»). Это говорит о том, что для построения зависимой функции из $\sum$-типа в семейство $C : \left(\sum_{(x:A)} B(x)\right) \to \mathcal{U}$ нам нужна функция

$$g : \prod_{(a:A)} \prod_{(b:B(a))} C((a, b))\,.$$

Тогда мы можем получить функцию

$$f : \prod_{p:\sum_{(x:A)} B(x)} C(p)$$

с определяющим уравнением

$$f((a,b)) :\equiv g\,(a)\,(b)\,.$$

Применяя эту функцию с $C(p) :\equiv B(\mathsf{pr}_1(p))$, мы можем определить $\mathsf{pr}_2 : \prod_{(p:\sum_{(x:A)} B(x))} B(\mathsf{pr}_1(p))$ с очевидным уравнением

$$\mathsf{pr}_2((a,b)) :\equiv b\,.$$

Чтобы убедиться, что это верно, заметим, используя определяющее уравнение для pr_1, что $B(\mathsf{pr}_1((a,b))) \equiv B(a)$, и действительно, $b : B(a)$.

Мы можем сконцентрировать принципы рекурсии и индукции в рекурсоре для $\sum$:

$$\mathsf{rec}_{\sum_{(x:A)} B(x)} : \prod_{(C:\mathcal{U})} \left(\prod_{(x:A)} B(x) \to C \right) \to \left(\sum_{(x:A)} B(x) \right) \to C$$

с определяющим уравнением

$$\mathsf{rec}_{\sum_{(x:A)} B(x)}(C, g, (a,b)) :\equiv g\,(a)\,(b)$$

и в соответствующем операторе индукции:

$$\mathsf{ind}_{\sum_{(x:A)} B(x)} : \prod_{(C:(\sum_{(x:A)} B(x))\to\mathcal{U})} \left(\prod_{(a:A)} \prod_{(b:B(a))} C((a,b)) \right) \prod_{(p:\sum_{(x:A)} B(x))} C(p)$$

с определяющим уравнением

$$\mathsf{ind}_{\sum_{(x:A)} B(x)}(C, g, (a,b)) :\equiv g\,(a)\,(b)\,.$$

Как и выше, рекурсор является частным случаем индукции, когда семейство C является постоянным.

В качестве еще одного примера рассмотрим следующий принцип, где A и B являются типами и $R : A \to B \to \mathcal{U}$.

$$\mathsf{ac} : \left(\prod_{(x:A)} \sum_{(y:B)} R(x,y) \right) \to \left(\sum_{(f:A\to B)} \prod_{(x:A)} R(x, f(x)) \right)\,.$$

Мы можем рассматривать R как «релевантно-доказательное отношение» между A и B, причем $R(a,b)$ является типом свидетельств связанности $a : A$ и $b : B$. Тогда, интуитивно, ac утверждает, что если мы имеем зависимую функцию g, назначающей каждому $a : A$ зависимую пару (b, r), где $b : B$ и $r : R(a,b)$, то мы имеем функцию $f : A \to B$ и зависимую функцию, назначающей каждому $a : A$ свидетельство о том, что $R(a, f(a))$. Наша интуиция говорит, что можно просто расщепить значения g на компоненты. Действительно, используя только что определенные проекции, мы можем определить:

$$\mathsf{ac}(g) :\equiv \left(\lambda x.\, \mathsf{pr}_1(g(x)), \lambda x.\, \mathsf{pr}_2(g(x))\right)\,.$$

Чтобы убедиться, что все хорошо типизировано, заметим, что если $g : \prod_{(x:A)} \sum_{(y:B)} R(x,y)$, то имеем

$$\begin{aligned}
&\lambda x.\, \mathsf{pr}_1(g(x)) : A \to B\,,\\
&\lambda x.\, \mathsf{pr}_2(g(x)) : \prod_{(x:A)} R(a, \mathsf{pr}_1(g(x))).
\end{aligned}$$

Более того, тип $\prod_{(x:A)} R(a, \mathsf{pr}_1(g(x)))$ является результатом подстановки функции $\lambda x.\, \mathsf{pr}_1(g(x))$ для f в семействе, суммируемом в кообласти для ac:

$$\prod_{(x:A)} R(x, \mathsf{pr}_1(g(x))) \equiv \left(\lambda f.\, \prod_{(x:A)} R(x, \mathsf{pr}_1(f(x)))\right) (\lambda x.\, \mathsf{pr}_1(g(x)))\ .$$

Таким образом, имеем

$$(\lambda x.\, \mathsf{pr}_1(g(x)), \lambda x.\, \mathsf{pr}_2(g(x))) : \left(\sum_{(f:A\to B)} \prod_{(x:A)} R(x, f(x))\right)\ ,$$

что и требовалось.

Если мы читаем $\prod$ как «для всех» и $\sum$ как «существует», то тип функции ac выражает следующее: *если для всех $x : A$ существует некоторое $y : B$ такое, что имеет место $R(x, y)$, то существует функция $f : A \to B$ такая, что для всех $x : A$ верно $R(x, f(x))$*. Так как это звучит как вариант аксиомы выбора, то функцию ac традиционно называют **теоретико-типовой аксиомой выбора**, и, как мы только что показали, ее можно подтвердить непосредственным использованием правил теории типов, а не воспринимать как аксиому. Однако обратите внимание, что на самом деле это имитация выбора, поскольку выбор уже предоставлен нам в предпосылке: все, что нам нужно сделать, состоит в том, чтобы разделить зависимую функцию на две функции, одна из которых представляет выбор, а другая — правильность этого выбора. В §3.8 мы дадим еще одну формулировку «аксиомы выбора», которая ближе к обычной.

Типы зависимых пар часто используются для определения типов математических структур, которые обычно состоят из нескольких зависимых фрагментов данных. В качестве простого примера предположим, что мы хотим определить **магму** как тип A вместе с бинарной операцией $m : A \to A \to A$. Точный смысл фразы «вместе с» (и синонима «снабженной») заключается в том, что «магмой» является *пара* (A, m), состоящая из типа $A : \mathcal{U}$ и операции $m : A \to A \to A$. Так как тип $A \to A \to A$ второй составляющей m этой пары зависит от ее первой компоненты A, то такие пары относятся к зависимому парному типу. Таким образом, определение «магма является типом A вместе с бинарной операцией $m : A \to A \to A$» следует интерпретировать как *тип магм*, представляемый следующим образом:

$$\mathsf{Magma} :\equiv \sum_{A:\mathcal{U}} (A \to A \to A)\,.$$

Для заданной магмы, мы извлекаем лежащий в ее основе тип (ее «несущую»), с помощью первой проекции pr_1, и ее действие с использованием второй проекции pr_2. Конечно, структуры, построенные из более чем двух данных, требуют дополнительных типов пар, которые могут быть частично зависимыми, например, тип точечных магм (магмы (A, m), снабженные отмеченной точкой $e : A$),

$$\mathsf{PointedMagma} :\equiv \sum_{A:\mathcal{U}} (A \to A \to A) \times A\,.$$

Как правило, у нас появляется необходимость дополнить аксиомами такую структуру, например, для встраивания точечных магм в моноид или группу. Это также можно сделать с использованием $\sum$-типов; см. §1.11.

В оставшейся части книги мы иногда будем делать определения такого рода явными, но в конечном итоге мы доверяем читателю перевести их с естественного языка на язык $\sum$-типов. Мы также обычно будем следовать общепринятой математической практике использования одного и того же символа для подобной структуры и ее носителя (что сводится к тому, что в обозначении подразумевается соответствующая функция проекции): то есть мы будем говорить о магме A с операцией $m : A \to A \to A$.

Заметим, что канонические элементы $\mathsf{PointedMagma}$ имеют вид $(A, (m, e))$, где $A : \mathcal{U}$, $m : A \to A \to A$ и $e : A$. Из-за частоты, с которой повторяются $\sum$-типы такого рода, мы будем использовать обычное обозначение упорядоченных троек, четверок и т.д., для вложенных вправо пар (возможно зависимых). То есть мы имеем $(x, y, z) :\equiv (x, (y, z))$, $(x, y, z, w) :\equiv (x, (y, (z, w)))$ и т.д.

1.7 Типы копроизведений

Для $A, B : \mathcal{U}$ введем тип их **копроизведения** $A + B : \mathcal{U}$. Оно соответствует *несвязному объединению* в теории множеств и мы также можем использовать этот термин. В теории типов, как и в случае с функциями и произведениями, копроизведение должно быть фундаментальной конструкцией, поскольку ранее не было введено понятия «объединение типов». Мы также вводим нульарную версию: **пустой тип** $\mathbf{0} : \mathcal{U}$.

Существуют два способа построения элементов $A + B$, либо как $\mathsf{inl}(a) : A + B$ для $a : A$, или как $\mathsf{inr}(b) : A + B$ для $b : B$. Нет способов построить элементы пустого типа.

Для построения обычной функции $f : A + B \to C$, нам требуются функции $g_0 : A \to C$ и $g_1 : B \to C$. Тогда f определяется через определяющие уравнения

$$f(\mathsf{inl}(a)) :\equiv g_0(a)\,,$$
$$f(\mathsf{inr}(b)) :\equiv g_1(b)\,.$$

То есть функция f определяется **анализом случаев**. Как и выше, мы можем получить рекурсор:

$$\mathsf{rec}_{A+B} : \prod_{(C:\mathcal{U})} (A \to C) \to (B \to C) \to A + B \to C$$

с определяющими уравнениями

$$\mathsf{rec}_{A+B}(C, g_0, g_1, \mathsf{inl}(a)) :\equiv g_0(a)\,,$$
$$\mathsf{rec}_{A+B}(C, g_0, g_1, \mathsf{inr}(b)) :\equiv g_1(b)\,.$$

Мы всегда можем построить функцию $f : \mathbf{0} \to C$, не требуя каких-либо определяющих уравнений, потому что нет элементов из $\mathbf{0}$, на которых можно определить f. Таким образом, рекурсор для для $\mathbf{0}$ есть

$$\mathsf{rec}_{\mathbf{0}} : \prod_{(C:\mathcal{U})} \mathbf{0} \to C\,,$$

который конструирует каноническую функцию от пустого типа к любому другому типу. Логически это соответствует принципу *ex falso quodlibet* (из лжи следует все).

Чтобы построить зависимую функцию $f : \prod_{(x:A+B)} C(x)$ из копроизведения, предполагаем, в качестве заданного семейства, $C : (A + B) \to \mathcal{U}$ и требуем

$$g_0 : \prod_{a:A} C(\mathsf{inl}(a))\,,$$
$$g_1 : \prod_{b:B} C(\mathsf{inr}(b))\,.$$

Это задает f с определяющими уравнениями:

$$f(\mathsf{inl}(a)) :\equiv g_0(a)\,,$$
$$f(\mathsf{inr}(b)) :\equiv g_1(b)\,.$$

Мы упаковываем эту схему в принцип индукции для копроизведений:

$$\mathsf{ind}_{A+B} : \prod_{(C:(A+B)\to\mathcal{U})} \left(\prod_{(a:A)} C(\mathsf{inl}(a))\right) \to \left(\prod_{(b:B)} C(\mathsf{inr}(b))\right) \to \prod_{(x:A+B)} C(x)\,.$$

Как и ранее, рекурсор возникает в случае, когда семейство C является постоянным.

Принцип индукции для пустого типа

$$\mathsf{ind}_{\mathbf{0}} : \prod_{(C:\mathbf{0}\to\mathcal{U})} \prod_{(z:\mathbf{0})} C(z)$$

дает нам способ определить тривиальную зависимую функцию из пустого типа.

1.8 Тип булевых значений

Тип булевых элементов $\mathbf{2} : \mathcal{U}$ должен иметь ровно два элемента $1_{\mathbf{2}}, 0_{\mathbf{2}} : \mathbf{2}$. Ясно, что мы могли бы построить этот тип из типов копроизведений и единиц как $\mathbf{1} + \mathbf{1}$. Однако, поскольку он используется часто, здесь приводятся явные правила. Действительно, мы будем свидетелями, что можно пойти другим путем и получить бинарные копроизведения из $\sum$-типов и $\mathbf{2}$.

Для вывода функции $f : \mathbf{2} \to C$ нам нужны $c_0, c_1 : C$ и необходимо добавить определяющие уравнения

$$f(0_{\mathbf{2}}) :\equiv c_0\,, \qquad f(1_{\mathbf{2}}) :\equiv c_1\,.$$

Рекурсор соответствует конструкции if-then-else в функциональном программировании:

$$\mathsf{rec}_{\mathbf{2}} : \prod_{C:\mathcal{U}} C \to C \to \mathbf{2} \to C$$

с определяющими уравнениями

$$\mathsf{rec}_{\mathbf{2}}(C, c_0, c_1, 0_{\mathbf{2}}) :\equiv c_0\,,$$
$$\mathsf{rec}_{\mathbf{2}}(C, c_0, c_1, 1_{\mathbf{2}}) :\equiv c_1\,,$$

С учетом $C : \mathbf{2} \to \mathcal{U}$, для получения зависимой функции $f : \prod_{(x:\mathbf{2})} C(x)$ нам понадобятся $c_0 : C(0_{\mathbf{2}})$ и $c_1 : C(1_{\mathbf{2}})$, и в этом случае мы можем предоставить определяющие уравнения

$$f(0_{\mathbf{2}}) :\equiv c_0\,,$$
$$f(1_{\mathbf{2}}) :\equiv c_1\,.$$

Мы вкладываем это в принцип индукции

$$\mathsf{ind}_{\mathbf{2}} : \prod_{(C:\mathbf{2}\to\mathcal{U})} C(0_{\mathbf{2}}) \to C(1_{\mathbf{2}}) \to \prod_{(x:\mathbf{2})} C(x)$$

с определяющими уравнениями

$$\mathsf{ind}_{\mathbf{2}}(C, c_0, c_1, 0_{\mathbf{2}}) :\equiv c_0\,,$$
$$\mathsf{ind}_{\mathbf{2}}(C, c_0, c_1, 1_{\mathbf{2}}) :\equiv c_1\,,$$

В качестве примера, используя принцип индукции, мы можем доказать, что, как мы ожидаем, каждый элемент из $\mathbf{2}$ равен либо $1_{\mathbf{2}}$, либо $0_{\mathbf{2}}$. Как и ранее, мы используем типы равенств, которые мы еще не вводили, но нам нужен только тот факт, что все равно самому себе, согласно $\mathsf{refl}_x : x = x$. Таким образом, мы строим элемент из

$$\prod_{x:\mathbf{2}} (x = 0_{\mathbf{2}}) + (x = 1_{\mathbf{2}}) \tag{1.8.1}$$

т.е. функцию, присваивающую каждому $x : \mathbf{2}$ либо равенство $x = 0_{\mathbf{2}}$, либо равенство $x = 1_{\mathbf{2}}$. Определим этот элемент, используя принцип индукции для $\mathbf{2}$, с $C(x) :\equiv (x = 0_{\mathbf{2}}) + (x = 1_{\mathbf{2}})$; двумя входами являются $\mathsf{inl}(\mathsf{refl}_{0_{\mathbf{2}}}) : C(0_{\mathbf{2}})$ и $\mathsf{inr}(\mathsf{refl}_{1_{\mathbf{2}}}) : C(1_{\mathbf{2}})$. Другими словами, наш элемент из (1.8.1) есть

$$\mathsf{ind}_{\mathbf{2}}\left(\lambda x.\,(x = 0_{\mathbf{2}}) + (x = 1_{\mathbf{2}}), \mathsf{inl}(\mathsf{refl}_{0_{\mathbf{2}}}), \mathsf{inr}(\mathsf{refl}_{1_{\mathbf{2}}})\right)\,.$$

Мы отметили, что $\sum$-типы можно рассматривать как аналогичные индексированным дизъюнктным объединениям, а копроизведения — бинарным несвязным объединениям. Естественно ожидать, что бинарное несвязное объединение $A + B$ может быть построено как индексированное по двухэлементному типу $\mathbf{2}$. Для этого нам понадобится *семейство типов* $P : \mathbf{2} \to \mathcal{U}$ такое, что $P(0_{\mathbf{2}}) = A$ и $\mathrm{P}(1_{\mathbf{2}}) = B$. Действительно, мы можем получить такое семейство именно по принципу рекурсии для $\mathbf{2}$ (возможность определять типы семейств по индукции и рекурсии, используя тот факт, что универсум $\mathcal{U}$ сам по себе является типом, является тонким и важным аспектом теории типов). Таким образом, мы могли бы определить

$$A + B :\equiv \sum_{x:\mathbf{2}} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)\,.$$

с

$$\mathsf{inl}(a) :\equiv (0_{\mathbf{2}}, a)\,,$$
$$\mathsf{inr}(b) :\equiv (1_{\mathbf{2}}, b)\,.$$

Мы оставляем, как упражнение, вывод индукционного принципа типа копроизведения из этого определения (см. также упражнение 1.5 и §5.2).

Мы можем применить ту же идею к произведениям и $\prod$-типам: мы могли бы определить

$$A \times B :\equiv \prod_{x:\mathbf{2}} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x) .$$

Затем пары могли бы быть построены с использованием индукции для $\mathbf{2}$:

$$(a, b) :\equiv \mathsf{ind}_{\mathbf{2}}(\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B), a, b)$$

в то время как проекции — это простые объявления

$$\mathsf{pr}_1(p) :\equiv p(0_{\mathbf{2}}) ,$$
$$\mathsf{pr}_2(p) :\equiv p(1_{\mathbf{2}}) .$$

Вывод принципа индукции для бинарных произведений, определенных таким образом, является более сложным и требует функциональной многогранности, которую мы введем в §2.9. Более того, мы не получаем дефинициальных суждений, см. упражнение 1.6. Это проблема, повторяющаяся каждый раз при кодировании одного типа другим, мы вернемся к этому в §5.5.

Мы можем иногда ссылаться на элементы $1_{\mathbf{2}}$ и $0_{\mathbf{2}}$ из $\mathbf{2}$ как «истинные» и «ложные», соответственно. Однако обратите внимание, что в отличие от классической математики, мы не используем элементы $\mathbf{2}$ как истинностные значения или высказывания (вместо этого мы определяем высказывания с типами, см. §1.11). В частности, тип $A \to \mathbf{2}$ обычно не является степенным множеством A, он представляет собой только «разрешенные» подмножества A (см. главу 3).

1.9 Натуральные числа

Пока что, мы имеем правила построения новых типов с помощью абстрактных операций, но для создания конкретной математики также требуются некоторые конкретные типы, такие как типы чисел. Основополагающим из них является тип натуральных чисел $\mathbb{N} : \mathcal{U}$; при его наличии можно создавать целые, рациональные, действительные числа и так далее (см. главу 11).

Элементы $\mathbb{N}$ строятся с использованием $0 : \mathbb{N}$ и операции следования $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$. Для обозначения натуральных чисел мы используем обычную десятичную нотацию:

$$1 :\equiv \mathsf{succ}(0),\ 2 :\equiv \mathsf{succ}(1),\ 3 :\equiv \mathsf{succ}(2),\ \dots .$$

Существенное свойство натуральных чисел состоит в том, что мы можем определять функции рекурсией и выполнять доказательства по индукции, где теперь слова «рекурсия» и «индукция» имеют более знакомый смысл. Для построения обычной функции $f : \mathbb{N} \to C$ из натуральных чисел путем рекурсии, достаточно указать начальную точку $c_0 : C$ и функцию «следующего шага» $c_s : N \to C \to C$. Это приводит к f с определяющими уравнениями

$$f(0) :\equiv c_0 ,$$
$$f(\mathsf{succ}(n)) :\equiv c_s(n, f(n)) .$$

Мы говорим, что f определяется **примитивной рекурсией**.

В качестве примера рассмотрим, как определить функцию на натуральных числах, которая удваивает свой аргумент. В этом случае имеем $C :\equiv \mathbb{N}$. Сначала нужно указать значение для

$\mathsf{double}(0)$, что легко: положим $c_0 :\equiv 0$. Затем, чтобы вычислить значение $\mathsf{double}(\mathsf{succ}(n))$ для натурального числа n, сначала вычислим значение $\mathsf{double}(n)$, а затем дважды выполним операцию следования. Это фиксируется рекурентностью $c_s(n, y) :\equiv \mathsf{succ}(\mathsf{succ}(y))$. Обратите внимание, что второй аргумент y в c_s означает результат *рекурсивного вызова* $\mathsf{double}(n)$.

Таким образом, определяя $\mathsf{double} : \mathbb{N} \to \mathbb{N}$ примитивной рекурсией, мы получаем определяющие уравнения:

$$\begin{aligned}\mathsf{double}(0) &:\equiv 0\\ \mathsf{double}(\mathsf{succ}(n)) &:\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{double}(n)))\,.\end{aligned}$$

И действительно имеет место правильное вычислительное поведение: например, мы имеем

$$\begin{aligned}\mathsf{double}(2) &\equiv \mathsf{double}(\mathsf{succ}(\mathsf{succ}(0)))\\ &\equiv c_s(\mathsf{succ}(0), \mathsf{double}(\mathsf{succ}(0)))\\ &\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{double}(\mathsf{succ}(0))))\\ &\equiv \mathsf{succ}(\mathsf{succ}(c_s(0, \mathsf{double}(0))))\\ &\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{double}(0)))))\\ &\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{succ}(c_0))))\\ &\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{succ}(\mathsf{succ}(0))))\\ &\equiv 4\,.\end{aligned}$$

Мы можем также определять примитивной рекурсией функции нескольких переменных, используя каррирование и допуская, что C может быть типом функции. Например, мы так определяем сложение $\mathsf{add} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$ с $C :\equiv \mathbb{N} \to \mathbb{N}$, «начальную точку» и функцию «следующего шага»:

$$\begin{aligned}c_0 &: \mathbb{N} \to \mathbb{N}\\ c_0(n) &:\equiv n\\ c_s &: \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})\\ c_s(m, g)(n) &:\equiv \mathsf{succ}(g(n))\,.\end{aligned}$$

Таким образом, получаем $\mathsf{add} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$, удовлетворяющие определяющим уравнениям

$$\begin{aligned}\mathsf{add}(0, n) &\equiv n\\ \mathsf{add}(\mathsf{succ}(m), n) &\equiv \mathsf{succ}(\mathsf{add}(m, n))\,.\end{aligned}$$

Как обычно, мы записываем $\mathsf{add}(m, n)$ как $m+n$. Читателю предлагается проверить, что $2+2 \equiv 4$.

Как и в предыдущих случаях, мы можем упаковать принцип примитивной рекурсии в рекурсор:

$$\mathsf{rec}_{\mathbb{N}} : \prod_{(C:\mathcal{U})} C \to (\mathbb{N} \to C \to C) \to \mathbb{N} \to C$$

с определяющими уравнениями

$$\begin{aligned}\mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, 0) &:\equiv c_0\,,\\ \mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, \mathsf{succ}(n)) &:\equiv c_s(n, \mathsf{rec}_{\mathbb{N}}(C, c_0, c_s, n))\,.\end{aligned}$$

С помощью $\mathsf{rec}_{\mathbb{N}}$, мы можем представить double и add следующим образом:

$$\mathsf{double} :\equiv \mathsf{rec}_{\mathbb{N}}\left(\mathbb{N}, 0, \lambda n.\, \lambda y.\, \mathsf{succ}(\mathsf{succ}(y))\right) \tag{1.9.1}$$

$$\mathsf{add} :\equiv \mathsf{rec}_{\mathbb{N}}\left(\mathbb{N} \to \mathbb{N}, \lambda n.\, n, \lambda n.\, \lambda g.\, \lambda m.\, \mathsf{succ}(g(m))\right). \tag{1.9.2}$$

Конечно, все функции, определяемые только с использованием принципа примитивной рекурсии, будут *вычислимы* (наличие типов функций высших порядков, то есть функций с другими функциями в качестве аргументов, означает, однако, что мы можем определить более широкий класс функций, чем класс обычных примитивно-рекурсивных функций, см., например, упражнение 1.10). Это уместно в конструктивной математике; в §§ 3.4 и 3.8 мы увидим, как расширить теорию типов, чтобы можно было определять более общие математические функции.

Теперь будем придерживаться того же подхода, что и для других типов, обобщая примитивную рекурсию на зависимые функции для получения *принципа индукции*. Таким образом, предположим, что задано семейство $C : \mathbb{N} \to \mathcal{U}$, элемент $c_0 : C(0)$ и функция$c_s : \prod_{(n:\mathbb{N})} C(n) \to$ C$(\mathsf{succ}(n))$, тогда мы можем построить $f : \prod_{(n:\mathbb{N})} C(n)$ с определяющими уравнениями:

$$f(0) :\equiv c_0\,,$$
$$f(\mathsf{succ}(n)) :\equiv c_s(n, f(n))\,.$$

Мы можем, также, упаковать их в одну функцию

$$\mathsf{ind}_{\mathbb{N}} : \prod_{(C:\mathbb{N}\to\mathcal{U})} C(0) \to \left(\prod_{(n:\mathbb{N})} C(n) \to C(\mathsf{succ}(n))\right) \to \prod_{(n:\mathbb{N})} C(n)$$

с определяющими уравнениями

$$\mathsf{ind}_{\mathbb{N}}(C, c_0, c_s, 0) :\equiv c_0\,,$$
$$\mathsf{ind}_{\mathbb{N}}(C, c_0, c_s, \mathsf{succ}(n)) :\equiv c_s(n, \mathsf{ind}_{\mathbb{N}}(C, c_0, c_s, n))\,.$$

Здесь мы наконец видим связь с классическим понятием доказательства по индукции. Напомним, что в теории типов мы представляем высказывания согласно типам и доказываем суждение путем установления обитаемости соответствующего типа. В частности, *свойство* натуральных чисел представлено семейством типов $P : \mathbb{N} \to \mathcal{U}$. С этой точки зрения вышеприведенный принцип индукции говорит, что если мы можем доказать $P(0)$, и если для любого n можно доказать $P(\mathsf{succ}(n))$, предполагая $P(n)$, то $P(n)$ имеет место для всех n. Это, конечно же, именно обычный принцип доказательства индукцией для натуральных чисел.

В качестве примера рассмотрим, как можно представить явное доказательство того, что «+» ассоциативна (мы фактически не будем выписывать доказательства в этом стиле, но это послужит полезным примером для понимания того, как индукция представляется формально в теории типов). Чтобы получить

$$\mathsf{assoc} : \prod_{i,j,k:\mathbb{N}} i + (j + k) = (i + j) + k\,,$$

достаточно предоставить

$$\mathsf{assoc}_0 : \prod_{j,k:\mathbb{N}} 0 + (j + k) = (0 + j) + k$$

и

$$\mathsf{assoc}_s : \prod_{i:\mathbb{N}} \left(\prod_{j,k:\mathbb{N}} i + (j+k) = (i+j)+k \right) \to \prod_{j,k:\mathbb{N}} \mathsf{succ}(i) + (j+k) = (\mathsf{succ}(i)+j)+k\,.$$

Для вывода assoc_0 напомним, что $0 + n \equiv n$, и, следовательно, $0 + (j+k) \equiv j+k \equiv (0+j)+k$. Таким образом, можно просто установить

$$\mathsf{assoc}_0(j,k) :\equiv \mathsf{refl}_{j+k}\,.$$

Для assoc_s напомним, что определение «+» дает$\mathsf{succ}(m) + n \equiv \mathsf{succ}(m+n)$ и, следовательно,

$$\begin{aligned} \mathsf{succ}(i) + (j+k) &\equiv \mathsf{succ}(i + (j+k)) \qquad \text{и} \\ (\mathsf{succ}(i) + j) + k &\equiv \mathsf{succ}((i+j)+k)\,. \end{aligned}$$

Таким образом, получающийся тип assoc_s эквивалентен $\mathsf{succ}(i+(j+k)) = \mathsf{succ}((i+j)+k)$. Но его входное значение («индуктивная гипотеза») дает $i+(j+k) = (i+j)+k$, поэтому достаточно сослаться на то, что если два натуральных числа равны, то и их следующие соответствующие значения также равны (мы докажем этот очевидный факт в лемме 2.2.1, используя принцип индукции тождественных типов). Обозначим этот последний факт как $\mathsf{ap}_{\mathsf{succc}} : (m =_N n) \to (\mathsf{succ}(m) =_N \mathsf{succ}(n))$, так что можно определить

$$\mathsf{assoc}_s(i,h,j,k) :\equiv \mathsf{ap}_{\mathsf{succ}}(h(j,k))\,.$$

Объединяя все с $\mathsf{ind}_{\mathbb{N}}$, мы получаем доказательство ассоциативности операции «+».

1.10 Сопоставление с образцом и рекурсия

Натуральные числа демонстрируют дополнительное тонкое отличие от типов, рассмотренных до сих пор. В случае копроизведений, например, мы могли бы определить функцию $f : A+B \to C$, либо с помощью рекурсора:

$$f :\equiv \mathsf{rec}_{A+B}(C, g_0, g_1)$$

или определяющими уравнениями:

$$\begin{aligned} f(\mathsf{inl}(a)) &:\equiv g_0(a) \\ f(\mathsf{inr}(b)) &:\equiv g_1(b)\,. \end{aligned}$$

Чтобы перейти от одного выражения f к последующему, мы просто используем правила вычисления для рекурсора. Обратно, любые определяющие уравнения

$$\begin{aligned} f(\mathsf{inl}(a)) &:\equiv \Phi_0 \\ f(\mathsf{inr}(b)) &:\equiv \Phi_1\,, \end{aligned}$$

где Φ_0 и Φ_1 — выражения, которые могут включать переменные a и b, соответственно, можно выразить эквивалентно в терминах рекурсора с использованием λ-абстракции:

$$f :\equiv \mathsf{rec}_{A+B}(C, \lambda a.\, \Phi_0, \lambda b.\, \Phi_1)\,.$$

Однако, в случае натуральных чисел, «определяющие уравнения» такой функции, как double:

$$\mathsf{double}(0) :\equiv 0 \tag{1.10.1}$$
$$\mathsf{double}(\mathsf{succ}(n)) :\equiv \mathsf{succ}(\mathsf{succ}(\mathsf{double}(n))) \tag{1.10.2}$$

включают *саму функцию* double в правой части. Однако, мы все равно хотели бы использовать эти уравнения, а не (1.9.1), в качестве определения double, так как они гораздо удобнее и читабельнее. Решение состоит в том, чтобы толковать выражение «$\mathsf{double}(n)$» в правой части (1.10.2), как стоящее за результатом рекурсивного вызова, которое в определении формы $\mathsf{double} :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}, c_0, c_s)$ будет вторым аргументом для c_s.

В более общем плане, если у нас есть «определение» функции $f : \mathbb{N} \to C$, такой как

$$f(0) :\equiv \Phi_0$$
$$f(\mathsf{succ}(n)) :\equiv \Phi_s$$

где Φ_0 — выражение типа C, а Φ_s — выражение типа C, которое может включать переменную n, а также символ «$f(n)$», мы можем перевести ее в определение

$$f :\equiv \mathsf{rec}_{\mathbb{N}}(C, \Phi_0, \lambda n.\, \lambda r.\, \Phi_s')$$

где Φ_s' получается из Φ_s заменой всех вхождений «$f(n)$» новой переменной r.

Этот стиль определения функций рекурсией (или, в более общем смысле, зависимых функций по индукции) настолько удобен, что мы часто используем его. Он называется определением путем **сопоставления с образцом**. Конечно, это очень похоже на то, как программист может определить рекурсивную функцию с телом, которое, буквально, содержит рекурсивные вызовы самого себя. Однако, в отличие от программиста, мы ограничены в том, какие рекурсивные вызовы мы можем сделать: для того, чтобы такое определение было повторно выражено с использованием принципа рекурсии, определяемая функция f может появляться только в теле $f(\mathsf{succ}(\mathbb{N}))$ как часть составного символа «$f(n)$». В противном случае мы могли бы записывать бессмысленные функции, такие как

$$f(0) :\equiv 0$$
$$f(\mathsf{succ}(n)) :\equiv f(\mathsf{succ}(\mathsf{succ}(n)))\,.$$

Если бы программист написал такую функцию, она просто постоянно вызывала бы себя при любом исходном положительном значении, перейдя в бесконечный цикл, никогда не возвращая результат. Однако в математике, чтобы соответствовать своему определению, функция всегда должна связывать уникальное значение результата с каждым входным значением, поэтому представленная ситуация была бы неприемлема.

Этот момент будет еще более важным, когда мы введем более сложные индуктивные типы в главах 5, 6 и 11. Всякий раз, когда мы вводим новый вид индуктивного определения, мы всегда начинаем с вывода его принципа индукции. Только после этого мы вводим подходящий тип «сопоставления с образцом», который может быть оправдан в качестве сокращения для принципа индукции.

1.11 Высказывания как типы

Как упоминалось во введении, истинность высказывания в теории типов соответствует предъявлению элемента типа, соответствующего этому высказыванию. Мы рассматриваем элементы этого типа в качестве *признаков* или *свидетельств* того, что высказывание истинно (иногда их даже называют *доказательствами*, но эта терминология может вводить в заблуждение, поэтому мы обычно избегаем ее). В общем, однако, мы не будем строить свидетельства явно, вместо этого мы будем приводить доказательства в обычном математическом стиле, чтобы они могли быть переведены в элемент типа. Это ничем не отличается от рассуждений в классической теории множеств, где мы не ожидаем увидеть явный вывод с использованием правил предикатной логики и аксиом теории множеств.

Однако, теоретико-множественная перспективы относительно доказательств, тем не менее, различаются существенным образом. Основной принцип логики теории типов состоит в том, что высказывание не просто истинно или ложно, а скорее может рассматриваться как совокупность всех возможных свидетельств его истины. Согласно этой концепции, доказательства являются не только средством выражения математики, но скорее являются математическими объектами сами по себе, наравне с более знакомыми объектами, такими как числа, отображения, группы и т.д. Таким образом, поскольку типы классифицируют доступные математические объекты и определяют, как они взаимодействуют, высказывания — это ничто иное, как специальные типы, а именно типы, элементы которых являются доказательствами.

Основное наблюдение, которое делает эту идентификацию выполнимой, состоит в том, что мы имеем следующее естественное соответствие между *логическими* операциями над высказываниями, выраженными на естественном языке, и *теоретико-типовыми* операциями над их соответствующими типами свидетельств.

Естественный язык	Теория типов
истина	$\mathbf{1}$
ложь	$\mathbf{0}$
A и B	$A \times B$
A или B	$A + B$
если A, то B	$A \to B$
A тогда и только тогда, когда B	$(A \to B) \times (B \to A)$
не A	$A \to \mathbf{0}$

Суть соответствия в том, что в каждом случае правила построения и использования элементов типа (в правом столбце таблицы) соответствуют правилам рассуждения о высказывании (левый столбец). Например, основным способом доказать высказывание вида «A и B» является доказательство A, а также доказательство B, а основной способ построения элемента из $A \times B$ заключается в предоставлении пары (a, b), где a является элементом (или свидетельством) A и b является элементом (или свидетельством) B. А если мы хотим использовать «A и B» для доказательтва чего-то еще, мы можем использовать при этом и A и B, аналогично тому, как принцип индукции для $A \times B$ позволяет нам построить функцию, используя элементы A и B.

Аналогичным образом, основной способ доказать импликацию «если A, то B» заключается в допущении A и доказательстве B, а основной способ построения элемента $A \to B$ — предоставление выражения, которое обозначает элемент (свидетельство) из B, который может включать неопределенный переменный элемент (свидетельство) типа A. Основной способ использования импликации «если A, то B» выводит B, если известно A, аналогично тому, как мы можем применить функцию $f : A \to B$ к элементу из A, чтобы получить элемент из B. Мы настоятельно рекомендуем читателю выполнить проверку того, что правила, управляющие конструкторами других типов, разумно переводятся в логику.

Особо следует отметить, что пустой тип $\mathbf{0}$ соответствует ложности. Говоря логически, мы относимся к обитатаемости $\mathbf{0}$ как к **противоречию**: при этом не существует никакого способа доказать противоречие[9], в то время как из противоречия можно вывести что угодно. Мы также определяем **отрицание** типа A как

$$\neg A :\equiv A \to \mathbf{0}\,.$$

Таким образом, свидетельством $\neg A$ является функция $A \to \mathbf{0}$, которую мы можем построить, предположив $x : A$ и получив элемент из $\mathbf{0}$. Заметим, что хотя логика, которую мы используем, является «конструктивной», как обсуждалось во введении, это «доказательство противоречия» (предполагаем, что A и выводим противоречие, заключающееся в $\neg A$) совершенно корректно конструктивно: оно просто вскрывает смысл «отрицания». Противоречивость, которая запрещена, заключается в том, чтобы предположить $\neg A$ и получить противоречие как способ доказательства A. Конструктивно такой аргумент позволил бы вывести $\neg\neg A$, и читатель может проверить, что не существует очевидного способа получить A из $\neg\neg A$ (то есть из $(A \to \mathbf{0}) \to \mathbf{0}$).

Вышеприведенный перевод логических связок в типо-формирующие операции оправдывает использование **высказываний как типов**: он дает нам способ выражения высказываний и их доказательств, записанных на естественном языке, типами и их элементами. Например, предположим, что мы хотим доказать следующую тавтологию (один из «законов де Моргана»):

$$\text{"Если не } A \text{ и не } B, \text{ то не } (A \text{ или } B)\text{".} \tag{1.11.1}$$

Обычное «словесное» доказательство этого факта может быть следующим.

> Предположим, что «не A», и «не B», а также пусть «A или B», получаем противоречие. Далее, существуют два случая. Если A выполняется, то, так как «не A», — противоречие. Аналогично, если B имеет место, то, так как «не B», — противоречие. Таким образом, противоречие имеется в любом случае, поэтому «не (A или B)».

Тип, соответствующий нашей тавтологии (1.11.1), согласно приведенным выше правилам, есть

$$(A \to \mathbf{0}) \times (B \to \mathbf{0}) \to (A + B \to \mathbf{0}) \tag{1.11.2}$$

поэтому мы имеем возможность перевести указанное выше доказательство в элемент этого типа.

В качестве примера того, как работает такой перевод, давайте опишем, как математик, читающий вышеприведенное «словесное» доказательство, мог бы одновременно строить в своей голове элемент (1.11.2). Вводная фраза «Предположим, что "не A", а "не B"» переводится в определение функции с неявным применением принципа рекурсии для декартова произведения в своей области $(A \to \mathbf{0}) \times (B \to \mathbf{0})$. При этом вводятся неименованные переменные (гипотезы) x и y типов $A \to \mathbf{0}$ и $B \to \mathbf{0}$. При переводе в теорию типов мы должны назначить имена этим переменным; назовем их x и y. На этом этапе наше частичное определение элемента из (1.11.2) можно записать в виде

$$f((x, y)) :\equiv \square : A + B \to \mathbf{0}$$

с «пусто́той» $\square$ типа $A + B \to \mathbf{0}$, указывающей на то, что еще что-то предстоит сделать (можно эквивалентно записать $f :\equiv \mathsf{rec}_{(A\to\mathbf{0})\times(B\to\mathbf{0})}(A+B \to \mathbf{0}, \lambda x.\,\lambda y.\,\square)$, используя рекурсор вместо сопоставления с образцом). Следующая фраза «а также пусть "A или B", получаем противоречие»

[9] Точнее, нет основополагающего способа доказать противоречие, т.е. $\mathbf{0}$ не имеет конструкторов. Если бы наша теория типов была противоречива, то возникли бы сложности при построении элемента из $\mathbf{0}$.

указывает на заполнение этой пусто́ты определением функции, вводя другую неименованную гипотезу $z : A + B$, приводящую к доказательству:

$$f((x, y)) :\equiv \square : \mathbf{0}\,.$$

Теперь, говоря «существуют два случая», обозначается разделение на случаи, т.е. применение принципа рекурсии для копроизведения $A + B$. Если мы запишем это с использованием рекурсора, то получим

$$f((x, y))(z) :\equiv \mathsf{rec}_{A+B}(\mathbf{0}, \lambda a.\,\square, \lambda b.\,\square, z)$$

в то время как, если мы запишем то же самое с помощью сопоставления с образцом, это будет

$$f((x, y))(\mathsf{inl}(a)) :\equiv \square : \mathbf{0}$$
$$f((x, y))(\mathsf{inr}(b)) :\equiv \square : \mathbf{0}\,.$$

Заметим, что в обоих случаях у нас теперь есть две «пусто́ты» типа $\mathbf{0}$ для заполнения, соответствующие двум случаям, когда мы должны получить противоречие. Наконец, заключение противоречия от $a : A$ и $x : A \to \mathbf{0}$ — простое применение функции x к a, и аналогично в другом случае (заметим удобное совпадение фразы «применение функции» с фразой «применение гипотезы», или теоремы). Таким образом, наше возможное определение есть

$$f((x, y))(\mathsf{inl}(a)) :\equiv x(a)$$
$$f((x, y))(\mathsf{inr}(b)) :\equiv y(b)\,.$$

В качестве упражнения вы должны проверить обратную тавтологию «Если "не (A или B)", то "не A" и "не B"», используя элемент

$$((A + B) \to \mathbf{0}) \to (A \to 0) \times (B \to 0)\,,$$

для любых типов A и B, используя только что введенные правила.

Однако, такой интерпретации придерживаются не все классические тавтологии. Например, правило «Если "не (A и B)", то "не A" или "не B"» недействительно: мы, вообще говоря, не сможем построить элемент соответствующего типа

$$((A \times B) \to \mathbf{0}) \to (A \to 0) + (B \to 0)\,.$$

Это отражает тот факт, что «естественная» логика высказываний как типов теории типов является *конструктивной*. Это означает, что она не включает в себя некоторые классические принципы, такие как закон исключенного третьего (the Law of Excluded Middle, LEM), или доказательство от противного, и другие, которые зависят от них, например, приведенный закон де Моргана.

С философской точки зрения, конструктивная логика так называется потому, что она ограничивается конструкциями, которые могут быть выполнены *эффективно*, то есть с вычислением значения. Не будучи слишком четким утверждением, это означает, что есть какой-то алгоритм, определяющий, шаг за шагом, как построить объект (и, в частности, увидеть, что теорема верна). Это требует игнорирования LEM, поскольку не существует *эффективной* процедуры для определения того, является ли высказывание истинным или ложным.

Конструктивность теоретико-типовой логики означает, что она имеет внутренний вычислительный смысл, который представляет интерес для ученых, работающих в области компьютинга. Это также означает, что теория типов обеспечивает *аксиоматическую свободу*. Например, хотя по умолчанию нет конструкции, свидетельствующей о LEM, логика все еще совместима с существованием LEM (см. §3.4). Таким образом, поскольку теория типов не отрицает LEM, мы можем согласовано добавлять этот закон в качестве предположения и работать традиционно без ограничений. В этом отношении теория типов обогащает, а не ограничивает, обычную математическую практику.

Мы призываем читателя, который не знаком с конструктивной логикой, проработать еще несколько примеров в качестве ознакомительного средства (в этой связи см. упражнения 1.12 и 1.13).

До сих пор мы обсуждали только пропозициональную логику. Теперь мы рассматрим *предикатную* логику, в которой помимо логических связок, таких как «и» и «или» у нас есть кванторы «существует» и «для всех». В этом случае типы играют двойную роль: они служат в качестве высказываний, а также как типы в обычном смысле, т.е. как области, которые мы количественно оцениваем. Предикат над типом A представляется как семейство $P : A \to \mathcal{U}$, присваивая каждому элементу $a : A$ тип $P(a)$, соответствующий высказыванию, что P выполняется для a. Теперь мы расширим приведенное выше соответствие объяснением кванторов:

Естественный язык	Теория типов
Для всех $x : A$ выполнено $P(x)$	$\prod_{(x:A)} P(x)$
Существует $x : A$ такое, что $P(x)$	$\sum_{(x:A)} P(x)$

Как и ранее, мы можем показать, что тавтология (конструктивной) предикатной логики переводится в обитаемые типы. Например, «Если для всех $x : A$, $P(x)$ и $Q(x)$, то (для всех $x : A$, $P(x)$) и (для всех $x : A$, $Q(x)$)» переводится в

$$\left(\prod_{(x:A)} P(x) \times Q(x)\right) \to \left(\prod_{(x:A)} P(x)\right) \times \left(\prod_{(x:A)} Q(x)\right) .$$

Неформальное доказательство этой тавтологии может выглядеть следующим образом:

> Предположим, что для всех x выполняются «$P(x)$ и $Q(x)$». Во-первых, предположим, что задан x и докажем выполнение $P(x)$. По предположению имеем выполнение «$P(x)$ и $Q(x)$», и, следовательно, выполняется $P(x)$. Во-вторых, предположим, что задан x и докажем выполнение $Q(x)$. Снова по предположению мы имеем выполнение «$P(x)$ и $Q(x)$», и, следовательно, выполнение $Q(x)$.

В первом предложении начинается определение импликации как функции, путем введения свидетельства для ее гипотезы:

$$f(p) :\equiv \square : \left(\prod_{(x:A)} P(x)\right) \times \left(\prod_{(x:A)} Q(x)\right) .$$

Здесь существует неявное использование конструктора спаривания для создания элемента типа произведения, который в этом примере отчасти обозначен словами «во-первых» и «во-вторых»:

$$f(p) :\equiv \left(\square : \prod_{(x:A)} P(x), \square : \prod_{(x:A)} Q(x)\right) .$$

Фраза «предположим, что задан x и докажем выполнение $P(x)$» теперь указывает на определение зависимой функции обычным способом, вводя для нее входную переменную. Так как это происходит внутри конструктора спаривания, естественно записать соответствующий фрагмент как λ-абстракцию:

$$f(p) :\equiv \left(\lambda x.\,(\square : P(x)), \square : \prod_{(x:A)} Q(x)\right).$$

Теперь «имеем выполнение "$P(x)$ и $Q(x)$"» обращается к гипотезе, получая $p(x) : P(x) \times Q(x)$, а «следовательно, выполняется $P(x)$» неявно применяет соответствующую проекцию:

$$f(p) :\equiv \left(\lambda x.\,\mathsf{pr}_1(p(x)), \square : \prod_{(x:A)} Q(x)\right).$$

Следующие два предложения заполняют оставшуюся пусто́ту аналогичным образом:

$$f(p) :\equiv (\lambda x.\,\mathsf{pr}_1(p(x)), \lambda x.\,\mathsf{pr}_2(p(x))).$$

Конечно, доказательства на естественном языке, которые мы использовали в качестве примеров, гораздо более многословны, чем те, которые математики обычно используют на практике, они больше похожи на язык, который используется в курсе «Введение в доказательства». Практикующий математик научился заполнять бреши, поэтому на практике мы можем опустить множество деталей, и мы это сделаем. Однако критерий обонованности для доказательств всегда состоит в том, что они могут быть переведены обратно в конструкцию элемента соответствующего типа.

В качестве более конкретного примера рассмотрим, как определить неравенства натуральных чисел. Одно из естественных определений состоит в том, что $n \leqslant m$, если существует такое $k : \mathbb{N}$, что $n + k = m$ (здесь снова используются тождественные типы, которые мы представим в следующем разделе, но пока нам не требуется дополнительных сведений о них). С учетом перевода высказываний-как-типов это даст:

$$(n \leqslant m) :\equiv \sum_{k:\mathbb{N}} (n + k = m).$$

Читателю предлагается доказать известные свойства «$\leqslant$» из этого определения. Для строгого неравенства существует пара естественных вариантов, таких как

$$(n < m) :\equiv \sum_{k:\mathbb{N}} (n + \mathsf{succ}(k) = m)$$

или

$$(n < m) :\equiv (n \leqslant m) \times \neg(n = m).$$

Первый вариант более естественен в конструктивной математике, но в этом случае он фактически эквивалентен второму, так как $\mathbb{N}$ содержит «разрешимое равенство» (см. §3.4 и теорему 7.2.6).

Имеется также другая интерпретация тип $\sum_{(x:A)} P(x)$. Поскольку его обитателем является элемент $x : A$ вместе со свидетельством того, что $P(x)$ выполняется, то, вместо того, чтобы рассматривать $\sum_{(x:A)} P(x)$ как высказывание «существует $x : A$ такое, что $P(x)$», можно рассматривать его как «тип всех элементов $x : A$ таких, что $P(x)$», то есть как «подтип» A.

Мы вернемся к этой интерпретации в §3.5. Теперь же, заметим, что это позволяет включать аксиомы в определение типов в качестве математических структур, используя $\sum$-типы, как описано в §1.6. Например, предположим, что мы хотим определить **полугруппу** как тип A, снабженный двоичной операцией $m : A \to A \to A$ (т.е. магмой) такой, что для всех $x, y, z : A$ имеем $m(x, m(y, z)) = m(m(x, y), z)$. Последнее высказывание представляется типом

$$\prod_{(x,y,z:A)} m(x, m(y, z)) = m(m(x, y), z)\,,$$

так что тип полугрупп есть

$$\mathsf{Semigroup} :\equiv \sum_{A:\mathcal{U}} \sum_{(m:A\to A\to A)} \prod_{(x,y,z:A)} m(x, m(y, z)) = m(m(x, y), z)\,.$$

т.е. подтип Magma, состоящий из полугрупп. От обитателя этого типа мы можем извлечь носитель A, операцию m и свидетельство аксиомы, применив соответствующие проекции. Мы вернемся к этому примеру в §2.14.

Обратите также внимание на то, что мы можем использовать универсумы в теории типов для представления «логики высшего порядка», то есть мы можем проводить оценивание над всеми высказываниями или над всем предикатам. Например, мы можем представить высказывание *для всех свойств* $P : A \to \mathcal{U}$, *если* $P(a)$, *то* $P(b)$ как

$$\prod_{P:A\to\mathcal{U}} P(a) \to P(b)$$

где $A : \mathcal{U}$ и $a, b : A$. Однако, *априори*, это высказывание обитает в другом универсуме, более высокого порядка, чем высказывание, которые мы оцениваем, то есть

$$\left(\prod_{P:A\to\mathcal{U}_i} P(a) \to P(b)\right) : \mathcal{U}_{i+1}\,.$$

Мы вернемся к этому вопросу в §3.5.

Мы описали здесь «релевантно-доказательный» перевод высказываний, в которых доказательства дизъюнкций и экзистенциальных высказываний содержат некоторую информацию. Например, если имеется обитатель из $A+B$, рассматриваемый как свидетельство «A или B», то нам известно, было ли оно из A или из B. Аналогично, если имеется обитатель из $\sum_{(x:A)} P(x)$, рассматриваемый как свидетельство «существует $x : A$ такое, что $P(x)$», то мы знаем, что это элемент x (это первая проекция данного обитателя).

Вследствие релевантно-доказательной природы этой логики мы можем иметь «A тогда и только тогда, когда B» (что, напомним, означает $(A \to B) \times (B \to A)$), но все же типы A и B проявляют различное поведение. Например, легко проверить, что «$\mathbb{N}$ тогда и только тогда, когда $\mathbf{1}$», и тем не менее ясно, что $\mathbb{N}$ и $\mathbf{1}$ различаются важными качествами. Высказывание «$\mathbb{N}$ тогда и только тогда, когда $\mathbf{1}$» говорит нам только о том, что, *рассматривая его как простое высказывание*, тип $\mathbb{N}$ представляет то же высказывание, что и $\mathbf{1}$ (в данном случае истинное высказывание). Мы иногда говорим «A тогда и только тогда, когда B», подразумевая, что A и B **логически эквивалентны**. Это следует отличать от более сильного понятия *эквивалентности типов*, которое будет введено в §2.4 и главе 4: хотя $\mathbb{N}$ и $\mathbf{1}$ логически эквивалентны, они не являются эквивалентными типами.

В главе 3 мы введем класс типов, называемых «простыми высказываниями», для которых эквивалентность и логическая эквивалентность совпадают. Используя эти типы, мы введем модификацию вышеописанной логики, иногда лучше подходящей к ситуациям, в которых дополнительная информация, содержащаяся в дизъюнкциях и экзистенциальных элементах, отбрасывается.

Наконец, отметим, что соответствие высказываний как типов можно рассматривать обратным образом, позволяя считать любой тип A как высказывание, которое доказывается предъявлением элемента из A. Иногда мы будем озвучивать такое высказывание как «A **обитаем**». . То есть, когда мы говорим, что A обитаем, то имеем в виду, что предъявили (частный) элемент из A, но решили не давать имя этому элементу. Точно так же, говоря, что A *не обитаем*, мы предъявляем элемент из $\neg A$. В частности, пустой тип $\mathbf{0}$, очевидно, не обитаем, так как $\neg\mathbf{0} = (\mathbf{0} \to \mathbf{0})$ заселен элементом $\mathsf{id}_{\mathbf{0}}$[10].

1.12 Типы тождественности

Хотя предыдущие построения можно рассматривать как обобщения стандартных теоретико-множественных конструкций, наш подход к обращению с тождественностью, по-видимому, специфичен для теории типов. Согласно концепции высказываний как типов, само *высказывание* о том, что два элемента одного типа $a, b : A$ равны, должно соответствовать некоторому *типу*. Поскольку это высказывание зависит от того, что из себя представляют a и b, эти **типы равенства** или **типы тождественности** должны быть семействами типов, зависящими от двух копий A.

Можно записать это семейство как $\mathsf{Id}_A : A \to A \to \mathcal{U}$ (не путайте с тождественной функцией id_A), так что $\mathsf{Id}_A(a, b)$ является типом, представляющим утверждение равенства между a и b. Однако, поскольку мы знакомы с высказываниями как типами, для этого удобно использовать стандартный символ равенства, таким образом, «$a = b$» также будет обозначать *тип* $\mathsf{Id}_A(a, b)$, соответствующий высказыванию, что a равно b. Для ясности можно записать «$a =_A b$», чтобы указать тип A. Если имеется элемент $a =_A b$, мы можем сказать, что a и b **равны** или, иногда, **пропозиционально равны**, когда хотим подчеркнуть, что оно отличается от дефиниციального равенства $a \equiv b$, обсуждавшегося в §1.1.

Так же, как мы отметили в §1.11, что версии высказываний как типов для «или» и «существует» могут содержать больше информации, чем просто факт, что высказывание истинно, ничто не мешает типу $a = b$ также включать в себя больше информация. На самом деле, это краеугольный камень гомотопической интерпретации, где мы рассматриваем свидетельства $a = b$ как *пути* или *эквивалентности* между a и b в пространстве A. Так же, как может существовать более одного пути между двумя точками пространства, может быть более одного свидетельства того, что два объекта равны. Иначе говоря, мы можем рассматривать $a = b$ как тип *отождествлений* a и b, и может быть много разных способов, которыми можно отождествлять a и b. Мы вернемся к такой интерпретации в главе 2, пока же мы сосредоточимся на основных правилах для типа тождественности. Как и у всех других типов, рассмотренных в этой главе, у него будут правила формирования, введения, исключения и вычисления, которые ведут себя формально точно так же.

[10]Это не следует путать с высказыванием, что теория типов согласована, являющимся метатеоретическим высказыванием о том, что невозможно получить элемент из $\mathbf{0}$ по правилам теории типов.

Правило формирования гласит, что для типа $A : \mathcal{U}$ и двух элементов $a, b : A$, можно сформировать тип $a =_A b : \mathcal{U}$ в том же универсуме. Основной способ построения элемента $a = b$ состоит в понимании, что a и b одинаковы. Таким образом, мы имеем зависимую функцию

$$\mathsf{refl} : \prod_{a:A} (a =_A a) ,$$

называемую **рефлексивностью**, которая означает, что каждый элемент из A равен самому себе (определенным образом). Мы рассматриваем refl_a как постоянный путь при точке a.

В частности, это означает, что если a и b являются *дефинициально* равными, $a \equiv b$, то также имеется элемент $\mathsf{refl}_a : a =_A b$. Все хорошо типизировано, потому что $a \equiv b$ означает, что тип $a =_A b$ также дефинициально равен $a =_A a$, который является типом refl_a.

Принцип индукции для тождественных типов является одним из самых тонких частей теории типов и имеет решающее значение для гомотопической интерпретации. Мы начнем с рассмотрения важного следствия этого, принципа, означающего, что «равное может быть заменено равным», который выражается следующим образом:

Неразличимость тождественностей: для каждого семейства

$$C : A \to \mathcal{U}$$

существует функция

$$f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x) \to C(y) ,$$

такая, что

$$f(x, x, \mathsf{refl}_x) :\equiv \mathsf{id}_{C(x)} .$$

Это говорит о том, что каждое семейство типов C соответствует равенству в том смысле, что применение C к *равным* элементам из A также приводит к функции между результирующими типами. Отображаемое равенство означает, что функция, ассоциированная с рефлексивностью, является тождественной функцией (и мы увидим, что в общем случае функция $f(x, y, p) : C(x) \to C(y)$ всегда является эквивалентностью типов).

Неразличимость тождественности можно рассматривать как принцип рекурсии для тождественного типа, аналогичный приведенным ранее для булевых и натуральных чисел. Так же, как $\mathsf{rec}_{\mathbb{N}}$ предоставляет указанное отображение $\mathbb{N} \to C$ для любого другого типа C некоторого сорта, неразличимость тождественностей предоставляет заданное отображение от $x =_A y$ к некоторым другим рефлексивным бинарным отношениям на A, а именно к формам $C(x) \to C(y)$ для некоторого унарного предиката $C(x)$. Мы могли бы также сформулировать более общий принцип рекурсии относительно рефлексивных отношений более общего вида $C(x, y)$. Однако, чтобы полностью охарактеризовать тип тождественности, мы должны обобщить его на принцип индукции, который рассматривает не только отображения из $x =_A y$, но и семейства над ними. Иными словами, мы рассматриваем не только разрешение замены равных на равные, но и принимаем во внимание свидетельство p для равенства.

1.12.1 Индукция пути

Принцип индукции для тождественного типа называется **индукцией пути** в связи с объяснением гомотопической интерпретации во введении к главе 2. Можно видеть, что семейство типов тождественности свободно порождается элементами вида $\mathsf{refl}_x : x = x$.

Индукция пути: Для заданного семейства

$$C : \prod_{x,y:A} (x =_A y) \to \mathcal{U}$$

и функции

$$c : \prod_{x:A} C(x, x, \mathsf{refl}_x)$$

существует функция

$$f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x, y, p)$$

такая, что

$$f(x, x, \mathsf{refl}_x) :\equiv c(x)\,.$$

Обратите внимание, что так же, как и принципы индукции для произведений, копроизведений, натуральных чисел и т.д., индукция пути позволяет нам определить заданные функции, которые проявляют соответствующее вычислительное поведение. Так же, как имеется функция $f : \mathbb{N} \to C$, определяемая рекурсией из $c_0 : C$ и $c_s : \mathbb{N} \to C \to C$, для которой верно $f(0) \equiv c_0$ и $f(\mathsf{succ}(n)) \equiv c_s(n, f(n))$, имеется и функция $f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x, y, p)$, определяемая индукцией пути из $c : \prod_{(x:A)} C(x, x, \mathsf{refl}_x)$ и удовлетворяющая $f(x, x, \mathsf{refl}_x) \equiv c(x)$.

Чтобы понять смысл этого принципа, сначала рассмотрим более простой случай, когда C не зависит от p. Тогда имеем $C : A \to A \to \mathcal{U}$, что можно рассматривать как предикат, зависящий от двух элементов из A. Нас интересует ситуация, когда высказывание $C(x, y)$ выполняется для некоторой пары элементов $x, y : A$. В этом случае гипотеза индукции пути подсказывает, что мы знаем, что $C(x, x)$ выполняется для всех $x : A$, т.е. если мы определим C на паре (x, x), мы получим истинное высказывание, так что, C — рефлексивное отношение. Этот вывод тогда говорит нам, что $C(x, y)$ выполняется всякий раз, когда $x = y$. Это есть как раз более общий принцип рекурсии для упомянутых выше рефлексивных отношений.

Вообще, индуктивная форма правила позволяет C также зависеть от свидетельства $p : x = y$ для тождества между x и y. В предпосылке правила мы не только меняем x, y на x, x, но и одновременно меняем p на рефлексивность: для доказательства свойства для всех элементов x, y и путей $p : x = y$ между ними, достаточно рассмотреть все случаи, когда элементами являются x, x и путь $\mathsf{refl}_x : x = x$. Если бы мы рассматривали типы так же, как и множества, было бы не ясно, что это нас подкупает, но так как может быть много различных отождествлений $p : x = y$ между x и y, имеет смысл отслеживать их при рассмотрении семейств по типу $x =_A y$. В главе 2 мы увидим, что это очень важно для гомотопической интерпретации.

Если мы упакуем индукцию пути в одну функцию, она примет форму:

$$\mathsf{ind}_{=_A} : \prod_{(C:\prod_{(x,y:A)}(x=_A y)\to\mathcal{U})} \left(\prod_{(x:A)} C(x, x, \mathsf{refl}_x)\right) \to \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x, y, p)$$

с равенством

$$\mathsf{ind}_{=_A}(C, c, x, x, \mathsf{refl}_x) :\equiv c(x)\,.$$

Функция $\mathsf{ind}_{=_A}$ традиционно обозначается $\mathcal{J}$. В лемме 2.3.1 мы покажем, что неразличимость тождеств является примером индукции пути, а также дадим этому новое имя и обозначение.

Для доказательства $p : a = b$, индукция пути требует, чтобы мы заменили a и b одним и тем же неизвестным элементом x. Таким образом, чтобы определить элемент семейства C, для всех пар элементов из A, достаточно определить его на диагонали. Однако в некоторых доказательствах проще использовать уравнение $p : a = b$, заменив все вхождения b на a (или наоборот), потому что иногда легче провести оставшуюся часть доказательства для конкретного элемента, упомянутого в равенстве, чем для общего неизвестного x. Это мотивирует введение второго принципа индукции для типов тождественности, в котором говорится, что семейство типов $a =_A x$ порождается элементом $\mathsf{refl}_a : a = a$. Как мы покажем ниже, этот второй принцип эквивалентен первому; иногда это просто более удобная формулировка.

Базированная индукция пути: зафиксируем элемент $a : A$ и предположим, что задано семейство

$$C : \prod_{x:A} (a =_A x) \to \mathcal{U}$$

и некоторый элемент

$$c : C(a, \mathsf{refl}_a)\,.$$

Тогда получим функцию

$$f : \prod_{(x:A)} \prod_{(p:a=x)} C(x, p)$$

такую, что

$$f(a, \mathsf{refl}_a) :\equiv c\,.$$

Здесь $C(x, p)$ — семейство типов, где x — элемент из A, а p — элемент типа равенства $a =_A x$, для фиксированного a в A. Принцип базированной индукции пути утверждает, что для определения элемента этого семейства для всех x и p, достаточно рассмотреть случай, когда x есть a, а p — $\mathsf{refl}_a : a = a$.

Упакованная как функция, базированная индукция пути принимает вид:

$$\mathsf{ind}'_{=_A} : \prod_{(a:A)} \prod_{(C:\prod_{(x:A)}(a=_A x)\to\mathcal{U})} C(a, \mathsf{refl}_a) \to \prod_{(x:A)} \prod_{(p:a=_A x)} C(x, p)$$

с равенством

$$\mathsf{ind}'_{=_A}(a, C, c, a, \mathsf{refl}_a) :\equiv c\,.$$

Ниже мы покажем, что индукция пути и базированная индукция пути эквивалентны. Поэтому иногда будем не очень строгими, ссылаясь на базированную индукцию пути просто как на индукцию пути, полагаясь на читателя, способного вывести, какой принцип подразумевается из формы доказательства.

Замечание 1.12.1. Интуитивно, принцип индукции для натуральных чисел выражает тот факт, что каждое натуральное число равно либо 0, либо $\mathsf{succ}(n)$ для некоторого натурального числа n, так что если мы докажем свойство для этих случаев (с предположением индукции во втором случае), то мы докажем это для всех натуральных чисел. Аналогично, принцип индукции для $A + B$ выражает тот факт, что каждый элемент из $A + B$ имеет либо вид $\mathsf{inl}(a)$, либо $\mathsf{inr}(b)$ и т.д. Применяя это же понимание к индукции пути, мы можем сказать, что индукция пути выражает тот факт, что каждый путь имеет вид refl_a, так что, если мы докажем свойство для путей рефлексивности, то мы докажем его для всех путей.

Однако такое понимание полностью сбивает с толку в контексте гомотопической интерпретации путей, где может быть множество различных способов выделения двух элементов a и b и, следовательно, много разных элементов тождественного типа! Как может быть много разных путей, если имеетя принцип индукции, утверждающий, что единственным путем является рефлексивность?

Главное наблюдение заключается в том, что это не *тип* тождественности, который определяется индуктивно, а *семейство* тождественности. В частности, индукция пути говорит о том, что семейство типов $(x =_A y)$ в качестве x, когда y меняется по всем элементам A, индуктивно определяется элементами вида refl_x. Это означает, что для того, чтобы предоставить элемент любого другого семейства $C(x, y, p)$, зависящего от *общего* элемента (x, y, p) тождественного семейства, достаточно рассмотреть случаи вида (x, x, refl_x). В гомотопической интерпретации это говорит о том, что тип троек (x, y, p), где x и y — крайние точки пути p (другими словами, $\sum$-тип $\sum_{(x,y:A)}(x = y)$), индуктивно порождается постоянными петлями в каждой точке x. Как мы увидим в главе 2, в гомотопической теории пространство, соответствующее $\sum_{(x,y:A)}(x = y)$, является *свободным пространством пути* — пространством путей в A, крайние точки которого могут меняться — и, на самом деле, любая точка этого пространство гомотопна постоянной петле в некоторой точке, так как мы можем просто отбросить один из концов данного пути. Аналогичный факт справедлив и в теории типов: мы можем доказать индукцией пути по $p : x = y$, что $(x, y, p) =_{\sum_{(x,y:A)}(x=y)} (x, x, \mathsf{refl}_x)$.

Аналогично, базированная индукция пути говорит о том, что для фиксированного $a : A$ *семейство* типов $(a =_A y)$ при изменении y по всем элементам A индуктивно определяется элементом refl_a. Таким образом, чтобы предоставить элемент любого другого семейства $C(y, p)$, зависящего от общего элемента (y, p) этого семейства, достаточно рассмотреть случай (a, refl_a). Гомотопически это выражает тот факт, что пространство путей, начинающихся в некоторой выбранной точке (*базированное пространство путей* в этой точке, которое теоретически есть $\sum_{(y:A)}(a = y)$), стягиваемо к постоянной петле в выбранной точке. Опять же, соответствующий факт справедлив и в теории типов: мы можем доказать базированной индукцией пути по $p : a = y$, что $(y, p) =_{\sum_{(y:A)}(a=y)} (a, \mathsf{refl}_a)$. Заметим также, что согласно интерпретации $\sum$-типов как подтипов, упомянутой в §1.11, тип $\sum_{(y:A)}(a = y)$ можно рассматривать как «тип всех элементов из A, которые равны a», теоретико-типовую версию «одноэлементного подмножества» $\{a\}$.

Ни один из этих двух принципов не дает возможности предоставить элемент семейства $C(p)$, где p имеет *две зафиксированные концевые точки* a и b. В частности, для семейства $C(p : a =_A a)$, зависящего от петли, мы не можем применять индукцию пути и рассматривать только случай для $C(\mathsf{refl}_a)$, и, следовательно, мы не можем доказать, что все петли являются рефлексивностью. Таким образом, индуктивное определение семейства тождественности не запрещает пути нерефлексивности в конкретных случаях типа тождественности. Другими словами, путь $p : x = x$ может быть не равен рефлексивности как элемент $(x = x)$, но пара (x, p) будет, тем не менее, равна паре (x, refl_x) в качестве элементов из $\sum_{(\mathrm{y:A})}(\mathrm{x = y})$.

В качестве топологического примера рассмотрим петлю в проколотом диске (круге) $\{(x, y) \mid 0 < x^2 + y^2 < 2\}$, которая начинается с точки $(1, 0)$ и проходит один раз вокруг отверстия в точке $(0, 0)$, прежде чем вернуться обратно к $(1, 0)$. Если мы удерживаем обе крайние точки, зафиксированные в $(1, 0)$, эта петля не может быть деформирована в постоянный путь, оставаясь внутри проколотого диска, так же, как невозможно натянуть веревку вокруг шеста, если мы удерживаем оба конца этой веревки. Тем не менее, петлю можно сжимать до постоянной величины, если мы позволяем одной из крайних точек меняться, точно так же, как мы всегда можем смотать веревку, если мы держим только один ее конец.

1.12.2 Эквивалентность индукции пути и базированной индукции пути

Два принципа индукции, для введенного выше типа тожественности, эквивалентны. Легко видеть, что индукция пути вытекает из принципа базированной индукции пути. Действительно, пусть даны предпосылки индукции пути:

$$
\begin{aligned}
C &: \prod_{x,y:A} (x =_A y) \to \mathcal{U}\,, \\
c &: \prod_{x:A} C(x, x, \mathsf{refl}_x)\,.
\end{aligned}
$$

Теперь, для элемента $x : A$, мы можем создать экземпляр обоих вышеизложенных случаев, получив

$$
\begin{aligned}
C' &: \prod_{y:A} (x =_A y) \to \mathcal{U}\,, \\
C' &:\equiv C(x)\,, \\
c' &: C'(x, \mathsf{refl}_x)\,, \\
c' &:\equiv c(x)\,.
\end{aligned}
$$

Ясно, что C' и c' соответствуют предпосылкам базированной индукции пути, и, следовательно, можно построить

$$
g : \prod_{(y:A)} \prod_{(p:x=y)} C'(y, p)
$$

с определяющим равенством

$$
g(x, \mathsf{refl}_x) :\equiv c'\,.
$$

Теперь заметим, что кообласть g равна $C(x, y, p)$. Таким образом, при предположении $x : A$, мы можем получить функцию

$$
f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} C(x, y, p)
$$

с требуемым дефинициальным равенством $f(x, x, \mathsf{refl}_x) \equiv g(x, \mathsf{refl}_x) :\equiv c' :\equiv c(x)$.

Другим доказательством этого факта является наблюдение, что любая такая f может быть получена как экземпляр $\mathsf{ind}'_{=_A}$, поэтому достаточно определить $\mathsf{ind}_{=_A}$ в обозначениях $\mathsf{ind}'_{=_A}$ как

$$
\mathsf{ind}_{=_A}(C, c, x, y, p) :\equiv \mathsf{ind}'_{=_A}(x, C(x), c(x), y, p)\,.
$$

Другое направление немного сложнее; неясно, как мы можем использовать конкретный экземпляр индукции пути для получения конкретного экземпляра базированной индукции пути. Вместо этого мы можем построить один экземпляр индукции пути, который демонстрирует сразу все возможные экземпляры базированной индукции пути. Определим

$$D : \prod_{x,y:A} (x =_A y) \to \mathcal{U}\,,$$
$$D(x,y,p) :\equiv \prod_{C:\prod_{(z:A)}\prod_{(p:x=_A z)}\mathcal{U}} C(x,\mathsf{refl}_x) \to C(y,p)\,.$$

Тогда мы можем построить функцию

$$d : \prod_{x:A} D(x,x,\mathsf{refl}_x)\,,$$
$$d :\equiv \lambda x.\, \lambda C.\, \lambda(c : C(x,\mathsf{refl}_x)).\, c$$

и, следовательно, используя индукцию пути, получим

$$f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} D(x,y,p)$$

с $f(x,x,\mathsf{refl}_x) :\equiv d(x)$. Развернув определение D, мы можем детализировать тип f:

$$f : \prod_{(x,y:A)} \prod_{(p:x=_A y)} \prod_{(C:\prod_{(z:A)}\prod_{(p:x=_A z)}\mathcal{U})} C(x,\mathsf{refl}_x) \to C(y,p)\,.$$

Теперь, принимая во внимание $x : A$ и $p : a =_A x$, можно получить вывод базированной индукции пути:

$$f(a,x,p,C,c) : C(x,p)\,.$$

Обратите внимание, что к тому же мы получаем корректное определение равенства.

Другое доказательство основано на наблюдении, что любое использование базированной индукци пути является экземпляром $\mathsf{ind}'_{=_A}$, и на определении

$$\mathsf{ind}'_{=_A}(a,C,c,x,p) :\equiv \mathsf{ind}_{=_A}\Big((\lambda x,y.\,\lambda p.\prod_{(C:\prod_{(z:A)}\prod_{(p:x=_A z)}\mathcal{U})} C(x,\mathsf{refl}_x) \to C(y,p)),$$
$$(\lambda x.\,\lambda C.\,\lambda d.\lambda d), a, x, p, C, c\Big)\,.$$

Обратите внимание, что приведенная выше конструкция использует универсумы. То есть, если мы хотим моделировать $\mathsf{ind}'_{=_A}$ с $C : \prod_{(x:A)}(a =_A x) \to U_i$, нам нужно использовать $\mathsf{ind}_{=_A}$ с

$$D : \prod_{x,y:A} (x =_A y) \to \mathcal{U}_{i+1}$$

так как D количественно оценивает все C данного типа. Хотя это совместимо с нашим определением универсумов, также можно выводить $\mathsf{ind}'_{=_A}$ и без их использования: мы можем показать, что $\mathsf{ind}_{=_A}$ влечет за собой леммы 2.3.1 и 3.11.8 и что два рассматриваемых принципа подразумевают $\mathsf{ind}'_{=_A}$ непосредственно. Мы оставляем детали читателю в качестве Упражнения 1.7.

Мы можем использовать любую из вышеперечисленных формулировок типов тождественности, чтобы установить, что равенство является отношением эквивалентности, что каждая функция сохраняет равенство и что каждое семейство не нарушает равенства. Мы оставляем изложение деталей до следующей главы, где это будет выведено и объяснено в контексте гомотопической теории типов.

1.12.3 Неэквивалентности

Наконец, позвольте нам также кое-что сказать об **неэквивалентности**, которая является отрицанием равенства[11]:

$$(x \neq_A y) :\equiv \neg(x =_A y)\,.$$

Если $x \neq y$, мы говорим, что x и y **не тождественны** или **не равны**. Как и отрицание, нетождественность играет здесь менее важную роль, чем в классической математике. Например, мы не можем доказать, что две вещи равны, доказывая, что они не являются неравными: это будет применение классического закона двойного отрицания, см. §3.4.

Иногда полезно позиционировать неэквивалентность в позитивном ключе. Например, в теореме 11.2.4 мы докажем, что вещественное число x имеет обратное, тогда и только тогда, когда его расстояние от 0 положительно, что является более сильным требованием, чем $x \neq 0$.

Примечания

Представленная здесь теория типов представляет собой версию теории интуиционистского типа Мартина-Лёфа (Martin-Löf) [ML98, ML75, ML82, ML84], которая сама основана на основополагающих работах Брауэра (Brouwer) [Bee85], Гейтинга (Heyting) [Hey66], Скотта (Scott) [Sco70], де Брёйна (de Bruijn) [dB73], Говарда (Howard) [How80], Тейта (Tait) [Tai67, Tai68] и Ловера (Lawvere) [Law06]. Три основных варианта теории типа Мартина-Лёфа лежат в основе компьютерных реализаций типа NuPRL [CAB+86], Coq [Coq12] и Agda [Nor07]. Приведенная здесь теория отличается от этих формулировок многими аспектами, некоторые из которых имеют решающее значение для гомотопической интерпретации, в то время как другие являются техническими удобствами или включают понятия, которые еще не изучались в гомотопической обстановке.

Наиболее важно, что описанная здесь теория типов получена из *интенсиональной* версии теории типов Мартина-Лёфа [ML75], а не *экстенсиональной* версии [ML82]. В то время как экстенсиональная теория не делает различий между дефинициальным и пропозициональным равенством, интенсиональная теория рассматривает субъективное равенство как чисто определительное и допускает гораздо более широкую, доказательную трактовку тождественного типа, который является центральным для гомотопической интерпретации. С гомотопической точки зрения экстенсиональная теория типов ограничивается гомотопически дискретными множествами (см. §3.1), тогда как интенсиональная теория допускает типы с высокоразмерной структурой. Система NuPRL [CAB+86] является экстенсиональной, тогда как Coq [Coq12] и Agda [Nor07] являются интенсиональными. Среди интенсиональных теорий типов существует ряд вариантов, которые различаются по структуре доказательств тождественности. Наиболее вольная интерпретация, на которую мы здесь опираемся, допускает *доказательную* интерпретацию равенства, тогда как более ограничительные варианты налагают такие ограничения, как *уникальность доказательств тождественности* (*UIP*) [Str93a], в которой утверждается, что любые два доказательства равенства дефинициально равны, и Аксиома K [Str93a], утверждающая, что единственным доказательством равенства является рефлексивность (вплоть до

[11] Мы используем термин «неравенство» при использовании «$<$» и «$\leqslant$». Кроме того, обратите внимание, что речь идет об отрицании типа пропозиционального тождества. Разумеется, нет смысла отрицать субъективное равенство $\equiv$, потому что суждения не подчиняются логическим операциям.

дефинициального равенства). Эти дополнительные требования могут быть выборочно введены в системы Coq и Agda.

Еще одна точка различия среди интенсиональных теорий — прочность дефинициального равенства, особенно в отношении объектов функционального типа. Здесь мы вводим принцип уникальности (η-конверсия) $f \equiv \lambda x.\, f(x)$, как принцип дефинициального равенства. Этот принцип используется, например, в §4.9, чтобы показать, что унивалентность подразумевает расширение пропозициональной функции. Принципы уникальности иногда рассматриваются для других типов. Например, принцип уникальности для декартовых произведений был бы дефинициальной версией пропозиционального равенства uppt, которое мы построили в §1.5, говоря, что $u \equiv (\mathsf{pr}_1(u), \mathsf{pr}_1(u))$. Это и соответствующая версия для зависимых пар были бы разумным выбором (чего мы не делали), но мы не можем включать все такие правила, потому что соответствующий принцип уникальности для типов тождественности будет превращать в тривиальную всю высшую гомотопическую структуру. Поэтому мы *вынуждены* не делать этого, но тогда возникает вопрос, где проводить черту. Что касается индуктивных типов, мы обсудим соответствующие моменты далее в §5.5.

Для наших целей важно, чтобы (пропозициональное) равенство функций принималось как *экстенсиональное* (в ином смысле, чем используемое выше!). Это не является следствием правил из этой главы и будет выражено Аксиомой 2.9.3. Такое решение важно для наших целей, поскольку оно устанавливает, что равенство функций соответствует предполагаемому в математике. Хотя в качестве аксиомы мы включаем Аксиому 2.9.3, она может быть получена из аксиомы унивалентности и принципа уникальности для функций (см. §4.9), а также из существования типа интервалов (см. лемму 6.3.2).

Что касается индуктивных типов, таких как произведения, $\sum$-типы, копроизведения, натуральные числа и т.д. (см. главу 5), существуют дополнительные варианты относительно того, как именно формулировать индукцию и рекурсию. Мы приняли *принципы индукции* в качестве базовых и *сопоставление с образцом* выведено из них, но можно сделать и по другому; см. Приложение A. Обычно в последнем случае допускается также глубокое сопоставление с образцом; см. [Coq92b]. Имеется несколько причин для нашего выбора. Одна из причин заключается в том, что принципы индукции — это то, что мы получаем естественно в категорной семантике. Другая причина заключается в том, что указание допустимых видов (глубокого) сопоставления с образцом довольно сложно; например, сопоставление с образцом в Agda может доказать Аксиому K по умолчанию, хотя флаг `--without-K` не допускает этого [CDP14]. Наконец, глубокое сопоставление с образцом недостаточно хорошо понятно для высших индуктивных типов (см. главу 6). Поэтому мы будем использовать только сопоставление с образцом, как оно описано в §1.10, которое непосредственно эквивалентно применению принципа индукции.

В отличие от теории типов Coq, мы не вводим примитивный тип высказываний. Вместо этого, как обсуждалось в §1.11, мы принимаем принцип *высказывания-как-типы* (*PAT*), идентифицируя высказывания по типам. Первоначально это было предложено де Брёйном [dB73], Говардом [How80], Тейтом [Tai68] и Мартином-Лёфом [ML98] (наше решение более полно объяснено в §§ 3.2 и 3.3).

Однако, мы включаем полную кумулятивную иерархию универсумов, так что построение типов и суждения о равенстве становятся примерами суждений о членстве и равенстве для универсума. Для удобства мы рассматриваем объекты универсума как типы, а не как коды для типов; в терминологии [ML84], это означает, что мы используем «универсумы в стиле Расселла», а не «универсумы в стиле Тарского». Альтернативой может быть использование универсумов в стиле Тарского с явной функцией принуждения, требуемой для того, чтобы привести элемент $A : \mathcal{U}$ универсума к типу $\mathsf{El}(A)$, и просто сказать, что принуждение не учитывается при неформальном использовании.

Мы рассматриваем иерархию универсумов как кумулятивную, так как каждый тип из $\mathcal{U}_i$ также находится в $\mathcal{U}_j$ для каждого $j \geqslant i$. Существуют различные способы формальной реализации кумулятивности, самым простым является включение правила: если $A : \mathcal{U}_i$, то $A : \mathcal{U}_i$. Однако оно имеет досадное следствие: для семейства типов $B : A \to \mathcal{U}_i$ мы не можем вывести заключение $B : A \to \mathcal{U}_j$, хотя можем вывести $\lambda a.\, B(a) : A \to \mathcal{U}_j$. Более сложный подход, который решает эту проблему, заключается в введении дефинициального отношения подтипирования $<:$, порожденного $\mathcal{U}_i <: \mathcal{U}_j$, но это усложняет изучение теории типов. Другой альтернативой было бы включение явной функции принуждения $\uparrow : \mathcal{U}_i \to \mathcal{U}_j$, которую можно было бы игнорировать при неформальных рассуждениях.

Также не обязательно, чтобы универсумы были проиндексированы натуральными числами и линейно упорядочены. В некоторых целях более уместно предполагать, что каждый универсум является элементом более крупного универсума, вместе со свойством «направленности», выражающем то, что любые две универсума совместно содержатся в каком-то более крупном. Имеется много других возможных вариантов, таких как включение универсума «$\mathcal{U}_w$», который содержит все $\mathcal{U}_i$ (или даже высшие «большие кардинальные» универсумы типов), или путем введения иерархии в семейство типов $\lambda i.\mathcal{U}_i$. Последнее фактически сделано в Agda.

Принцип индукции пути для типов тождественности был сформулирован Мартином-Лёфом [ML98]. Это правило применено Мартином-Лёфом для теории типов благодаря Паулин-Морингу (Paulin-Mohring) [PM93]; его можно рассматривать как интенсиональное обобщение понятия «поточечной функциональности» для гипотетических суждений из NuPRL [CAB+86, раздел 8.1]. Тот факт, что правило Мартина-Лёфа содержит намек от Паулин-Моринга, было доказано Штрайхером (Streicher) с использованием Аксиомы K (см. §7.2), Альтенкирхом (Altenkirch) и Гогуеном (Goguen) (как в §1.12), и, наконец, Хофманом (Hofmann), без универсумов (как в упражнении 1.7); см. [Str93b, §1.3 и Addendum].

Упражнения

Упражнение 1.1. Для функций $f : A \to B$ и $g : B \to C$ определите их композицию $g \circ f : A \to C$. Покажите, что имеет место $h \circ (g \circ f) \equiv (h \circ g) \circ f$.

Упражнение 1.2. Выведите принцип рекурсии для произведения $\mathsf{rec}_{A \times B}$, используя только проекции, и убедитесь, что справедливы определяющие равенства. Сделайте то же самое для $\sum$-типов.

Упражнение 1.3. Выведите принцип индукции для произведений $\mathsf{ind}_{A \times B}$, используя только проекции и принцип пропозициональной уникальности uppt. Убедитесь, что определяющие равенства верны. Обобщите uppt для $\sum$-типов и сделайте то же самое для самих $\sum$-типов (для этого требуются понятия из главы 2).

Упражнение 1.4. Предполагая, что задан только *итератор* для натуральных чисел

$$\mathsf{iter} : \prod_{C:\mathcal{U}} C \to (C \to C) \to \mathbb{N} \to C$$

с определяющими равенствами

$$\mathsf{iter}(C, c_0, c_s, 0) :\equiv c_0\,,$$
$$\mathsf{iter}(C, c_0, c_s, \mathsf{succ}(n)) :\equiv c_s(\mathsf{iter}(C, c_0, c_s, n))$$

выведите функцию, имеющую тип рекурсора $\mathsf{rec}_{\mathbb{N}}$. Покажите, что определяющие уравнения этого рекурсора сохраняются только пропозиционально для этой функции, и требуют использования принципа индукции для $\mathbb{N}$.

Упражнение 1.5. Покажите, что если определить $A+B :\equiv \sum_{(x:\mathbf{2})} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, то можно дать определение ind_{A+B}, для которого выполнены определяющие равенства, указанные в §1.7.

Упражнение 1.6. Покажите, что если определить $A \times B :\equiv \sum_{(x:\mathbf{2})} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$, то можно дать определение $\mathsf{ind}_{A\times B}$, для которого определяющие равенства, указанные в §1.5, выполняются пропозиционально (т.е. с использованием типов равенства; для этого требуется аксиома функциональной экстенсиональности, которая вводится в §2.9).

Упражнение 1.7. Дайте альтернативный вывод $\mathsf{ind}'_{=_A}$ из $\mathsf{ind}_{=_A}$ без использования универсумов (*проще всего использовать понятия из последующих глав*).

Упражнение 1.8. Определите умножение и возведение в степень с помощью $\mathsf{rec}_{\mathbb{N}}$. Убедитесь, используя только $\mathsf{ind}_{\mathbb{N}}$, что $(\mathbb{N}, +, 0, \times, 1)$ является полукольцом (вероятно, понадобится использовать симметрию и транзитивность равенства, леммы 2.1.1 и 2.1.2)

Упражнение 1.9. Определите тип семейства $\mathsf{Fin} : \mathbb{N} \to \mathcal{U}$, упомянутое в конце §1.3, и зависимую функцию $\mathsf{fmax} : \prod_{(n:\mathbb{N})} \mathsf{Fin}(n+1)$, упомянутую в §1.4.

Упражнение 1.10. Покажите, что, используя только $\mathsf{rec}_{\mathbb{N}}$, можно определить функцию Аккермана $\mathsf{ack} : \mathbb{N} \to \mathbb{N} \to \mathbb{N}$, удовлетворяющую следующим уравнениям:

$$\begin{aligned} \mathsf{ack}(0, n) &\equiv \mathsf{succ}(n)\,, \\ \mathsf{ack}(\mathsf{succ}(m), 0) &\equiv 1\,, \\ \mathsf{ack}(\mathsf{succ}(m), \mathsf{succ}(n)) &\equiv \mathsf{ack}(m, \mathsf{ack}(\mathsf{succ}(m), n))\,. \end{aligned}$$

Упражнение 1.11. Покажите, что для любого типа A, имеет место $\neg\neg\neg A \to \neg A$.

Упражнение 1.12. Используя интерпретацию высказываний как типов, выведите следующие тавтологии.

(i) Если A, то (если B, то A).

(ii) Если A, то не (не A).

(iii) Если (не A или не B), то не (A и B).

Упражнение 1.13. Используя высказывания как типы, выведите двойное отрицание принципа исключения третьего, т.е. докажите *не* (*не* (P *или не* P)).

Упражнение 1.14. Почему принципы индукции для типов тождественности не позволяют построить функцию $f : \prod_{(x:A)} \prod_{(p:x=x)} (p = \mathsf{refl}_x)$ с определяющим уравнением

$$f(x, \mathsf{refl}_x) :\equiv \mathsf{refl}_{\mathsf{refl}_x}?$$

Упражнение 1.15. Покажите, что неразличимость тождественности следует из индукции пути.

Упражнение 1.16. Покажите, что сложение натуральных чисел коммутативно: $\prod_{(i,j:\mathbb{N})}(i + j = j + i)$.

Глава 2

Гомотопическая теория типов

Центральная новая идея в гомотопической теории типов заключается в том, что типы можно рассматривать как пространства в гомотопической теории или как высшие группоиды в теории категорий.

Начнем с краткого изложения связи между теорией гомотопий и теорией высших категорий. В классической гомотопической теории пространство X представляет собой множество точек, снабженных топологией, а путь между точками x и y представляется непрерывным отображением $p : [0,1] \to X$, где $p(0) = x$, а $p(1) = y$. Эту функцию можно рассматривать как указание точки в X в каждый «момент времени». Для многих целей строгое равенство путей (означающее поточечно равные функции) слишком тонкое понятие. Например, можно определить операции конкатенации путей (если p — путь от x до y, а q — путь от y до z, то конкатенация $p \cdot q$ — это путь от x до z) и обратные пути (p^{-1} — путь от y до x). Однако между этими операциями существуют естественные уравнения, которые не соблюдаются для строгого равенства: например, путь $p \cdot p^{-1}$ (который идет от x к y, а затем обратно по тому же маршруту, как время идет от 0 до 1), является не строго равным тождественному пути (который всегда остается в точке x).

Средством преодоления этого является рассмотрение более грубого представления о равенстве путей, называемого *гомотопией*. Гомотопия между парой непрерывных отображений $f : X_1 \to X_2$ и $g : X_1 \to X_2$ — непрерывное отображение $H : X_1 \times [0,1] \to X_2$, удовлетворяющее условиям $H(x,0) = f(x)$ и $H(x,1) = g(x)$. В конкретном случае путей p и q от x к y гомотопия является непрерывным отображением $H : [0,1] \times [0,1] \to X$ такой, что $H(s,0) = p(s)$ и $H(s,1) = q(s)$ для всех $s \in [0,1]$. В этом случае мы также требуем, чтобы $H(0,t) = x$ и $H(1,t) = y$ для всех $t \in [0,1]$, так что для каждого t функция $H(\_,t)$ снова является путем от x к y; считается, что гомотопия такого рода является *сохраняющей конечные точки* или *связывающей конечные точки*. В простых случаях мы можем представить себе образ квадрата $[0,1] \times [0,1]$ при H как «заполнение пространства» между p и q, хотя для произвольного X это не имеет смысла; лучше думать о H как о непрерывной деформации p в q, которая не перемещает конечные точки. Так как $[0,1] \times [0,1]$ является двумерным, мы также говорим о H как о двумерном *пути между путями*.

Например, поскольку $p \cdot p^{-1}$ выходит и возвращается по одному тому же маршруту, вы знаете, что можете непрерывно сжимать $p \cdot p^{-1}$ до тождественного пути — он, например, не будет цепляться за какое-нибудь отверстие в пространстве. Гомотопия является отношением эквивалентности, и такие операции, как конкатенация, обратные элементы и т.д., не нарушают ее. Более того, классы гомотопической эквивалентности петель в некоторой точке x_0 (где две петли p и q считаются равными, когда между ними существует гомотопия H, как указанная выше, которая дополнительно удовлетворяет условию $H(0,t) = H(1,t) = x_0$ для всех t) образуют группу, называемую *фундаментальной группой*. Эта группа является *алгебраическим инвариантом* пространства, которая может быть использована для исследования того, являются ли два пространства *гомотопическими эквивалентными* (существуют непрерывные отображения назад и вперед, композиции которых гомотопны тождественному отображению), поскольку эквивалентные пространства имеют изоморфные фундаментальные группы.

Поскольку гомотопии сами по себе являются своего рода двумерным путем, существует естественное понятие трехмерной *гомотопии между гомотопиями*, затем *гомотопии между гомотопиями между гомотопиями* и т.д. Эта бесконечная башня точек, путей, гомотопий, гомотопий между гомотопиями ..., снабженная алгебраическими операциями, такими как фундаментальная группа, является экземпляром алгебраической структуры, называемой (слабым) *∞-группоидом*. ∞-группоид состоит из набора объектов, затем набора *морфизмов* между объектами, *морфизмов между морфизмами* и т.д., снабженных некоторой сложной алгебраической структурой; морфизм на уровне k называется **k-морфизмом**. Морфизмы на каждом уровне имеют тождественные, их можно компоновать и применять обратные операции, и которые являются слабыми в том смысле, что они удовлетворяют группоидным законам (ассоциативность композиции, тождественный морфизм является единицей для композиции, инверсия морфизмов отсутствует) только для морфизмов данного уровня, и эта слабость приводит к дальнейшему усложнению структуры. Например, поскольку ассоциативность композиции морфизмов $p \cdot (q \cdot r) = (p \cdot q) \cdot r$ сама по себе является многомерным морфизмом, нужна дополнительная операция, связанная с различными доказательствами ассоциативности: различные способы повторного связывания $p \cdot (q \cdot (r \cdot s))$ в $((p \cdot q) \cdot r) \cdot s$ порождают пятиугольник МакЛейна. Слабость, к тому же, создает нетривиальные взаимодействия между уровнями.

Каждое топологическое пространство X имеет *фундаментальный ∞-группоид*, k-морфизмы которого являются k-мерными путями в X. Слабость ∞-группоида напрямую связана с тем, что пути образуют группу только до гомотопии, с $(k+1)$-путями, служащими гомотопиями между k-путями. Более того, рассмотрение пространства как ∞-группоида сохраняет достаточные аспекты пространства, с тем чтобы создать гомотопическую теорию: фундаментальная конструкция ∞-группоида сопряжена с геометрической реализацией ∞-группоида как пространства, и это сопряжение охраняет гомотопическую теорию (оно называется *гомотопической гипотезой/теоремой*, потому что, рассмотрение ее в качестве гипотезы или теоремы зависит от того, как вы определяете ∞-группоид). Например, вы можете легко определить фундаментальную группу ∞-группоида, а если вы вычислите фундаментальную группу фундаментального ∞-группоида пространства, это согласуется с классическим определением фундаментальной группы этого пространства. Из-за этого соответствия теория гомотопий и теория высших категорий тесно связаны между собой.

Теперь, в гомотопического теории типов каждый тип может рассматриваться, как имеющий структуру ∞-группоида. Напомним, что для любого типа A и любых $x, y : A$ имееся тождественный тип $x =_A y$, также записываемый как $\mathsf{Id}_A(x,y)$ или просто $x = y$. Логически мы можем представить элементы из $x = y$ в качестве доказательства того, что x и y равны, или как идентификаторы из x с y. Кроме того, теория типов (в отличие, скажем, от логики первого порядка) позволяет рассматривать такие элементы из x $=_\mathrm{A}$ y также как индивидуумы, которые могут являться предметом дальнейших исследований. Поэтому мы можем итерировать тип тождественности: мы можем образовать тип $p =_{(x =_A y)} q$ отождествлений между отождествлениями p, q и типом $r =_{(p =_{(x =_A y)} q)} s$ и т.д. Структура этой башни типов тождественности точно соответствует структуре непрерывных путей и (высших) гомотопий между ними в пространстве или

∞-группоиду.

Таким образом, мы будем часто ссылаться на элемент $p : x =_A y$ как на **путь** от x до y; мы называем x его **начальной точкой**, а y — его **конечной точкой**. Два пути $p, q : x =_A y$ с одинаковой начальной и конечной точкой называются **параллельными**, и в этом случае элемент $r : p =_{(x=_A y)} q$ можно рассматривать как гомотопию или морфизм между морфизмами; мы будем часто ссылаться на него как на **2-путь** или **2-мерный путь**. Аналогично, $r =_{(p=_{(x=_A y)}q)} s$ является типом **3-мерных путей** между двумя параллельными двумерными путями, и т.д. Если тип A является «подобным множеству», например $\mathbb{N}$, эти итерированные типы тождественности становятся неинтересными (см. §3.1), но в общем случае они могут моделировать нетривиальные гомотопические типы.

Важное различие между гомотопическогй теорией типов и классической гомотопической теорией состоит в том, что гомотопическая теория типов содержит *синтетическое* описание пространств в следующем смысле. Синтетическая геометрия — это геометрия в стиле Евклида [EucBC]: она начинается с некоторых основных понятий (точек и линий), конструкций (линия, соединяющая любые две точки), аксиом (все прямые углы равны) и логического вывода следствий. Это контрастирует с аналитической геометрией, где такие понятия, как точки и линии, представлены конкретно с использованием декартовых координат в $\mathbb{R}^n$ — линии являются множествами точек — и из этого представления выводятся основные конструкции и аксиомы. Хотя классическая гомотопическая теория аналитична (пространства и пути составлены из точек), гомотопическая теория типов является синтетической: точки, пути и пути между путями являются основными, неделимыми, примитивными понятиями.

Более того, одна из удивительных особенностей гомотопической теории типов состоит в том, что все основные конструкции и аксиомы — все от структур высших группоидов — возникают автоматически из принципа индукции для типов тождественности. Напомним из §1.12, что это говорит о том, что если

- для любых $x, y : A$ и любого $p : x =_A y$, имеется тип $D(x, y, p)$ и
- для любого $a : A$, имеется элемент $d(a) : D(a, a, \mathsf{refl}_a)$,

то

- существует элемент $\mathsf{ind}_{=_A}(D, d, x, y, p) : D(x, y, p)$, для *любых* двух элементов $x, y : A$ и $p : x =_A y$, такой что $\mathsf{ind}_{=_A}(D, d, a, a, \mathsf{refl}_a) \equiv d(a)$.

Другими словами, для зависимых функций

$$D : \prod_{(x,y:A)} \prod_{(p:x=y)} \mathcal{U}$$
$$d : \prod_{a:A} D(a, a, \mathsf{refl}_a)$$

существует зависимая функция

$$\mathsf{ind}_{=_A}(D, d) : \prod_{(x,y:A)} \prod_{(p:x=y)} D(x, y, p)$$

такая, что

$$\mathsf{ind}_{=_A}(D, d, a, a, \mathsf{refl}_a) \equiv d(a) \tag{2.0.1}$$

для любого $a : A$. Обычно, каждый раз, когда мы применяем это правило индукции, мы либо не заботимся о конкретной заданной функции, либо сразу же даем ей другое имя.

Неформально, принцип индукции для типов тождественности гласит, что если мы хотим построить объект (или доказать высказывание), который зависит от обитателя $p : x =_A y$ некоторого тождественного типа, то достаточно выполнить конструкцию (или доказательство) в частном случае, когда x и y одинаковы (дефинициально), а p — элемент рефлексивности $\mathsf{refl}_x : x = x$ (дефинициально). В неформальной форме мы можем выразить это фразой типа «по индукции, достаточно предположить ...». Это сведение к «случаю рефлексивности» аналогично сведению к «базовому случаю» и «индуктивному шагу» в обычном доказательстве индукцией на натуральных числах, а также «левому случаю» и «правому случаю» в доказательстве анализом случаев на несвязном объединении или дизъюнкции.

«Правило преобразования» (2.0.1) в меньшей степени знакомо в контексте доказательства индукцией на натуральных числах, но имеется аналогичное понятие в связанной концепции определения с помощью рекурсии. Если последовательность $(a_n)_{n\in\mathbb{N}}$ определяется заданием a_0 и уточнением a_{n+1} в терминах a_n, то фактически 0-й член результирующей последовательности является заданным, а заданное рекуррентное отношение, связывающее a_{n+1} с a_n, позволяет получить всю результирующую последовательность (это может показаться настолько очевидным, что и не стоит об этом говорить, но если мы рассмотрим определение с помощью рекурсии, как алгоритма вычисления значений последовательности, то это будет именно процесс выполнения данного алгоритма). Правило (2.0.1) аналогично: оно говорит, что если мы определим объект $f(p)$ для всех $p : x = y$, указав, какое значение должно быть, когда p является $\mathsf{refl}_x : x = x$, то указанное нами значение фактически является значением $f(\mathsf{refl}_x)$.

Этот принцип индукции наделяет каждый тип структурой ∞-группоида, а каждую функцию между двумя типами — структурой ∞-функтора между двумя такими группоидами. Это интересно с математической точки зрения, поскольку дает новый способ работы с ∞-группоидами. Это интересно и со стороны теоретико-множественного точечного представления, поскольку раскрывает новые операции, связанные с каждым типом и функцией. В оставшейся части этой главы мы начинаем исследовать эту структуру.

2.1 Типы — высшие группоиды

С помощью принципа индукции мы выясним теперь структуру группоида высшего порядка. Начнем с симметрии равенства, которое в топологическом языке означает, что «пути можно обращать».

Лемма 2.1.1. *Для любого типа A и любых $x, y : A$, существует функция*

$$(x = y) \to (y = x)$$

обозначаемая $p \mapsto p^{-1}$, такая что $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$, для любого $x : A$. Назовем p^{-1} ***обратным*** *к p.*

Поскольку мы первый раз говорим о чем-то как о «лемме» или «теореме», остановимся для осмысления, что это значит. Напомним, что высказывания (утверждения, восприимчивые к доказательству) отождествляются с типами, тогда как леммы и теоремы (высказывания, которые были доказаны) отождествляются с обитаемыми типами. Таким образом, утверждение леммы или теоремы должно быть переведено в тип, как в §1.11, а его доказательство переведено на обитателя этого типа. Согласно интерпретации универсального квантора «для каждого», тип, соответствующий лемме 2.1.1, есть

$$\prod_{(A:\mathcal{U})} \prod_{(x,y:A)} (x = y) \to (y = x) .$$

Доказательство леммы 2.1.1 будет состоять в построении элемента этого типа, т.е. выведения суждения $f : \prod_{(A:\mathcal{U})} \prod_{(x,y:A)} (x = y) \to (y = x)$ для некоторого f. Затем мы вводим обозначение $(\_)^{-1}$ для этого элемента f, в котором аргументы A, x и y опущены и выведены из контекста (как отмечено в §1.1, вторичное высказывание «$\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$ для каждого $x : A$» следует рассматривать как отдельное суждение).

Доказательство. (*первое*) Предположим, что $A : \mathcal{U}$, и пусть $D : \prod_{(x,y:A)}(x = y) \to \mathcal{U}$ — семейство типов, определяемое как $D(x, y, p) :\equiv (y = x)$. Другими словами, D — функция, назначающая любым $x, y : A$ и $p : x = y$ тип, а именно тип $y = x$. Тогда имеется элемент

$$d :\equiv \lambda x.\, \mathsf{refl}_x : \prod_{x:A} D(x, x, \mathsf{refl}_x) .$$

Таким образом, принцип индукции для типов тождественности дает элемент $\mathsf{ind}_{=_A}(D, d, x, y, p) : (y = x)$ для каждого $p : (x = y)$. Теперь можно определить искомую функцию $(\_)^{-1}$ как $\lambda p.\, \mathsf{ind}_{=_A}(D, d, x, y, p)$, т.е. положим $p^{-1} :\equiv \mathsf{ind}_{=_A}(D, d, x, y, p)$. Правило преобразования (2.0.1) дает $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$, что требовалось. □

Мы выписали это доказательство в том формальном стиле, который может быть полезен, в то время как правило индукции для типов тождественности малознакомо. Еще более формально можно сказать, что лемма 2.1.1 и ее доказательство вместе состоят из суждения

$$\lambda A.\, \lambda x.\, \lambda y.\, \lambda p.\, \mathsf{ind}_{=_A}((\lambda x.\, \lambda y.\, \lambda p.\, (y = x)), (\lambda x.\, \mathsf{refl}_x), x, y, p) : \prod_{(A:\mathcal{U})} \prod_{(x,y:A)} (x = y) \to (y = x)$$

(наряду с дополнительным суждением о равенстве). Тем не менее, в итоге мы предпочитаем снова использовать естественный язык, например, в следующем эквивалентном доказательстве.

Доказательство. (*второе*) Мы хотим построить для любых $x, y : A$ и $p : x = y$ элемент $p^{-1} : y = x$. По индукции достаточно сделать это в случае, когда y равен x, а p есть refl_x. Но в этом случае, и тип $x = y$ из p, и тип $y = x$ из p^{-1}, который мы пытаемся построить, являются простым типом $x = x$. Таким образом, в «случае рефлексивности» мы можем определить refl_x^{-1} как просто refl_x. Тогда общий случай следует из принципа индукции, а правило преобразования $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$ — это именно доказательство для случая рефлексивности, которое мы привели. □

Мы запишем следующие несколько доказательств в обоих стилях, чтобы помочь читателю привыкнуть к последнему. Далее докажем транзитивность равенства или, что то же самое, «конкатенируем пути».

Лемма 2.1.2. *Для любого типа A и любых $x, y, z : A$ существует функция*

$$(x = y) \to (y = z) \to (x = z) ,$$

обозначаемая $p \mapsto q \mapsto p \cdot q$, такая что $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$, для любого $x : A$. Назовем $p \cdot q$ ***конкатенацией*** *или* ***композицией*** *p и q.*

Обратите внимание, что мы решили обозначать конкатенацию путей в порядке, обратном функциональной композиции: из $p : x = y$ и $q : y = z$ мы получаем $p \cdot q : x = z$, тогда как $f : A \to B$ и $g : B \to C$ дает $g \circ f : A \to C$ (см. упражнение 1.1).

Первое доказательство. Требуемая функция имеет тип $\prod_{(x,y,z:A)}(x = y) \to (y = z) \to (x = z)$. Вместо этого определим функцию с эквивалентным типом $\prod_{(x,y:A)}(x = y) \to \prod_{(z:A)}(y = z) \to (x = z)$, что позволяет дважды применить индукцию пути. Пусть $D : \prod_{(x,y:A)} \prod_{(p:x=y)} \mathcal{U}$ — семейство типов

$$D(x, y, p) :\equiv \prod_{(z:A)} \prod_{(q:y=z)} (x = z)\,.$$

Отметим, что $D(x, x, \mathsf{refl}_x) \equiv \prod_{(z:A)} \prod_{(q:x=z)}(x = z)$. Таким образом, для применения принципа индукции для типов тождественности к этому D нам нужна функция типа

$$\prod_{x:A} D(x, x, \mathsf{refl}_x) \tag{2.1.3}$$

т.е. типа

$$\prod_{(x,z:A)} \prod_{(q:x=z)} (x = z)\,.$$

Теперь, пусть $E : \prod_{(x,z:A)} \prod_{(q:x=y)} \mathcal{U}$ — семейство типов $E(x, z, q) :\equiv (x = z)$. Заметим, что $E(x, x, \mathsf{refl}_x) \equiv (x = x)$. Таким образом, имеем функцию

$$e(x) :\equiv \mathsf{refl}_x : E(x, x, \mathsf{refl}_x)\,.$$

По принципу индукции для типов тождественности, примененного к E, получаем функцию

$$d : \prod_{(x,z:A)} \prod_{(q:x=z)} E(x, z, q)\,.$$

Но $E(x, z, q) \equiv (x = z)$, так что это (2.1.3). Таким образом, мы можем использовать эту функцию d и применить принцип индукции для типов тождественности к D, чтобы получить искомую функцию типа

$$\prod_{(x,y:A)} (x = y) \to \prod_{(z:A)} (y = z) \to (x = z)$$

и, следовательно, $\prod_{(x,y,z:A)}(x = y) \to (y = z) \to (x = z)$. Правила преобразования для двух принципов индукции дают $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$, для любого $x : A$. $\square$

Второе доказательство. Мы хотим построить для любых $x, y, z : A$ и любых $p : x = y$ и $q : y = z$ элемент $x = z$. По индукции по p достаточно считать, что y есть x, а p — refl_x. В этом случае тип $y = z$ из q равен $x = z$. Теперь по индукции по q достаточно предположить также, что z есть x, а q — refl_x. Но в этом случае $x = z$ есть $x = x$, и мы имеем $\mathsf{refl}_x : (x = x)$. $\square$

Читатель может почувствовать, что мы дали излишне подробное доказательство этой леммы. Фактически, мы можем остановиться после индукции по p, так как в этой точке мы хотим получить равенство $x = z$, и мы уже имеем такое равенство, а именно q. Почему же мы выполняем еще одну индукцию по q?

Ответ заключается в том, что, как описано во введении, мы создаем *доказательную* математику. Когда мы доказываем лемму, мы определяем обитателя некоторого типа, и может иметь значение, какой *конкретный* элемент мы определили в ходе доказательства, а не только тип, в котором этот элемент обитает (т.е. формулировка леммы). Лемма 2.1.2 имеет три очевидных доказательства: мы могли бы сделать индукцию по p, индукцию по q или индукцию по обоим из них. Если бы мы доказали это тремя различными способами, у нас было бы три разных элемента одного типа. Нетрудно показать, что эти три элемента равны (см. упражнение 2.1), но поскольку они не являются дефиницильно равными, все же могут быть причины предпочесть один другому.

В случае леммы 2.1.2 это различие зависит от правила вычисления. Если бы мы доказали лемму, используя единственную индукцию по p, то мы получили бы правило вычисления формы $\mathsf{refl}_y \cdot q \equiv q$. Если бы мы доказали эту лемму единственной индукцией по q, мы имели бы вместо этого $p \cdot \mathsf{refl}_y \equiv p$, в то время как доказательство двойной индукцией (как мы это сделали), дает $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$.

Асимметричные правила вычислений иногда могут быть удобными при следовании правилам формальной математики, поскольку они позволяют компьютеру провести больше упрощений автоматически. Однако в неформальной математике и, возможно, даже в формализованном случае, может возникнуть путаница с операцией конкатенации, которая ведет себя асимметрично и должна «помнить», какая сторона является «специальной». Симметричная обработка обеих сторон делает доказательства более надежными; вот почему мы дали доказательство того, что сделали (тем не менее, это, по общему признанию, стилистический выбор).

В приведенной ниже таблице суммируется то, что мы делали до сих пор, с точки зрения «равенства», «гомотопической» и «высших группоидов».

Равенство	Гомотопия	∞-Группоид
рефлексивность	постоянный путь	тождественный морфизм
симметричность	инверсия пути	обратный морфизм
транзитивность	конкатенация путей	композиция морфизмов

На практике транзитивность часто применяется для доказательства равенства цепью промежуточных этапов. Для этого мы будем использовать общие обозначения, такие как $a = b = c = d$. Если промежуточные выражения длинны или мы хотим указать обоснование каждого равенства, мы можем написать

$$\begin{aligned} a &= b && \text{(по } p\text{)} \\ &= c && \text{(по } q\text{)} \\ &= d && \text{(по } r\text{)} \end{aligned}$$

В любом случае эта нотация обозначает конструкцию элемента $(p \cdot q) \cdot r : (a = d)$ (мы использовали левую ассоциативность для конкретности, хотя в силу нижеприводимой леммы 2.1.4 (iv) это несущественно). Если должно быть, что b и c, скажем, дефинииально равны, то мы можем написать

$$\begin{aligned} a &= b && \text{(по } p\text{)} \\ &\equiv c && \\ &= d && \text{(по } r\text{)} \end{aligned}$$

для обозначения построения $p \cdot r : (a = d)$. Мы также следуем общепринятой математической практике, не требуя обоснований этих обозначений («по p» и «по r») для предоставления точного свидетельства; вместо этого мы позволяем им просто упомянуть о самой важной (или наименее очевидной) компоненте в построении этого свидетельства. Например, если «Лемма А» утверждает, что для всех x и у имеет место $f(x) = g(y)$, то мы можем записать «по лемме А» в качестве обоснования для шага $f(a) = g(b)$, доверяя читателю вывести, что мы применяем Лемму А с $x :\equiv a$ и $y :\equiv b$. Мы также можем полностью опустить обоснование, если мы верим, что читатель сам использует его.

Из-за доказательной релевантности мы не можем останавливаться после доказательства «симметрии» и «транзитивности» равенства: нам нужно убедиться, что эти *операции* над равенствами ведут себя предсказуемо (эта проблема не проявляется в теории множеств, где симметрия и транзитивность — это просто *свойства* равенства, а не структура на путях). С теоретико-гомотопической точки зрения конкатенация и инверсия — это всего лишь «первый уровень» высшей группоидной структуры — нам также необходимы законы согласованности этих операций и аналогичных операций при больших размерностях. Например, нам нужно знать, что конкатенация *ассоциативна*, и что инверсия обеспечивает *обратные связи* относительно конкатенации.

Лемма 2.1.4 (ω-группоидная структура типов)**.** *Предположим, что $A : \mathcal{U}$, $x, y, z, w : A$, $p : x = y$, $q : y = z$ и $r : z = w$. Тогда имеет место следующее*:

(i) $p = p \cdot \mathsf{refl}_y$ и $p = \mathsf{refl}_x \cdot p$.

(ii) $p^{-1} \cdot p = \mathsf{refl}_y$ и $p \cdot p^{-1} = \mathsf{refl}_x$.

(iii) $(p^{-1})^{-1} = p$.

(iv) $p \cdot (q \cdot r) = (p \cdot q) \cdot r$.

Отметим, в частности, что (i)-(iv) сами являются пропозициональными равенствами, обитающими в типах тождественности типов тождественности, таких как $p =_{x=y} q$ для $p, q : x = y$. Топологически это *пути путей*, т.е. гомотопии. В топологии известен факт, что, когда мы конкатенируем путь p с обратным путем p^{-1}, мы не получаем буквально постоянный путь (который соответствует равенству refl в теории типов) — вместо этого мы имеем гомотопию или путь более высокого порядка, от $p \cdot p^{-1}$ к постоянному пути.

Доказательство леммы 2.1.4. Все доказательства используют принцип индукции для равенств.
(i)
Первое доказательство: пусть $D : \prod_{(x,y:A)} \prod_{p:x=y} \mathcal{U}$ — семейство типов, заданое как

$$D(x, y, p) :\equiv (p = p \cdot \mathsf{refl}_y)\,.$$

Тогда $D(x, x, \mathsf{refl}_x)$ есть $\mathsf{refl}_x = \mathsf{refl}_x \cdot \mathsf{refl}_x$. Поскольку $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$, то отсюда следует, что $D(x, x, \mathsf{refl}_x) \equiv (\mathsf{refl}_x = \mathsf{refl}_x)$. Таким образом, существует функция

$$d :\equiv \lambda x.\, \mathsf{refl}_{\mathsf{refl}_x} : \prod_{x:A} D(x, x, \mathsf{refl}_x)\,.$$

Теперь принцип индукции для типов тождественности дает элемент $\mathsf{ind}_{=_A}(D, d, x, y, p) : (p = p \cdot \mathsf{refl}_y)$ для каждого $p : x = y$. Другое равенство доказывается аналогично.

Второе доказательство: Индукцией по p достаточно предположить, что y есть x, а p есть refl_x. Но в этом случае имеем $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$.
(ii)
Первое доказательство: пусть $D : \prod_{(x,y:A)} \prod_{p:x=y} \mathcal{U}$ — семейство типов, заданое как

$$D(x,y,p) :\equiv (p^{-1} \cdot p = \mathsf{refl}_y)\,.$$

Тогда $D(x,x,\mathsf{refl}_x)$ есть $\mathsf{refl}_x^{-1} \cdot \mathsf{refl}_x = \mathsf{refl}_x$. Поскольку $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$ и $\mathsf{refl}_x \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$, мы получаем, что $D(x,x,\mathsf{refl}_x) \equiv (\mathsf{refl}_x = \mathsf{refl}_x)$. Отсюда находим функцию

$$d :\equiv \lambda x.\, \mathsf{refl}_{\mathsf{refl}_x} : \prod_{x:A} D(x,x,\mathsf{refl}_x)\,.$$

Теперь индукция пути дает некоторый элемент $\mathsf{ind}_{=_A}(D,d,x,y,p) : p^{-1} \cdot p = \mathsf{refl}_y$ для каждого $p : x = y$ в A. Другое равенство получается аналогично.
Второе доказательство: По индукции достаточно предположить, что p есть refl_x. Но в этом случае мы имеем $p^{-1} \cdot p \equiv \mathsf{refl}_x^{-1} \cdot \mathsf{refl}_x \equiv \mathsf{refl}_x$.
(iii)
Первое доказательство: пусть $D : \prod_{(x,y:A)} \prod_{p:x=y} \mathcal{U}$ — семейство типов, заданое как

$$D(x,y,p) :\equiv (p^{-1^{-1}} = p)\,.$$

Тогда $D(x,x,\mathsf{refl}_x)$ есть тип$\left(\mathsf{refl}_x^{-1^{-1}} = \mathsf{refl}_x\right)$. Но, поскольку $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$ для каждого $x : A$, имеем $\mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x^{-1} \equiv \mathsf{refl}_x$, и поэтому $D(x,x,\mathsf{refl}_x) \equiv (\mathsf{refl}_x = \mathsf{refl}_x)$. Отсюда находим функцию

$$d :\equiv \lambda x.\, \mathsf{refl}_{\mathsf{refl}_x} : \prod_{x:A} D(x,x,\mathsf{refl}_x)\,.$$

Теперь индукция пути дает некоторый элемент $\mathsf{ind}_{=_A}(D,d,x,y,p) : p^{-1^{-1}} = p$ для каждого $p : x = y$.
Второе доказательство: По индукции достаточно предположить, что p есть refl_x. Но в этом случае имеем $p^{-1^{-1}} \equiv \mathsf{refl}_x^{-1^{-1}} \equiv \mathsf{refl}_x$.
(iv)
Первое доказательство: пусть $D_1 : \prod_{(x,y:A)} \prod_{(p:x=y)} \mathcal{U}$ — семейство типов, заданное следующим образом

$$D_1(x,y,p) :\equiv \prod_{(z,w:A)} \prod_{(q:y=z)} \prod_{(r:z=w)} \big(p \cdot (q \cdot r) = (p \cdot q) \cdot r\big)\,.$$

Тогда $D_1(x,x,\mathsf{refl}_x)$ есть

$$\prod_{(z,w:A)} \prod_{(q:y=z)} \prod_{(r:z=w)} \big(\mathsf{refl}_x \cdot (q \cdot r) = (\mathsf{refl}_x \cdot q) \cdot r\big)\,.$$

Для построения элемента этого типа, пусть $D_2 : \prod_{(x,z:A)} \prod_{(q:x=z)} \mathcal{U}$ семейство типов

$$D_2(x,z,q) :\equiv \prod_{(w:A)} \prod_{(r:z=w)} \big(\mathsf{refl}_x \cdot (q \cdot r) = (\mathsf{refl}_x \cdot q) \cdot r\big)\,.$$

Тогда $D_2(x, x, \mathsf{refl}_x)$ есть

$$\prod_{(w:A)} \prod_{(r:z=w)} (\mathsf{refl}_x \cdot (\mathsf{refl}_x \cdot r) = (\mathsf{refl}_x \cdot \mathsf{refl}_x) \cdot r)\,.$$

Для построения элемента последнего типа, пусть $D_3 : \prod_{(x,w:A)} \prod_{(r:x=w)} \mathcal{U}$ семейство типов

$$D_3(x, w, r) :\equiv (\mathsf{refl}_x \cdot (\mathsf{refl}_x \cdot r) = (\mathsf{refl}_x \cdot \mathsf{refl}_x) \cdot r)\,.$$

Тогда $D_3(x, x, \mathsf{refl}_x)$ есть

$$(\mathsf{refl}_x \cdot (\mathsf{refl}_x \cdot \mathsf{refl}_x) = (\mathsf{refl}_x \cdot \mathsf{refl}_x) \cdot \mathsf{refl}_x)$$

который по определению равен типу $(\mathsf{refl}_x = \mathsf{refl}_x)$, и поэтому он заселен $\mathsf{refl}_{\mathsf{refl}_x}$. Поэтому, применяя правило исключения тождественности три раза, мы получаем элемент общего желаемого типа.

Второе доказательство: по индукции достаточно положить, что все p, q и r есть refl_x. Но в этом случае мы имеем

$$\begin{aligned} p \cdot (q \cdot r) &\equiv \mathsf{refl}_x \cdot (\mathsf{refl}_x \cdot \mathsf{refl}_x) \\ &\equiv \mathsf{refl}_x \\ &\equiv (\mathsf{refl}_x \cdot \mathsf{refl}_x) \cdot \mathsf{refl}_x \\ &\equiv (p \cdot q) \cdot r \end{aligned}$$

Таким образом, мы имеем $\mathsf{refl}_{\mathsf{refl}_x}$, населяющий этот тип. □

Замечание 2.1.5. Существуют и другие способы определения этих высших путей. Например, в лемме 2.1.4 (iv) мы можем проводить индукцию только по одному или двум путям, а не по всем трем. Каждая возможность даст *дефинициально* отличное доказательство, но все они будут эквивалентны. Такое равенство между любыми двумя конкретными доказательствами можно, опять же, доказать индукцией, сводя все рассматриваемые пути к рефлексивности, а затем заметив, что оба доказательства сводятся к рефлексивности.

Ввиду леммы 2.1.4 (iv) мы будем часто писать $p \cdot q \cdot r$ для $(p \cdot q) \cdot r$, аналогично $p \cdot q \cdot r \cdot s$ для $((p \cdot q) \cdot r) \cdot s$ и т.д. (как отмечалось выше, мы используем левую ассоциативность для определенности).

Мы до сих пор не справились с высшей группоидной структурой: пути (i)-(iv) также должны удовлетворять своим собственным высшим законам согласованности, которые сами являются высшими путями и т.д. «вплоть до бесконечности» (это можно сделать более четко, используя, например, понятие глобулярной операды). Однако для большинства целей нет необходимости делать всю бесконечномерную структуру явной. Одна из хороших новостей о гомотопической теории типов заключается в том, что вся эта структура может быть *доказана*, отталкиваясь только от индуктивного свойства типов тождественности, поэтому мы можем сделать ее явной, как большую, так и малую, как нам требуется.

В частности, в этой книге нам не понадобится какая-либо сложная комбинаторика, участвующая в создании точных понятий, таких как «когерентная структура на всех высших уровнях». В дополнение к обычным путям мы будем использовать пути путей (т.е. элементы типа $p =_{x=_A y} q$), которые, как отмечалось ранее, мы называем 2-*путями* или 2-*мерными путями* и, возможно, иногда пути путей путей (т.е. элементов типа $r =_{p=_{x=_A y}q} s$), которые мы называем 3-*путями* или 3-*мерными путями*. Можно определить общее понятие n-*мерного пути* (см. упражнение 2.4), но оно нам не понадобится.

Однако мы будем использовать один особенно важный и простой случай высших путей, когда начальная и конечная точки одинаковы. В теории множеств высказывание $a = a$ совершенно неинтересно, но в теории гомотопий пути от точки к себе называются *петлями* и содержат много интересных высших структур. Таким образом, для заданного типа A с точкой $a : A$, мы определяем его **пространство петель** $\Omega(A, a)$ как тип $a =_A a$. Иногда можем писать просто ΩA, если точка a понятна из контекста.

Так как любые два элемента из ΩA — это пути с одинаковыми начальными и конечными точками, они могут быть объединены; таким образом, мы имеем операцию $\Omega A \times \Omega A \to \Omega A$. В более общем плане, высшая группоидная структура A является источником ΩA, аналогичной структуре «высшей группы».

Также полезно рассмотреть пространство петель *для* пространства петель A, являющееся пространством 2-мерных петель на тождественной петле в точке a. Это записывается как $\Omega^2(A, a)$ и представлено в теории типов типом $\mathsf{refl}_a =_{(a=_A a)} \mathsf{refl}_a$. В то время как $\Omega^2(A, a)$, как пространство петель, снова является «высшей группой», теперь она также имеет некоторую дополнительную структуру, связанную с тем, что ее элементы представляют собой 2-мерные петли между 1-мерными петлями.

Теорема 2.1.6 (Eckmann-Hilton)**.** *Операция композиции в пространстве 2-мерных петель*

$$\Omega^2(A) \times \Omega^2(A) \to \Omega^2(A)$$

является коммутативной: $\alpha \cdot \beta = \beta \cdot \alpha$*, для любых* $\alpha, \beta ; \Omega^2(A)$.

Доказательство. Во-первых, заметим, что композиция 1-петель $\Omega A \times \Omega A \to \Omega A$ индуцирует операцию

$$\star : \Omega^2(A) \times \Omega^2(A) \to \Omega^2(A)$$

следующим образом: рассмотрим элементы $a, b, c : A$ и 1- и 2-пути,

$$\begin{aligned}
p &: a = b, & r &: b = c, \\
q &: a = b, & s &: b = c, \\
\alpha &: p = q, & \beta &: r = s,
\end{aligned}$$

как показано на следующей диаграмме (с путями, изображенными стрелками).

$$\begin{tikzcd}
a \arrow[r, bend left, "p"] \arrow[r, bend right, "q"'] \arrow[r, phantom, "\Downarrow\alpha"] & b \arrow[r, bend left, "r"] \arrow[r, bend right, "s"'] \arrow[r, phantom, "\Downarrow\beta"] & c
\end{tikzcd}$$

Компонуя верхние и нижние 1-пути, соответственно, получаем два пути $p \cdot r, q \cdot s : a = c$ и тогда существует «горизонтальная композиция»

$$a \star b : p \cdot r = q \cdot s$$

между ними, определяемая следующим образом. Во-первых, определим $\alpha \cdot_{\mathsf{r}} r : p \cdot r = q \cdot r$ индукцией пути по r, так что

$$\alpha \cdot_{\mathsf{r}} \mathsf{refl}_b \equiv \mathsf{ru}_p^{-1} \cdot \alpha \cdot \mathsf{ru}_q$$

где $\mathsf{ru}_p : p = p \cdot \mathsf{refl}_b$ — закон правой единицы из Леммы 2.1.4.(i). Мы могли бы аналогичным образом определить $\cdot_{\mathsf{r}}$ индукцией по α, или по всем путям в поле зрения, что привело бы к разным дефинициальным равенствам, но для настоящих целей определение индукцией по r упростит ситуацию. Аналогично, мы определяем $q \cdot_{\mathsf{l}} \beta : q \cdot r = q \cdot s$ индукцией по q, так что

$$\mathsf{refl}_b \cdot_{\mathsf{l}} \beta \equiv \mathsf{lu}_r^{-1} \cdot \beta \cdot \mathsf{lu}_s\,,$$

где lu_r обозначает закон левой единицы. Операции $\cdot_{\mathsf{l}}$ и $\cdot_{\mathsf{r}}$ называются **вискерингом**. Далее, поскольку $\alpha \cdot_{\mathsf{r}} r$ и $q \cdot_{\mathsf{l}} \beta$ являются компонуемыми 2-путями, мы можем определить **горизонтальную композицию** так:

$$\alpha \star \beta :\equiv (\alpha \cdot_{\mathsf{r}} r) \cdot (q \cdot_{\mathsf{l}} \beta)\,.$$

Предположим теперь, что $a \equiv b \equiv c$, так что все 1-пути p, q, r, и s являются элементами $\Omega(A, a)$ и, кроме того, предположим, что $p \equiv q \equiv r \equiv s \equiv \mathsf{refl}_a$, так что $\alpha : \mathsf{refl}_a = \mathsf{refl}_a$ и $\beta : \mathsf{refl}_a = \mathsf{refl}_a$ компонуемы в обоих порядках. В этом случае имеем

$$\begin{aligned}
\alpha \star \beta &\equiv (\alpha \cdot_{\mathsf{r}} \mathsf{refl}_a) \cdot (\mathsf{refl}_a \cdot_{\mathsf{l}} \beta)\\
&= \mathsf{ru}_{\mathsf{refl}_a}^{-1} \cdot \alpha \cdot \mathsf{ru}_{\mathsf{refl}_a} \cdot \mathsf{lu}_{\mathsf{refl}_a}^{-1} \cdot \beta \cdot \mathsf{lu}_{\mathsf{refl}_a}\\
&\equiv \mathsf{refl}_{\mathsf{refl}_a}^{-1} \cdot \alpha \cdot \mathsf{refl}_{\mathsf{refl}_a} \cdot \mathsf{refl}_{\mathsf{refl}_a}^{-1} \cdot \beta \cdot \mathsf{refl}_{\mathsf{refl}_a}\\
&= \alpha \cdot \beta\,.
\end{aligned}$$

(напомним, что $\mathsf{ru}_{\mathsf{refl}_a} \equiv \mathsf{lu}_{\mathsf{refl}_a} \equiv \mathsf{refl}_{\mathsf{refl}_a}$ по правилу вычисления для индукции пути). С другой стороны, мы можем определить другую горизонтальную композицию аналогичным образом

$$\alpha \star' \beta :\equiv (p \cdot_{\mathsf{l}} \beta) \cdot (\alpha \cdot_{\mathsf{r}} s)$$

и мы также знаем, что

$$\alpha \star' \beta = \beta \cdot \alpha\,.$$

Но вообще, два способа определения горизонтальной композиции согласуются, $\alpha \star \beta = \alpha \star' \beta$, как мы можем видеть индукцией по a и b, а затем на двух оставшихся 1-путях, сводя все к рефлексивности. Таким образом, имеем

$$\alpha \cdot \beta = \alpha \star \beta = \alpha \star' \beta = \beta \cdot \alpha\,.$$

□

Вышеупомянутый факт, известный как *аргумент Экманна-Хилтона*, исходит из классической гомотопической теории, и действительно, он используется далее в главе 8, чтобы показать, что высшие гомотопические группы типа всегда являются абелевыми группами. Операции вискеринга и горизонтальной композиции, определенные в доказательстве, также являются общей частью ∞-группоидной структуры типов. Они удовлетворяют своим собственным законам (с точностью до высшей гомотопии), таким как

$$\alpha \cdot_{\mathsf{r}} (p \cdot q) = (\alpha \cdot_{\mathsf{r}} p) \cdot_{\mathsf{r}} q$$

и т.д. С этого момента мы надеемся, что читатель сможет применять индукцию пути, когда это необходимо, для определения дальнейших операций такого рода и проверки их свойств.

Как следует из этого примера, алгебра типов высших путей намного сложнее, чем просто группоидоподобная структура на каждом уровне; уровни взаимодействуют, давая много дополнительных операций и законов, как при рассмотрении итерированных пространств петель в теории гомотопий. Действительно, как и в классической гомотопической теории, мы можем дать следующие общие определения.

Определение 2.1.7. **Точечный тип** (A, a) является типом $A : U$ вместе с точкой $a : A$, называемой его **отмеченной точкой**. Будем записывать $\mathcal{U}_\bullet :\equiv \sum_{(A:U)} A$ для типа точечных типов в универсуме $\mathcal{U}$.

Определение 2.1.8. Для точечного типа (A, a) мы определяем **пространство петель** (A, a) следующим точечным типом:

$$\Omega(A, a) :\equiv ((a =_A a), \mathsf{refl}_a)\,.$$

Его элемент будет называться **петлей** в точке a. Для $n : \mathbb{N}$ **пространство n-кратных итерированных петель** $\Omega^n(A, a)$ точечного типа (A, a) определяется рекурсивно:

$$\Omega^0(A, a) :\equiv (A, a)$$
$$\Omega^{n+1}(A, a) :\equiv \Omega^n(\Omega(A, a))\,.$$

Его элемент будет называться **n-петлей** или **n-мерной петлей** при a.

Мы вернемся к пространствам итерированных петель в главах 6-8.

2.2 Функции являются функторами

Теперь мы хотим показать, что функции $f : A \to B$ ведут себя на путях функториально. В традиционной теории типов это эквивалентно утверждению, что функции не нарушают равенств. Топологически это соответствует тому, что каждая функция является «непрерывной», т.е. сохраняет пути.

Лемма 2.2.1. *Предположим, что $f : A \to B$ — функция. Тогда для любых $x, y : A$ существует операция*

$$\mathsf{ap}_f : (x =_A y) \to (f(x) =_B f(y))\,.$$

Более того, для любого $x : A$ имеем $\mathsf{ap}_f(\mathsf{refl}_x) \equiv \mathsf{refl}_{f(x)}$.

Обозначение ap_f может быть истолковано либо как применение (application) f к пути, либо как действие (action) на путях (paths) f.

Доказательство. (*первое*) Пусть $D : \prod_{(x,y:A)}(x = y) \to \mathcal{U}$ — семейство типов, определенное посредством

$$D(x, y, p) :\equiv (f(x) = f(y))\,.$$

Тогда имеем

$$d :\equiv \lambda x.\, \mathsf{refl}_{f(x)} : \prod_{x:A} D(x, x, \mathsf{refl}_x)\,.$$

По индукции пути получаем $\mathsf{ap}_f : \prod_{(x,y:A)}(x = y) \to (f(x) = f(y))$. Из правила вычисления следует $\mathsf{ap}_f(\mathsf{refl}_x) \equiv \mathsf{refl}_{f(x)}$ для любого $x : A$. □

Доказательство. (*второе*) По индукции достаточно предположить, что p есть refl_x. В этом случае мы можем определить $\mathsf{ap}_f(p) :\equiv \mathsf{refl}_{f(x)} : f(x) = f(x)$. □

Мы будем часто записывать $\mathsf{ap}_f(p)$ как просто $f(p)$. Это, строго говоря, двусмысленно, но вообще-то, путаницы не возникает. Такая запись соответствует общему соглашению в теории категорий использования одного и того же символа для применения функтора к объектам и к морфизмам.

Заметим, что ap ведет себя функториально на всех путях, чего и можно было ожидать.

Лемма 2.2.2. *Для функций $f : A \to B$ и $g : B \to C$, путей $p : x =_A y$ и $q : y =_A z$ имеет место следующее:*

(i) $\mathsf{ap}_f(p \cdot q) = \mathsf{ap}_f(p) \cdot \mathsf{ap}_f(q)$.

(ii) $\mathsf{ap}_f(p^{-1}) = \mathsf{ap}_f(p)^{-1}$.

(iii) $\mathsf{ap}_g(\mathsf{ap}_f(p)) = \mathsf{ap}_{g \circ f}(p)$.

(iv) $\mathsf{ap}_{\mathsf{id}_A}(p) = p$.

Доказательство. Предоставляется читателю. □

Как и в случае равенств в лемме 2.1.4, они в лемме 2.2.2 сами по себе являются путями, которые удовлетворяют своим собственным законам когерентности (что доказывается аналогично) и т.д.

2.3 Семейства типов являются расслоениями

Так как в теории типов существенны *зависимые типизированные* функции, для них нам также понадобится версия леммы 2.2.1. Однако это не так просто потому, что, если $f : \prod_{(x:A)} B(x)$ и $p : x = y$, то $f(x) : B(x)$ и $f(y) : B(y)$ являются элементами разных типов, поэтому даже априори мы не можем требовать их равенства. Недостающим компонентом является то, что сама p дает возможность связать типы $B(x)$ и $B(y)$.

Мы уже сталкивались с этим в разделе 1.12, где назвали эту ситуацию «неразличимостью тождественности». Теперь введем другое название и обозначение, которые будем использовать далее.

Лемма 2.3.1 (транспортирование)**.** *Допустим, что P — семейство типов над A и что $p : x =_A y$. Тогда существует функция $p_* : P(x) \to P(y)$.*

Доказательство. (*первое*) Пусть $D : \prod_{(x,y:A)} (x = y) \to \mathcal{U}$ — семейство типов, определенное посредством

$$D(x, y, p) :\equiv P(x) \to P(y) .$$

Тогда имеем функцию

$$d :\equiv \lambda x.\, \mathsf{id}_{P(x)} : \prod_{x:A} D(x, x, \mathsf{refl}_x) .$$

Поэтому, принцип индукции дает $\mathsf{ind}_{=_A}(D, d, x, y, p) : P(x) \to P(y)$, для $p : x = y$, которое мы и определяем в качестве p_*. □

Доказательство. (*второе*) По индукции достаточно положить p равным refl_x. Но в этом случае можно считать $(\mathsf{refl}_x)_* : P(x) \to P(x)$ тождественной функцией. $\square$

Иногда необходимо указать тип семейства P, в котором имеет место операция транспортировки. В этом случае мы можем записать

$$\mathsf{transport}^P(p, \_) : P(x) \to P(x)\,.$$

Напомним, что семейство типов P над типом A можно рассматривать как свойство элементов из A, которое выполняется в точке x из A, если $P(x)$ обитаемо. Тогда в лемме транспортирования говорится, что P соответствует равенству в том смысле, что если x равно y, то $P(x)$ выполняется тогда и только тогда, когда выполняется $P(y)$. В самом деле, мы увидим позже, что если $x = y$, то фактически $P(x)$ и $P(y)$ *эквивалентны.*

Топологически, лемма транспортировки может рассматриваться как операция «поднятия пути» в расслоении. Мы думаем о типе семейства $P : A \to \mathcal{U}$ как о *расслоении* с пространством расслоения A, причем $P(x)$является слоем над x, и с $\sum_{(x:A)} P(x)$, являющимся **полным пространством** расслоения с первой проекцией $\sum_{(x:A)} P(x) \to A$. Определяющим свойством расслоения является то, что при заданном пути $p : x = y$ в пространстве расслоения A и точке $u : P(x)$ в слое над x, мы можем поднять путь p к пути в полном пространстве, начинающееся с u. Точку $p_*(u)$ можно рассматривать как другую, конечную точку этого поднятого пути. Мы также можем определить сам путь в теории типов:

Лемма 2.3.2 (свойство поднятия пути). *Пусть $P : A \to \mathcal{U}$ — семейство типов над A и предположим, что $u : P(x)$ для некоторого $x : A$. Тогда для любого $p : x = y$ имеем:*

$$\mathsf{lift}(u, p) : (x, u) = (y, p_*(u))$$

в $\sum_{(x:A)} P(x)$, так что $\mathsf{pr}_1(\mathsf{lift}(u, p)) = p$.

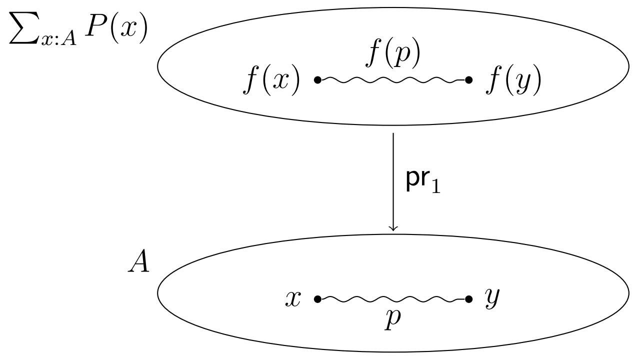
Доказательство. Предоставляется читателю. Мы докажем более общую теорему в §2.7. $\square$

В классической теории гомотопий расслоение определяется как отображение, для которого *существуют* поднятия путей; в отличие от этого, мы только что показали, что в теории типов каждое семейство типов имеет *заданную* «функцию поднятия пути». Это согласуется с философией конструктивной математики, согласно которой мы не можем показать, что что-то существует, кроме как путем его предъявления. При этом также автоматически гарантируется, что поднятия путей выбираются «непрерывно», так как, как мы видели, все функции теории типов «непрерывны».

Замечание 2.3.3. Хотя мы можем думать о семействе типов $P : A \to \mathcal{U}$, как о расслоении, обычно не рекомендуется употреблять такие фразы, как «расслоение $P : A \to \mathcal{U}$», так как это звучит так, как будто речь идет о расслоении с базой $\mathcal{U}$ и пространством расслоений A. Акцентируем, когда семейство типов $P : A \to \mathcal{U}$ рассматривается как расслоение, базой является A, а пространство расслоений — это $\sum_{(x:A)} P(x)$.

Иногда мы можем использовать другую топологическую терминологию, говоря о типах семейств. Например, мы можем ссылаться на зависимую функцию $f : \prod_{(x:A)} P(x)$ как на **сечение** расслоения P, и мы можем говорить, что что-то происходит **послойно**, если оно происходит для каждого $P(x)$. Например, сечение $f : \prod_{(x:A)} P(x)$ показывает, что P «послойно обитаемо».

Теперь мы можем доказать зависимую версию леммы 2.2.1. Топологическая интуиция такова, что для $f : \prod_{(x:A)} P(x)$ и пути $p : x =_A y$ мы должны были бы применить f к p и получить путь в пространстве расслоений P, который «лежит над» p, как показано на диаграмме:



Мы *можем* получить это из леммы 2.2.1. Для $f : \prod_{(x:A)} P(x)$ можно определить независимую функцию $f' : A \to \sum_{(x:A)} P(x)$, полагая $f'(x) :\equiv (x, f(x))$, а затем учитывая, что $f'(p) : f'(x) = f'(y)$. Поскольку $\mathsf{pr}_1 \circ f' \equiv \mathsf{id}_A$, по лемме 2.2.2 имеем $\mathsf{pr}_1(f'(p)) = p$; таким образом, $f'(p)$ «лежит над» p в этом смысле. Однако из типа такого пути не очевидно, что он лежит над определенным путем в A (в данном случае, p), что иногда важно.

Решение состоит в использовании леммы транспортировки. Так как существует канонический путь от $u : P(x)$ до $p_*(u) : P(y)$, который (по крайней мере интуитивно) лежит над p, любой путь от u к $v : P(y)$, лежащий над p, должен быть фактором по $\mathsf{lift}(u, p)$, по существу, однозначным образом, согласно пути от $p_*(u)$ к v, целиком лежащего в слое $P(y)$. Таким образом, с точностью до эквивалентности, имеет смысл определить «путь от u к v, лежащий над $p : x = y$», подразумевая путь $p_*(u) = v$ в $P(y)$. И действительно, мы можем показать, что зависимые функции порождают такие пути.

Лемма 2.3.4 (зависимое отображение). *Предположим $f : \prod_{(x:A)} P(x)$; тогда имеется отображение*

$$\mathsf{apd}_f : \prod_{p:x=y} \big(p_*(f(x)) =_{P(x)} f(y)\big) \,.$$

Доказательство. (*первое*) Пусть $D : \prod_{(x,y:A)} (x = y) \to \mathcal{U}$ — семейство типов, определенное как

$$D(x, y, p) :\equiv p_*(f(x)) = f(y) \,.$$

Тогда $D(x, x, \mathsf{refl}_x)$ есть $(\mathsf{refl}_x)_*(f(x)) = f(x)$. Но, поскольку $(\mathsf{refl}_x)_*(f(x)) \equiv f(x)$, то имеем $D(x, x, \mathsf{refl}_x) \equiv (f(x) = f(x))$. Таким образом, мы получаем функцию

$$d :\equiv \lambda x.\, \mathsf{refl}_{f(x)} : \prod_{x:A} D(x, x, \mathsf{refl}_x)$$

и теперь индукция пути дает нам $\mathsf{apd}_f(p) : p_*(f(x)) = f(y)$ для каждого $p : x = y$. □

Доказательство. (*второе*) По индукции достаточно считать, что p есть refl_x. Но в этом случае искомым уравнением является $(\mathsf{refl}_x)_*(f(x)) \equiv f(x)$, которое выполняется дефинициально. □

Мы будем иметь дело в основном с путями, которые «находятся над другим путям», в этом смысле, с *зависимыми путями*. Они будут играть все более важную роль, начиная с главы 6. В §2.5 мы увидим, что для некоторых отдельных видов семейств типов существуют эквивалентные способы представления понятия зависимых путей, некоторые из которых иногда более предпочтительны.

Теперь напомним из §1.4, что независимая от типа функция $f : A \to B$ является лишь частным случаем зависимой от типа функции $f : \prod_{(x:A)} P(x)$, когда P — семейство с постоянным типом, $P(x) :\equiv B$. В этом случае apd_f и ap_f тесно связаны между собой по следующей лемме:

Лемма 2.3.5. *Если $P : A \to \mathcal{U}$ определяется через $P(x) :\equiv B$ для фиксированного $B : \mathcal{U}$, то для любых $x, y : A$, $p : x = y$ и $b : B$ имеется путь*

$$\mathsf{transportconst}^B_p(b) : \mathsf{transport}^P(p, b) = b\,.$$

Доказательство. (*первое*) Зафиксируем $b : B$, и пусть $D : \prod_{(x,y:A)}(x = y) \to \mathcal{U}$ — семейство типов, пределяемое как

$$D(x, y, p) :\equiv (\mathsf{transport}^P(p, b) = b)\,.$$

Тогда $D(x, x, \mathsf{refl}_x)$ есть $(\mathsf{transport}^P(\mathsf{refl}_x, b) = b)$, которое дефинициально равно $(b = b)$ по правилу вычисления для транспортировки. Таким образом, имеем функцию

$$d :\equiv \lambda x.\, \mathsf{refl}_b : \prod_{x:A} D(x, x, \mathsf{refl}_x)\,.$$

Теперь индукция пути дает нам элемент из $\prod_{(x,y:A)} \prod_{(p:x=y)}(\mathsf{transport}^P(p, b) = b)$, что и требовалось. □

Доказательство. (*второе*) По индукции достаточно считать, что y есть x, а p есть refl_x. Но $\mathsf{transport}^P(\mathsf{refl}_x, b) \equiv b$, поэтому в данном случае нам нужно доказать, что $b = b$, а для этого мы имеем refl_b. □

Таким образом, для любых $x, y : A$, $p : x = y$ и $f : A \to B$ путем конкатенации с $\mathsf{transportconst}^B_p(b)$ и его обратным, получаем функции, соответтвенно

$$(f(x) = f(y)) \to (p_*(f(x)) = f(y))\,, \tag{2.3.6}$$

$$(p_*(f(x)) = f(y)) \to (f(x) = f(y))\,. \tag{2.3.7}$$

Фактически эти функции являются обратными эквивалентностями (в смысле, который будет введен в §2.4) и они связывают $\mathsf{ap}_f(p)$ с $\mathsf{apd}_f(p)$.

Лемма 2.3.8. *Для $f : A \to B$ и $p : x =_A y$ имеем*

$$\mathsf{apd}_f(p) = \mathsf{transportconst}^B_p(f(x)) \cdot \mathsf{ap}_f(p)\,.$$

Доказательство. (*первое*) Пусть $D : \prod_{(x,y:A)}(x = y) \to \mathcal{U}$ — семейство типов, пределяемое как

$$D(x, y, p) :\equiv (\mathsf{apd}_f(p) = \mathsf{transportconst}^B_p(f(x)) \cdot \mathsf{ap}_f(p))\,.$$

Таким образом, имеем

$$D(x, x, \mathsf{refl}_x) :\equiv (\mathsf{apd}_f(\mathsf{refl}_x) = \mathsf{transportconst}^B_{\mathsf{refl}_x}(f(x)) \cdot \mathsf{ap}_f(\mathsf{refl}_x)) .$$

Но, по определению, каждый из трех путей в этом типе есть $\mathsf{refl}_{f(x)}$, так что, имеем

$$\mathsf{refl}_{\mathsf{refl}_{f(x)}} : D(x, x, \mathsf{refl}_x) .$$

Теперь индукция пути дает нам элемент из $\prod_{(x,y:A)} \prod_{(p:x=y)} D(x, y, p)$, что и требовалось. □

Доказательство. (*второе*) По индукции достаточно считать, что y есть x, а p есть refl_x. В этом случае мы должны доказать, что $\mathsf{refl}_{f(x)} = \mathsf{refl}_{f(x)} \cdot \mathsf{refl}_{f(x)}$, а это истинно дефинициально. □

Поскольку типы apd_f и ap_f различны, часто бывает проще использовать для них разные обозначения.

На этом этапе мы надеемся, что читатель начинает ощущать доказательства по индукции на типах тождественности. Отныне мы перестанем давать оба стиля доказательств, позволяя себе использовать то, что наиболее ясно и удобно (и часто второе, как более сжатое). Приведем еще несколько полезных лемм о транспортировке; мы оставляем читателю их доказательства (в любом стиле).

Лемма 2.3.9. *Для $P : A \to \mathcal{U}$ с $p : x =_A y$ и $q : y =_A z$, пока $u : P(x)$, имеем*

$$q_*(p_*(u)) = (p \cdot q)_*(u) .$$

Лемма 2.3.10. *Для функции $f : A \to B$, семейства типов $P : B \to \mathcal{U}$, любых $p : x =_A y$ и $u : P(x)$, имеем*

$$\mathsf{transport}^{P \circ f}(p, u) = \mathsf{transport}^P(\mathsf{ap}_f(p), u) .$$

Лемма 2.3.11. *Для $P, Q : A \to \mathcal{U}$, семейства типов $f : \prod_{(x:A)} P(x) \to Q(x)$, любых $p : x =_A y$ и $u : P(x)$, имеем*

$$\mathsf{transport}^Q(p, f_x(u)) = f_y(\mathsf{transport}^P(p, u)) .$$

2.4 Гомотопии и эквивалентности

До сих пор мы рассматривали, каким образом тип тождественности $x =_A y$ можно считать типом *идентификаций*, *путей* или *эквивалентностей* между двумя элементами x и y типа A. Теперь мы исследуем соответствующие понятия «отождествление» или «тождественность» между *функциями* и между *типами*. В §§ 2.9 и 2.10 мы увидим, что теория гомотопического типа позволяет отождествлять их с экземплярами типа тождественности, но прежде чем мы сможем сделать это, нам нужно разобраться с ними как с самостоятельными сущностями.

Традиционно мы считаем две функции одинаковыми, если они принимают равные значения на всех входных значениях рагументов. В соответствии с интерпретацией высказываний как типов это предполагает, что две функции f и g (возможно, в зависимости от типа) должны быть одинаковыми, если тип $\prod_{(x:A)}(f(x) = g(x))$ обитаем. При гомотопической интерпретации этот зависимый функциональный тип состоит из *непрерывных* путей или *функториальных* эквивалентностей и, следовательно, может рассматриваться как тип *гомотопий* или *естественных изоморфизмов*. Мы примем для этого топологическую терминологию.

Определение 2.4.1. Пусть $f, g : \prod_{(x:A)} P(x)$ — два сечения семейства типов $P : A \to \mathcal{U}$. **Гомотопия** от f к g — это зависимый функциональный тип

$$(f \sim g) :\equiv \prod_{x:A} (f(x) = g(x)) .$$

Заметим, что гомотопия не совпадает с отождествлением $(f = g)$. Однако, в §2.9 мы введем аксиому, делающую гомотопии и отождествления «эквивалентными».

Доказательство следующего утверждения предоставляется читателю.

Лемма 2.4.2. *Гомотопия является отношением эквивалентности для каждой функционального типа $\prod_{(x:A)} P(x)$. То есть, имеются элементы типов*

$$\prod_{f:\prod_{(x:A)} P(x)} (f \sim f)$$

$$\prod_{f,g:\prod_{(x:A)} P(x)} (f \sim g) \to (g \sim f)$$

$$\prod_{f,g,h:\prod_{(x:A)} P(x)} (f \sim g) \to (g \sim h) \to (f \sim h) .$$

Подобно тому, как функции в теории типов автоматически являются «функторами», гомотопии автоматически являются «естественными преобразованиями». Мы сформулируем и докажем это только для независимых функций $f, g : A \to B$; в упражнении читателю предлагается сделать обобщение на зависимые функции.

Напомним, что для $f : A \to B$ и $p : x =_A y$, можно писать $f(p)$ для обозначения $\mathsf{ap}_f(p)$.

Лемма 2.4.3. *Предположим, что $H : f \sim g$ является гомотопией между функциями $f, g : A \to B$ и пусть $p : x =_A y$. Тогда*

$$H(x) \cdot g(p) = f(p) \cdot H(y) .$$

Это иллюстрирует следующая коммутативная диаграмма:

$$\begin{tikzcd}
f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\
g(x) \arrow[r, equal, "g(p)"'] & g(y)
\end{tikzcd}$$

Доказательство. По индукции можно считать, что p есть refl_x. Поскольку вычисление ap_f и ap_g рефлексивно, то в этом случае мы должны показать, что

$$H(x) \cdot \mathsf{refl}_{g(x)} = \mathsf{refl}_{f(x)} \cdot H(x) .$$

Но это следует из эквивалентности $H(x)$ обеим частям этого выражения. $\square$

Следствие 2.4.4. *Пусть $H : f \sim \mathsf{id}_A$ — гомотопия с $f : A \to A$. Тогда для любого $x : A$*

$$H(f(x)) = f(H(x))\,.$$

Здесь $f(x)$ обозначает обычное применение f к x, а $f(H(x))$ обозначает $\mathsf{ap}_f(H(x))$.

Доказательство. По причине естественности H следующая диаграмма путей коммутативна:

$$\begin{tikzcd}
ffx \arrow[r, equal, "f(Hx)"] \arrow[d, equal, "H(fx)"'] & fx \arrow[d, equal, "Hx"] \\
fx \arrow[r, equal, "Hx"'] & x
\end{tikzcd}$$

То есть $f(Hx) \cdot Hx = H(fx) \cdot Hx$. Теперь мы можем, используя $(Hx)^{-1}$, избавиться от Hx, получая

$$f(Hx) = f(Hx) \cdot Hx \cdot (Hx)^{-1} = H(fx) \cdot Hx \cdot (Hx)^{-1} = H(fx)$$

что и трбовалось (с подавлением некоторых путей ассоциативности). □

Разумеется, как и в случае функториальности функций (лемма 2.2.2), равенство в лемме 2.4.3 является путём, удовлетворяющим его собственным законам когерентности и т.д.

Переходя к типам, с традиционной точки зрения можно сказать, что функция $f : A \to B$ является *изоморфизмом*, если существует функция $g : B \to A$, так что обе композиции $f \circ g$ и $g \circ f$ поточечно равны единице, то есть такие, что $f \circ g \sim \mathsf{id}_B$ и $g \circ f \sim \mathsf{id}_A$. Гомотопическая перспектива предполагает, что эту ситуацию следует называть *гомотопической эквивалентностью*, а с категорной точки зрения ее следует назвать *эквивалентностью* (*высших*) *группоидов*. Однако при проведении доказательств, соответствующий тип

$$\sum_{g:B \to A} ((f \circ g \sim \mathsf{id}_B) \times (g \circ f \sim \mathsf{id}_A)) \tag{2.4.5}$$

обладает плохим поведением. Например, для одной функции $f : A \to B$ могут существовать несколько не одинаковых обитателей (2.4.5) (это тесно связано с наблюдением в теории высших категорий, что часто требуется рассматривать не простые, а *сопряженные* эквивалентности). По этой причине мы даем типу (2.4.5) следующее исторически точное, но немного уничижительное звучание.

Определение 2.4.6. Для функции $f : A \to B$ ее **квазиобратной** является тройка (g, α, β), содержащая функцию $g : B \to A$ и гомотопии $f \circ g \sim \mathsf{id}_B$ и $g \circ f \sim \mathsf{id}_A$.

Таким образом, (2.4.5) есть *тип квазиобратной к* f; мы обозначаем ее $\mathsf{qinv}(f)$.

Пример 2.4.7. Тождественная функция $\mathsf{id}_A : A \to A$ является квазиобратной сама к себе с гомотопиями $\alpha(y) :\equiv \mathsf{refl}_y$ и $\beta(x) :\equiv \mathsf{refl}_x$.

Пример 2.4.8. Для любых $p : x =_A y$ и $z : A$ функции

$$(p \cdot \_) : (y =_A z) \to (x =_A z)$$
$$(\_ \cdot p) : (z =_A x) \to (z =_A y)$$

имеют квазиобратные, задаваемые как $(p^{-1} \cdot \_)$ и $(\_ \cdot p^{-1})$, соответственно; см. упражнение 2.6.

Пример 2.4.9. Для любых $p : x =_A y$ и $P : A \to \mathcal{U}$ функция

$$\mathsf{transport}^P(p, \_) : P(x) \to P(y)$$

имеет квазиобратную, задаваемую как $\mathsf{transport}^P(p^{-1}, \_)$; это следует из леммы 2.3.11.

Вообще, мы будем использовать слово *изоморфизм* (и подобные слова, такие как *биекция*, с соответствующим обозначением $A \cong B$) в частном случае, когда типы A и B «ведут себя как множества» (см. §3.1). В этом случае тип (2.4.5) не является проблематичным. Мы зарезервируем слово *эквивалентность* для усовершенствованного понятия $\mathsf{isequiv}(f)$ со следующими свойствами:

(i) для любой функции $f : A \to B$ существует функция $\mathsf{qinv}(f) \to \mathsf{isequiv}(f)$;

(ii) аналогично, для любой f имеем $\mathsf{isequiv}(f) \to \mathsf{qinv}(f)$; таким образом, оба свойства логически эквивалентны (см. §1.11);

(iii) для любых двух обитателей $e_1, e_2 : \mathsf{isequiv}(f)$ имеем $e_1 = e_2$.

В главе 4 мы увидим, что существует множество различных определений $\mathsf{isequiv}(f)$, которые удовлетворяют этим трем свойствам, но все они эквивалентны. Пока же, чтобы убедить читателя, что такие существуют, мы упомянем только самое легкое такое определение:

$$\mathsf{isequiv}(f) :\equiv \left(\sum_{g:B\to A} (f \circ g \sim \mathsf{id}_B)\right) \times \left(\sum_{h:B\to A} (h \circ f \sim \mathsf{id}_A)\right). \tag{2.4.10}$$

Мы можем продемонстрировать (i) и (ii) для этого определения. Функцию $\mathsf{qinv}(f) \to \mathsf{isequiv}(f)$ легко определить, если взять (g, α, β) в (g, α, g, β). В другом направлении, (g, α, h, β), пусть γ — составная гомотопия

$$g \overset{\beta}{\sim} h \circ f \circ g \overset{\alpha}{\sim} h$$

означающая, что $\gamma(x) :\equiv \beta(g(x)) \cdot h(\alpha(x))$. Теперь определим $\beta' : g \circ f \sim \mathsf{id}_A$ с помощью $\beta'(x) :\equiv \gamma(f(x)) \cdot \beta(x)$. Тогда $(g, \alpha, \beta') : \mathsf{qinv}(f)$.

Свойство (iii) для этого определения не слишком сложно доказать, но оно требует идентификации тождественных типов декартовых произведений и зависимых типов пар, которые мы обсудим в §§ 2.6 и 2.7. Таким образом, мы также откладываем и доказательство этого свойства; см. §4.3. Сейчас же главное, это то, что существует хорошо подобранный тип, который мы можем декларировать как «f — эквивалентность», и что мы можем доказать, что f является эквивалентностью, привлекая для этого квазиобратность. На практике это самый распространенный способ доказать, что функция является эквивалентностью.

В соответствии с доказательной философией, *эквивалентность* от A к B определяется как функция $f : A \to B$ вместе с обитателем $\mathsf{isequiv}(f)$, т.е. доказательством этой эквивалентности. Мы пишем $(A \simeq B)$ для типа эквивалентностей от A к B, т.е. типа

$$(A \simeq B) :\equiv \sum_{f:A\to B} \mathsf{isequiv}(f). \tag{2.4.11}$$

Вышеупомянутое свойство (iii) гарантирует, что если две эквивалентности равны как функции (т.е. лежащие в основе элементы из $A \to B$ равны), то они также равны как эквивалентности (см. §2.7). Таким образом, мы часто злоупотребляем нотацией и размываем различие между эквивалентностями и лежащими в их основе функциями. Например, если у нас есть функция $f : A \to B$, и мы знаем, что $e : \mathsf{isequiv}(f)$, мы можем записать $f : A \simeq B$, а не (f, e). Или, наоборот, если мы имеем эквивалентность $g : A \simeq B$, то можем написать $g(a)$, если дано $a : A$, но не $(\mathsf{pr}_1 g)(a)$.

Мы заканчиваем наблюдением:

Лемма 2.4.12. *Эквивалентность типов является отношением эквивалентности на $\mathcal{U}$. Более конкретно*:

(i) *для любого A тождественная функция id_A является эквивалентностью*; *отсюда $A \simeq A$*;

(ii) *для любой $f : A \simeq B$ имеется эквивалентность $f^{-1} : B \simeq A$*;

(iii) *для любых $f : A \simeq B$ и $g : B \simeq C$ имеет место $g \circ f : A \simeq C$.*

Доказательство. Очевидно, что функция тождества является квазиобратной себе; следовательно, она является эквивалентностью.

Если $f : A \to B$ — эквивалентность, то она имеет квазиобратную, скажем $f^{-1} : B \to A$. Тогда f также является квазиобратной к f^{-1}, так что f^{-1} есть эквивалентность $B \to A$.

Наконец, для $f : A \simeq B$ и $g : B \simeq C$ с квазиобратными f^{-1} и g^{-1}, тогда для любого $a : A$ имеем $f^{-1}g^{-1}gfa = f^{-1}fa = a$, а для любого $C : C$ — $gff^{-1}g^{-1}c = gg^{-1}c = c$. Так что $f^{-1} \circ g^{-1}$ есть квазиобратное к $g \circ f$, следовательно последнее есть эквивалентность. $\square$

2.5 Высшая группоидная структура конструкторов типов

В главе 1 мы ввели ряд способов формирования новых типов: декартовых произведений, несвязных объединений, зависимых произведений, зависимых сумм и т.д. В §§ 2.1-2.3 мы видели, что *все* типы в гомотопической теории типов ведут себя как пространства или высшие группоиды. Наша цель в оставшейся части главы — тщательно разобраться, как ведет себя эта высшая структура в случае конкретных типов, определенных в главе 1.

Оказывается, для многих типов A типы равенства $x =_A y$ могут быть охарактеризованы с точностью до эквивалентности в терминах любых данных, которые использовались для построения A. Например, если A является декартовым произведением $B \times C$, а $x \equiv (b, c)$ и $y \equiv (b', c')$, то мы имеем эквивалентность

$$((b, c) = (b', c')) \simeq ((b = b') \times (c = c')) . \tag{2.5.1}$$

На более традиционном языке две упорядоченные пары равны, когда их компоненты равны (но эквивалентность (2.5.1) говорит, пожалуй, больше этого). Высшая структура типов тождественности также может быть выражена через эти эквивалентности; например, объединение двух равенств между парами соответствует попарной конкатенации.

Аналогично, когда семейство типов $P : A \to \mathcal{U}$ строится послойно, используя правила формирования типов из главы 1, операция $\mathsf{transport}^P(p, \_)$ может быть охарактеризована с точностью до гомотопии в терминах соответствующих операций над данными, которые входят в P. Например, если $P(x) \equiv B(x) \times C(x)$, то имеем

$$\mathsf{transport}^P(p, (b, c)) = (\mathsf{transport}^B(p, b), \mathsf{transport}^C(p, c)) .$$

Наконец, правила формирования типов к тому же функториальны, и если функция f строится с учетом этой функториальности, то операции ap_f и apd_f могут быть вычислены на основе соответствующих функториальностей на данных, входящих в f. Например, если $g : B \to B'$ и $h : C \to C'$ и мы определяем $f : B \times C \to B' \times C'$ через $f(b, c) :\equiv (g(b), h(c))$, то по модулю эквивалентности (2.5.1) мы можем отождествить ap_f с «$(\mathsf{ap}_g, \mathsf{ap}_h)$».

Следующие несколько разделов (§§ 2.6-2.13) будут посвящены такого рода теоремам формулировки и доказательства для всех правил формирования базового типа с отдельным рассмотрением каждого базового конструктора типа. Здесь мы сталкиваемся с определенным явным недостатком в доступных ныне теориях типов; как станет ясно в последующих главах, казалось бы, было бы более удобным и интуитивным, если бы эти характеристики типов тождественности, транспортировки и т.д. были *дефинициальными* равенствами. Однако в теории, представленной в главе 1, типы тождествености определяются одинаково для всех типов по их принципу индукции, поэтому мы не можем «переопределять» их на разные сущности разных типов. Таким образом, характеристики для конкретных типов, которые будут обсуждаться в этой главе, по большей части являются *теоремами*, которые мы должны выявить и доказать, если это возможно.

На самом деле, теория типов главы 1 недостаточна для доказательства искомых теорем для двух из конструкторов типов: $\prod$-типов и универсумов. По этой причине мы вынуждены вводить аксиомы в нашу теорию типов, чтобы сделать эти «теоремы» истинными. Теоретически, *аксиома* (сравните с §1.1) является «атомарным» элементом, который объявляется населенным определенным типом, без каких-либо правил, регулирующих его поведение, отличных от тех, которые относятся к типу его обитателей.

Аксиома для $\prod$-типов (§2.9) знакома теоретикам: она называется *функциональной экстенсиональностью*, и утверждает (грубо говоря), что если две функции гомотопны в смысле §2.4, то они равны. Однако, аксиома для универсумов (§2.10) является новым вкладом в теорию гомотопических типов от Воеводского: она называется *аксиомой унивалентности* и утверждает (примерно), что если два типа эквивалентны в смысле §2.4, то они равны. Мы уже отметили эту аксиому во введении; она будет играть очень важную роль в этой книге[1].

Важно отметить, что *не все* типы тождественности могут быть «определены» индукцией по построению типов. Контрпримеры включают большинство нетривиальных высших индуктивных типов (см. главы 6 и 8). Например, вычисление тождественных типов для типов $\mathcal{S}^n$ (см. §6.4) эквивалентно вычислению высших гомотопических групп сфер, глубокого и важного поля исследований в алгебраической топологии.

2.6 Типы декартова произведения

Для типов A и B рассмотрим тип декартова произведения $A \times B$. Для любых элементов $x, y : A \times B$ и пути $p : x =_{A\times B} y$ в силу функториальности можно извлечь пути $\mathsf{pr}_1(p) : \mathsf{pr}_1(x) =_A \mathsf{pr}_1(y)$ и $\mathsf{pr}_2(p) : \mathsf{pr}_2(x) =_B \mathsf{pr}_2(y)$. Таким образом, имеем функцию

$$(x =_{A\times B} y) \to (\mathsf{pr}_1(x) =_A \mathsf{pr}_1(y)) \times (\mathsf{pr}_2(x) =_B \mathsf{pr}_2(y))\,. \tag{2.6.1}$$

Теорема 2.6.2. *Для любых x и y функция* (2.6.1) *является эквивалентностью.*

Логическое понимание этого выражает то, что две пары равны, если они равны покомпонентно. Теоретико-категорный взгляд утверждает, что морфизмы в группоидном произведении

[1] Мы решили ввести эти принципы в качестве аксиом, но потенциально существуют другие способы формулировки теории типов, содержащие эти принципы. См. примечания к этой главе.

являются парами морфизмов. Теоретико-гомотопическое прочтение говорит о том, что пути в пространстве произведений являются парами путей.

Доказательство. Нам требуется функция другого направления:

$$(\mathsf{pr}_1(x) =_A \mathsf{pr}_1(y)) \times (\mathsf{pr}_2(x) =_B \mathsf{pr}_2(y)) \to (x =_{A\times B} y). \tag{2.6.3}$$

По правилу индукции для декартовых произведений мы можем принять, что x и y являются парами, т.е. $x \equiv (a,b)$ и $y \equiv (a',b')$ для некоторых $a,a' : A$ и $b,b' : B$. В этом случае имеем функцию

$$(a =_A a') \times (b =_B b') \to ((a,b) =_{A\times B} (a',b')).$$

Теперь, по индукции для декартова произведения в его области, можно считать, что $p : a = a'$ и $q : b = b'$. А по индукции пути, мы можем предположить, что $a \equiv a'$ и $b \equiv b'$, так что p и q являются рефлексивностью. Но в этом случае имеем $(a,b) \equiv (a',b')$, и поэтому получаем на выходе также рефлексивность.

Остается доказать, что (2.6.3) квазиобратно (2.6.1). Это простая последовательность индукций, но их нужно делать в правильном порядке.

В одном направлении начнем с $r : x =_{A\times B} y$. Сначала применим индукцию пути по r, чтобы предположить, что $x \equiv y$ и r — рефлексивность. В этом случае, поскольку $\mathsf{ap}_{\mathsf{pr}_1}$ и $\mathsf{ap}_{\mathsf{pr}_2}$ определяются индукцией пути, (2.6.1) использует $r \equiv \mathsf{refl}_x$ к паре $(\mathsf{refl}_{\mathsf{pr}_1 x}, \mathsf{refl}_{\mathsf{pr}_2 x})$. Теперь индукцией по x можно считать x $\equiv (a,b)$, так что это есть $(\mathsf{refl}_a, \mathsf{refl}_b)$. Таким образом, (2.6.3) использует его, по определению, к $\mathsf{refl}_{(a,b)}$, который (при наших текущих предположениях) есть r.

В другом направлении, если мы начнем с $s : (\mathsf{pr}_1(x) =_A \mathsf{pr}_1(y)) \times (\mathsf{pr}_2(x) =_B \mathsf{pr}_2(y))$, то сначала сделаем индукцию по x и y, чтобы предположить, что они являются парами (a,b) и (a',b'), а затем индукцией по $s : (a =_A a') \times (b =_B b')$, чтобы свести его к паре (p,q), где $p : a = a'$ и $q : b = b'$. Теперь индукцией по p и q можно считать, что они являются рефлексивностью refl_a и refl_b, и в этом случае (2.6.3) дает $\mathsf{refl}_{(a,b)}$, а затем (2.6.1) возвращает нас к $(\mathsf{refl}_a, \mathsf{refl}_b) \equiv (p,q) \equiv s$. □

В частности, мы показали, что (2.6.1) имеет обратный (2.6.3), который мы можем обозначить через

$$\mathsf{pair}^= : (\mathsf{pr}_1(x) = \mathsf{pr}_1(y)) \times (\mathsf{pr}_2(x) = \mathsf{pr}_2(y)) \to (x = y).$$

Заметим, что в частном случае это является принципом пропозициональной уникальности для произведений: $z = (\mathsf{pr}_1(z), \mathsf{pr}_2(z))$.

Может быть полезно рассматривать $\mathsf{pair}^=$ как *конструктор* или *правило введения* для $x = y$, аналогичное конструктору «спаривания» $A \times B$, который вводит пару (a,b) для заданных $a : A$ и $b : B$. В этой перспективе две компоненты (2.6.1):

$$\mathsf{ap}_{\mathsf{pr}_1} : (x = y) \to (\mathsf{pr}_1(x) = \mathsf{pr}_1(y))$$
$$\mathsf{ap}_{\mathsf{pr}_2} : (x = y) \to (\mathsf{pr}_2(x) = \mathsf{pr}_2(y))$$

являются *правилами исключения*. Аналогично, две гомотопии, служащие доказательством (2.6.3) как квазиобратной к (2.6.1), состоят соответственно из *правил вычисления высказываний*:

$$\mathsf{ap}_{\mathsf{pr}_1}(\mathsf{pair}^=(p,q)) = p \qquad \text{для } \mathsf{pr}_1 x = \mathsf{pr}_1 y$$
$$\mathsf{ap}_{\mathsf{pr}_2}(\mathsf{pair}^=(p,q)) = q \qquad \text{для } \mathsf{pr}_2 x = \mathsf{pr}_2 y$$

и принципа *пропозициональной уникальности*:

$$r = \mathsf{pair}^{=}(\mathsf{ap}_{\mathsf{pr}_1}(r), \mathsf{ap}_{\mathsf{pr}_2}(r)) \qquad \text{для } r : x =_{A\times B} y\,.$$

Мы также можем охарактеризовать рефлексивность, обратные и композицию путей в $A \times B$ покомпонентно:

$$\begin{aligned}
\mathsf{refl}_{(z:A\times B)} &= \mathsf{pair}^{=}(\mathsf{refl}_{\mathsf{pr}_1 z}, \mathsf{refl}_{\mathsf{pr}_2 z}) \\
p^{-1} &= \mathsf{pair}^{=}\left(\mathsf{ap}_{\mathsf{pr}_1}(p)^{-1}, \mathsf{ap}_{\mathsf{pr}_2}(p)^{-1}\right) \\
p \cdot q &= \mathsf{pair}^{=}\left(\mathsf{ap}_{\mathsf{pr}_1}(p) \cdot \mathsf{ap}_{\mathsf{pr}_1}(q), \mathsf{ap}_{\mathsf{pr}_2}(p) \cdot \mathsf{ap}_{\mathsf{pr}_2}(q)\right)\,.
\end{aligned}$$

Или, записывая по-другому:

$$\begin{aligned}
\mathsf{ap}_{\mathsf{pr}_i}(\mathsf{refl}_{(z:A\times B)}) &= \mathsf{refl}_{\mathsf{pr}_i z} && (i = 1, 2) \\
\mathsf{pair}^{=}(p^{-1}, q^{-1}) &= \mathsf{pair}^{=}(p, q)^{-1} \\
\mathsf{pair}^{=}(p \cdot q, p' \cdot q') &= \mathsf{pair}^{=}(p, p') \cdot \mathsf{pair}^{=}(q, q')\,.
\end{aligned}$$

Все эти уравнения можно получить, используя индукцию пути по заданным путям и возвращающие затем рефлексивность. То же самое справедливо для остальной части высшей группоидной структуры, рассмотренной в §2.1, хотя начинает утомлять вставка достаточного количества других путей когерентности, чтобы получить уравнение, которое будет проверяться. Например, если мы обозначим обратный путь в лемме 2.1.4 (iv) через $\mathsf{assoc}(p, q, r)$, а последний путь, показанный выше, как $\mathsf{pair}^{\cdot}(p, q, p', q')$, то для любых $u, v, z, w : A \times B$ и p, q, r, p', q', r' соответствующих типов получим

$$\begin{aligned}
&\mathsf{pair}^{\cdot}(p \cdot q, r, p' \cdot q', r') \\
&\cdot (\mathsf{pair}^{\cdot}(p, q, p', q') \cdot_{\mathsf{r}} \mathsf{pair}^{=}(r, r')) \\
&\cdot \mathsf{assoc}(\mathsf{pair}^{=}(p, p'), \mathsf{pair}^{=}(q, q'), \mathsf{pair}^{=}(r, r')) \\
&= \quad \mathsf{ap}_{\mathsf{pair}^{=}}(\mathsf{pair}^{=}(\mathsf{assoc}(p, q, r), \mathsf{assoc}(p', q', r'))) \\
&\qquad \cdot \mathsf{pair}^{\cdot}(p, q \cdot r, p', q' \cdot r') \\
&\qquad \cdot (\mathsf{pair}^{=}(p, p') \cdot_{\mathsf{l}} \mathsf{pair}^{\cdot}(q, r, q', r'))\,.
\end{aligned}$$

К счастью, нам никогда не придется использовать такие высокоразмерные когерентности.

Перейдем теперь к транспортированию в поточечном произведении семейств типов. Для семейств типов $A, B : Z \to \mathcal{U}$, определяемых посредством $(A \times B)(z) :\equiv A(z) \times B(z)$, мы упрощенно пишем $A \times B : Z \to \mathcal{U}$. Теперь для заданных $p : z =_Z w$ и $x : A(z) \times B(z)$, мы можем транспортировать x вдоль p, чтобы получить элемент из $A(w) \times \mathrm{B}(w)$.

Теорема 2.6.4. *Для приведенной выше ситуации имеем*

$$\mathsf{transport}^{A\times B}(p, x) =_{A(w)\times B(w)} (\mathsf{transport}^{A}(p, \mathsf{pr}_1 x), \mathsf{transport}^{B}(p, \mathsf{pr}_2 x))\,.$$

Доказательство. По индукции пути мы можем считать p рефлексивностью, и в этом случае имеем

$$\begin{aligned}
\mathsf{transport}^{A\times B}(p, x) &\equiv x \\
\mathsf{transport}^{A}(p, \mathsf{pr}_1 x) &\equiv \mathsf{pr}_1 x \\
\mathsf{transport}^{A}(p, \mathsf{pr}_2 x) &\equiv \mathsf{pr}_2 x\,.
\end{aligned}$$

Таким образом, остается показать, что $x = (\mathsf{pr}_1 x, \mathsf{pr}_2 x)$. Но это пропозициональный принцип единственности для типов произведений, который, как отмечалось выше, следует из теоремы 2.6.2. $\square$

Наконец, рассмотрим функториальность ap под декартовыми произведениями. Пусть заданы типы A, B, A', B' и функции $g : A \to A'$ и $h : B \to B'$; тогда мы можем определить функцию $f : A \times B \to A' \times B'$ через $f(x) :\equiv (g(\mathsf{pr}_1 x), h(\mathsf{pr}_2 x))$.

Теорема 2.6.5. *Для приведенной выше ситуации, при $x, y : A \times B$, $p : \mathsf{pr}_1 x = \mathsf{pr}_1 y$, и $q : \mathsf{pr}_2 x = \mathsf{pr}_2 y$ имеет место*

$$f(\mathsf{pair}^{=}(p, q)) =_{(f(x)=f(y))} \mathsf{pair}^{=}(g(p), h(q)).$$

Доказательство. Заметим сначала, что приведенное выше уравнение хорошо типизировано. С одной стороны, поскольку $\mathsf{pair}^{=}(p, q) : x = y$, мы имеем $f(\mathsf{pair}^{=}(p, q)) : f(x) = f(y)$. С другой стороны, поскольку $\mathsf{pr}_1(f(x)) \equiv g(\mathsf{pr}_1 x)$ и $\mathsf{pr}_2(f(x)) \equiv h(\mathsf{pr}_2 x)$, мы также имеем $\mathsf{pair}^{=}(g(p), h(q)) : f(x) = f(y)$.

Теперь, по индукции, мы можем считать $x \equiv (a, b)$ и $y \equiv (a', b')$, в этом случае имеем $p : a = a'$ и $q : b = b'$. Таким образом, по индукции пути можно считать p и q рефлексивностью, и в этом случае требуемое уравнение выполняется дефиниционально. $\square$

2.7 $\sum$-типы

Пусть A — тип и $B : A \to \mathcal{U}$ — семейство типов. Напомним, что $\sum$-тип или тип зависимой пары $\sum_{(x:A)} B(x)$ является обобщением типа декартова произведения. Таким образом, мы ожидаем, что его высшая группоидная структура также будет обобщением предыдущего раздела. В частности, его пути должны быть парами путей, но требуется немного подумать, чтобы указать правильные типы этих путей.

Предположим, что имеется путь $p : w = w'$ в $\sum_{(x:A)} P(x)$. Тогда получим $\mathsf{pr}_1(p) : \mathsf{pr}_1(w) = \mathsf{pr}_1(w')$. Однако мы не можем напрямую спросить, является ли $\mathsf{pr}_2(w)$ тождественным $\mathsf{pr}_2(w')$, поскольку они могут быть не одного типа. Но мы можем переносить (транспортировать) $\mathsf{pr}_2(w)$ вдоль пути $\mathsf{pr}_1(p)$, и это дает нам элемент того же типа, что и $\mathsf{pr}_2(w')$. По индукции пути мы фактически получаем путь $\mathsf{pr}_1(p)_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w')$.

Напомним из обсуждения, предшествующего лемме 2.3.4, что $\mathsf{pr}_1(p)_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w')$ можно рассматривать как тип путей от $\mathsf{pr}_2(w)$ к $\mathsf{pr}_2(w')$, которые лежат над путём $\mathsf{pr}_1(p)$ в A. Таким образом, мы говорим, что путь $w = w'$ в общем пространстве определяет (и определяется как) путь $p : \mathsf{pr}_1(w) = \mathsf{pr}_1(w')$ в A вместе с путем от $\mathsf{pr}_2(w)$ к $\mathsf{pr}_2(w')$, лежащих над p, что кажется разумным.

Замечание 2.7.1. Отметим, что если имеются $x : A$ и $u, v : P(x)$ такие, что $(x, u) = (x, v)$, из этого не следует, что $u = v$. Все, что мы можем заключить, это то, что существует $p : x = x$ такое, что $p_*(u) = v$. Это известный источник путаницы для новичков в теории типов, но имеет смысл с топологической точки зрения: существование пути $(x, u) = (x, v)$ в пространстве расслоений между двумя точками, которые входят в один и тот же слой, не означает существования пути $u = v$, лежащего целиком *внутри* этого слоя.

Следующая теорема утверждает, что мы также можем обратить вспять этот процесс. Так как это прямое обобщение теоремы 2.6.2, мы будем более краткими.

Теорема 2.7.2. *Предположим, что $P : A \to \mathcal{U}$ - семейство типов над типом A и пусть $w, w' : \sum_{(x:A)} P(x)$. Тогда существует эквивалентность*

$$(w = w') \simeq \sum_{(p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w'))} p_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w') \,.$$

Доказательство. Определим функцию

$$f : \prod_{w,w':\sum_{(x:A)} P(x)} (w = w') \to \sum_{(p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w'))} p_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w')$$

используя индукции пути, с

$$f(w, w, \mathsf{refl}_w) :\equiv (\mathsf{refl}_{\mathsf{pr}_1(w)}, \mathsf{refl}_{\mathsf{pr}_2(w)}) \,.$$

Мы хотим показать, что f является эквивалентностью.

В обратном направлении определим

$$g : \prod_{w,w':\sum_{(x:A)} P(x)} \left(\sum_{(p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w'))} p_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w') \right) \to (w = w')$$

сначала используя индукцию на w и w', которая разбивает их на (w_1, w_2) и (w_1', w_2') соответственно, поэтому достаточно показать, что

$$\left(\sum_{p:w_1=w_1'} p_*(w_2) = w_2' \right) \to ((w_1, w_2) = (w_1', w_2')) \,.$$

Далее, для пары $\sum_{(p:w_1=w_1')} p_*(w_2) = w_2'$ мы можем использовать $\sum$-индукцию для получения $p : w_1 = w_1'$ и $q : p_*(w_2) = w_2'$. Проводя индукцию на p, получаем $q : (\mathsf{refl}_{w_1})_*(w_2) = w_2'$, и достаточно показать , что $(w_1, w_2) = (w_1, w_2')$. Но $(\mathsf{refl}_{w_1})_*(w_2) = w_2$, поэтому индукция на q приводит к цели к $(w_1, w_2) = (w_1, w_2)$, которую мы можем доказать с помощью $\mathsf{refl}_{(w_1,w_2)}$.

Затем мы показываем, что $f(g(r)) = r$ для всех w, w' и r, где r имеет тип

$$\sum_{(p:\mathsf{pr}_1(w)=\mathsf{pr}_1(w'))} p_*(\mathsf{pr}_2(w)) = \mathsf{pr}_2(w') \,.$$

Во-первых, мы разбиваем пары w, w', и r парной индукцией, как и в определении g, а затем используем два пути индукции, чтобы свести обе составляющие r к refl. Тогда достаточно показать, что $f(g(\mathsf{refl}_{w_1}, \mathsf{refl}_{w_2})) = (\mathsf{refl}_{w_1}, \mathsf{refl}_{w_2})$, что верно по определению.

Аналогично, чтобы показать, что $g(f(p)) = p$ для всех w, w', и $p : w = w'$, мы можем провести индукцию пути по p, а затем парной индукцией разбить w, где достаточно показать что $g(f(\mathsf{refl}_{w_1}, \mathsf{refl}_{w_2})) = (\mathsf{refl}_{w_1}, \mathsf{refl}_{w_2})$, что верно по определению.

Таким образом, функция f имеет квазиобратную и, следовательно, является эквивалентностью. □

Как и в случае декартовых произведений, мы можем вывести принцип пропозициональной уникальности как частный случай.

Следствие 2.7.3. *Для* $z : \sum_{(x:A)} P(x)$ *имеем* $z = (\mathsf{pr}_1(z), \mathsf{pr}_2(z))$.

Доказательство. Мы имеем $\mathsf{refl}_{\mathsf{pr}_1}(z) : \mathsf{pr}_1(z) = \mathsf{pr}_1(\mathsf{pr}_1(z), \mathsf{pr}_2(z))$, поэтому по теореме 2.7.2 достаточно предъявить путь $(\mathsf{refl}_{\mathsf{pr}_1}(z))_*(\mathsf{pr}_2(z)) = \mathsf{pr}_2(\mathsf{pr}_1(z), \mathsf{pr}_2(z))$. Но обе стороны его дефиниционально равны $\mathsf{pr}_2(z)$. □

Как и в случае с бинарными декартовыми произведениями, мы можем думать об обратном направлении теоремы 2.7.2, как о введении форме представления ($\mathsf{pair}^=$), о прямом направлении — как о формах исключения ($\mathsf{ap}_{\mathsf{pr}_1}$ и $\mathsf{ap}_{\mathsf{pr}_2}$), а об эквивалентности — как о предъявлении правила пропозиционального вычисления и принципа уникальности для них.

Заметим, что поднятый путь $\mathsf{lift}(u, p)$ для $p : x = y$ при $u : P(x)$, определенный в лемме 2.3.2, можно отождествить со специальным случаем формы представления

$$\mathsf{pair}^=(p, \mathsf{refl}_{p_*(u)}) : (x, u) = (y, p_*(u)) \,.$$

Это проявляется в формулировке о действии транспортировки на $\sum$-типах, что также является обобщением действия для бинарных декартовых произведений:

Теорема 2.7.4. *Предположим, что имеются семейства типов*

$$P : A \to \mathcal{U}, \qquad Q : \left(\sum_{x:A} p(x)\right) \to \mathcal{U} \,.$$

Тогда мы можем построить семейство типов над A*, определенное с помощью*

$$x \mapsto \sum_{u:P(x)} Q(x, u) \,.$$

Для любого пути $p : x = y$ *и любой пары* $(u, z) : \sum_{(u:P(x))} Q(x, u)$ *имеем*

$$p_*(u, z) = (p_*(u), \mathsf{pair}^=(p, \mathsf{refl}_{p_*(u)})_*(z)) \,.$$

Доказательство. Непосредственно следует из индукции пути. □

Мы оставляем читателю формулировку и доказательство обобщения теоремы 2.6.5 (см. упражнение 2.7), и покомпонентную характеризацию рефлексивности, инверсий и композиции $\sum$-типов.

2.8 Единичный тип

Иногда важны тривиальные случаи, поэтому мы вкратце коснемся единичного типа $\mathbf{1}$.

Теорема 2.8.1. *Для любых* $x, y : \mathbf{1}$ *имеем* $(x = y) \simeq \mathbf{1}$.

Может возникнуть соблазн начать это доказательство с помощью $\mathbf{1}$-индукции по x и y, сводя задачу к $(\star = \star) \simeq \mathbf{1}$. Однако в этот момент мы бы застряли, так как мы не смогли бы провести индукцию пути по $p : \star = \star$. Таким образом, мы вместо этого работаем с общими x и y как можно дольше, сводя их к $\star$, по индукции, только в последний момент.

Доказательство. Функцию $(x = y) \to \mathbf{1}$ легко определить, направив все к $\star$. Обратно, для любых $x, y : \mathbf{1}$ по индукции можно считать, что $x \equiv \star \equiv$ y. В этом случае имеем $\mathsf{refl}_\star : x = y$, что дает постоянную функцию $\mathbf{1} \to (x = y)$.

Чтобы показать, что эти функции обратны друг другу, рассмотрим сначала элемент $u : \mathbf{1}$. Можно считать, что $u \equiv \star$, но это также результат комбинирования $\mathbf{1} \to (x = y) \to \mathbf{1}$.

С другой стороны, пусть $p : x = y$. По индукции пути можно считать, что $x \equiv y$ и p есть refl_x. Тогда мы можем предположить, что x есть $\star$, и в этом случае компоновка $(x = y) \to \mathbf{1} \to (x = y)$ применяет p к refl_x, т. е. к p. $\square$

В частности, любые два элемента из $\mathbf{1}$ равны. Мы предоставляем читателю формулировку этой эквивалентности в терминах правил приведения, исключения, вычисления и единственности. Лемма транспортировки для $\mathbf{1}$ является просто леммой транспортировки для семейств с постоянным типом (лемма 2.3.5).

2.9 $\prod$-типы и аксиома функциональной экстенсиональности

Для типа A и семейства типов $B : A \to \mathcal{U}$ рассмотрим функциональный тип $\prod_{(x:A)} B(x)$. Мы ожидаем, что тип $f = g$ путей от f к g в $\prod_{(x:A)} B(x)$ эквивалентен типу поточечных путей:

$$(f = g) \simeq \left(\prod_{x:A}(f(x) =_{B(x)} g(x))\right). \tag{2.9.1}$$

С традиционной точки зрения это говорит о том, что две функции, которые равны в каждой точке, равны как функции. С топологической точки зрения можно сказать, что путь в функциональном пространстве совпадает с непрерывной гомотопией. А с категорной точки зрения можно будет сказать, что изоморфизм в категории функторов является естественным семейством изоморфизмов.

Однако, в отличие от предыдущего раздела, базисной теории типов, представленной в главе 1, недостаточно для доказательства (2.9.1). Все, что мы можем сказать, это то, что существует определенная функция

$$\mathsf{happly} : (f = g) \to \prod_{x:A}(f(x) =_{B(x)} g(x)) \tag{2.9.2}$$

которая легко определяется индукцией пути. Поэтому на данный момент мы примем:

Аксиома 2.9.3 (Функциональная экстенсиональность)**.** *Для любых A, B, f и g функция* (2.9.2) *является эквивалентностью.*

В последующих главах мы увидим, что эта аксиома следует как из унивалентности (см. §§ 2.10, 4.9), так и из типа интервалов (см. §6.3 и упражнение 6.10).

В частности, из аксиомы 2.9.3 следует, что функция (2.9.2) имеет квазиобратную

$$\mathsf{funext} : \left(\prod_{x:A}(f(x) = g(x))\right) \to (f = g).$$

Эта функция также упоминается как «функциональная экстенсиональность». Как и в случае с $\mathsf{pair}^=$ в §2.6, мы можем рассматривать funext как *правило выведения* для типа $f = g$. С этой точки зрения, happly — это *правило исключения*, в то время как гомотопии, свидетельствующие о том, что funext квазиинверсна к happly, становятся правилом пропозиционального вычисления

$$\mathsf{happly}(\mathsf{funext}(h), x) = h(x) \qquad \text{для } h : \prod_{x:A}(f(x) = g(x))$$

и принципом пропозициональной уникальности:

$$p = \mathsf{funext}(x \mapsto \mathsf{happly}(p, x)) \qquad \text{для } p : f = g\,.$$

Мы также можем вычислить тождественность, обратные функции и композицию в $\prod$-типах; они просто задаются поточечными операциями:

$$\begin{aligned}
\mathsf{refl}_f &= \mathsf{funext}(x \mapsto \mathsf{refl}_{f(x)})\\
\alpha^{-1} &= \mathsf{funext}(x \mapsto \mathsf{happly}(\alpha, x)^{-1})\\
\alpha \cdot \beta &= \mathsf{funext}(x \mapsto \mathsf{happly}(\alpha, x) \cdot \mathsf{happly}(\beta, x))\,.
\end{aligned}$$

Первое из этих равенств следует из определения happly, в то время как второе и третье являются просто индукциями пути.

Поскольку независимый функциональный тип $A \to B$ является частным случаем зависимого функционального типа $\prod_{(x:A)} B(x)$, когда B не зависит от x, все, о чем мы говорили выше, применимо и в независимых случаях. Однако правила транспортировки в независимом случае несколько упрощаются. Для типа X путь p: $x_1 =_X x_2$, семейств типов $A, B : X \to \mathcal{U}$ и функции $f : A(x_1) \to B(x_1)$, имеем

$$\mathsf{transport}^{A\to B}(p, f) = \left(x \mapsto \mathsf{transport}^B(p, f(\mathsf{transport}^A(p^{-1}, x)))\right)\,, \tag{2.9.4}$$

где $A \to B$ обозначает (уничижительно) семейство типов $X \to \mathcal{U}$, определяемое как

$$(A \to B)(x) :\equiv (A(x) \to B(x))\,.$$

Другими словами, когда мы транспортируем функцию $f : A(x_1) \to B(x_1)$ вдоль пути $p : x_1 = x_2$, получаем функцию $A(x_2) \to B(x_2)$, которая транспортирует свой аргумент обратно по p (в семейство типов A), применяет f, а затем транспортирует результат вперед вдоль p (в семейство типов B). Это может быть легко доказано путем индукции пути.

Транспортирование зависимых функций аналогично, но сложнее. Пусть заданы X и p, как и раньше, семейства типов $A : X \to \mathcal{U}$ и $B : \prod_{(x:X)}(A(x) \to \mathcal{U})$, а также зависимая функция $f : \prod_{(a:A(x_1))} B(x_1, a)$. Тогда для $a : A(x_2)$ имеем

$$\mathsf{transport}^{\Pi_A(B)}(p, f)(a) = \mathsf{transport}^{\hat{B}}\left((\mathsf{pair}^=(p^{-1}, \mathsf{refl}_{p_*^{-1}(a)}))^{-1}, f(\mathsf{transport}^A(p^{-1}, a))\right)$$

где, $\prod_A(B)$ и $\hat{B}$ обозначают семейства типов, соответственно

$$\begin{aligned}
\prod_A(B) &:\equiv \left(x \mapsto \prod_{(a:A(x))} B(x, a)\right) && : X \to \mathcal{U}\\
\hat{B} &:\equiv (w \mapsto B(\mathsf{pr}_1(w), \mathsf{pr}_2(w))) && : \left(\textstyle\sum_{(x:X)} A(x)\right) \to \mathcal{U}\,.
\end{aligned} \tag{2.9.5}$$

Если эти формулы выглядят немного пугающе, не беспокойтесь о деталях. Основная идея такая же, как и для независящего функционального типа: мы переносим аргумент назад, применяем функцию, а затем снова переносим результат вперед.

Теперь напомним, что для общего семейства типов $P : X \to \mathcal{U}$, в §2.2 мы определили тип *зависимых путей* над $p : x =_X y$ от $u : P(x)$ к $v : P(y)$ как $p_*(u) =_{P(y)} v$. Когда P — семейство функциональных типов, существует эквивалентный способ представить это, что часто бывает более удобно.

Лемма 2.9.6. *Для семейств типов $A, B : X \to \mathcal{U}$, $p : x =_X y$ и $f : A(x) \to B(x)$, $g : A(y) \to B(y)$ имеется эквивалентность*

$$(p_*(f) = g) \simeq \prod_{a:A(x)} (p_*(f(a)) = g(p_*(a))) .$$

Более того, если $q : p_*(f) = g$ соответствует этой эквивалентности к $\hat{q}$, то для $a : A(x)$ путь

$$\mathsf{happly}(q, p_*(a)) : (p_*(f))(p_*(a)) = g(p_*(a))$$

равен составному

$$\begin{aligned}
(p_*(f))(p_*(a)) &= p_*\left(f(p_*^{-1}(p_*(a)))\right) && \text{(по (2.9.4))}\\
&= p_*(f(a))\\
&= g(p_*(a)) . && \text{(по } \hat{q}\text{)}
\end{aligned}$$

Доказательство. По индукции пути мы можем считать p рефлексивностью, и в этом случае искомая эквивалентность сводится к функциональной экстенсиональности. Второе утверждение затем следует правилу вычисления для функциональной экстенсиональности. □

По-прежнему, версия леммы 2.9.6 зависимых функций похожа, но устроена сложнее.

Лемма 2.9.7. *Для семейств типов $A : X \to \mathcal{U}$ и $B : \prod_{(x:X)} A(x) \to \mathcal{U}$, $p : x =_X y$, $f : \prod_{(a:A(x))} B(x, a)$ и $f : \prod_{(a:A(y))} B(y, a)$ имеется эквивалентность*

$$(p_*(f) = g) \simeq \left(\prod_{a:A(x)} \mathsf{transport}^{\hat{B}}(\mathsf{pair}^{=}(p, \mathsf{refl}_{p_*(a)}), f(a)) = g(p_*(a)) \right)$$

с $\hat{B}$ как в (2.9.5).

Мы оставляем читателю доказательство этой леммы и формулировку подходящего правила вычисления.

2.10 Универсумы и аксиома унивалентности

Имея два типа, A и B, можно рассматривать их как элементы некоторого универсума типов $\mathcal{U}$ и тем самым сформировать тип тождественности $A =_{\mathcal{U}} B$. Как упоминалось во введении, *унивалентность* — это идентификация $A =_{\mathcal{U}} B$ типом $(A \simeq B)$ эквивалентностей от A к B, которая описана в §2.4. Мы выполняем эту идентификацию с помощью следующей канонической функции.

Лемма 2.10.1. *Для типов $A, B : \mathcal{U}$ существует функция*

$$\mathsf{idtoeqv} : (A =_{\mathcal{U}} B) \to (A \simeq B), \tag{2.10.2}$$

определенная в доказательстве.

Доказательство. Мы могли бы построить требуемое непосредственно индукцией по равенству, но следующее описание более удобно. Заметим, что функцию $id_{\mathcal{U}} : \mathcal{U} \to \mathcal{U}$ можно рассматривать как семейство типов, индексированное универсумом $\mathcal{U}$; она назначает каждому типу $X : \mathcal{U}$ сам тип X (если рассматривать это как расслоение, то его пространство расслоений есть тип $\sum_{(A:\mathcal{U})} A$ «точечных типов», см. также §4.8). Таким образом, для заданного пути $p : A =_{\mathcal{U}} B$ имеется функция транспортирования $p_* : A \to B$. Мы утверждаем, что p_* является эквивалентностью. Но по индукции достаточно предположить, что p является refl_A, а в этом случае $p_* \equiv \mathsf{id}_A$, что является эквивалентностью согласно примеру 2.4.7. Таким образом, мы можем определить $\mathsf{idtoeqv}(p)$ как p_* (вместе с приведенным выше доказательством эквивалентности).

Мы хотели бы сказать, что $\mathsf{idtoeqv}$ является эквивалентностью. Однако, как и в случае с функциональными типами, теория типов, описанная в главе 1, недостаточна, чтобы гарантировать это. Таким образом, так же, как мы поступали с функциональной эксистенсиональностью, мы формулируем это свойство как аксиому: аксиому унивалентности Воеводского. □

Аксиома 2.10.3 (Унивалентность)**.** *Для любых $A, B : \mathcal{U}$ функция (2.10.2) является эквивалентностью.*

В частности, поэтому имеем

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B).$$

Технически аксиома унивалентности является утверждением о конкретном универсуме $\mathcal{U}$. Если универсум $\mathcal{U}$ удовлетворяет этой аксиоме, мы говорим, что он **унивалентен**. Если не указано иное (например, в §4.9), мы будем предполагать, что все универсумы унивалентны.

Замечание 2.10.4. Для аксиомы унивалентности важно, чтобы мы определили $A \simeq B$, используя «хорошую» версию $\mathsf{isequiv}$, как описано в §2.4, а не (скажем) как $\sum_{(f:A\to B)} \mathsf{qinv}(f)$. См. упражнение 4.6.

В частности, унивалентность означает, что *могут быть идентифицированы эквивалентные типы*. Как и в предыдущих разделах, полезно разбить эту эквивалентность на:

- Правило выведения для $(A =_{\mathcal{U}} B)$, обозначаемое ua, для «аксиомы унивалентности»:

$$\mathsf{ua} : (A \simeq B) \to (A =_{\mathcal{U}} B).$$

- Правило исключения, которое есть $\mathsf{idtoeqv}$,

$$\mathsf{idtoeqv} \equiv \mathsf{transport}^{X \mapsto X} : (A =_{\mathcal{U}} B) \to (A \simeq B).$$

- Правило пропозиционального вычисления,

$$\mathsf{transport}^{X \mapsto X}(\mathsf{ua}(f), x) = f(x).$$

- Принцип пропозициональной уникальности: для любого $p : A = B$,

$$p = \mathsf{ua}(\mathsf{transport}^{X \mapsto X}(p)).$$

Мы также можем идентифицировать рефлексивность, конкатенацию и инверсии равенств в универсуме с соответствующими операциями над эквивалентностями:

$$\begin{aligned} \mathsf{refl}_A &= \mathsf{ua}(\mathsf{id}_A) \\ \mathsf{ua}(f) \cdot \mathsf{ua}(g) &= \mathsf{ua}(g \circ f) \\ \mathsf{ua}(f)^{-1} &= \mathsf{ua}(f^{-1}) \,. \end{aligned}$$

Первое из них следует из того, что $\mathsf{id}_A = \mathsf{idtoeqv}(\mathsf{refl}_A)$ по определению $\mathsf{idtoeqv}$, а ua является инверсией $\mathsf{idtoeqv}$. Для второго, если мы определим $p :\equiv \mathsf{ua}(f)$ и $q :\equiv \mathsf{ua}(g)$, то имеем

$$\mathsf{ua}(g \circ f) = \mathsf{ua}(\mathsf{idtoeqv}(q) \circ \mathsf{idtoeqv}(p)) = \mathsf{ua}(\mathsf{idtoeqv}(p \cdot q)) = p \cdot q$$

используя лемму 2.3.9 и определение $\mathsf{idtoeqv}$. Третье доказывается похожим образом.

Следующее наблюдение, которое является частным случаем леммы 2.3.10, часто полезно при применении аксиомы унивалентности.

Лемма 2.10.5. *Для любых семейства типов $B : A \to \mathcal{U}$, $x, y : A$ с путем $p : x = y$ и $u : B(x)$ имеем*

$$\begin{aligned} \mathsf{transport}^B(p, u) &= \mathsf{transport}^{X \mapsto X}(\mathsf{ap}_B(p), u) \\ &= \mathsf{idtoeqv}(\mathsf{ap}_B(p))(u) \,. \end{aligned}$$

2.11 Тип тождественности

В то время как тип $a =_A a'$ характеризуется изоморфизмом, с отдельным «определением» для каждого A, нет простой характеризации типа $p =_{a =_A a'} q$ путей между путями $p, q : a =_A a'$. Однако наши другие общие классы теорем распространяются на типы тождественности, не нарушая эквивалентности.

Теорема 2.11.1. *Если $f : A \to B$ — эквивалентность, то для всех $a, a' : A$*

$$\mathsf{ap}_f : (a =_A a') \to (f(a) =_B f(a')) \,.$$

Доказательство. Пусть f^{-1} — квазиобратна к f, с гомотопиями

$$\alpha : \prod_{b:B} (f(f^{-1}(b)) = b) \qquad \text{и} \qquad \beta : \prod_{a:A} (f^{-1}(f(a)) = a)$$

Квазиобратный к ap_f есть, по существу,

$$\mathsf{ap}_{f^{-1}} : (f(a) = f(a')) \to (f^{-1}(f(a)) = f^{-1}(f(a'))) \,.$$

Однако для того, чтобы получить элемент $a =_A a'$ из $\mathsf{ap}_{f^{-1}}(q)$, мы должны провести конкатенацию с путями β_a^{-1} и $\beta_{a'}$ с двух сторон. Чтобы показать, что это дает квазиобратный к ap_f, с одной стороны, мы должны показать, что для любого $p : a =_A a'$ имеем

$$\beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(p)) \cdot \beta_{a'} = p \,.$$

Это следует из функториальности ap и естественности гомотопий, леммы 2.2.2 и 2.4.3. С другой стороны, мы должны показать, что для любого $q : f(a) =_B f(a')$ имеем

$$\mathsf{ap}_f(\beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \beta_{a'}) = q\,.$$

Доказательство этого немногим более важно, но каждый шаг снова является применением лемм 2.2.2 и 2.4.3 (или просто устраняет обратные пути):

$$\begin{aligned}
\mathsf{ap}_f(\beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \beta_{a'}) &= \alpha_{f(a)}^{-1} \cdot \alpha_{f(a)} \cdot \mathsf{ap}_f(\beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \beta_{a'}) \cdot \alpha_{f(a')}^{-1} \cdot \alpha_{f(a')} \\
&= \alpha_{f(a)}^{-1} \cdot \mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(\mathsf{ap}_f(\beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \beta_{a'}))) \cdot \alpha_{f(a')} \\
&= \alpha_{f(a)}^{-1} \cdot \mathsf{ap}_f(\beta_a \cdot \beta_a^{-1} \cdot \mathsf{ap}_{f^{-1}}(q) \cdot \beta_{a'} \cdot \beta_{a'}^{-1}) \cdot \alpha_{f(a')} \\
&= \alpha_{f(a)}^{-1} \cdot \mathsf{ap}_f(\mathsf{ap}_{f^{-1}}(q)) \cdot \alpha_{f(a')} \\
&= q\,.
\end{aligned}$$

□

Таким образом, если для некоторого типа A мы имеем полную характеризацию $a =_A a'$, то также определяется тип $p =_{a =_A a'} q$. Например:

- Пути $p = q$, где $p, q : w =_{A \times B} w'$, эквивалентны парам путей

$$\mathsf{ap}_{\mathsf{pr}_1} p =_{\mathsf{pr}_1 w =_A \mathsf{pr}_1 w'} \mathsf{ap}_{\mathsf{pr}_1} q \qquad \text{и} \qquad \mathsf{ap}_{\mathsf{pr}_2} p =_{\mathsf{pr}_2 w =_B \mathsf{pr}_2 w'} \mathsf{ap}_{\mathsf{pr}_2} q\,.$$

- Пути $p = q$, где $p, q : f =_{\prod_{(x:A)} B(x)} g$, эквивалентны гомотопиям

$$\prod_{x:A} (\mathsf{happly}(p)(x) =_{f(x)=g(x)} \mathsf{happly}(q)(x))\,.$$

Далее рассматривается транспортирование в семействах путей, т.е. транспортирование в $C : A \to \mathcal{U}$, где каждый $C(x)$ является типом тождественности. В простейшем случае $C(x)$ является типом путей в самом A, возможно, с фиксированной финальной точкой.

Лемма 2.11.2. *Для любых A и $a : A$ с $p : x_1 = x_2$ имеем*

$$\begin{aligned}
&\mathsf{transport}^{x \mapsto (a = x)}(p, q) = q \cdot p && \textit{для } q : a = x_1 \\
&\mathsf{transport}^{x \mapsto (x = a)}(p, q) = p^{-1} \cdot q && \textit{для } q : x_1 = a \\
&\mathsf{transport}^{x \mapsto (x = x)}(p, q) = p^{-1} \cdot q \cdot p && \textit{для } q : x_1 = x_1
\end{aligned}$$

Доказательство. Индукция пути по p, за которой следуют законы единиц для композиции. □

Другими словами, транспортировка с помощью $x \mapsto c = x$ — это посткомпозиция, а перенос с помощью $x \mapsto x = c$ — контравариантная прекомпозиция. Они могут быть известны как функториальные действия ковариантных и контравариантных hom-функторов $hom(c, \_)$ и $hom(\_, c)$ в теории категорий.

Аналогично, мы можем доказать следующий более общую формулировку леммы 2.11.2, которая связана с леммой 2.3.10.

Теорема 2.11.3. *Для* $f, g : A \to B$ *с* $p : a =_A a'$ *и* $q : f(a) =_B g(a)$ *имеем*

$$\mathsf{transport}^{x \mapsto f(x) =_B g(x)}(p, q) =_{f(a') = g(a')} (\mathsf{ap}_f p)^{-1} \cdot q \cdot \mathsf{ap}_g p\,.$$

Поскольку $\mathsf{ap}_{(x \mapsto x)}$ — тождественная функция, а $\mathsf{ap}_{(x \mapsto c)}$ (где c — константа) есть $p \mapsto \mathsf{refl}_c$, лемма 2.11.2 является частным случаем. Еще более общая версия — это когда B может быть семейством типов, индексированных на A:

Теорема 2.11.4. *Пусть* $B : A \to \mathcal{U}$ *и* $f, g : \prod_{(x:A)} B(x)$ *с* $p : a =_A a'$ *и* $q : f(a) =_{B(a)} g(a)$. *Тогда имеет место*

$$\mathsf{transport}^{x \mapsto f(x) =_{B(x)} g(x)}(p, q) = (\mathsf{apd}_f p)^{-1} \cdot \mathsf{ap}_{(\mathsf{transport}^B p)} \cdot \mathsf{apd}_g p\,.$$

Наконец, как и в §2.9, для семейств типов тождественности существует другая эквивалентная характеризация зависимых путей.

Теорема 2.11.5. *Для* $p : a =_A a'$ *с* $q : a = a$ *и* $r : a' = a'$ *имеем*

$$(\mathsf{transport}^{x \mapsto (x = x)}(p, q) = r) \simeq (q \cdot p = p \cdot r)\,.$$

Доказательство. Индукция пути по p, после чего следует, что композиция с единичными равенствами $q \cdot 1 = q$ и $r = 1 \cdot r$ является эквивалентностью. □

Существуют более общие эквивалентности, связанные с применением функций, аналогичные теоремам 2.11.3 и 2.11.4.

2.12 Копроизведения

До сих пор большинство конструкторов типов, которые мы рассматривали, были так называемыми *негативными*. Интуитивно это означает, что их элементы определяются их поведением по правилам исключения: (зависимая) пара определяется своими проекциями, а (зависимая) функция определяется своими значениями. Типы тождественности негативных типов почти всегда можно охарактеризовать прямо, вместе со всей их высшей структурой, как это было сделано в §§ 2.6–2.9. Универсум не является точно негативным типом, но его типы тождественности ведут себя одинаково: у нас есть простая характеризация (унивалентность) и описание структуры высшего порядка. Конечно, сами типы тождественности являются особым случаем.

Теперь рассмотрим первый пример *позитивного* конструктора типов. Опять же, неформально, позитивный тип — это тот, который «представлен» определенными конструкторами, причем универсальное свойство представления выражается правилом исключения (говоря категорным языком, позитивный тип имеет универсальное свойство «отображение наружу», а негативный тип — универсальное свойство «отображение внутрь»). Поскольку вычисления с представлениями — это, вообще говоря, проблема невычислимости, для позитивных типов мы не всегда можем ожидать простой характеризации типа тождественности. Однако во многих частных случаях характеризация или частичная характеризация существуют и могут быть получены с помощью общего метода, который мы введем в этом примере.

(Технически наше выбранное представление декартовых произведений и $\sum$-типов также позитивно. Однако, поскольку эти типы, к тому же, допускают негативное представление, которые мало чем отличаются, их типы тождественности имеют прямую характеризацию, которая не требует описанного здесь метода).

Рассмотрим тип копроизведения $A+B$, который «представлен» инъекциями $\mathsf{inl} : A \to A+B$ и $\mathsf{inr} : B \to A+B$. Интуитивно мы ожидаем, что $A+B$ содержит точные копии A и B дизъюнктно, так что мы должны иметь

$$(\mathsf{inl}(a_1) = \mathsf{inl}(a_2)) \simeq (a_1 = a_2)\,, \tag{2.12.1}$$

$$(\mathsf{inr}(b_1) = \mathsf{inr}(b_2)) \simeq (b_1 = b_2)\,, \tag{2.12.2}$$

$$(\mathsf{inl}(a) = \mathsf{inr}(b)) \simeq \mathbf{0}\,. \tag{2.12.3}$$

Докажем это следующим образом. Зафиксируем элемент $a_0 : A$; мы будем проводить характеризацию типа семейства

$$(x \mapsto (\mathsf{inl}(a_0) = x)) : A + B \to \mathcal{U}\,. \tag{2.12.4}$$

Аналогичный аргумент будет характеризовать аналогичное семейство $x \mapsto (x = \mathsf{inr}(b_0))$ для любого $b_0 : B$. Вместе эти характеризации подразумевают (2.12.1)–(2.12.3).

Чтобы охарактеризовать (2.12.4), мы определим семейство типов $\mathsf{code} : A + B \to \mathcal{U}$ и покажем, что $\prod_{(x:A+B)}((\mathsf{inl}(a_0) = x) \simeq \mathsf{code}(x))$. Поскольку мы хотим получить (2.12.1) из этого, мы должны иметь $\mathsf{code}(\mathsf{inl}(a)) = (a_0 = a)$, и поскольку мы также хотим вывести (2.12.3), мы должны иметь $\mathsf{code}(\mathsf{inr}(b)) = \mathbf{0}$. Весьма важная интуиция заключается в том, что мы можем использовать принцип рекурсии по $A + B$ для *определения* $\mathsf{code} : A + B \to \mathcal{U}$ следующими двумя уравнениями:

$$\begin{aligned}
\mathsf{code}(\mathsf{inl}(a)) &:\equiv (a_0 = a)\,,\\
\mathsf{code}(\mathsf{inr}(b)) &:\equiv \mathbf{0}\,.
\end{aligned}$$

Это очень простой пример метода доказательства, который довольно часто используется при оперировании гомотопической теорией в гомотопической теории типов; см., например, §§ 8.1 и 8.9. Теперь мы можем заявить:

Теорема 2.12.5. *Для всех $x : A + B$ имеет место* $(\mathsf{inl}(a_0) = x) \simeq \mathsf{code}(x)$.

Доказательство. Ключевой особенностью доказательства является то, что мы проводим его для всех точек x одновременно, что позволяет использовать принцип исключения для копроизведения. Сначала определим функцию

$$\mathsf{encode} : \prod_{(x:A+B)} \ \prod_{(p:\mathsf{inl}(a_0)=x)} \mathsf{code}(x)$$

транспортируя рефлексивность вдоль p:

$$\mathsf{encode}(x, p) :\equiv \mathsf{transport}^{\mathsf{code}}(p, \mathsf{refl}_{a_0})\,.$$

Отметим, что $\mathsf{refl}_{a_0} : \mathsf{code}(\mathsf{inl}(a_0))$ поскольку $\mathsf{code}(\mathsf{inl}(a_0)) \equiv (a_0 = a_0)$ по определению code. Далее, определим функцию

$$\mathsf{decode} : \prod_{(x:A+B)} \ \prod_{(c:\mathsf{code}(x))} (\mathsf{inl}(a_0) = x)\,.$$

Чтобы определить $\mathsf{decode}(x, c)$, мы можем сначала использовать принцип исключения для $A+B$, чтобы разделить случаи, основанные на том, является ли x формой $\mathsf{inl}(a)$ или формой $\mathsf{inr}(b)$.

В первом случае, когда $x \equiv \mathsf{inl}(a)$, будет $\mathsf{code}(x) \equiv (a_0 = a)$, так что c является идентификацией между a_0 и a. Таким образом, $\mathsf{ap}_{\mathsf{inl}}(c) : (\mathsf{inl}(a_0) = \mathsf{inl}(a))$, поэтому мы можем заключить, что это — $\mathsf{decode}(\mathsf{inl}(a), c)$.

Во втором случае, когда $x \equiv \mathsf{inr}(b)$, будет $code(x) \equiv \mathbf{0}$, так что c обитает в пустом типе. Таким образом, правило исключения для $\mathbf{0}$ дает значение для $\mathsf{decode}(\mathsf{inr}(b), c)$.

Это завершает определение decode; мы теперь показываем, что $\mathsf{encode}(x, \_)$ и $\mathsf{decode}(x, \_)$ являются квазиобратными для всех x. С одной стороны, пусть $x : A + B$ и $p : \mathsf{inl}(a_0) = x$; мы хотим показать, что $\mathsf{decode}(x, \mathsf{encode}(x, p)) = p$. Но теперь по индукции (на основе) пути достаточно учесть, что $x \equiv \mathsf{inl}(a_0)$ и $p \equiv \mathsf{refl}_{\mathsf{inl}(a_0)}$:

$$\begin{aligned}
\mathsf{decode}(x, \mathsf{encode}(x, p)) &\equiv \mathsf{decode}(\mathsf{inl}(a_0), \mathsf{encode}(x, \mathsf{refl}_{\mathsf{inl}(a_0)})) \\
&\equiv \mathsf{decode}(\mathsf{inl}(a_0), \mathsf{transport}^{\mathsf{code}}(\mathsf{refl}_{\mathsf{inl}(a_0)}, \mathsf{refl}_{a_0})) \\
&\equiv \mathsf{decode}(\mathsf{inl}(a_0), \mathsf{refl}_{a_0}) \\
&\equiv \mathsf{ap}_{\mathsf{inl}}(\mathsf{refl}_{a_0}) \\
&\equiv \mathsf{refl}_{\mathsf{inl}(a_0)} \\
&\equiv p\,.
\end{aligned}$$

С другой стороны, пусть $x : A + B$ и $c : code(x)$; мы хотим показать, что $\mathsf{encode}(x, \mathsf{decode}(x, c)) = c$. Мы снова можем выделить случаи, связанные с x. Если $x \equiv \mathsf{inl}(a)$, то $c : a_0 = a$ и $\mathsf{decode}(x, c) \equiv \mathsf{ap}_{\mathsf{inl}}(c)$, так что

$$\begin{aligned}
\mathsf{encode}(x, \mathsf{decode}(x, c)) &\equiv \mathsf{transport}^{\mathsf{code}}(\mathsf{ap}_{\mathsf{inl}}(c), \mathsf{refl}_{a_0}) \\
&\equiv \mathsf{transport}^{a \mapsto (a_0 = a)}(c, \mathsf{refl}_{a_0}) && \text{(по лемме 2.3.10)} \\
&\equiv \mathsf{refl}_{a_0} \cdot c && \text{(по лемме 2.11.2)} \\
&\equiv c\,.
\end{aligned}$$

Наконец, если $x \equiv \mathsf{inr}(b)$, то $c : \mathbf{0}$, поэтому отсюда следует все, что угодно. □

Конечно, существует соответствующая теорема, если мы зафиксируем $b_0 : B$ вместо $a_0 : A$.

В частности, из теоремы 2.12.5 следует, что для любых $a : A$ и $b : B$ существуют функции

$$\mathsf{encode}(\mathsf{inl}(a), \_) : (\mathsf{inl}(a_0) = \mathsf{inl}(a)) \to (a_0 = a)$$

и

$$\mathsf{encode}(\mathsf{inr}(b), \_) : (\mathsf{inl}(a_0) = \mathsf{inr}(b)) \to \mathbf{0}\,.$$

Вторая из этих форм «$\mathsf{inl}(a_0)$ не равно $\mathsf{inr}(b)$», то есть образы inl и inr не пересекаются. Традиционное чтение первой формы, где типы тождественности рассматриваются как высказывания, является просто инъективностью inl. Полная гомотопическая формулировка теоремы 2.12.5 дает больше информации: типы $\mathsf{inl}(a_0) = \mathsf{inl}(a)$ и $a_0 = a$ фактически эквивалентны, так как $\mathsf{inr}(b_0) = \mathsf{inr}(b)$ и $b_0 = b$.

Замечание 2.12.6. В частности, поскольку двухэлементный тип $\mathbf{2}$ эквивалентен $\mathbf{1} + \mathbf{1}$, имеем $0_{\mathbf{2}} \neq 1_{\mathbf{2}}$.

Приведенное доказательство иллюстрирует общий метод описания пространств путей, который мы будем часто использовать. Чтобы охарактеризовать пространство путей, первым шагом является определение «кодового словаря» сравнения расслоений, которое обеспечивает более четкое описание путей. Существует несколько различных методов доказательства того, что такое сравнительное расслоение эквивалентно траекториям (мы показываем несколько различных доказательств того же результата в §8.1). Тот, который мы здесь использовали, называется **методом кодирования-декодирования**: ключевой идеей является определение декодирования в общем случае для всех экземпляров расслоения (т.е. как функции

$\prod_{(x:A+B)} \mathsf{code}(x) \to (\mathsf{inl}(a_0) = x))$, так что индукция пути может использоваться для анализа $\mathsf{decode}(x, \mathsf{encode}(x,p))$.

Как обычно, мы можем также охарактеризовать действие транспортирования в типах копроизведений. Для типа X, пути $p : x_1 =_X x_2$ и семейств типов $A, B : X \to \mathcal{U}$, имеем

$$\mathsf{transport}^{A+B}(p, \mathsf{inl}(a)) = \mathsf{inl}(\mathsf{transport}^{A}(p,a))\,,$$
$$\mathsf{transport}^{A+B}(p, \mathsf{inr}(b)) = \mathsf{inr}(\mathsf{transport}^{B}(p,b))\,,$$

где, как обычно, $A+B$ в верхнем индексе уничижительно обозначает семейство типов $x \mapsto A(x)+B(x)$. Доказательством является простая индукция пути.

2.13 Натуральные числа

Мы используем метод кодирования-декодирования для характеризации пространства путей натуральных чисел, которое также является позитивным типом. В этом случае, не фиксируя одну концевую точку, мы охарактеризуем пути пространства одновременно с двух сторон. Таким образом, коды для тождественностей являются семейством типов

$$\mathsf{code} : \mathbb{N} \to \mathbb{N} \to \mathcal{U}\,,$$

определенными двойной рекурсией над $\mathbb{N}$ следующим образом

$$\begin{aligned}
\mathsf{code}(0,0) &:\equiv \mathbf{1}\\
\mathsf{code}(\mathsf{succ}(m),0) &:\equiv \mathbf{0}\\
\mathsf{code}(0,\mathsf{succ}(n)) &:\equiv \mathbf{0}\\
\mathsf{code}(\mathsf{succ}(m),\mathsf{succ}(n)) &:\equiv \mathsf{code}(m,n)\,.
\end{aligned}$$

Мы также определяем по рекурсии зависимую функцию $r : \prod_{(n:\mathbb{N})} \mathsf{code}(n,n)$, с

$$\begin{aligned}
r(0) &:\equiv \star\\
r(\mathsf{succ}(n)) &:\equiv r(n)\,.
\end{aligned}$$

Теорема 2.13.1. *Для любых $m, n : \mathbb{N}$ имеем $(m = n) \simeq \mathsf{code}(m,n)$.*

Доказательство. Определим

$$\mathsf{encode} : \prod_{m,n:\mathbb{N}} (m = n) \to \mathsf{code}(m,n)$$

с помощью транспортировки, $\mathsf{encode}(m,n,p) :\equiv \mathsf{transport}^{\mathsf{code}(m,-)}(p, r(m))$. Также определим

$$\mathsf{decode} : \prod_{m,n:\mathbb{N}} \mathsf{code}(m,n) \to (m = n)$$

двойной индукцией по m, n. Когда m и n одновременно равны 0, нам требуется функция $\mathbf{1} \to (0 = 0)$, которую мы определяем, чтобы направить все к refl_0. Когда m является следующим значением, а n равно 0 или наоборот, область $\mathsf{code}(m,n)$ равна $\mathbf{0}$, так что выделителя для $\mathbf{0}$ достаточно. А когда оба являются следующими значениями, мы можем определить $\mathsf{decode}(\mathsf{succ}(m), \mathsf{succ}(n))$ по композиции

$$\mathsf{code}(\mathsf{succ}(m), \mathsf{succ}(n)) \equiv \mathsf{code}(m, n) \xrightarrow{\mathsf{decode}(m,n)} (m = n) \xrightarrow{\mathsf{ap}_{\mathsf{succ}}} (\mathsf{succ}(m) = \mathsf{succ}(n)) .$$

Далее мы показываем, что $\mathsf{encode}(m, n)$ и $\mathsf{decode}(m, n)$ являются квазиобратными для всех m, n.

С одной стороны, если мы начнем с $p : m = n$, то индукцией по p достаточно показать, что

$$\mathsf{decode}(n, n, \mathsf{encode}(n, n, \mathsf{refl}_n)) = \mathsf{refl}_n .$$

Но $\mathsf{encode}(n, n, \mathsf{refl}_n) \equiv r(n)$, поэтому достаточно показать, что $\mathsf{decode}(n, n, r(n)) = \mathsf{refl}_n$. Мы можем доказать это индукцией по n. Если $n \equiv 0$, затем $\mathsf{decode}(0, 0, r(0)) = \mathsf{refl}_0$ по определению decode. А в случае следующих значений по индуктивному предположению мы имеем $\mathsf{decode}(n, n, r(n)) = \mathsf{refl}_n$, поэтому достаточно заметить, что $\mathsf{ap}_{\mathsf{succ}}(\mathsf{refl}_n) \equiv \mathsf{refl}_{\mathsf{succ}}(n)$.

С другой стороны, если мы начнем с $c : \mathsf{code}(m, n)$, то применим двойную индукцию по m и n. Если они оба равны 0, то $\mathsf{decode}(0, 0, c) \equiv \mathsf{refl}_0$, а $\mathsf{encode}(0, 0, \mathsf{refl}_0) \equiv r(0) \equiv \star$. Таким образом, достаточно вспомнить из §2.8, что каждый обитатель из $\mathbf{1}$ равен $\star$. Если m равно 0, а n является следующим значением или наоборот, то $c : \mathbf{0}$, так что здесь разбор случаев завершен. А в случае двух следующих значений мы имеем

$$\begin{aligned}
&\mathsf{encode}(\mathsf{succ}(m), \mathsf{succ}(n), \mathsf{decode}(\mathsf{succ}(m), \mathsf{succ}(n), c)) \\
&\qquad = \mathsf{encode}(\mathsf{succ}(m), \mathsf{succ}(n), \mathsf{ap}_{\mathsf{succ}}(\mathsf{decode}(m, n, c))) \\
&\qquad = \mathsf{transport}^{\mathsf{code}(\mathsf{succ}(m), -)}(\mathsf{ap}_{\mathsf{succ}}(\mathsf{decode}(m, n, c)), r(\mathsf{succ}(m))) \\
&\qquad = \mathsf{transport}^{\mathsf{code}(\mathsf{succ}(m), \mathsf{succ}(-))}(\mathsf{decode}(m, n, c), r(\mathsf{succ}(m))) \\
&\qquad = \mathsf{transport}^{\mathsf{code}(m, -)}(\mathsf{decode}(m, n, c), r(m)) \\
&\qquad = \mathsf{encode}(m, n, \mathsf{decode}(m, n, c)) \\
&\qquad = c
\end{aligned}$$

используя предположение индуктивности. □

В частности имеем

$$\mathsf{encode}(\mathsf{succ}(m), 0) : (\mathsf{succ}(m) = 0) \to \mathbf{0} \tag{2.13.2}$$

что выражает «0 не является следующим значением какого-либо натурального числа». Также имеет место композиция

$$(\mathsf{succ}(m) = \mathsf{succ}(n)) \xrightarrow{\mathsf{encode}} \mathsf{code}(\mathsf{succ}(m), \mathsf{succ}(n)) \equiv \mathsf{code}(m, n) \xrightarrow{\mathsf{decode}} (m = n) \tag{2.13.3}$$

которая показывает, что функция succ является инъективной.

Мы рассмотрим более общие позитивные типы в главах 5 и 6. В главе 8 мы увидим, что для вычисления гомотопических групп сфер также можно использовать метод, который используется здесь для характеризации типов тождественности копроизведений и $\mathbb{N}$.

2.14 Пример: равенство структур

Рассмотрим теперь один пример, иллюстрирующий взаимодействие между структурой группоида на типе и конструкторами типов. Во вступлении мы отметили, что одно из преимуществ унивалентности состоит в том, что две изоморфные вещи взаимозаменяемы в том смысле, что каждое свойство или конструкция, связанные с одним, также относятся к другому. Обычные «злоупотребления обозначениями» становятся формально истинными. Само понятие «унивалентность» говорит о том, что эквивалентные типы равны и, следовательно, взаимозаменяемы, что включает, например, общая практика идентификации изоморфных множеств. Более того, когда мы определяем другие математические объекты как множества или даже общие типы, снабженные структурой или свойствами, мы можем вывести правильное понятие равенства для них из унивалентности. Мы проиллюстрируем этот момент показательным примером в главе 9, где мы определяем основные понятия теории категорий таким образом, чтобы равенство категорий было эквивалентностью, равенство функторов было естественным изоморфизмом и т.д.; см., в частности, §9.8. В этом разделе мы опишем очень простой пример, происходящий из алгебры.

Для простоты мы используем *полугруппы* в качестве примера, где полугруппа — это тип, снабженный ассоциативной операцией «умножения». Те же идеи применяются и к другим алгебраическим структурам, таким как моноиды, группы и кольца. Напомним из §§ 1.6 и 1.11, что определение какой-либо математической структуры должно интерпретироваться как определяющее тип таких структур, как определенный итерированный $\sum$-тип. В случае полугрупп это дает следующее.

Определение 2.14.1. Для типа A тип $\mathsf{SemigroupStr}(A)$ **полугрупповых структур** с носителем A определяется как

$$\mathsf{SemigroupStr}(A) :\equiv \sum_{(m:A\to A\to A)} \prod_{(x,y,z:A)} m(x, m(y,z)) = m(m(x,y),z).$$

Полугруппа есть тип вместе со структурой:

$$\mathsf{Semigroup} :\equiv \sum_{A:U} \mathsf{SemigroupStr}(A).$$

В следующих двух разделах мы опишем два способа, с помощью которых унивалентность облегчает работу с такими полугруппами.

2.14.1 Поднятие эквивалентностей

При нестогом подходе, можно сказать, что биекция между множествами A и B «очевидно» индуцирует изоморфизм между полугрупповыми структурами на A и полугрупповыми структурами на B. С унивалентностью это действительно очевидно, поскольку, учитывая эквивалентность между типами A и B, мы можем автоматически получить полугрупповую структуру на B от подобной структуры на A и, кроме того, показать, что этот вывод является эквивалентностью полугрупповых структур. Причина в том, что $\mathsf{SemigroupStr}$ — это семейство типов и, следовательно, обладает действием на пути между типами, заданными посредством $\mathsf{transport}$:

$$\mathsf{transport}^{\mathsf{SemigroupStr}}(\mathsf{ua}(e)) : \mathsf{SemigroupStr}(A) \to \mathsf{SemigroupStr}(B).$$

Более того, это отображение является эквивалентностью, так как $\mathsf{transport}^C(\alpha)$ всегда является эквивалентностью с обратным $\mathsf{transport}^C(\alpha^{-1})$, см. леммы 2.1.4 и 2.3.9.

В то время как аксиома унивалентности гарантирует, что это отображение существует, нам нужно использовать факты о $\mathsf{transport}$, доказанные в предыдущих разделах, для понимания того, что она на самом деле делает. Пусть (m, a) — полугрупповая структура на A, и мы исследуем индуцированную полугрупповую структуру на B, заданную формулой

$$\mathsf{transport}^{\mathsf{SemigroupStr}}(\mathsf{ua}(e), (m, a)).$$

Сначала, поскольку $\mathsf{SemigroupStr}(X)$ определяется как $\sum$-тип, по теореме 2.7.4

$$\mathsf{transport}^{\mathsf{SemigroupStr}}(\mathsf{ua}(e), (m, a)) = (m', a')$$

где m' — индуцированная операция умножения на B

$$\begin{aligned} m' &: B \to B \to B \\ m'(b_1, b_2) &:\equiv \mathsf{transport}^{X \mapsto (X \mapsto X \mapsto X)}(\mathsf{ua}(e), m)(b_1, b_2) \end{aligned}$$

а a' — индуцированное доказательство того, что m' ассоциативно. Снова по теореме 2.7.4, имеем

$$\begin{aligned} a' &: \mathsf{Assoc}(B, m') \\ a' &:\equiv \mathsf{transport}^{(X,m) \mapsto \mathsf{Assoc}(X,m)}((\mathsf{pair}^{=}(\mathsf{ua}(e), \mathsf{refl}_{m'})), a), \end{aligned} \tag{2.14.2}$$

где $\mathsf{Assoc}(X, m)$ есть тип $\prod_{(x,y,z:X)} m(x, m(y, z)) = m(m(x, y), z)$. По функциональной экстенсиональности, достаточно исследовать поведение m' при применении к аргументам $b_1, b_2 : B$. Применяя (2.9.4) дважды, получаем, что $m'(b_1, b_2)$ равно

$$\mathsf{transport}^{X \mapsto X}(\mathsf{ua}(e), m(\mathsf{transport}^{X \mapsto X}(\mathsf{ua}(e)^{-1}, b_1), \mathsf{transport}^{X \mapsto X}(\mathsf{ua}(e)^{-1}, b_2))).$$

Тогда, так как ua квазиобратно к $\mathsf{transport}^{X \mapsto X}$, это равно

$$e(m(e^{-1}(b_1), e^{-1}(b_2))).$$

Таким образом, для двух элементов из B индуцированное умножение m' передает их в A, используя эквивалентность e, умножает их в A и затем возвращает результат в B по e, как и следовало ожидать.

Более того, хотя мы не приводим доказательства, можно вывести, что индуцированное доказательство ассоциативности m' (см. (2.14.2)) равно функции, передающей $b_1, b_2, b_3 : B$ к пути, следующими шагами:

$$\begin{aligned} m'(m'(b1, b2), b3) &= e(m(e^{-1}(m'(b_1, b_2)), e^{-1}(b_3))) \\ &= e(m(e^{-1}(e(m(e^{-1}(b_1), e^{-1}(b_2)))), e^{-1}(b_3))) \\ &= e(m(m(e^{-1}(b_1), e^{-1}(b_2)), e^{-1}(b_3))) \\ &= e(m(e^{-1}(b_1), m(e^{-1}(b_2), e^{-1}(b_3)))) \\ &= e(m(e^{-1}(b_1), e^{-1}(e(m(e^{-1}(b_2), e^{-1}(b_3)))))) \\ &= e(m(e^{-1}(b_1), e^{-1}(m'(b_2, b_3)))) \\ &= m'(b_1, m'(b_2, b_3)). \end{aligned} \tag{2.14.3}$$

Эти шаги используют доказательство a, что m ассоциативно, и обратные законы для e. С точки зрения алгебры может показаться странным исследовать тождество доказательства того, что операция ассоциативна, но это имеет смысл, если мы думаем о A и B как об обычных пространствах с нетривиальными гомотопиями между путями. В главе 3 мы введем понятие *множества*, которое является типом с только тривиальными гомотопиями, и если мы рассмотрим полугрупповые структуры на множествах, то любые два таких доказательства ассоциативности автоматически равны.

2.14.2 Равенство полугрупп

Используя уравнения для пространств путей, рассмотренные в предыдущих разделах, мы можем исследовать, когда две полугруппы равны. Для полугрупп (A, m, a) и (B, m', a'), по теореме 2.7.2, тип путей $(A, m, a) =_{\mathsf{Semigroup}} (B, m', a')$ равен типу пар

$$\begin{aligned} &p_1 : A =_{\mathcal{U}} B \qquad \text{и} \\ &p_2 : \mathsf{transport}^{\mathsf{SemigroupStr}}(p_1, (m, a)) = (m', a')\,. \end{aligned}$$

По унивалентности, p_1 есть $\mathsf{ua}(e)$ для некоторой эквивалентности e. По теореме 2.7.2, функциональной экстенсиональности и приведенному выше анализу транспортировки в семействе типов $\mathsf{SemigroupStr}$, p_2 эквивалентно паре доказательств, первое из которых показывает, что

$$\prod_{y_1, y_2 : B} e(m(e^{-1}(y_1), e^{-1}(y_2))) = m'(y_1, y_2)$$

а второе из них показывает, что a' равно индуцированному доказательству ассоциативности, построенному от a в (2.14.3). Но аннулирование инверсий (2.14.2) эквивалентно

$$\prod_{x_1, x_2 : A} e(m(x_1, x_2)) = m'(e(x_1), e(x_2))\,.$$

Это говорит о том, что e коммутирует с бинарной операцией в том смысле, что она применяет умножение в A (т.е. m) к умножению в B (т.е. m'). Аналогичное действие возможно для уравнения, связывающего a и a'. Таким образом, равенство полугрупп состоит в точности из эквивалентности на типах носителей, коммутирующих с полугрупповой структурой.

Для всеобщих типов, доказательство ассоциативности рассматривается как часть структуры полугруппы. Однако, если мы ограничимся множественно-подобными типами (опять же, см. главу 3), уравнение, связывающее a и a', тривиально верно. Более того, в этом случае эквивалентность между множествами является в точности биекцией. Таким образом, мы пришли к стандартному определению *изоморфизма полугрупп*: биекция на множествах носителей, которая сохраняет операцию умножения. Можно также использовать категорное определение изоморфизма, определяя *гомоморфизм полугрупп* как отображение, сохраняющее умножение, и прийти к выводу, что равенство полугрупп совпадает с двумя взаимно обратными гомоморфизмами; но здесь мы не будем показывать подробности; см. §9.8.

Вывод состоит в том, что благодаря унивалентности полугруппы одинаковы, когда они изоморфны как алгебраические структуры. Как мы увидим в §9.8, подобное заключение применяется в более общем плане: в теории гомотопических типов все конструкции математических структур автоматически учитывают изоморфизмы, без каких-либо утомительных доказательств или злоупотреблений обозначениями.

2.15 Универсальные свойства

Комбинируя правила вычисления пути, описанные в предыдущих разделах, мы можем показать, что различные операции формирования типа удовлетворяют ожидаемым универсальным свойствам, интерпретируемым гомотопически как эквивалентность. Например, для заданных типов X, A, B имеем функцию

$$(X \to A \times B) \to (X \to A) \times (X \to B)\,, \tag{2.15.1}$$

определяемую как $f \mapsto (\mathsf{pr}_1 \circ f, \mathsf{pr}_2 \circ f)$.

Теорема 2.15.2. (2.15.1) *является эквивалентностью.*

Доказательство. Определим квазиобратность, подавая (g, h) в $\lambda x. (g(x), h(x))$ (технически мы использовали принцип индукции для декартова произведения $(X \to A) \times (X \to B)$, чтобы сводить к случаю пары; с этого момента мы часто будем применять этот принцип без явного упоминания).

Теперь, для $f : X \to A \times B$ взаимообратные составляющие образуют функцию

$$\lambda x. (\mathsf{pr}_1(f(x)), \mathsf{pr}_2(f(x))) . \tag{2.15.3}$$

По теореме 2.6.2, для любого $x : X$ имеем $(\mathsf{pr}_1(f(x)), \mathsf{pr}_2(f(x))) = f(x)$. Таким образом, по функциональной экстенсиональности, функция (2.15.3) равна f.

С другой стороны, для (g, h), взаимообратные составляющие образуют пару $(\lambda x. g(x), \lambda x. h(x))$. По принципу единственности для функций она (дефинициально) равна (g, h). □

Фактически, у нас также есть зависимая типизированная версия этого универсального свойства. Предположим, что задан тип X и семейства типов $A, B : X \to \mathcal{U}$. Тогда имеем функцию

$$\left(\prod_{x:X}(A(x) \times B(x))\right) \to \left(\prod_{x:X}A(x)\right) \times \left(\prod_{x:X}B(x)\right) , \tag{2.15.4}$$

определенную, как и выше, в виде $f \mapsto (\mathsf{pr}_1 \circ f, \mathsf{pr}_2 \circ f)$.

Теорема 2.15.5. (2.15.4) *является эквивалентностью.*

Доказательство. Предоставляется читателю. □

Когда $\sum$-типы являются обобщением декартовых произведений, они удовлетворяют обобщенному варианту этого универсального свойства. Переключаясь на зависимую типизированную версию, предположим, что у нас есть тип X и семейства типов $A : X \to \mathcal{U}$ и $P : \prod_{(x:X)} A(x) \to \mathcal{U}$. Тогда имеем функцию

$$\left(\prod_{x:X} \sum_{(a:A(x))} P(x, a)\right) \to \left(\sum_{(g:\prod_{(x:X)} A(x))} \prod_{(x:X)} P(x, g(x))\right) . \tag{2.15.6}$$

Заметим, что если $P(x, a) :\equiv B(x)$ для некоторого $B : X \to \mathcal{U}$, то (2.15.6) сводится к (2.15.4).

Теорема 2.15.7. (2.15.6) *является эквивалентностью.*

Доказательство. Как и выше, мы определяем квазиобратность, подавая (g, h) в $\lambda x. (g(x), h(x))$. Теперь, для $f : \prod_{(x:X)} \sum_{(a:A(x))} P(x, a)$ взаимообратные составляющие образуют функцию

$$\lambda x. (\mathsf{pr}_1(f(x)), \mathsf{pr}_2(f(x))) . \tag{2.15.8}$$

Для любого $x : X$, по следствию 2.7.3 (принцип единственности для $\sum$-типов) имеем

$$(\mathsf{pr}_1(f(x)), \mathsf{pr}_2(f(x))) = f(x) .$$

Таким образом, по функциональной экстенсиональности, (2.15.8) равен f. С другой стороны, для (g, h), взаимообратные составляющие образуют пару $(\lambda x. g(x), \lambda x. h(x))$, которая, как и выше, дефинициально равна (g, h). □

Это примечательно, потому что интерпретация высказываний как типов для (2.15.6) является «аксиомой выбора». Если мы будем воспринимать $\sum$ как «существует», а $\prod$ (иногда) как «для всех», то можно интерпретировать:

- $\prod_{(x:X)} \sum_{(a:A(x))} P(x,a)$ как «для всех $x : X$существует $a : A(x)$ такое, что $P(x,a)$» и
- $\sum_{(g:\prod_{(x:X)} A(x))} \prod_{(x:X)} P(x, g(x))$ как «существует функция выбора $g : \prod_{(x:X)} A(x)$ такая, что для всех $x : X$ имеем $P(x, g(x))$».

Таким образом, в теореме 2.15.7 говорится, что аксиома выбора не только «истинна», ее условие фактически эквивалентно ее заключению (с другой стороны, классический математик может обнаружить, что (2.15.6) не несет обычного значения аксиомы выбора, так как мы уже указали значения g, и нет выбора, который еще предстоит сделать; мы вернемся к этому в §3.8).

Вышеупомянутое универсальное свойство для типов пар относится к «отображению в», что хорошо известно из теоретико-категорного понятия произведений. Однако типы пар также имеют универсальное свойство для «отображения наружу», которое может выглядеть менее знакомым. В случае декартовых произведений независимая версия просто выражает декартово замкнутое сопряжение:

$$((A \times B) \to C) \simeq (A \to (B \to C)).$$

Зависимая версия этого формулируется для семейства типов $C : A \times B \to \mathcal{U}$:

$$\left(\prod_{w:A\times B} C(w)\right) \simeq \left(\prod_{(x:A)} \prod_{(y:B)} C(x,y)\right).$$

Здесь функция направления справа-налево является просто принципом индукции для $A \times B$, а направления слева-направо — оценкой пары. Мы оставляем читателю доказательство их квазиобратности. Существует также версия для $\sum$-типов:

$$\left(\prod_{w:\sum_{(x:A)} B(x)} C(w)\right) \simeq \left(\prod_{(x:A)} \prod_{(y:B(x))} C(x,y)\right). \tag{2.15.9}$$

Опять же, функция направления справа-налево является принципом индукции.

Некоторые другие принципы индукции также являются частью универсальных свойств такого рода. Например, индукция пути является эквивалентностью направления справа-налево следующим образом:

$$\left(\prod_{(x:A)} \prod_{(p:a=x)} B(x,p)\right) \simeq B(a, \mathsf{refl}_a) \tag{2.15.10}$$

для любого $a : A$ и семейства типов $B : \prod_{(x:A)}(a = x) \to \mathcal{U}$. Однако индуктивные типы с рекурсией, такие как натуральные числа, имеют более сложные универсальные свойства; см. главу 5.

Так как теорема 2.15.2 выражает обычное универсальное свойство декартова произведения (в соответствующем гомотопическом смысле), категорно предрасположенный читатель может заинтересоваться другими пределами и копределами типов. В упражнении 2.9 мы просим читателя показать, что тип копроизведения $A + B$ также имеет ожидаемое универсальное свойство, а нулевые случаи $\mathbf{1}$ (терминальный объект) и $\mathbf{0}$ (начальный объект) просты.

Предполагаемая явная конструкция для обратных образов: для $f : A \to C$ и $g : B \to C$ определяется

$$A \times_C B :\equiv \sum_{(a:A)} \sum_{(b:B)} (f(a) = g(b)) \,. \tag{2.15.11}$$

В упражнении 2.11 мы просим читателя проверить это. Некоторые общие гомотопические пределы могут быть построены аналогичным образом, но для копределов нам понадобится новый ингредиент; см. главу 6.

Примечания

Определение типов тождественности с их принципом индукции связано с именем Мартина-Лёфа (Martin-Löf) [ML98]. Как упоминалось в примечаниях к главе 1, нашими типами тождественности являются те, которые относятся к *интенсиональной* теории типов, а не к *экстенсиональной* теории типов. В общем случае понятие равенства называется «интенсиональным», если оно отличает объекты, основанные на их конкретных определениях, и «экстенсиональным», если оно не различает объекты, которые имеют одно и то же «расширение» или «наблюдаемое поведение». В терминологии Фреге (Frege) интенсиональное равенство сопоставляет *смысл*, в то время как экстенсиональное сравнивает только *ссылку*. Мы можем также говорить о том, что одно равенство является «более» или «менее» экстенсиональным, чем другое, что означает, что оно учитывает меньше или больше интенсивных аспектов объектов, соответственно.

Интенсиональная теория типов названа так потому, что ее *дефинициальное* равенство, $x \equiv y$, является предельным интенсиональным равенством: в нем говорится, что x и y «имеют одно и то же определение», после того как мы распишем определяющие уравнения функций. Напротив, пропозициональный тип равенства $x = y$ является более экстенсиональным даже в без-аксиоматической интенсиональной теории типов главы 1: например, по индукции мы можем доказать, что $n+m = m+n$ для всех $m, n : \mathbb{N}$, но мы не можем сказать, что $n+m \equiv m+n$ для всех $m, n : \mathbb{N}$, так как *определение* сложения трактует его аргументы асимметрично. Мы можем сделать тип тождественности интенсиональной теории типов еще более экстенсиональным, добавив аксиомы, такие как функциональная экстенсиональность (две функции равны, если они имеют одинаковое поведение на всех входах, независимо от того, как они определены) и унивалентность (которую можно рассматривать как свойство экстенсиональности для универсума: два типа равны, если они ведут себя одинаково во всех контекстах). Аксиомы функциональной экстенсиональности и унивалентности в частном случае простых высказываний («пропозициональная экстенсивность») появились уже в первых теориях типов Рассела (Russell) и Чёрча (Church).

Как упоминалось ранее, *экстенсиональная* теория типов включает также «правило отражения», гласящее, что если $p : x = y$, то фактически $x \equiv y$. Таким образом, экстенсиональная теория типов названа так потому, что она не допускает какого-либо чисто *интенсионального* равенства: правило отражения вынуждает дефинициальное равенство совпадать с более экстенсиональным типом тождественности. Более того, из правила отражения можно вывести функциональную экстенсивность (по крайней мере, при наличии принципа дефинициальной уникальности для функций). Однако правило отражения также подразумевает, что вся высшая группоидная структура рушится (см. упражнение 2.14) и, следовательно, несовместима с аксиомой унивалентности (см. пример 3.1.9). Поэтому, в отношении унивалентности как свойства расширения, можно сказать, что интенсиональная теория типов допускает типы тождественности, которые в ней «более экстенсиональны», чем в экстенсиональной теория типов.

Доказательства симметрии (инверсии) и транзитивности (конкатенации) для равенств хорошо известны в теории типов. Тот факт, что они вносят каждый тип в 1-группоид (вплоть до гомотопии), использовался в [HS98], чтобы дать первую семантику «гомотопического» стиля для теории типов.

Фактическая гомотопическая интерпретация, с типами тождественности как пространствами путей, и семействами типов, как расслоений, связана с [AW09], где использован формализм категорий моделей Квиллена (Quillen). Интерпретация в (строгих) ∞-группоидах также была дана в статье [War08]. Про построение всех высших операций и когерентностей ∞-группоида в теории типов см. в [Lum10] и [vdBG11].

Операции, такие как $\mathsf{transport}^P(p, \_)$ и ap_f, и одно хорошее представление эквивалентности, впервые были широко изучены в теории типов Воеводским, с использованием системы доказательств Coq. Впоследствии было найдено много других эквивалентных определений эквивалентности, которые сравниваются в главе 4.

«Вычисляемая» интерпретация типов тождественности, транспортировки и т.д., описанных в §2.5, была подчеркнута в [LH12]. Там также построена теория «1-усеченных» типов (см. главу 7), в которой правила являются дефинициальными равенствами. Возможность распространить это на всю необработанную теорию является предметом текущих исследований.

Наивная форма функциональной экстенсиональности, которая гласит, что «если две функции поточечно равны, то они в целом равны», является общепринятой аксиомой теории типов, полностью возвращаясь к [WR27]. Некоторые более сильные формы функциональной экстенсиональности были рассмотрены в [Gar09]. Используемая нами версия, которая идентифицирует типы тождественности функциональных типов с точностью до эквивалентности, была впервые изучена Воеводским, который также доказал, что это подразумевается наивной версией (и унивалентностью, см. §4.9).

Аксиома унивалентности также связана с Воеводским. Первоначально она была мотивирована семантическими соображениями по модели симплициальных множеств; см. [KLV12]. Аналогичная аксиома, мотивированная группоидной моделью, была предложена Хофманом (Hofmann) и Штрайхером (Streicher) [HS98] под названием «расширение универсума». Они использовали квазиобратность (2.4.5), а не хорошее понятие «эквивалентности», что, следовательно, является корректным (и эквивалентным унивалентности) только для универсумов 1-типов (см. определение 3.1.7).

В теории типов, которую мы используем в этой книге, функциональная экстенсиональность и унивалентность должны приниматься как аксиомы, т.е. элементы, утверждаемые на принадлежность некоторому типу, но не построенные в соответствии с правилами для этого типа. Несмотря на работоспособность, такой подход имеет несколько недостатков. Например, теория типов формально лучше себя ведет, если мы можем основывать ее целиком на правилах, а не на утверждении аксиом. Иногда также неудобно, что теоремы из §§ 2.6–2.13 являются только пропозициональными равенствами (путями) или эквивалентносями, так как в этом случае мы должны явно указывать всякий раз, когда мы проходим через них. Одним из направлений современных исследований в гомотопической теории типов является описание системы типов, в которой эти правила являются *дефинициальными* равенствами, одновременно решая обе эти проблемы. Пока это сделано только в некоторых простых случаях, хотя предварительные результаты, такие как [LH12], являются многообещающими. Существуют также другие потенциальные способы введения унивалентности и функциональной экстенсиональности в теорию типов, таких, как предполагающих наличие достаточно сильного представления о «высших частных» или «высших индуктивно-рекурсивных типах».

Простые выводы в §§ 2.12–2.13, такие как «inl и inr являются инъективными и непересекающимися», хорошо известны в теории типов, а построение функции encode является обычным способом их доказательства. Более изощренный подход, который мы описали, характеризующий весь тип тождественности позитивного типа (вплоть до эквивалентности), является более поздним развитием; см., например, [LS13].

Теоретико-типовая аксиома выбора (2.15.6) была упомянута в оригинальной статье Уильяма Говарда (William Howard) [How80] о соответствии высказываний как типов и изучалась далее Мартином-Лёфом с введением его теории зависимых типов. Эта аксиома упоминается как «закон распределения» в теории множеств Бурбаки (Bourbaki) [Bou68].

Для более полного (и формализованного) обсуждения обратных образов и более общих гомотопических пределов в гомотопической теории типов см. [AKL13]. Пределы диаграмм над направленными графами — это самый простой общий вид предела для формализации; проблема с диаграммами над категориями (или, вообще говоря, $(\infty, 1)$-категории) состоит в том, что в понятие бесконечно многих условий связности входит понятие (гомотопически связной) диаграммы. Решение этой проблемы является важным открытым вопросом в гомотопической теории типов.

Упражнения

Упражнение 2.1. Покажите, что три очевидных доказательства леммы 2.1.2 попарно равноправны.

Упражнение 2.2. Покажите, что три равноправных доказательства, построенные в предыдущем упражнении, образуют коммутативный треугольник. Другими словами, если три определения конкатенации обозначаются $(p \cdot_1 q)$, $(p \cdot_2 q)$ и $(p \cdot_3 q)$, то конкатенированное равенство

$$(p \cdot_1 q) = (p \cdot_2 q) = (p \cdot_3 q)$$

равно равенству $(p \cdot_1 q) = (p \cdot_3 q)$.

Упражнение 2.3. Дайте четвертое, другое, доказательство леммы 2.1.2 и докажите, что оно равно остальным.

Упражнение 2.4. Определить индукцией по n общее понятие n-**мерного пути** в типе A, одновременно с типом границ для таких путей.

Упражнение 2.5. Докажите, что функции (2.3.6) и (2.3.7) являются обратными эквивалентностями.

Упражнение 2.6. Докажите, что если $p : x = y$, то функция $(p \cdot \_) : (y = z) \to (x = z)$ является эквивалентностью.

Упражнение 2.7. Сформулируйте и докажите обобщение теоремы 2.6.5 от декартовых произведений к $\sum$-типам.

Упражнение 2.8. Сформулируйте и докажите аналог теоремы 2.6.5 для копроизведений.

Упражнение 2.9. Докажите, что копроизведения имеют ожидаемое универсальное свойство,

$$(A + B \to X) \simeq (A \to X) \times (B \to X).$$

Можете ли вы обобщить это на эквивалентность с участием зависимых функций?

Упражнение 2.10. Докажите, что $\sum$-типы являются «ассоциативными», поскольку для любых $A : \mathcal{U}$ и семейств $B : A \to \mathcal{U}$ и $C : (\sum_{(x:A)} B(x)) \to \mathcal{U}$, имеет место

$$\left(\sum_{(x:A)} \sum_{(y:B(x))} C((x,y)) \right) \simeq \left(\sum_{p:\sum_{(x:A)} B(x)} C(p) \right).$$

Упражнение 2.11. (Гомотопический) **коммутативный квадрат**

$$\begin{tikzcd} P \arrow[r, "h"] \arrow[d, "k"'] & A \arrow[d, "f"] \\ B \arrow[r, "g"'] & C \end{tikzcd}$$

содержит функции f, g, h и k с путем $f \circ h = g \circ k$. Обратите внимание, что это как раз элемент обратного образа $(P \to A) \times_{P \to C} (P \to B)$, как определено в (2.15.11). Коммутативный квадрат называется (гомотопическим) **квадратом обратного образа**, если для любого X индуцированное отображение

$$(X \to P) \to (X \to A) \times_{(X \to C)} (X \to B)$$

является эквивалентностью. Докажите, что обратный образ $P :\equiv A \times_C B$, определенный в (2.15.11), является углом квадрата обратного образа.

Упражнение 2.12. Пусть заданы два коммутативных квадрата

$$\begin{tikzcd} A \arrow[r] \arrow[d] & C \arrow[r] \arrow[d] & E \arrow[d] \\ B \arrow[r] & D \arrow[r] & F \end{tikzcd}$$

и предположим, что правый квадрат является квадратом обратного образа. Докажите, что левый квадрат является квадратом обратного образа тогда и только тогда, когда внешний прямоугольник является квадратом обратного образа.

Упражнение 2.13. Покажите, что $(\mathbf{2} \simeq \mathbf{2}) \simeq \mathbf{2}$.

Упражнение 2.14. Предположим, мы добавим к теории типов правило отражения равенства, которое гласит, что если есть элемент $p : x = y$, то фактически $x \equiv$ y. Докажите, что для любого $p : x = x$ имеем $p \equiv \mathsf{refl}_x$ (это означает, что каждый тип является множеством в том смысле, который должен быть введен в §3.1; см. §7.2).

Упражнение 2.15. Покажите, что лемма 2.10.5 может быть усилена до

$$\mathsf{transport}^B(p, \_) =_{B(x) \to B(y)} \mathsf{idtoeqv}(\mathsf{ap}_B(p))$$

без использования функциональной экстенсиональности (в этом и других подобных случаях, по-видимому, более слабая формулировка была выбрана для удобочитаемости и согласованности).

Упражнение 2.16. Предположим, что вместо функциональной экстенсиональности (аксиома 2.9.3) мы предполагаем существование только элемента

$$\mathsf{funext} : \prod_{(A:\mathcal{U})} \prod_{(B:A\to\mathcal{U})} \prod_{(f,g:\prod_{(x:A)} B(x))} (f \sim g) \to (f = g)$$

(предполагается без связи с happly). Докажите, что на самом деле этого достаточно, чтобы заключить в себе всю аксиому функциональной экстенсиональности (т.е., что happly является эквивалентностью). Это связано с Воеводским; его доказательство сложно и может потребовать привлечения понятий из последующих глав.

Упражнение 2.17.

(i) Покажите, что если $A \simeq A'$ и $B \simeq B'$, то $(A \times B) \simeq (A' \times B')$.

(ii) Приведите два доказательства (i), с использованием и без использования унивалентности, и покажите, что эти два доказательства равны.

(iii) Сформулируйте и докажите аналогичный результат для других конструкторов типов: $\sum$, $\to$, $\prod$ и $+$.

Упражнение 2.18. Сформулируйте и докажите версию леммы 2.4.3 для зависимых функций.

Глава 3

Множества и логика

Теория типов, формальная или неформальная, представляет собой набор правил для манипулирования типами и их элементами. Но при неформальном описании математики на естественном языке мы обычно используем знакомые слова, в частности логические связки, такие как «и» и «или», и логические кванторы, такие как «для всех» и «существует». В отличие от теории множеств теория типов предлагает нам более одного способа рассматрения этих фраз естественного языка в качестве операций над типами. Эта потенциальная двусмысленность должна быть решена путем определения локальных или глобальных соглашений введением новых аннотаций к неформальному описанию математики, или того и другого. Это требует некоторого привыкания, но компенсируется тем фактом, что, поскольку теория типов дает возможность проведения более тонкого анализа логики, мы можем более точно представлять математику с меньшим количеством «злоупотреблений естественным языком», чем в теоретико-множественных основах. В этой главе мы объясним затронутые проблемы и оправдаем сделанный нами выбор.

3.1 Множества и n-типы

Чтобы объяснить связь между логикой теории типов и логикой теории множеств, полезно иметь понятие *множества* в теории типов. В то время как типы вообще ведут себя как пространства или высшие группоиды, они содержат подкласс, который ведет себя как множества в традиционной теоретико-множественной системе. С категорной точки зрения мы можем рассматривать *дискретные* группоиды, которые определяются множеством объектов и только морфизмами тождественности в качестве высших морфизмов; топологически, мы можем рассматривать пространства с дискретной топологией. В более общем плане мы можем рассматривать группоиды или пространства, которые *эквивалентны* таким типам; поскольку все, что мы делаем в теории типов, зависит от гомотопии, мы не можем ожидать различия.

Интуитивно мы ожидали бы, что тип «будет множеством» в этом смысле, если он не имеет высшей гомотопической информации: любые два параллельных пути равны (вплоть до гомотопии), это верно и для параллельных высших путей во всех измерениях. К счастью, потому что все в гомотопической теории типов автоматически функториально/непрерывно, этого оказывается достаточно, чтобы задаться соответствующим вопросом на нижнем уровне.

Определение 3.1.1. Тип A является **множеством**, если для всех $x, y : A$ и всех $p, q : x = y$ имеет место $p = q$.

Точнее, $\mathsf{isSet}(A)$определяется как тип

$$\mathsf{isSet}(A) :\equiv \prod_{(x,y:A)} \prod_{(p,q:x=y)} (p = q)\,.$$

Как упоминалось в §1.1, множества в гомотопической теории типов не похожи на множества в теории множеств ZF, поскольку нет глобального «предиката принадлежности» $\in$. Они больше похожи на множества, используемые в структурной математике и теории категорий, элементы которых являются «абстрактными точками», которых мы наделяем структурой с функциями и отношениями. Это все, что нам нужно, чтобы использовать их в качестве базовой системы для большинства математических разделов, базирующихся на множествах; мы рассмотрим несколько примеров в главе 10.

Какие типы являются множествами? В главе 7 мы рассмотрим детальнее более общую формулировку этого вопроса, а на данный момент мы можем привести несколько простых примеров.

Пример 3.1.2. Тип $\mathbf{1}$ — это множество. В силу теоремы 2.8.1, для любых $x, y : \mathbf{1}$ тип $(x = y)$ эквивалентен $\mathbf{1}$. Так как любые два элемента из $\mathbf{1}$ равны, это означает, что любые два элемента из $x = y$ равны.

Пример 3.1.3. Тип $\mathbf{0}$ является множеством, для любых $x, y : \mathbf{0}$ можно вывести все, что угодно, по принципу индукции для $\mathbf{0}$.

Пример 3.1.4. Тип $\mathbb{N}$ натуральных чисел также является множеством. Это следует из теоремы 2.13.1, так как все типы тождественности $x =_{\mathbb{N}} y$ эквивалентны либо $\mathbf{1}$, либо $\mathbf{0}$, а любые два обитателя из $\mathbf{1}$ или из $\mathbf{0}$ равны. Мы рассмотрим еще одно доказательство этого факта в главе 7.

Большинство операций формирования типов, рассмотренные до сих пор, также сохраняют и множества.

Пример 3.1.5. Если A и B — множества, то таковым является и $A \times B$. Для $x, y : A \times B$ и $p, q : x = y$, по теореме 2.6.2, имеем $p = \mathsf{pair}^{=}(\mathsf{ap}_{\mathsf{pr}_1}(p), \mathsf{ap}_{\mathsf{pr}_2}(p))$ и $q = \mathsf{pair}^{=}(\mathsf{ap}_{\mathsf{pr}_1}(q), \mathsf{ap}_{\mathsf{pr}_2}(q))$. Но $\mathsf{ap}_{\mathsf{pr}_1}(p) = \mathsf{ap}_{\mathsf{pr}_1}(q)$, т.к. A — множество, и $\mathsf{ap}_{\mathsf{pr}_2}(p) = \mathsf{ap}_{\mathsf{pr}_2}(q)$, т.к. B — множество; следовательно $p = q$.

Так же, если A — множество, а $B : A \to \mathcal{U}$ таково, что $B(x)$ — множество, то $\sum_{(x:A)} B(x)$ также является множеством.

Пример 3.1.6. Если A — *произвольный* тип и $B : A \to \mathcal{U}$ таково, что каждое $B(x)$ является множеством, то тип $\prod_{(x:A)} B(x)$ — множество. Предположим, что $f, g : \prod_{(x:A)} B(x)$ и $p, q : f = g$. По функциональной экстенсиональности имеем

$$p = \mathsf{funext}(x \mapsto \mathsf{happly}(p, x)) \qquad \text{и} \qquad q = \mathsf{funext}(x \mapsto \mathsf{happly}(q, x))\,.$$

Но, для любого$x : A$ имеет место

$$\mathsf{happly}(p, x) : f(x) = g(x) \qquad \text{и} \qquad \mathsf{happly}(q, x) : f(x) = g(x)\,,$$

так как $B(x)$ — множество, то $\mathsf{happly}(p, x) = \mathsf{happly}(q, x)$. Теперь, снова используя функциональную экстенсиональность, зависимые функции $(x \mapsto \mathsf{happly}(p, x))$ и $(x \mapsto \mathsf{happly}(q, x))$ равны, и, следовательно, (применяя $\mathsf{ap}_{\mathsf{funext}}$), так же равны p и q.

Дополнительные примеры см. в упражнениях 3.2 и 3.3. Более систематическое исследование подсистемы (категории) всех множеств в гомотопической теории типов приведено в главе 10.

Множества — это только первая ступенька на лестнице так называемых *гомотопических n-типов*. Следующая ступенька состоит из 1-*типов*, которые аналогичны 1-группоидам в теории категорий. Определяющее свойство множества (которое мы также можем назвать 0-*типом*) состоит в том, что оно не имеет нетривиальных путей. Аналогично, определяющее свойство 1-типа состоит в том, что у него нет нетривиальных путей между путями:

Определение 3.1.7. Тип A называется **1-типом**, если для всех $x, y : A$, $p, q : x = y$ и $r, s : p = q$ имеет место $r = s$.

Аналогично, можно определить 2-типы, 3-типы и т.д. Мы определим общее понятие n-типа индуктивно в главе 7 и изучим отношения между такими типами.

Однако на данный момент полезно иметь в виду два обстоятельства. Во-первых, уровни замкнуты вверх: если A является n-типом, то A является и $(n+1)$-типом.

Лемма 3.1.8. *Если A — множество (т.е. $\mathsf{isSet}(A)$ обитаем), то A есть 1-тип.*

Доказательство. Предположим $f : \mathsf{isSet}(A)$; тогда для любых $x, y : A$ и $p, q : x = y$ имеем $f(x, y, p, q) : p = q$. Зафиксируем x, y, p и определим $g : \prod_{(q:x=y)}(p = q)$ через $g(q) :\equiv f(x, y, p, q)$. Тогда для любого $r : q = q'$ имеем $\mathsf{apd}_g(r) : r_*(g(q)) = g(q')$. По лемме 2.11.2, следовательно, имеем $g(q) \cdot r = g(q')$.

В частности, предположим, что заданы x, y, p, q и $r, s : p = q$, как в определении 3.1.7, и определим g, как указано выше. Тогда $g(p) \cdot r = g(q)$, а также $g(p) \cdot s = g(q)$, следовательно, путем сокращения, получаем $r = s$. □

Во-вторых, это наслоение типов по уровню не является вырожденным, в том смысле, что не все типы являются множествами:

Пример 3.1.9. Универсум $\mathcal{U}$ не является множеством. Чтобы доказать это, достаточно предъявить тип A и путь $p : A = A$, который не равен refl_A. Возьмем $A = \mathbf{2}$ и пусть $f : A \to A$ определяется посредством $f(0_{\mathbf{2}}) :\equiv 1_{\mathbf{2}}$ и $f(1_{\mathbf{2}}) :\equiv 0_{\mathbf{2}}$. Тогда $f(f(x)) = x$ для всех x (путем простого анализа), поэтому f является эквивалентностью. Следовательно, по унивалентности, f порождает путь $p : A = A$.
Если p был бы равен refl_A, то (опять-таки по унивалентности) f будет равна тождественной функции из A. Но это означало бы, что $0_{\mathbf{2}} = 1_{\mathbf{2}}$, что противоречит замечанию 2.12.6.

В главах 6 и 8 мы покажем, что для любого n существуют типы, которые не являются n-типами.

Заметим, что A точно является 1-типом, когда для любых $x, y : A$ тип тождественности $x =_A y$ является множеством (таким образом, лемму 3.1.8 можно было бы эквивалентно воспринимать как утверждающую, что типы тождественности множества также являютя множествами). Это будет основой рекурсивного определения n-типов, которое мы дадим в главе 7.

Мы также можем расширить эту характеризацию «вниз» от множеств. То есть, тип A является множеством только тогда, когда для любых $x, y : A$ любые два элемента из $x =_A y$ равны. Так как множества эквивалентны 0-типам, естественно назвать тип (-1)-*типом*, если он имеет это последнее свойство (любые два его элемента равны). Такие типы можно рассматривать как *высказывания в узком смысле*, и их изучение — это то, что обычно называют «логикой»; оно будет занимать нас до конца этой главы.

3.2 Высказывания как типы?

До сих пор мы следовали простой философии «высказываний как типов», описанной в §1.11, согласно которой фразы естественного языка, такие как «существует $x : A$, такой, что $P(x)$» интерпретируются соответствующими типами, такими как $\sum_{(x:A)} P(x)$, причем доказательство высказывания рассматривается как суждение о каком-то конкретном элементе, обитающем в этом типе. Однако, мы также наблюдали некоторые ситуации, в которых «логика», возникающая в результате такого прочтения, кажется чуждой классическому математику. Например, в теореме 2.15.7 мы видели, что высказывание

$$\begin{aligned}&\text{«Если для всех } x : X \text{ существует } a : A(x) \text{ такое, что} P(x, a)\,, \text{ то существует}\\ &\text{функция } g : \textstyle\prod_{(x:A)} A(x) \text{ такая, что для всех } x : X \text{ верно } P(x, g(x))\text{»},\end{aligned} \tag{3.2.1}$$

которое выглядит как классическая *аксиома выбора*, соответствует этому пониманию. Это примечательная, и часто полезная, особенность логики высказываний как типов, но она также иллюстрирует, насколько она существенно отличается от классической интерпретации логики, при которой аксиома выбора является не логической истиной, а дополнительной «аксиомой».

С другой стороны, теперь мы можем также показать, что соответствующие высказывания, подобные классическим закону *двойного отрицания* и закону *исключения третьего*, несовместимы с аксиомой унивалентности.

Теорема 3.2.2. *Неверно, что для всех $A : \mathcal{U}$ имеет место $\neg(\neg A) \to A$.*

Доказательство. Напомним, что $\neg A \equiv (A \to \mathbf{0})$. Также мы интерпретируем «неверно, что ...» как оператор $\neg$. Таким образом, для доказательства этого высказывания достаточно взять значение $f : \prod_{(A:U)} (\neg\neg A \to A)$ и построить элемент из $\mathbf{0}$.

Идея доказательства состоит в том, что функция f, как и любая функция теории типов, является «непрерывной». В связи с унивалентностью это означает, что f естественна относительно эквивалентности типов. Из этого и автоэквивалентности с фиксированной точкой мы сможем извлечь противоречие.

Пусть $e : \mathbf{2} \simeq \mathbf{2}$ — эквивалентность, определенная посредством $e(1_\mathbf{2}) :\equiv 0_\mathbf{2}$ и $e(0_\mathbf{2}) :\equiv 1_\mathbf{2}$, как в примере 3.1.9. Пусть $p : \mathbf{2} = \mathbf{2}$ — путь, соответствующий e по унивалентности, т.е. $p :\equiv \mathsf{ua}(e)$. Тогда имеем $f(\mathbf{2}) : \neg\neg\mathbf{2} \to \mathbf{2}$ и

$$\mathsf{apd}_f(p) : \mathsf{transport}^{A \mapsto (\neg\neg A \to A)}(p, f(\mathbf{2})) = f(\mathbf{2})\,.$$

Следовательно, для любого $u : \neg\neg\mathbf{2}$ имеем

$$\mathsf{happly}(\mathsf{apd}_f(p), u) : \mathsf{transport}^{A \mapsto (\neg\neg A \to A)}(p, f(\mathbf{2}))(u) = f(\mathbf{2})(u)\,.$$

Теперь, по (2.9.4), транспортировка $f(\mathbf{2}) : \neg\neg\mathbf{2} \to \mathbf{2}$ вдоль p в семействе типов $A \mapsto (\neg\neg A \to A)$ равна функции, которая переносит свои аргументы вдоль p^{-1} в семействе типов $A \mapsto \neg\neg A$, применяет $f(\mathbf{2})$, затем переносит результат вдоль p в семействе типов $A \mapsto A$:

$$\mathsf{transport}^{A \mapsto (\neg\neg A \to A)}(p, f(\mathbf{2}))(u) = \mathsf{transport}^{A \mapsto A)}(p, f(\mathbf{2})(\mathsf{transport}^{A \mapsto \neg\neg A)}(p^{-1}, u)))\,.$$

Однако, любые две точки $u, v : \neg\neg 2$ равны по функциональной экстенсиональности, так как для любого $x : \neg 2$ имеем $u(x) : \mathbf{0}$ и, следовательно, можно получить любой вывод, в частности $u(x) = v(x)$. Таким образом, имеем $\mathsf{transport}^{A \mapsto \neg\neg A)}(p^{-1}, u) = u$, и поэтому из $\mathsf{happly}(\mathsf{apd}_f(p), u)$ получаем равенство

$$\mathsf{transport}^{A \mapsto A)}(p, f(\mathbf{2})(u)) = f(\mathbf{2})(u)\,.$$

Наконец, как обсуждалось в §2.10, транспортировка в семействе типов $A \mapsto A$ вдоль пути $p :\equiv \mathsf{ua}(e)$ эквивалентна применению эквивалентности e; таким образом, имеем

$$e(f(\mathbf{2})(u)) = f(\mathbf{2})(u)\,. \tag{3.2.3}$$

Однако мы также можем доказать, что

$$\prod_{x:\mathbf{2}} \neg(e(x) = x)\,. \tag{3.2.4}$$

Это следует из анализа случаев по x: оба случая непосредственно вытекают из определения e и того факта, что $0_{\mathbf{2}} \neq 1_{\mathbf{2}}$ (замечание 2.12.6). Таким образом, применяя (3.2.4) к $f(\mathbf{2})(u)$ и (3.2.3), получаем элемент из $\mathbf{0}$. □

Замечание 3.2.5. В частности, это означает, что здесь не может быть «оператора выбора» в стиле Гильберта, который выбирает элемент каждого непустого типа. Дело в том, что такой оператор не может быть естественным, а по аксиоме унивалентности все функции, действующие на типы, должны быть естественными относительно эквивалентностей.

Замечание 3.2.6. Тем не менее, есть случай, когда $\neg\neg\neg A \to \neg A$ для любого A; см. упражнение 1.11.

Следствие 3.2.7. *Неверно, что для всех $A : \mathcal{U}$ имеет место $A + (\neg A)$.*

Доказательство. Предположим, что имеется функция $g : \prod_{(A:\mathcal{U})}(A+(\neg A))$. Покажем, что тогда $\prod_{(A:\mathcal{U})}(\neg\neg A \to A)$, так что мы можем применить теорему 3.2.2. Таким образом, полагая, что $A : \mathcal{U}$ и $u : A$, мы хотим построить элемент A.

Теперь $g(A) : A + (\neg A)$, поэтому, анализируя этот случай, мы можем считать, что, либо $g(A) \equiv \mathsf{inl}(a)$ для некоторого $a : A$, или$g(A) \equiv \mathsf{inr}(w)$ для некоторого $w : \neg A$. В первом случае имеем $a : A$, а во втором случае — $u(w) : \mathbf{0}$ и поэтому мы можем получить все, что пожелаем (например, A). Таким образом, в обоих случаях мы имеем элемент A, что и требовалось. □

Таким образом, если мы хотим принять аксиому унивалентности (что, конечно же, мы и делаем), и по-прежнему оставляем себе возможность классического рассуждения (что также желательно), мы не можем использовать немодифицированный принцип «высказывания как типы» для интерпретации всех неформальных математических утверждений в теории типов, поскольку тогда закон исключения третьего будет ложным. Однако мы не хотим полностью отказаться от высказываний как типов из-за его многих хороших свойств (таких как простота, конструктивность и вычислимость). Теперь мы обсудим модификацию высказываний-как-типов, которые решают эти проблемы; в §3.10 мы вернемся к вопросу о том, какую логику когда использовать.

3.3 Простые высказывания

Мы видели, что логика высказываний как типов имеет как хорошие, так и плохие свойства. Проявление этих свойств имеет общую причину: когда типы рассматриваются как высказывания, они могут содержать больше информации, чем просто правдивость или ложность, и все «логические» конструкции на них не должны нарушать эту дополнительную информацию.

Это означает, что мы могли бы получить более традиционную логику, ограничив внимание теми типами, которые *не* содержат информации, большей чем истинностные значения, и только относительно них рассматривая логические высказывания.

Такой тип A будет «истинным», если он обитаем, и «ложным», если его обитание дает противоречие (то есть, если: $\neg A \equiv (A \to \mathbf{0})$ обитаемо). То, чего мы хотим избежать, чтобы получить более традиционную логику, рассматривает в качестве логических высказываний те типы, для которых предоставление элемента на них дает больше информации, чем просто знание того, что этот тип обитаем. Например, если нам задан элемент из $\mathbf{2}$, мы получаем больше информации, чем тот факт, что $\mathbf{2}$ содержит некоторый элемент. В самом деле, мы получаем ровно на *один бит* больше информации: мы знаем, какой элемент из $\mathbf{2}$ мы получили. Напротив, если нам задан элемент из $\mathbf{1}$, мы получаем больше информации, чем тот факт, что $\mathbf{1}$ содержит элемент, поскольку любые два элемента из $\mathbf{1}$ равны друг другу. Это подсказывает следующее определение.

Определение 3.3.1. Тип P является **простым высказыванием**, если для любых $x, y : P$ имеет место $x = y$.

Заметим, что, поскольку мы все еще занимаемся математикой в теории типов, это является определением в теории типов, что означает, что оно — тип или, вернее, семейство типов. В частности, для любого $P : \mathcal{U}$ тип $\mathsf{isProp}(P)$ определяется как

$$\mathsf{isProp}(P) :\equiv \prod_{x,y:P} (x = y).$$

Таким образом, заявление «P — простое высказывание» означает предъявление обитателя $\mathsf{isProp}(P)$, который является зависимой функцией, соединяющей любые два элемента P на пути. Из непрерывности/естественности этой функции следует, что не только любые два элемента из P равны, но и что P не содержит высшей гомотопии.

Лемма 3.3.2. *Если P — простое высказывание и $x_0 : P$, то $P \simeq \mathbf{1}$.*

Доказательство. Определим $f : P \to \mathbf{1}$ как $f(x) :\equiv \star$ и $g : \mathbf{1} \to P$ как $g(u) :\equiv x_0$. Заключение данной леммы следует из следующей леммы и наблюдения, что $\mathbf{1}$ — простое высказывание по теореме 2.8.1. □

Лемма 3.3.3. *Если P и Q — простые высказывания такие, что $P \to Q$ и $Q \to P$, то $P \simeq Q$.*

Доказательство. Предположим, что заданы $f : P \to Q$ и $g : Q \to P$. Тогда для любого $x : P$ имеем $g(f(x)) = x$, поскольку P — простое высказывание. Аналогично, для любого $y : Q$ имеем $f(g(y)) = y$, поскольку Q — простое высказывание; так что f и g квази-обратны. □

То есть, как было обещано в §1.11, если два простых высказывания логически эквивалентны, то они эквивалентны.

В гомотопической теории пространство, гомотопически эквивалентное $\mathbf{1}$, называется *стягиваемым*. Таким образом, любое простое высказывание, которое обитаемо, является стягиваемым (см. также §3.11). С другой стороны, необитаемый тип $\mathbf{0}$ также (несомненно) является простым высказыванием. В классической математике, по крайней мере, это единственные две возможности.

Простые высказывания также называются *подтерминальными объектами* (если мыслить категорно), *подсинглетонами* (если оперировать языком теории множеств) или *h-высказываниями*.

Обсуждение в §3.1 подсказывает, что мы можем также называть их (-1)-*типами*; мы вернемся к этому в главе 7. Прилагательное «простой» подчеркивает, что хотя любой тип можно рассматривать как высказывание (которое мы доказываем, предъявляя его обитателя), тип, являющийся простым высказыванием, нельзя рассматривать претендующим на *большее*: нет дополнительной информации, содержащейся в свидетельстве его истинности.

Заметим, что тип A является множеством, тогда и только тогда, когда для любых $x, y : A$ тип тождественности $x =_A y$ является простым высказыванием. С другой стороны, копируя и упрощая доказательство леммы 3.1.8, имеем:

Лемма 3.3.4. *Каждое простое высказывание является множеством.*

Доказательство. Предположим, что $f : \mathsf{isProp}(A)$; таким образом, для всех $x, y : A$ имеем $f(x, y) : x = y$. Зафиксируем $x : A$ и определим $g(y) :\equiv f(x, y)$. Тогда для всех $y, z : A$ и $p : y = z$ имеет место $\mathsf{apd}_g(p) : p_*(g(y)) = g(z)$. Следовательно, по лемме 2.11.2, имеем $g(y) \cdot p = g(z)$, что означает $p = g(y)^{-1} \cdot g(z)$. Таким образом, для любых $p, q : x = y$ получаем $p = g(x)^{-1} \cdot g(y) = q$. $\square$

В частности, это означает:

Лемма 3.3.5. *Для любого типа A, типы* $\mathsf{isProp}(A)$ *и* $\mathsf{isSet}(A)$ *являются простыми высказываниями.*

Доказательство. Предположим, что $f, g : \mathsf{isProp}(A)$. В связи с функциональной экстенсиональностью, чтобы показать, что $f = g$, достаточно показать, что $f(x, y) = g(x, y)$ для любых $x, y : A$. Но $f(x, y)$ и $g(x, y)$ — пути в A и, следовательно, они равны, потому что либо на f, либо на g, A является простым высказыванием и, следовательно, по лемме 3.3.4, является множеством. Аналогично, предположим, что $f, g : \mathsf{isSet}(A)$, а это значит, что для всех $a, b : A$ и $p, q : a = b$ имеем $f(a, b, p, q) : p = q$ и $g(a, b, p, q) : p = q$. Но к тому же, так как A является множеством (либо по f, либо по g), и, следовательно, 1-типом, то $f(a, b, p, q) = g(a, b, p, q)$; следовательно, $f = g$ по функциональной экстенсиональности. $\square$

Пока мы видели еще один пример: условие (iii) в §2.4 утверждает, что для любой функции f тип $\mathsf{isequiv}(f)$ должен быть простым высказыванием.

3.4 Классическая и интуиционистская логики

Имея понятие простого высказывания, мы можем теперь дать подходящую формулировку **закона исключения третьего** в гомотопической теории типов:

$$\mathsf{LEM} :\equiv \prod_{A:\mathcal{U}} \left(\mathsf{isProp}(A) \to (A + \neg A)\right) . \tag{3.4.1}$$

Аналогично, **закон двойного отрицания** есть

$$\prod_{A:\mathcal{U}} \left(\mathsf{isProp}(A) \to (\neg\neg A \to A)\right) . \tag{3.4.2}$$

Эти два закона также легко воспринимаются как эквивалентные друг другу — см. упражнение 3.18 — поэтому далее мы будем говорить только о LEM.

Приведенная формулировка LEM позволяет избежать «парадоксов» теоремы 3.2.2 и следствия 3.2.7, поскольку $\mathbf{2}$ не является простым высказыванием. Чтобы отличить этот закон от более общей формулировки высказываний как типов, мы переименовываем последнее:

$$\mathsf{LEM}_\infty :\equiv \prod_{A:\mathcal{U}}(A + \neg A)\,.$$

Для акцентирования, правильная версия (3.4.1) может быть обозначена как LEM_{-1}; см. также упражнение 7.7. Хотя LEM не является следствием базовой теории типов, описанной в главе 1, ее можно последовательно рассматривать как аксиому (в отличие от его ∞-двойника). Например, мы будем использовать ее в §10.4.

Однако, что может быть неожиданным, как долго мы можем обходиться без использования LEM. Довольно часто простая переформулировка определения или теоремы позволяет избежать использования исключения третьего. Это приводит к более элегантным и более общим доказательствам, хотя немного отвлекает и требует определенных усилий. Мы обсудили некоторые преимущества этого во введении.

Например, в классической математике двойные отрицания часто используются без необходимости. Очень простой пример — это общее предположение, что множество A является «непустым», что буквально означает, что это *не* тот случай, когда A *не* содержит элементов. Почти всегда, на самом деле, подразумевается позитивное суждение, что A *содержит* хотя бы один элемент, и, удалив двойное отрицание, мы делаем высказывание менее зависимым от LEM. Напомним, что мы говорим, что тип A *населен* (или, обитаем), когда мы предъявляем A в качестве высказывания (т.е. мы строим элемент, обычно неназванный, из A). Таким образом, часто при переводе классического доказательства в конструктивную логику мы заменяем слово «непустой» на «населенный» (или, «обитаемый», хотя иногда мы вместо этого заменяем его на «просто обитаемым», см. §3.7).

Точно так же в классической математике нередко приходится заменять необходимые доказательства сведением к противоречию. Разумеется, классическая форма доказательства от противоречия соответствует закону двойного отрицания: мы предполагаем $\neg A$ и получаем противоречие, выводя $\neg\neg A$ и, таким образом, двойным отрицанием получаем A. Однако часто вывод противоречия из $\neg A$ можно слегка перефразировать, чтобы дать прямое доказательство A, избегая необходимости в LEM.

Важно также отметить, что если целью является доказательство *отрицания*, то «доказательство от противного» не связано с LEM. Действительно, поскольку $\neg A$ по определению является типом $A \to \mathbf{0}$, по определению доказать $\neg A$ — это доказать противоречие ($\mathbf{0}$) в предположении A. Аналогично, закон двойного отрицания имеет место для негативных высказываний: $\neg\neg\neg A \to \neg A$. Приобретая опыт, вы учитесь более тщательно проводить различие между негативными и не негативными высказываниями и замечать, когда используется LEM, а когда — нет.

Таким образом, вопреки тому, как это может выглядеть внешне, «конструктивно» математика обычно не предполагает отказа от важных теорем, а скорее занимается поиском лучшего способа изложения определений, чтобы сделать важные теоремы конструктивно доказуемыми. То есть, мы можем свободно использовать LEM при начальном исследовании предмета, но как только этот предмет будет лучше понят, мы можем надеяться уточнить его определения и доказательства, чтобы избежать использования этой аксиомы. Подобное наблюдение еще более выражено в *гомотопической* теории типов, где мощные инструменты унивалентности и высших индуктивных типов позволяют нам конструктивно браться за решение многих проблем, которые традиционно потребуют классических рассуждений. Мы увидим несколько примеров этого в части II.

Стоит также отметить, что даже в конструктивной математике закон исключения третьего может выполняться для *некоторых* высказываний. Название, традиционно даваемое таким высказываниям, — *разрешимые*.

Определение 3.4.3.

(i) Тип A называется **разрешимым**, если $A + \neg A$.

(ii) Аналогично, семейство типов $B : A \to \mathcal{U}$ **разрешимо**, если $\prod_{(a:A)}(B(a) + \neg B(a))$.

(iii) В частности, A имеет **разрешимое равенство**, если $\prod_{(a,b:A)}((a = b) + \neg(a = b))$.

Таким образом, LEM — это именно заявление, что все простые высказывания разрешимы, и, следовательно, это относится и ко всем семействам простых высказываний. В частности, LEM означает, что все множества (в смысле §3.1) имеют разрешимое равенство. Обладание разрешимым равенством в этом смысле является очень сильным; см. теорему 7.2.5.

3.5 Подмножества и пропозициональное изменение размера

В качестве еще одного примера полезности простых высказываний мы обсудим подмножества (и более обще, подтипы). Предположим, что $P : A \to \mathcal{U}$ — семейство типов, причем каждый тип $P(x)$ рассматривается как высказывание. Тогда сам P является *предикатом* на A или *свойством* элементов из A.

В теории множеств всякий раз, когда имеется предикат P на множестве A, можно составить подмножество $\{f \in A \mid P(x)\}$. Как уже упоминалось в §1.11, его очевидным аналогом в теории типов является $\sum$-тип $\sum_{(x:A)} P(x)$. Обитатель из $\sum_{(x:A)} P(x)$ — это, конечно, пара (x, p), где $x : A$ и p — доказательство для $P(x)$. Однако для общего P элемент $a : A$ может порождать более одного отдельного элемента из $\sum_{(x:A)} P(x)$, если высказывание $P(a)$ имеет более одного отдельного доказательства. Это противоречит обычной интуиции подмножества. Но если P — простое высказывание, то этого не может произойти.

Лемма 3.5.1. *Предположим, что $P : A \to \mathcal{U}$ — семейство типов такое, что $P(x)$ является простым высказыванием для всех $x : A$. Если $u, v : \sum_{(x:A)} P(x)$ такое, что $\mathsf{pr}_1(u) = \mathsf{pr}_1(v)$, то $u = v$.*

Доказательство. Предположим $p : \mathsf{pr}_1(u) = \mathsf{pr}_1(v)$. По теореме 2.7.2, чтобы показать $u = v$, достаточно показать, что $p_*(\mathsf{pr}_2(u)) = \mathsf{pr}_2(v)$. Но $p_*(\mathsf{pr}_2(u))$ и $\mathsf{pr}_2(v)$ являются элементами из $P(\mathsf{pr}_1(v))$, которое есть простое высказывание; следовательно, они равны. □

Например, напомним, что в §2.4 мы определили

$$(A \simeq B) :\equiv \sum_{f:A\to B} \mathsf{isequiv}(f)\,,$$

где каждый тип $\mathsf{isequiv}(f)$ должен был быть простым высказыванием. Отсюда следует, что если две эквивалентности имеют равные основные функции, то они равны как эквивалентности.

В дальнейшем, если $P : A \to \mathcal{U}$ — семейство простых высказываний (т.е. каждое $P(x)$ — простое высказывание), мы можем писать

$$\{x : A \mid P(x)\}, \tag{3.5.2}$$

в качестве альтернативной записи для $\sum_{(x:A)} P(x)$ (нет никаких технических причин не использовать эту нотацию для произвольного P, но такое ее использование может внести путаницу из-за непреднамеренных сопутствующих обозначений). Если A — множество, то мы называем (3.5.2) **подмножеством** A; для A произвольной природы мы можем назвать его **подтипом**. Мы можем также ссылаться на P как на *подмножество* или *подтип* A; это на самом деле более правильно, так как тип (3.5.2) вне контекста не помнит своего отношения к A.

Для таких P и $a : A$ мы можем записать $a \in P$ или $a \in \{x : A \mid P(x)\}$ для обозначения простого высказывания $P(a)$. Если оно выполнено, мы можем сказать, что a является **членом** P. Аналогично, если $\{x : A \mid Q(x)\}$ — другое подмножество A, то мы говорим, что P **содержится** в Q и записываем $P \subseteq Q$, если при этом имеет место $\prod_{(x:A)}(P(x) \to Q(x))$.

В качестве дополнительных примеров подтипов мы можем определить «подуниверсумы» множеств и простых высказываний в универсуме $\mathcal{U}$:

$$\mathsf{Set}_{\mathcal{U}} :\equiv \{A : U \mid \mathsf{isSet}(A)\},$$
$$\mathsf{Prop}_{\mathcal{U}} :\equiv \{A : U \mid \mathsf{isProp}(A)\}.$$

Элемент $\mathsf{Set}_{\mathcal{U}}$ является типом $A : \mathcal{U}$ вместе со свидетельством $s : \mathsf{isSet}(A)$, аналогично для $\mathsf{Prop}_{\mathcal{U}}$. Из леммы 3.5.1 следует, что $(A, s) =_{\mathsf{Set}_{\mathcal{U}}} (B, t)$ эквивалентно $A =_{\mathcal{U}} B$ (и, следовательно, эквивалентно $A \simeq B$). Таким образом, мы будем часто злоупотреблять нотацией и писать просто $A : \mathsf{Set}_{\mathcal{U}}$ вместо $(A, s) : \mathsf{Set}_{\mathcal{U}}$. Мы также можем отбросить индекс $\mathcal{U}$, если нет необходимости указывать рассматриваемый универсум.

Напомним, что для любых универсумов $\mathcal{U}_i$ и $\mathcal{U}_{i+1}$, если $A : \mathcal{U}_i$, то также $A : \mathcal{U}_{i+1}$. Таким образом, для любого $(A, s) : \mathsf{Set}_{\mathcal{U}_i}$ верно $(A, s) : \mathsf{Set}_{\mathcal{U}_{i+1}}$, и аналогично для $\mathsf{Prop}_{\mathcal{U}_i}$, давая в результате естественные отображения

$$\mathsf{Set}_{\mathcal{U}_i} \to \mathsf{Set}_{\mathcal{U}_{i+1}}, \tag{3.5.3}$$

$$\mathsf{Prop}_{\mathcal{U}_i} \to \mathsf{Prop}_{\mathcal{U}_{i+1}}. \tag{3.5.4}$$

Отображение (3.5.3) не может быть эквивалентностью, так как тогда мы могли бы воспроизвести парадоксы самоотносимости, знакомые из канторовской теории множеств. Однако, хотя (3.5.4) не является автоматически эквивалентностью в представленной теории типов, можно предположить, что это так. То есть, мы можем рассмотреть возможность добавления к теории типов следующей аксиомы.

Аксиома 3.5.5 (Пропозициональное изменение размера)**.** *Отображение* $\mathsf{Prop}_{\mathcal{U}_i} \to \mathsf{Prop}_{\mathcal{U}_{i+1}}$ *является эквивалентностью.*

Мы ссылаемся на эту аксиому как на **пропозициональное изменение размера**, так как это означает, что любое простое высказывание в универсуме $\mathcal{U}_{i+1}$ может быть «изменено» до эквивалентного в меньшем универсуме $\mathcal{U}_i$. Это получается автоматически, если $\mathcal{U}_{i+1}$ удовлетворяет LEM (см. упражнение 3.10). Мы не будем считать эту аксиому всеобщей, хотя в некоторых местах мы будем использовать ее как явную гипотезу. Это форма *непредикативности* для простых высказываний, и, избегая ее использования, говорят, что теория типов остается *предикативной*.

На практике то, что нам наиболее часто требуется, является немного отличающейся формулировкой: рассматриваемый универсум $\mathcal{U}$ содержит тип, который «классифицирует все простые высказывания». Другими словами, нам нужен тип $\Omega : \mathcal{U}$ вместе с Ω-индексированным семейством простых высказываний, которое содержит каждое простое высказывание вплоть до эквивалентности. Это утверждение следует из пропозиционального изменения размера, как указано выше, если $\mathcal{U}$ не является наименьшим универсумом $\mathcal{U}_0$, так как тогда мы можем определить $\Omega :\equiv \mathsf{Prop}_{\mathcal{U}_0}$.

Одно из использований непредикативности — определение степенных множеств. Естественно определить **степенное множество** множества A как $A \to \mathsf{Prop}_{\mathcal{U}}$; но в отсутствие непредикативности это определение зависит (даже вплоть до эквивалентности) от выбора универсума $\mathcal{U}$. А с пропозициональным изменением размера мы можем определить степенное множество таким образом:

$$\mathcal{P}(A) :\equiv (A \to \Omega)\,,$$

которое тогда не зависит от $\mathcal{U}$. См. также §10.1.4.

3.6 Логика простых высказываний

Мы упоминали в §1.1, что в отличие от теории типов, которая имеет только одно основное понятие (типы), теоретико-множественные основания содержат два основных понятия: множества и высказывания. Поэтому классический математик привык манипулировать этими двумя видами объектов по отдельности.

В теории типов можно восстановить подобную дихотомию, причем роль теоретико-множественных высказываний играют типы (и семейства типов), которые являются *простыми* высказываниями. Во многих случаях логические связки и кванторы могут быть представлены в этой логике путем простого ограничения соответствующего конструктора типов простых высказываний. Конечно, это требует знания того, что конструктор типов при этом сохраняет простые высказывания.

Пример 3.6.1. Если A и B — простые высказывания, то и $A \times B$ является таковым. Это легко показать, используя характеризацию путей в произведениях, как в примере 3.1.5, но проще. Таким образом, связка «и» сохраняет простые высказывания.

Пример 3.6.2. Если A — любой тип и $B : A \to \mathcal{U}$ таково, что для всех $x : A$ тип $B(x)$ является простым высказыванием, то и $\prod_{(x:A)} B(x)$ — простое высказывание. Доказательство такое же, как и в примере 3.1.6, но проще: для $f, g : \prod_{(x:A)} B(x)$, для любого $x : A$ имеем $f(x) = g(x)$, так как $B(x)$ — простое высказывание. Но тогда, по функциональной экстенсиональности, получаем $f = g$.

В частности, если B — простое высказывание, то $A \to B$ также является простым высказыванием, независимо от того, что из себя представляет A. Еще более конкретно, поскольку $\mathbf{0}$ — простое высказывание, таково же и $\neg A \equiv (A \to \mathbf{0})$. Таким образом, связки «следует» и «не», как и квантор «для всех», сохраняют простые высказывания.

С другой стороны, некоторые конструкторы типов не сохраняют простые высказывания. Даже если и A и B являются простыми высказываниями, $A + B$ таковым вообще-то не будет. Например, $\mathbf{1}$ — простое высказывание, но $\mathbf{2} = \mathbf{1} + \mathbf{1}$ — нет. Логически говоря, $A + B$ является «чисто конструктивным» видом связки «или»: свидетельство этого содержит дополнительную информацию о том, какой из дизъюнктов (A или B) истинен. Иногда это очень полезно, но если нам нужен более классический вид «или», который сохраняет простые высказывания, нам нужен способ «усечь» этот тип в простое высказывание, путем потери этой дополнительной информации.

Эта же проблема возникает с $\sum$-типом $\sum_{(x:A)} P(x)$. Это чисто конструктивная интерпретация «существует такое $x : A$, что $P(x)$», которое помнит свидетельство x, и, следовательно, обычно не является простым высказыванием, даже если каждый тип $P(x)$ таков (напомним, что в §3.5 мы заметили, что $\sum_{(x:A)} P(x)$ также можно рассматривать как «подмножество $x : A$ таких, что $P(x)$»).

3.7 Пропозициональное усечение

Пропозициональное усечение, также называемое (-1)-*усечением*, *скобочным типом* или *типом сквоша*, является дополнительным типом, который «сжимает» или «обрезает» тип до простого высказывания, забывая всю информацию, содержащуюся в обитателях этого типа, кроме их существования.

Точнее, для любого типа A существует тип $\|A\|$. Он имеет два конструктора:

- Для любого $a : A$ имеет место $|a| : \|A\|$.
- Для любых $x, y : \|A\|$ имеет место $x = y$.

Первый конструктор означает, что если A обитаем, то обитаем и $\|A\|$. Второй конструктор гарантирует, что $\|A\|$ — простое высказывание; обычно мы оставляем свидетельство этого факта безымянным.

Принцип рекурсии для $\|A\|$ утверждает, что:

- Если B — простое высказывание и $f : A \to B$, то существует индуцированная функция $g : \|A\| \to B$ такая, что $g(|a|) \equiv f(a)$ для всех $a : A$.

Другими словами, любое простое высказывание, которое следует из (обитаемости) A, уже следует из $\|A\|$. Таким образом, $\|A\|$, как простое высказывание, не содержит больше информации, чем факта обитаемости A (существует также принцип индукции для $\|A\|$, который не является особенно полезным, см. упражнение 3.17).

В упражнениях 3.14 и 3.15 и §6.9 мы опишем некоторые способы построения $\|A\|$ в терминах более общих понятий. На данный момент мы просто принимаем это как дополнительное правило наряду с тем, что описано в главе 1.

С пропозициональным усечением мы можем расширить «логику простых высказываний», чтобы охватить дизъюнкцию и квантор существования. В частности, $\|A + B\|$ — простая пропозициональная версия «A или B», которая не «запоминает» информацию о том, какой дизъюнкт истинен.

Принцип рекурсии для усечения подразумевает, что мы все еще можем провести анализ случая на $\|A + B\|$ при попытке доказать простое высказывание. То есть допустим, что у нас есть предположение $u : \|A + B\|$, и мы пытаемся доказать простое высказывание Q. Другими словами, мы пытаемся определить элемент из $\|A + B\| \to Q$. Так как Q — простое высказывание, то по принципу рекурсии для пропозиционального усечения достаточно построить функцию $A + B \to Q$. Но теперь мы можем использовать анализ случая на $A + B$.

Аналогично, для семейства типов $P : A \to \mathcal{U}$ можно рассматривать $\left\|\sum_{(x:A)} P(x)\right\|$, что является простым пропозициональным вариантом «существует такое $x : A$, что P(x)». Что касается дизъюнкции, то, объединяя принципы индукции усечения и $\sum$-типы, если у нас есть предположение типа $\left\|\sum_{(x:A)} P(x)\right\|$, мы можем ввести новые предположения $x : A$ и $y : P(x)$ при попытке доказать простое высказывание. Другими словами, если мы знаем, что существует некоторое $x : A$ такое, что верно $P(x)$, но не имеем такого конкретного x в наличии, то мы можем использовать произвольный x, пока не попытаемся построить что-либо, что может зависеть от конкретного значения этого x. Требуемый кодомен в качестве простого высказывания выражает эту независимость результата от свидетельства, поскольку все возможные обитатели такого типа должны быть равны.

Для целей математики на уровне множества в главах 10 и 11, где мы будем иметь дело главным образом с множествами и простыми высказываниями, удобно использовать традиционную логическую нотацию для обозначения только «пропозиционально усеченной логики».

Определение 3.7.1. Мы определяем **традиционную логическую нотацию**, используя усечение, следующим образом, где P и Q обозначают простые высказывания (или их семейства):

$$
\begin{aligned}
\top &:\equiv \mathbf{1}\\
\bot &:\equiv \mathbf{0}\\
P \wedge Q &:\equiv P \times Q\\
P \Rightarrow Q &:\equiv P \to Q\\
P \Leftrightarrow Q &:\equiv P = Q\\
\neg P &:\equiv \mathbf{0}\\
P \vee Q &:\equiv \|P + Q\|\\
\forall (x : A).P(x) &:\equiv \textstyle\prod_{(x:A)} P(x)\\
\exists (x : A).P(x) &:\equiv \left\|\textstyle\sum_{(x:A)} P(x)\right\|
\end{aligned}
$$

Обозначения $\wedge$ и $\vee$ также используются в гомотопической теории для скрещенного произведения и клина остроконечных пространств, которые мы введем в главе 6. Технически это создает возможность для противоречия, но путаницы обычно не возникает.

Аналогично, при обсуждении подмножеств, как в §3.5, мы можем использовать традиционную нотацию для пересечений, объединений и дополнений:

$$
\begin{aligned}
\{x : A \mid P(x)\} \cap \{x : A \mid Q(x)\} &:\equiv \{x : A \mid P(x) \wedge Q(x)\}\ ,\\
\{x : A \mid P(x)\} \cup \{x : A \mid Q(x)\} &:\equiv \{x : A \mid P(x) \vee Q(x)\}\ ,\\
A \setminus \{x : A \mid P(x)\} &:\equiv \{x : A \mid \neg P(x)\}\ .
\end{aligned}
$$

Конечно, в отсутствие LEM последнее не является «дополнением» в обычном смысле: у нас может не быть $B \cup (A \setminus B) = A$ для любого подмножества B из A.

3.8 Аксиома выбора

Теперь мы можем собственно сформулировать аксиому выбора в гомотопической теории типов. Предположим, что заданы тип X и семейства типов

$$A : X \to \mathcal{U} \qquad \text{и} \qquad P : \prod_{x:X} A(x) \to \mathcal{U}\,,$$

и, кроме того, что

- X — множество,
- $A(x)$ — множество для любого $x : X$ и
- $P(x, a)$ — простое высказывание для всех $x : X$ и $a : A(x)$.

Аксиома выбора AC утверждает, что в этих предположениях,

$$\left(\prod_{x:X}\left\|\sum_{a:A(x)} P(x,a)\right\|\right) \to \left\|\sum_{(g:\prod_{(x:X)} A(x))}\ \prod_{(x:X)} P(x, g(x))\right\| . \tag{3.8.1}$$

Конечно, это прямой перевод (3.2.1), где мы толкуем «существует $x : A$ такое, что $B(x)$» как $\left\|\sum_{(x:A)} B(x)\right\|$, поэтому мы могли бы написать приведенное выражение в привычной логической нотации, в виде

$$\Big(\forall(x : X).\exists(a : A(x).P(x,a)\Big) \Rightarrow \Big(\exists(g : \textstyle\prod_{(x:X)} A(x)).\forall(x : X).P(x, g(x))\Big) .$$

В частности, обратите внимание, что пропозициональное усечение появляется дважды. Усечение в области означает, что мы предполагаем, что для каждого x существует некоторое $a : A(x)$ такое, что $P(x, a)$, но эти значения не выбраны или не специфицированы каким-либо известным способом. Усечение в кообласти означает, что мы заключаем, что существует некоторая функция g, но эта функция не определена или не специфицирована каким-либо известным способом.

Действительно, по теореме 2.15.7 эта аксиома также может быть выражена в более простой форме.

Лемма 3.8.2. *Аксиома выбора* (3.8.1) *эквивалентна формулировке, что для любого множества X и любого $Y : X \to \mathcal{U}$ такого, что каждое $Y(x)$ является множеством, имеет место*

$$\left(\prod_{x:X}\|Y(x\|\right) \to \left\|\prod_{x:X} Y(x)\right\| . \tag{3.8.3}$$

Это соответствует известной эквивалентной форме классической аксиомы выбора, а именно: «декартово произведение семейства непустых множеств непусто».

Доказательство. По теореме 2.15.7 кообласть из (3.8.1) эквивалентна

$$\left\|\prod_{(x:X)}\ \sum_{(a:A(x))} P(x,a)\right\| .$$

Таким образом, (3.8.1) эквивалентно экземпляру (3.8.3), где $Y(x) :\equiv \sum_{(a:A(x))} P(x, a)$. Обратно, (3.8.3) эквивалентно экземпляру (3.8.1), где $A(x) :\equiv Y(x)$ и $P(x, a) :\equiv \mathbf{1}$. Таким образом, они логически эквивалентны. Так как оба являются простыми высказываниями, то по лемме 3.3.3 они эквивалентны. □

Как и в случае LEM, эквивалентные формы (3.8.1) и (3.8.3) не являются следствием нашей базисной теории типов, но их можно последовательно рассматривать как аксиомы.

Замечание 3.8.4. Легко показать, что правая часть (3.8.3) всегда влечет за собой левую. Так как оба являются простыми высказываниями, то по лемме 3.3.3 аксиома выбора также эквивалентна заданию эквивалентности

$$\left(\prod_{x:X} \|Y(x)\|\right) \simeq \left\|\prod_{x:X} Y(x)\right\| .$$

Это иллюстрирует общую ошибку: хотя типы зависимых функций сохраняют простые высказывания (пример 3.6.2), они не коммутативны с усечением: $\left\|\prod_{(x:A)} P(x)\right\|$ обычно не эквивалентен $\prod_{(x:A)} \|P(x)\|$. Аксиома выбора, если мы ее принимаем, говорит, что она верна *для множеств*; как мы увидим ниже, в общем случае она не работает.

Требование использования в аксиоме выбора типов, которые являются множествами, может быть в некоторой степени ослаблено. Например, мы можем допустить, чтобы A и P в (3.8.1) или Y в (3.8.3) были произвольными семействами типов; это приводит к кажущемуся более сильному утверждению, которое также согласованно. Мы можем также заменить пропозициональное усечение на более общие n-усечения, которые будут рассмотрены в главе 7, получив спектр аксиом AC_n, интерполированными между (3.8.1), которую мы называем просто AC (или, акцентируя, AC_{-1}) и теоремой 2.15.7, которую мы будем обозначать AC_∞. См. также упражнения 7.8 и 7.10. Однако заметим, что мы не можем ослабить требование к X быть множеством.

Лемма 3.8.5. *Существуют тип X и семейство $Y : X \to \mathcal{U}$ такие, что каждое $Y(x)$ является множеством, но таким, что* (3.8.3) *не выполняется.*

Доказательство. Определим $X :\equiv \sum_{(A:\mathcal{U})} \|\mathbf{2} = A\|$ и пусть $x_0 :\equiv (2, |\mathsf{refl}_{\mathbf{2}}|) : X$. Тогда, отождествляя пути в $\sum$-типах, учитывая факт, что $\|A = \mathbf{2}\|$ является простым высказыванием и унивалентностью, для любых $(A,p),(B,q) : X$ имеем $((A,p) =_X (B,q)) \simeq (A \simeq B)$. В частности, $(x_0 =_X x_0) \simeq (\mathbf{2} \simeq \mathbf{2})$, так что, как и в примере 3.1.9, X не является множеством.

С другой стороны, если $(A,p) : X$, то A — множество; это следует индукцией по усечению для $p : \|\mathbf{2} = A\|$ и тем, что $\mathbf{2}$ — множество. Так как $A \simeq B$ является множеством, когда A и B — множества, то $x_1 =_X x_2$ является множеством для любых $x_1, x_2 : X$, т.е. X является 1-типом. В частности, если определить $Y : X \to \mathcal{U}$ посредством $Y(x) :\equiv (x_0 = x)$, то каждое $Y(x)$ будет множеством.

Теперь, по определению, для любого $(A,p) : X$ имеем $\|\mathbf{2} = A\|$ и, следовательно, $\|x_0 = (A,p)\|$. Таким образом, мы имеем $\prod_{(x:X)} \|Y(x)\|$. Если для этих X и Y выполнено (3.8.3), то также было бы быть и $\left\|\prod_{(x:X)} Y(x)\right\|$. Поскольку мы пытаемся получить противоречие ($\mathbf{0}$), которое является простым высказыванием, можно считать $\prod_{(x:X)} Y(x)$, т.е. что $\prod_{(x:X)} (x_0 = x)$. Но это означает, что X — простое высказывание и, следовательно, множество, что является противоречием.

□

3.9 Принцип единственности выбора

Следующее наблюдение тривиально, но очень полезно.

Лемма 3.9.1. *Если P — простое высказывание, то $P \simeq \|P\|$.*

Доказательство. Конечно же, по определению, $P \to \|P\|$. А поскольку P — простое высказывание, двойственное свойство для $\|P\|$, примененное к $\mathsf{id}_P : P \to P$, дает $\|P\| \to P$. Эти функции являются квази-обратными по лемме 3.3.3. □

Среди важных следствий — отметим следующее.

Следствие 3.9.2 (Принцип единственности выбора)**.** *Предположим, что семейство типов $P : A \to \mathcal{U}$ таково, что*

(i) *для любого x тип $P(x)$ является простым высказыванием и*

(ii) *для любого x имеет место $\|P\|$.*

Тогда $\prod_{(x:A)} P(x)$.

Доказательство. Непосредственно следует из двух предположений и предыдущей леммы. □

Следствие также инкапсулирует очень полезный метод рассуждения. А именно, предположим, что мы знаем, что $\|A\|$, и хотим использовать это для построения элемента другого типа B. Мы хотели бы использовать элемент из A в нашей конструкции элемента из B, но это допускается, только если B — простое высказывание, так что мы можем применить принцип индукции для пропозиционального усечения $\|A\|$; самое большее, что мы надеемся сделать вообще, это предъявить $\|B\|$. Вместо этого мы можем расширить B дополнительными данными, которые однозначно характеризуют объект, который мы хотим построить. В частности, мы определяем предикат $Q : B \to \mathcal{U}$ такой, что $\sum_{(x:B)} Q(x)$ является простым высказыванием. Тогда из элемента из A строим элемент $b : B$ такой, что $Q(b)$, поэтому из $\|A\|$ мы можем построить $\left\|\sum_{(x:B)} Q(x)\right\|$ и потому, что $\left\|\sum_{(x:B)} Q(x)\right\|$ эквивалентно $\sum_{(x:B)} Q(x)$, из него может быть проектирован элемент из B. Пример приведен в упражнении 3.19.

Аналогичная проблема возникает в теоретико-множественной математике, хотя она проявляется несколько иначе. Если мы пытаемся определить функцию $f : A \to B$, и в зависимости от элемента $a : A$ доказать просто существование некоторого $b : B$, на этом работа еще не заканчивается, потому что еще нам необходимо фактически определить элемент из B. Одним из вариантов является, конечно, доработка аргумента до уникального существования $b : B$, как это было в теории типов. Но в теории множеств проблему часто можно разрешить проще, применяя аксиому выбора, которая помогает выбрать для нас необходимые элементы. Однако, в гомотопической теории типов, несмотря на желание избежать выбора, доступные формы выбора просто неприменимы, поскольку они требуют, чтобы выбранная область была множеством. Таким образом, если A не является множеством (например, возможно, универсумом $\mathcal{U}$), то не существует согласованной формы выбора, которая позволила бы нам просто выбрать элемент из B для каждого $a : A$ для использования в определении $f(a)$.

3.10 Когда высказывания усекаются?

На первый взгляд может показаться, что усеченные версии $+$ и $\sum$ на самом деле ближе к неформальному математическому смыслу «или» и «существует», чем неусеченные. Разумеется, они ближе к точному значению «или» и «существует» в логике первого порядка, которая лежит в основе формальной теории множеств, поскольку последняя не пытается запомнить ни одного свидетельства истинности высказываний. Однако может показаться неожиданным осознание того, что практика *неформальной* математики часто более точно описывается неусеченными формами.

Например, рассмотрим высказывание типа «каждое простое число либо 2, либо нечетное». Работающий математик не чувствует угрызений совести по использованию этого факта, не только для доказательства теорем о простых числах, но и для выполнения *конструкций* на простых числах, возможно, делая одно в случае 2, и другое в случае нечетного простого числа. Конечный результат построения — это не просто истинность некоторого высказывания, а часть данных, которая может зависеть от четности простого числа. Таким образом, с теоретико-множественной точки зрения такая конструкция естественно формулируется с использованием принципа индукции для типа копроизведения «$(p = 2) + (p$ нечетно$)$», а не его пропозиционального усечения.

По общему признанию, это не идеальный пример, поскольку «$p = 2$» и «p нечетно» являются взаимоисключающими, поэтому $(p = 2) + (p$ нечетно$)$ на самом деле уже является простым высказыванием и, следовательно, эквивалентно его усечению (см. упражнение 3.7). Более убедительные примеры взяты из экзистенциального квантора. Нередко можно доказать теорему вида «существует такое x, что ...», а затем ссылаться далее на «x, построенное по теореме Y». Более того, при получении дополнительных свойств этого x можно использовать такие фразы, как «построение x в доказательстве теоремы Y».

Очень распространенный пример: «A изоморфен B», что, строго говоря, означает только, что существует *некоторый* изоморфизм между A и B. Но почти всегда при доказательстве такого высказывания проявляется определенный изоморфизм или доказывается, что некоторое уже известное отображение является изоморфизмом, и в последнее время часто возникает вопрос о том, какой именно изоморфизм был использован.

Математики-теоретики часто испытывают чувство вины за такие «злоупотребления языком». Мы можем попытаться извиниться за подобные обороты речи, изъять их из завершенных проектов или отгородиться от них, используя такие неопределенные слова, как «канонические». Проблема усугубляется тем фактом, что в формализованной теории множеств технически нет возможности вообще «конструировать» объекты — мы можем только доказать, что существует объект с определенными свойствами. Таким образом, неусеченная логика в теории типов фиксирует некоторые распространенные практики неформальной математики, которые скрывают теоретическую реконструкцию (это похоже на то, как аксиома унивалентности закрепляет обычную, но формально необоснованную практику идентификации изоморфных объектов).

С другой стороны, иногда усеченная логика имеет важное значение. Мы видели это в выражениях LEM и AC; некоторые другие примеры появятся в книге позже. Таким образом, мы сталкиваемся с проблемой: при описании неформальной теории типов, что мы подразумеваем под словами «или» и «существует» (наряду с обычными синонимами, такими как «есть» и «мы имеем»)?

Универсальный консенсус может быть недостижим. Возможно, в зависимости от области математики, то или иное соглашение может быть более полезным — или, возможно, выбор соглашения может быть неактуальным. В этом случае примечания в начале математической статьи может быть достаточно для информирования читателя о существующих в нем лингвистических условностях. Однако даже после того, как выбрано одно общее соглашение, обычно возникает другая логика, по крайней мере, иногда, поэтому нам нужен способ обращения с ней. В более общем плане можно рассмотреть замену пропозиционального усечения на другую операцию на типах, которая ведет себя аналогично, например, операция двойного отрицания

$A \mapsto \neg\neg A$ или n-усечения, которые будут рассмотрены в главе 7. В качестве эксперимента в изложении в дальнейшем мы будем иногда использовать *наречия*, чтобы обозначить применение таких «модальностей» как пропозициональное усечение.

Например, если неусеченная логика является стандартным соглашением, мы можем использовать наречие **просто** для обозначения пропозиционального усечения. Таким образом, фраза

$$\text{«просто существует такое } x : A, \text{ что } P(x)\text{»}$$

означает тип $\left\| \sum_{(x:A)} P(x) \right\|$. Аналогично, мы скажем, что тип A **просто обитаем** (или, **просто заселен**), подразумевая, что его пропозициональное усечение $\|A\|$ заселено (то есть, что у нас имеется неименованный его элемент). Обратите внимание, что это есть *определение* наречия «просто», поскольку оно должно использоваться на нашем неформальном математическом разговорном языке, так же, как мы определяем существительные типа «группа» и «кольцо», прилагательные типа «регулярные» и «нормальные», чтобы иметь точные математические значения. Мы не утверждаем, что словарное определение «просто» относится к пропозициональному усечению; выбор слова предназначен только для того, чтобы напомнить читателю-математику, что простое высказывание содержит «просто» информацию об истинностном значении и ничего больше.

С другой стороны, если усеченная логика является текущим соглашением по умолчанию, мы можем использовать наречие, такое как **чисто** или **конструктивно**, чтобы отметить отсутствие усечения, так что

$$\text{«конструктивно существует такое } x : A, \text{ что } P(x)\text{»}$$

будет обозначать тип $\sum_{(x:A)} P(x)$. Мы также можем использовать «чисто» или «фактически», чтобы подчеркнуть отсутствие усечения, даже если это соглашение по умолчанию.

В этой книге мы продолжим использовать неусеченную логику как соглашение по умолчанию в силу ряда причин.

(1) Мы хотим побудить новичка экспериментировать с ней, а не придерживаться усеченной логики, просто потому, что это более знакомо.

(2) Использование усеченной логики по умолчанию в теории типов страдает от того же сорта проблем «злоупотребления языком» в качестве теоретико-множественных оснований, которые избегает неусеченная логика. Например, наше определение «$A \simeq B$» как типа эквивалентности между A и B, а не его пропозиционального усечения, означает, что доказательство теоремы вида «$A \simeq B$» — это буквально построение определенной такой эквивалентности. Эту конкретную эквивалентность можно затем упомянуть позже.

(3) Мы хотим подчеркнуть, что понятие «простого высказывания» не является фундаментальной частью теории типов. Как мы увидим в главе 7, простые высказывания — это всего лишь вторая ступень на бесконечной лестнице, и существует также множество других модальностей, не лежащих на этой лестнице вообще.

(4) Многие высказывания, которые классически являются простыми высказываниями, уже не относятся к гомотопической теории типов. Конечно, прежде всего, это равенство.

(5) С другой стороны, одно из наиболее интересных наблюдений гомотопической теории типов заключается в том, что удивительное количество типов автоматически является простыми

высказываниями, может быть слегка измененными, чтобы считаться таковыми, без необходимости какого-либо усечения (см. лемму 3.3.5 и главы 4, 7, 9 и 10). Таким образом, хотя эти типы не содержат данных вне истинностных значений, мы тем не менее можем использовать их для построения неусеченных объектов, поскольку нет необходимости использовать принцип индукции пропозиционального усечения. Этот полезный факт более неудобен для выражения того случая, когда пропозициональное усечение применяется ко всем высказываниям по умолчанию.

(6) Наконец, усечения не очень полезны для большей части математики, которой мы будем заниматься в этой книге, поэтому их проще указывать явно, когда они встречаются.

3.11 Стягиваемость

В лемме 3.3.2 мы заметили, что простое высказывание, которое обитаемо, должно быть эквивалентно $\mathbf{1}$, и нетрудно видеть, что верно и обратное. Тип с этим свойством называется *стягиваемым*. Другим эквивалентным определением стягиваемости, которое также иногда удобно, является следующее.

Определение 3.11.1. Тип A является **стягиваемым** или **синглетоном**, если существует $a : A$, называемое **центром стягивания**, такое что $a = x$ для всех $x : A$. Обозначим указанный путь $a = x$ через contr_x.

Другими словами, тип $\mathsf{isContr}(A)$ определяется как

$$\mathsf{isContr}(A) :\equiv \sum_{(a:A)} \prod_{(x:A)} (a = x)\,.$$

Обратите внимание, что при обычном толковании высказываний как типов мы можем произносить $\mathsf{isContr}(A)$ как «A содержит ровно один элемент», или, точнее, «A содержит некоторый элемент, и любой элемент из A равен этому элементу».

Замечание 3.11.2. Мы также можем понимать $\mathsf{isContr}(A)$ топологически как «существует точка $a : A$ такая, что для всех $x : A$ существует путь от a к x». Обратите внимание, что в классическом восприятии это звучит как определение *связности*, а не стягиваемости. Дело в том, что смысл «существует» в этом предложении является непрерывным/естественным.

Лучшим способом выразить связность будет запись $\sum_{(a:A)} \prod_{(x:A)} \|a = x\|$. Это действительно так, если считать A точечным — см. замечание после леммы 7.5.11, но вообще тип может быть связан, не являясь точечным. В §7.5 мы будем определять связность, как случай с $n = 0$ общего понятия n-связности, а в упражнении 7.6 читателю будет предложено показать, что такое определение эквивалентно как $\|A\|$, так и $\prod_{(x,y:A)} \|x = y\|$.

Лемма 3.11.3. *Для типа A, следующие высказывания логически эквивалентны.*

(i) A является стягиваемым в смысле определения 3.11.1.

(ii) A — простое высказывание и существует точка $a : A$.

(iii) A эквивалентен $\mathbf{1}$.

Доказательство. Если A является стягиваемым, то он конечно же имеет точку $a : A$ (центр стягивания), а для любых $x, y : A$ имеет место $x = a = y$; так что A — простое высказывание. Обратно, если $a : A$ и A — простое высказывание, то для любого $x : A$ имеем $x = a$; таким образом, A стягиваемо. Мы показали (ii)⇒(iii) в лемме 3.3.2, а обратное следует из того, что $\mathbf{1}$ очевидно обладает свойством (ii). □

Лемма 3.11.4. *Для любого типа A тип* $\mathsf{isContr}(A)$ *является простым высказыванием.*

Доказательство. Пусть заданы $c, c' : \mathsf{isContr}(A)$. Мы можем предположить, что $c \equiv (a, p)$ и $c' \equiv (a', p')$, для $a, a' : A$, и $p : \prod_{(x:A)}(a = x)$, а $p' : \prod_{(x:A)}(a' = x)$. Согласно характеризацие путей в $\sum$-типах, чтобы показать, что $c = c'$, достаточно предъявить $q : a = a'$ такое, что $q_*(p) = p'$. Выберем $q :\equiv p(a')$. Теперь, поскольку A стягиваемо (по c или c'), по лемме 3.11.3 это простое высказывание. Следовательно, в силу леммы 3.3.4 и примера 3.6.2 имеем $\prod_{(x:A)}(a' = x)$; таким образом, $q_*(p) = p'$ выполняется автоматически. □

Следствие 3.11.5. *Если A стягиваемо, то таковым является и* $\mathsf{isContr}(A)$.

Доказательство. Следует из леммы 3.11.4 и леммы 3.11.3(ii). □

Как и простые высказывания, стягиваемые типы сохраняются многими конструкторами типов. Например:

Лемма 3.11.6. *Если $P : A \to \mathcal{U}$ - семейство типов такое, что каждое $P(a)$ стягиваемо, то и $\prod_{(x:A)} P(x)$ стягиваемо.*

Доказательство. В силу примера 3.6.2 $\prod_{(x:A)} P(x)$ является простым высказыванием, так как таково каждое $P(x)$. Но оно также имеет элемент, а именно функцию, отображающую каждый $x : A$ в центр стягивания $P(x)$. Таким образом, по лемме 3.11.3(ii), $\prod_{(x:A)} P(x)$ стягиваемо. □

(Фактически, утверждение леммы 3.11.6 эквивалентно аксиоме функциональной экстенсиональности; см. §4.9)

Конечно, если A эквивалентно B, и A стягиваемо, то и B стягиваемо. В общем случае достаточно, чтобы B было *ретрактом* A. По определению, **ретракция** является функцией $r : A \to B$ такой, что существует функция $s : B \to A$, называемая ее **сечением**, и гомотопия $e : \prod_{(y:B)}(r(s(y)) = y)$; тогда мы говорим, что B является **ретрактом** A.

Лемма 3.11.7. *Если B является ретрактом A, и A стягиваемо, то и B стягиваемо.*

Доказательство. Пусть $a_0 : A$ — центр стягивания. Покажем, что $b_0 :\equiv r(a_0) : B$ — центр стягивания для B. Пусть $b : B$; нам нужен путь $b = b_0$. Но мы имеем $\epsilon_b : r(s(b)) = b$ и $\mathsf{contr}_{s(b)} : s(b) = a_0$, поэтому с помощью композиции

$$\epsilon_b^{-1} \cdot r\left(\mathsf{contr}_{s(b)}\right) : b = r(a_0) \equiv b_0 .$$

□

Стягиваемые типы могут показаться не очень интересными, так как все они эквивалентны $\mathbf{1}$. Одна из причин, по которой это понятие полезно, заключается в том, что иногда сбор индивидуально нетривиальных данных будет в совокупности формировать стягиваемый тип. Важным примером является пространство путей с одной конечной точкой. Как мы увидим в §5.8, этот факт по существу инкапсулирует основанный на принципе индукции путь для типов тождественности.

Лемма 3.11.8. *Для любого A и любого $a : A$ тип $\sum_{(x:A)}(a = x)$ является стягиваемым.*

Доказательство. Выберем в качестве центра точку (a, refl_a). Предположим теперь, $(x, p) : \sum_{(x:A)}(a = x)$; мы должны показать, что $(a, \mathsf{refl}_a) = (x, p)$. По характеризации путей в $\sum$-типах достаточно предъявить $q : a = x$ такое, что $q_*(\mathsf{refl}_a) = p$. Но мы можем взять $q :\equiv p$, и в этом случае $q_*(\mathsf{refl}_a) = p$ следует из характерзации транспортировки в типах путей. □

Когда это происходит, то может позволить нам упростить сложную конструкцию до эквивалентности, используя неформальный принцип, согласно которому стягиваемые данные могут свободно игнорироваться. Этот принцип состоит из многих лемм, большинство из которых мы оставляем читателю; далее приведен пример.

Лемма 3.11.9. *Пусть $P : A \to \mathcal{U}$ — семейство типов.*

(i) Если каждое $P(x)$ стягаемо, то $\sum_{(x:A)} P(x)$ эквивалентно A.

(ii) Если A стягиваемо с центровм a, то $\sum_{(x:A)} P(x)$ эквивалентно $P(a)$.

Доказательство. В ситуации (i) покажем, что $\mathsf{pr}_1 : \sum_{(x:A)} P(x) \to A$ — эквивалентность. Для квазиобратности определим $g(x) :\equiv (x, c_x)$, где c_x — центр $P(x)$. Составное выражение $\mathsf{pr}_1 \circ g$, очевидно, есть id_A, тогда как обратное выражение гомотопно тождественности, используя стягивания каждого $P(x)$.

Мы оставляем доказательство (ii) читателю (см. упражнение 3.20). □

Другая причина стягиваемых типов интересна в том, что они расширяют лестницу n-типов, упомянутых в §3.1, еще на одну ступеньку.

Лемма 3.11.10. *Тип A является простым высказыванием тогда и только тогда, когда для всех $x, y : A$ тип $x =_A y$ является стягиваемым.*

Доказательство. Для «тогда» мы просто замечаем, что любой стягиваемый тип обитаем. Для «только тогда» в §3.3 мы отметили, что каждое простое высказывание является множеством, так что каждый тип $x =_A y$ является простым высказыванием. Но он также обитаем (так как A — простое высказывание), и, следовательно, по лемме 3.11.3 (ii), является стягиваемым. □

Таким образом, стягиваемые типы также можно назвать (-2)**-типами**. Они являются нижней ступенькой лестницы n-типов и будут базовым случаем рекурсивного определения n-типов в главе 7.

Примечания

Тот факт, что можно определить множества, простые высказывания и стягиваемые типы в теории типов, со всеми высшими гомотопиями, о которых шла речь в §§ 3.1, 3.3 и 3.11, впервые был исследован Воеводским. Фактически, он определил всю иерархию n-типов индукцией, как мы это сделаем в главе 7.

Теорема 3.2.2 и следствие 3.2.7 в основном базируются на классической теореме Хедберга (Hedberg), которую мы докажем в §7.2. Подтекст того, что форма LEM высказываний как типов противоречит унивалентности, отмечался Мартином Эскардо в списке рассылки Agda. Доказательство, приведенное в теореме 3.2.2, принадлежит Тьерри Кокванду.

Пропозициональное усечение было введено в экстенсиональной теории типов NuPRL в 1983 году Констеблем (Constable) [Con85] как применение типов «подмножество» и «частное». То, что здесь называется «пропозициональным усечением», называлось «сплющиванием» (squashing) в теории типов NuPRL [CAB+86]. Правила, характеризующие пропозициональное усечение непосредственно, все еще в экстенсиональноой теории типов, приведены в [AB04]. Интенсиональная версия в гомотопической теории типов была построена Воеводским с использованием непредикативной квантификации, а позднее Лемсдейном (Lumsdaine) с использованием высших индуктивных типов (см. §6.9).

Воеводский [Voe12] предложил правила изменения размера типа, рассмотренные в §3.5. Они явно связаны с пресловутой *аксиомой приводимости*, предложенной Расселом (Russell) и Уайтхедом (Whitehead) в «Принципах математики» [WR27].

Наречие «чисто», используемое для обозначения неусеченной логики, является ссылкой на использование монадических модальностей для моделирования эффектов в языках программирования; см. §7.7 и примечания к главе 7.

Существует множество различных способов, с помощью которых логику можно рассматривать относительно теории типов. Например, помимо простой логики высказываний как типов, описанной в §1.11, и альтернативы, которая использует простые высказывания только как описано в §3.6, можно ввести отдельный «вид» высказываний, которые ведут себя как типы, но не идентифицированы с ними. Это подход, применяемый в логической теории обогащенных типов [AG02] и в некоторых представлениях внутренних языков топосов и связанных категорий (например, [Jac99, Joh02]), а также в помощнике доказательств Coq. Такой подход является более общим, но менее мощным. Например, принцип единственности выбора (§3.9) терпит неудачу в категории так называемых сетоидов в Coq [Spi11], в логике теории обогащенного типа [AG02] и в теории минимальных типов [MS05]. Таким образом, аксиома унивалентности делает нашу теорию типов более похожей на внутреннюю логику топоса; см. также главу 10.

Мартин Лёф в [ML06] обсуждает историю аксиом выбора. Разумеется, конструктивная и интуиционистская математики имеют долгую и сложную историю, в которую мы здесь углубляться не будем; см., например, [TvD88a, TvD88b].

Упражнения

Упражнение 3.1. Докажите, что если $A \simeq B$ и A — множество, то и B — множество.

Упражнение 3.2. Докажите, что если A и B — множества, то $A + B$ также является множеством.

Упражнение 3.3. Докажите, что если A — множество, а $B : A \to \mathcal{U}$ — семейство типов такое, что $B(x)$ — множество для всех $x : A$, то $\sum_{(x:A)} B(x)$ — множество.

Упражнение 3.4. Покажите, что A — простое высказывание тогда и только тогда, когда $A \to A$ стягиваемо.

Упражнение 3.5. Покажите, что $\mathsf{isProp}(A) \simeq (A \to \mathsf{isContr}(A))$.

Упражнение 3.6. Покажите, что если A — простое высказывание, то таким же является и $A + (\neg A)$. Таким образом, нет необходимости вставлять пропозициональное усечение в (3.4.1).

Упражнение 3.7. Более обще, покажите, что если A и B — простые высказывания и $\neg(A \times B)$, то $A + B$ также является простым высказыванием.

Упражнение 3.8. Предполагая, что некоторый тип $\mathsf{isequiv}(f)$ удовлетворяет условиям (i)-(iii) из §2.4, покажите, что тип $\|\mathsf{qinv}(f)\|$ удовлетворяет тем же условиям и эквивалентен $\mathsf{isequiv}(f)$.

Упражнение 3.9. Покажите, что если выполняется LEM, то тип $\mathsf{Prop} :\equiv \sum_{(A:\mathcal{U})} \mathsf{isProp}(A)$ эквивалентен $\mathbf{2}$.

Упражнение 3.10. Покажите, что если $\mathcal{U}_{i+1}$ удовлетворяет LEM, то каноническое включение $\mathsf{Prop}_{\mathcal{U}_i} \to \mathsf{Prop}_{\mathcal{U}_{i+1}}$ является эквивалентностью.

Упражнение 3.11. Покажите, что неверно, что для всех $A : \mathcal{U}$ имеет место $\|A\| \to A$ (однако могут существовать особые типы, которые удовлетворяют этому; из упражнения 3.8 следует, что таковым является $\mathsf{qinv}(f)$).

Упражнение 3.12. Покажите, что если выполнено LEM, то для всех $A : \mathcal{U}$ имеем $\|(\|A\| \to A)\|$ (это свойство является очень простой формой аксиомы выбора, которая может быть неработоспособной при отсутствии LEM, см. [KECA13]).

Упражнение 3.13. В следствии 3.2.7 мы показали, что следующая наивная форма LEM несовместима с унивалентностью:

$$\prod_{A:\mathcal{U}} (A + (\neg A)) .$$

В отсутствие унивалентности эта аксиома является согласующейся. Однако, покажите, что она означает аксиому выбора (3.8.1).

Упражнение 3.14. Покажите, что, предполагая LEM, двойное отрицание $\neg\neg A$ имеет то же универсальное свойство, что и пропозициональное усечение $\|A\|$, и поэтому эквивалентно ему. Таким образом, с LEM пропозициональное усечение может быть определено, а не принято как отдельный конструктор типа.

Упражнение 3.15. Покажите, что если мы принимаем пропозициональное изменение размера, как в §3.5, то тип

$$\prod_{P:\mathsf{Prop}} ((A \to P) \to P)$$

то тип имеет то же универсальное свойство, что и $\|A\|$. Таким образом, мы можем также определить пропозициональное усечение и в этом случае.

Упражнение 3.16. Предполагая LEM, покажите, что двойное отрицание коммутирует с универсальной квантификацией простых высказываний над множествами. То есть, покажите, что если X — множество и каждое $Y(x)$ — простое высказывание, то LEM подразумевает

$$\left(\prod_{x:X} \neg\neg Y(x)\right) \simeq \left(\neg\neg \prod_{x:X} Y(x)\right) . \tag{3.11.11}$$

Заметим, что если предположить, что каждое Y (x) является множеством, то (3.11.11) становится эквивалентом аксиомы выбора (3.8.3).

Упражнение 3.17. Покажите, что правила для пропозиционального усечения, заданные в §3.7, достаточны для того, чтобы ввести следующий принцип индукции: для любого семейства типов $B : \|A\| \to \mathcal{U}$ такого, что каждый $B(x)$ является простым высказыванием, если для любого $a : A$ имеем $B(|a|)$, то для каждого $x : \|A\|$ имеем $B(x)$.

Упражнение 3.18. Покажите, что закон исключения третьего (3.4.1) и закон двойного отрицания (3.4.2) логически эквивалентны.

Упражнение 3.19. Предположим, что $P : \mathbb{N} \to \mathcal{U}$ — разрешимое семейство (см. определение 3.4.3 (ii)) простых высказываний. Докажите, что

$$\left\| \sum_{n:\mathbb{N}} P(n) \right\| \to \sum_{n:\mathbb{N}} P(n)\,.$$

Упражнение 3.20. Докажите лемму 3.11.9(ii): если A стягиваемо с центром a, то $\sum_{(x:A)} P(x)$ эквивалентно $P(a)$.

Упражнение 3.21. Докажите, что $\mathsf{isProp}(P) \simeq (P \simeq \|P\|)$.

Упражнение 3.22. Как и в классической теории множеств, конечная версия аксиомы выбора является теоремой. Докажите, что аксиома выбора (3.8.1) выполняется, когда X — конечный тип $\mathsf{Fin}(n)$ (как определено в упражнении 1.9).

Упражнение 3.23. Покажите, что заключение упражнения 3.19 верно, если $P : \mathbb{N} \to \mathcal{U}$ — любое разрешимое семейство.

Глава 4

Эквивалентность

Теперь более подробно рассмотрим понятие *эквивалентности типов*, которое было кратко введено в §2.4. В частности, дадим несколько разных способов определения типа $\mathsf{isequiv}(f)$, обладающего указанными здесь свойствами. Напомним, что мы хотели бы, чтобы тип $\mathsf{isequiv}(f)$ имел следующие свойства, которые здесь переформулируем:

(i) $\mathsf{qinv}(f) \to \mathsf{isequiv}(f)$.

(ii) $\mathsf{isequiv}(f) \to \mathsf{qinv}(f)$.

(iii) $\mathsf{isequiv}(f)$ — простое высказывание.

Здесь $\mathsf{qinv}(f)$ обозначает тип квазиобратных к f:

$$\sum_{g:B\to A} \left((f \circ g \sim \mathsf{id}_B) \times (g \circ f \sim \mathsf{id}_A) \right) .$$

Из функциональной экстенсиональности следует, что $\mathsf{qinv}(f)$ эквивалентен типу

$$\sum_{g:B\to A} \left((f \circ g = \mathsf{id}_B) \times (g \circ f = \mathsf{id}_A) \right) .$$

Мы определим три различных типа, имеющих свойства (i)-(iii), которые мы называем

- полусопряженные эквивалентности,
- би-обратимые отображения и
- стягиваемые функции.

Мы также покажем, что все эти типы эквивалентны. Их имена преднамеренно несколько громоздки, потому что, когда мы узнаем, что все они эквивалентны и имеют свойства (i)-(iii), мы вернемся к тому, чтобы использовать просто термин «эквивалентность», не указывая, какое конкретное определение мы выберем. Но для целей сравнений в этой главе нам нужны разные имена для каждого определения.

Однако, прежде чем мы рассмотрим различные понятия эквивалентности, мы дадим более подробные объяснения, почему требуется другое понятие вместо квазиобратимости.

4.1 Квазиобратные

Мы сказали, что $\mathsf{qinv}(f)$ неудовлетворительно, потому что это не простое высказывание, тогда как мы предпочли бы, чтобы данная функция могла «быть эквивалентностью» не более чем одним способом. Однако мы не дали никаких доказательств того, что $\mathsf{qinv}(f)$ не является простым высказыванием. В этом разделе мы приводим конкретный контрпример.

Лемма 4.1.1. *Если $f : A \to B$ такая, что тип $\mathsf{qinv}(f)$ — обитаемый, то*

$$\mathsf{qinv}(f) \simeq \left(\prod_{x:A}(x = x)\right).$$

Доказательство. По предположению f является эквивалентностью; т.е. $e : \mathsf{isequiv}(f)$ и, следовательно, $(f, e) : A \simeq B$. В силу унивалентности $\mathsf{idtoeqv} : (A = B) \to (A \simeq B)$ является эквивалентностью, поэтому мы можем предположить, что (f, e) имеет вид $\mathsf{idtoeqv}(p)$ для некоторого $p : A = B$. Тогда по индукции пути мы можем предположить, что p является refl_A, и в этом случае f — id_A. Таким образом, мы сводим заключение леммы к доказательству $\mathsf{qinv}(\mathsf{id}_A) \simeq (\prod_{(x:A)}(x = x))$. Теперь по определению имеем

$$\mathsf{qinv}(\mathsf{id}_A) \equiv \sum_{g:A\to A} ((g \sim \mathsf{id}_A) \times (g \sim \mathsf{id}_A)) .$$

По функциональной экстенсиональности это эквивалентно

$$\sum_{g:A\to A} ((g = \mathsf{id}_A) \times (g = \mathsf{id}_A)) .$$

А с учетом упражнения 2.10, это эквивалентно

$$\sum_{h:\sum_{(g:A\to A)}(g=\mathsf{id}_A)} (\mathsf{pr}_1(h) = \mathsf{id}_A) .$$

Однако по лемме 3.11.8 $\sum_{(g:A\to A)}(g = \mathsf{id}_A)$ является стягиваемым с центром $(\mathsf{id}_A, \mathsf{refl}_{\mathsf{id}_A})$; поэтому по лемме 3.11.9 этот тип эквивалентен типу $\mathsf{id}_A = \mathsf{id}_A$. А по функциональной экстенсиональности, $\mathsf{id}_A = \mathsf{id}_A$ эквивалентен $\prod_{(x:A)}(x = x)$. □

В упражнении 4.3 предлагается доказать эту лемму, не прибегая к унивалентности.

Таким образом, нам нужен некоторый тип A, допускающий нетривиальный элемент из $\prod_{(x:A)}(x = x)$. Представляя A высшим группоидом, получаем, что обитатель из $\prod_{(x:A)}(x = x)$ является естественным преобразованием из тождественного функтора A в себя. Говорят, что такие преобразования образуют **центр категории**, так как аксиома естественности требует, чтобы они коммутировали со всеми морфизмами. Классически, если A — группа, рассматриваемая как однообъектный группоид, то это дает именно ее центр в обычном теоретико-групповом смысле. Это также обеспечивает некоторую мотивацию для утверждения:

Лемма 4.1.2. *Предположим, что имеется тип A с $a : A$ и $q : a = a$ такие, что*

(i) Тип $a = a$ является множеством.

(ii) Для всех $x : A$ имеет место $\|a = x\|$.

(iii) Для любых $p : a = a$ *имеем* $p \cdot q = q \cdot p$.

Тогда существует $f : \prod_{(x:A)}(x = x)$ *с* $f(a) = q$.

Доказательство. Пусть $g : \prod_{(x:A)} \|a = x\|$ задано формулой (ii). Сначала отметим, что каждый тип $x =_A y$ является множеством. Поскольку «быть множеством» есть простое высказывание, мы можем применить принцип индукции пропозиционального усечения и предположить, что $g(x) = |p|$ и $g(y) = |p'|$ для $p : a = x$ и $p' : a = y$. В этом случае композиция с p и p'^{-1} дает эквивалентность $(x = y) \simeq (a = a)$. Но $(a = a)$ является множеством по (i), поэтому $(x = y)$ также является множеством.

Теперь мы хотим определить f, назначая каждому x путь $g(x)^{-1} \cdot q \cdot g(x)$, но это не работает, потому что $g(x)$ не обитает в $a = x$, но обитает в $\|a = x\|$, а тип $(x = x)$ не может быть простым высказыванием, поэтому мы не можем использовать индукцию по пропозициональному усечению. Вместо этого мы можем применить метод, упомянутый в §3.9: мы однозначно характеризуем объект, который мы хотим построить. Определим для каждого $x : A$ тип

$$B(x) :\equiv \sum_{(r:x=x)} \prod_{(s:a=x)} (r = s^{-1} \cdot q \cdot s)\,.$$

Покажем, что $B(x)$ является простым высказыванием для каждого $x : A$. Поскольку это утверждение само по себе является простым высказыванием, мы можем снова применить индукцию по усечению и считать, что $g(x) = |p|$ для некоторого $p : a = x$. Предположим теперь, что заданы (r, h) и (r', h') в $B(x)$; тогда

$$h(p) \cdot h'(p)^{-1} : r = r'\,.$$

Остается показать, что h отождествляется с h' при транспортировке по этому равенству, которое посредством транспортировки в типах тождественности и функциональных типах (§§ 2.9 и 2.11) сводится к

$$h(s) = h(p) \cdot h'(p)^{-1} \cdot h'(s)\,.$$

для любого $s : a = x$. Но каждая сторона этого выражения является равенством между элементами $(x = x)$, поэтому из вышеизложенного следует, что $(x = x)$ является множеством.

Таким образом, каждый $B(x)$ является простым высказыванием; мы утверждаем, что $\prod_{(x:A)} B(x)$. Для $x : A$ можно теперь сослаться на принцип индукции пропозиционального усечения, чтобы предположить, что $g(x) = |p|$ для $p : a = x$. Определим $r :\equiv p^{-1} \cdot q \cdot p$; для обитаемости $B(x)$ остается показать, что для любого $s : a = x$ имеет место $r = s^{-1} \cdot q \cdot s$. Манипулирование путями сводит это к $q \cdot (p \cdot s^{-1}) = (p \cdot s^{-1}) \cdot q$. Но это есть именно экземпляр (iii). □

Теорема 4.1.3. *Существуют типы* A *и* B *и функция* $f : A \to B$ *такие, что* $\mathsf{qinv}(f)$ *не является простым высказыванием.*

Доказательство. Достаточно предъявить такой тип X, чтобы $\prod_{(x:X)}(x = x)$ не являлось простым высказыванием. Определим $X :\equiv \sum_{(A:\mathcal{U})} \|\mathbf{2} = A\|$, как в доказательстве леммы 3.8.5. Достаточно предъявить такое $f : \prod_{(x:X)}(x = x)$, которое не равно $\lambda x.\, \mathsf{refl}_x$.

Пусть $a :\equiv (\mathbf{2}, |\mathsf{refl}_{\mathbf{2}}|) : X$ и пусть $q : a = a$ — путь, соответствующий не-тождественной эквивалентности $e : \mathbf{2} \simeq \mathbf{2}$, определяемой как $e(0_{\mathbf{2}}) :\equiv 1_{\mathbf{2}}$ и $e(1_{\mathbf{2}}) :\equiv 0_{\mathbf{2}}$. Мы хотим применить лемму 4.1.2 для построения f. По определению X, равенств в типах подмножеств (§3.5) и унивалентности, имеем $(a = a) \simeq (\mathbf{2} \simeq \mathbf{2})$, что является множеством, поэтому (i) выполняется. Аналогично, по определению X и равенств в типах подмножеств имеем (ii). Наконец, из упражнения 2.13 следует, что каждая эквивалентность $\mathbf{2} \simeq \mathbf{2}$ равна либо $\mathsf{id}_{\mathbf{2}}$, либо e, поэтому мы можем показать (iii) анализом четырех случаев.

Таким образом, имеем $f : \prod_{(x:X)}(x = x)$, так что $f(a) = q$. Так как e не равно $\mathsf{id}_{\mathbf{2}}$, то q не равно refl_a, и, следовательно, f не равно $\lambda x.\, \mathsf{refl}_x$. Следовательно, $\prod_{(x:X)}(x = x)$ не является простым высказыванием. $\square$

Более обще, из леммы 4.1.2 следует, что любое «пространство Эйленберга-МакЛейна» $K(G, 1)$, где G — нетривиальная абелева группа, предоставляет контрпример; см. главу 8. Используемый нами тип X оказывается эквивалентным $K(\mathbb{Z}_2, 1)$. В главе 6 мы увидим, что окружность $\mathbb{S}^1 = K(\mathbb{Z}, 1)$ является еще одним простым примером.

Перейдем теперь к описанию более подходящих нотаций эквивалентности.

4.2 Полусопряженные эквивалентности

В §4.1 мы пришли к выводу, что тип $\mathsf{qinv}(f)$ эквивалентен $\prod_{(x:A)}(x = x)$, путем отбрасывания стягиваемого типа. Говоря упрощенно, тип $\mathsf{qinv}(f)$ содержит данные g, h и e, из которых два (g и h) вместе могут рассматриваться как стягиваемый тип, когда f является эквивалентностью. Проблема в том, что после удаления этих данных, остается еще один элемент (e). Идея решения этой проблемы состоит в том, чтобы добавить еще один, *дополнительный* элемент данных, который вместе с e образует стягиваемый тип.

Определение 4.2.1. Функция $f : A \to B$ называется **полусопряженной эквивалентностью**, если существует функция $g : B \to A$ и гомотопии $\eta : g \circ f \sim \mathsf{id}_A$ и $\epsilon : f \circ g \sim \mathsf{id}_B$ такие, что существует гомотопия

$$\tau : \prod_{x:A} f(\eta x) = \epsilon(fx)\,.$$

Таким образом, мы имеем тип $\mathsf{ishae}(f)$, определяемый так, чтобы выполнялось

$$\sum_{(g:B\to A)}\ \sum_{(\eta:g\circ f\sim \mathsf{id}_A)}\ \sum_{(\epsilon:f\circ g\sim \mathsf{id}_B)}\ \prod_{(x:A)} f(\eta x) = \epsilon(fx)\,.$$

Заметим, что в вышеприведенном определении условие когерентности, связывающее h и e, включает только f. Вместо этого мы могли бы рассмотреть аналогичное условие когерентности с g:

$$\upsilon : \prod_{y:B} g(\epsilon y) = \eta(gy)$$

и результирующее аналогичное определение $\mathsf{ishae}'(f)$.

Кстати, оказывается, что каждое из условий предполагает другое:

Лемма 4.2.2. *Для функций $f : A \to B$ и $g : B \to A$, гомотопий $\eta : g \circ f \sim \mathsf{id}_A$ и $\epsilon : f \circ g \sim \mathsf{id}_B$, следующие условия логически эквивалентны:*

- $\prod_{(x:A)} f(\eta x) = \epsilon(fx)$
- $\prod_{(y:B)} g(\epsilon y) = \eta(gy)$

Доказательство. Достаточно показать одно направление; другое получается путем замены A, f и h на B, g и e соответственно. Пусть $\tau : \prod_{(x:A)} f(\eta x) = \epsilon(fx)$. Зафиксируем $y : B$. Используя естественность e и применяя g, получаем следующую коммутативную диаграмму путей:

$$\begin{tikzcd}
gfgfgy \arrow[r, equal, "gfg(\epsilon y)"] \arrow[d, equal, "g(\epsilon(fgy))"'] & gfgy \arrow[d, equal, "g(\epsilon y)"] \\
gfgy \arrow[r, equal, "g(\epsilon y)"'] & gy
\end{tikzcd}$$

Использование $\tau(gy)$ в левой части диаграммы дает нам

$$\begin{tikzcd}
gfgfgy \arrow[r, equal, "gfg(\epsilon y)"] \arrow[d, equal, "gf(\eta(gy))"'] & gfgy \arrow[d, equal, "g(\epsilon y)"] \\
gfgy \arrow[r, equal, "g(\epsilon y)"'] & gy
\end{tikzcd}$$

Используя коммутативность η с $g \circ f$ (следствие 2.4.4), имеем

$$\begin{tikzcd}
gfgfgy \arrow[r, equal, "gfg(\epsilon y)"] \arrow[d, equal, "\eta(gfgy)"'] & gfgy \arrow[d, equal, "g(\epsilon y)"] \\
gfgy \arrow[r, equal, "g(\epsilon y)"'] & gy
\end{tikzcd}$$

Однако, по естественности η, также имеем

$$\begin{tikzcd}
gfgfgy \arrow[r, equal, "gfg(\epsilon y)"] \arrow[d, equal, "\eta(gfgy)"'] & gfgy \arrow[d, equal, "\eta(gy)"] \\
gfgy \arrow[r, equal, "g(\epsilon y)"'] & gy
\end{tikzcd}$$

Таким образом, сокращая все, кроме правой гомотопии, имеем $g(\epsilon y) = \eta(gy)$, что и требовалось. □

Однако, важно, что мы не включали *одновременно* и t и u в определение $\mathsf{ishae}(f)$ (откуда и термин «*полу*сопряженная эквивалентность»). Если бы мы это сделали, то после аннулирования стягиваемых типов мы все равно имели бы один оставшийся элемент данных — пока мы не добавили бы еще одно условие высшей когерентности. Вообще, мы ожидаем, что получим хороший тип, если будем усекать его после нечетного числа когерентностей.

Конечно, очевидно, что $\mathsf{ishae}(f) \to \mathsf{qinv}(f)$: просто забудьте о привязке когерентности. Другое направление — это версия стандартного аргумента из теории гомотопий и теории категорий.

Теорема 4.2.3. *Для любой функции $f : A \to B$ имеет место* $\mathsf{qinv}(f) \to \mathsf{ishae}(f)$.

Доказательство. Предположим, что (g, η, ϵ) является квазиобратной для f. Мы должны предоставить четверку $(g', \eta', \epsilon', \tau)$, свидетельствующую, что f является полусопряженной эквивалентностью. Чтобы определить g' и η', мы можем сделать очевидный выбор, установив $g' :\equiv g$ и $\eta' :\equiv \eta$. Однако, в определении ϵ' нам нужно сначала позаботиться о построении τ, поскольку нельзя полагаться просто на свое чутье и считать ϵ' равным ϵ. Вместо этого мы принимаем

$$\epsilon'(b) :\equiv \epsilon(f(g(b)))^{-1} \cdot (f(\eta(g(b))) \cdot \epsilon(b)) .$$

Теперь нам нужно найти

$$\tau(a) : f(\eta(a)) = \epsilon(f(g(f(a))))^{-1} \cdot (f(\eta(g(f(a)))) \cdot \epsilon(f(a)))\,.$$

Заметим сначала, что в силу следствия 2.4.4, имеем $\eta(g(f(a))) = g(f(\eta(a)))$. Поэтому мы можем применить лемму 2.4.3 для вычисления

$$\begin{aligned} f(\eta(g(f(a)))) \cdot \epsilon(f(a)) &= f(g(f(\eta(a)))) \cdot \epsilon(f(a)) \\ &= \epsilon(f(g(f(a)))) \cdot f(\eta(a)) \end{aligned}$$

из которого получаем искомый путь $\tau(a)$. □

Объединяя этот результат с леммой 4.2.2 (или симметризируя доказательство), мы также получаем $\mathsf{qinv}(f) \to \mathsf{ishae}'(f)$.

Остается показать, что $\mathsf{ishae}(f)$ является простым высказыванием. Для этого нам необходимо знать, что слои эквивалентности стягиваются.

Определение 4.2.4. Слоем отображения $f : A \to B$ над точкой $y : B$ является

$$\mathsf{fib}_f(y) :\equiv \sum_{x:A} (f(x) = y)\,.$$

В теории гомотопий это то, что мы будем называть *гомотопическим слоем* f. Леммы пути в §2.5 дают следующую характеризацию путей в слоях:

Лемма 4.2.5. *Для любых* $f : A \to B$*,* $y : B$ *и* $(x,p),(x',p') : \mathsf{fib}_f(y)$ *имеет место*

$$((x,p) = (x',p')) \simeq \left(\sum_{\gamma : x = x'} f(\gamma) \cdot p' = p\right)\,.$$

Теорема 4.2.6. *Если* $f : A \to B$ *— полусопряженная эквивалентность, то для любого* $y : B$ *слой* $\mathsf{fib}_f(y)$ *является стягиваемым.*

Доказательство. Пусть $(g,\eta,\epsilon,\tau) : \mathsf{ishae}(f)$ и зафиксируем $y : B$. В качестве центра стягивания для $\mathsf{fib}_f(y)$ выберем $(gy, \epsilon y)$. Возьмем теперь любое $(x,p) : \mathsf{fib}_f(y)$; мы хотим построить путь от $(gy,\epsilon y)$ к (x,p). По лемме 4.2.5 достаточно указать путь $\gamma : gy = x$ такой, что $f(\gamma) \cdot p = \epsilon y$. Положим $\gamma :\equiv g(p)^{-1} \cdot \eta x$. Тогда

$$\begin{aligned} f(\gamma) \cdot p &= fg(p)^{-1} \cdot f(\eta x) \cdot p \\ &= fg(p)^{-1} \cdot \epsilon(fx) \cdot p \\ &= \epsilon y \end{aligned}$$

где второе равенство следует по τx, а третье равенство является естественностью для ϵ. □

Теперь мы определим типы, которые инкапсулируют стягиваемые пары данных. Следующие типы объединяют квазиобратную g с одной из гомотопий.

Определение 4.2.7. Для функции $f : A \to B$ определим типы

$$\begin{aligned}\mathsf{linv}(f) &:\equiv \sum_{g:B\to A} (g \circ f \sim \mathsf{id}_A)\\ \mathsf{rinv}(f) &:\equiv \sum_{g:B\to A} (f \circ g \sim \mathsf{id}_B)\end{aligned}$$

левых обратных и **правых обратных** к f, соответственно. Назовем f **левой обратимой**, если $\mathsf{linv}(f)$ обитаем, и аналогично, **правой обратимой**, если $\mathsf{rinv}(f)$ обитаем.

Лемма 4.2.8. *Если $f : A \to B$ имеет квазиобратную, то*

$$\begin{aligned}(f \circ \_) &: (C \to A) \to (C \to B)\\ (\_ \circ f) &: (B \to C) \to (A \to C)\,.\end{aligned}$$

Доказательство. Если g является квазиобратной к f, то $(g \circ \_)$ и $(\_ \circ g)$ являются квазиобратными к $(f \circ \_)$ и $(\_ \circ f)$ соответственно. □

Лемма 4.2.9. *Если $f : A \to B$ имеет квазиобратную, то типы $\mathsf{rinv}(f)$ и $\mathsf{linv}(f)$ являются стягиваемыми.*

Доказательство. По функциональной экстенсиональности имеем

$$\mathsf{linv}(f) \simeq \sum_{g:B\to A} (g \circ f \sim \mathsf{id}_A)\,.$$

Но это есть слой $(\_ \circ f)$ над id_A, поэтому, по лемме 4.2.8 и теоремам 4.2.3 и 4.2.6, он является стягиваемым. Аналогично, $\mathsf{rinv}(f)$ эквивалентен слою $(f \circ \_)$ над id_B и, следовательно, тоже является стягиваемым. □

Далее мы определяем типы, которые объединяют другую гомотопию с дополнительной привязкой когерентности.

Определение 4.2.10. Для $f : A \to B$, левого обратного $(g, \eta) : \mathsf{linv}(f)$ и правого обратного $(g, \epsilon) : \mathsf{rinv}(f)$, введем обозначения

$$\begin{aligned}\mathsf{lcoh}_f(g, \eta) &:\equiv \sum_{(\epsilon: f\circ g \sim \mathsf{id}_B)} \prod_{(y:B)} g(\epsilon y) = \eta(gy)\,,\\ \mathsf{rcoh}_f(g, \epsilon) &:\equiv \sum_{(\eta: g\circ f \sim \mathsf{id}_A)} \prod_{(x:A)} f(\eta x) = \epsilon(fx)\,.\end{aligned}$$

Лемма 4.2.11. *Для любых f, g, ϵ, η имеет место*

$$\begin{aligned}\mathsf{lcoh}_f(g, \eta) &\simeq \prod_{y:B} (fgy, \eta(gy)) =_{\mathsf{fib}_g(gy)} (y, \mathsf{refl}_{gy})\,,\\ \mathsf{rcoh}_f(g, \epsilon) &\simeq \prod_{x:A} (gfx, \epsilon(fx)) =_{\mathsf{fib}_f(fx)} (x, \mathsf{refl}_{fx})\,.\end{aligned}$$

Доказательство. С использованием леммы 4.2.5. □

Лемма 4.2.12. *Если f — полусопряженная эквивалентность, то для любого $(g, \epsilon) : \mathsf{rinv}(f)$ тип $\mathsf{rcoh}_f(g, \epsilon)$ является стягиваемым.*

Доказательство. В силу леммы 4.2.11 и того факта, что зависимые типы функций сохраняют стягиваемые пространства, достаточно показать, что для каждого $x : A$ тип

$$(gfx, \epsilon(fx)) =_{\mathsf{fib}_f(fx)} (x, \mathsf{refl}_{fx})$$

является стягиваемым. Но по теореме 4.2.6 слой $\mathsf{fib}_f(fx)$ является стягиваемым, а любое пространство пути стягиваемого пространства само стягиваемо. □

Теорема 4.2.13. *Для любой $f : A \to B$ тип $\mathsf{ishae}(f)$ является простым высказыванием.*

Доказательство. В силу упражнения 3.5 достаточно предположить, что f является полусопряженной эквивалентностью и показать, что $\mathsf{ishae}(f)$ является стягиваемым. Теперь, в связи с ассоциативностью $\sum$ (упражнение 2.10), тип $\mathsf{ishae}(f)$ эквивалентен типу

$$\sum_{u:\mathsf{rinv}(f)} \mathsf{rcoh}_f(\mathsf{pr}_1(u), \mathsf{pr}_2(u)) .$$

Но в силу лемм 4.2.9 и 4.2.12 и того факта, что $\sum$ сохраняет стягиваемость, последний тип также является стягиваемым. □

Таким образом, мы показали, что $\mathsf{ishae}(f)$ имеет все три желаемые качества для типа $\mathsf{isequiv}(f)$. В следующих двух разделах мы рассмотрим пару других возможностей.

4.3 Би-обратимые отображения

Используя язык, введенный в §4.2, мы можем переформулировать определение, предложенное в §2.4, следующим образом.

Определение 4.3.1. Скажем, что $f : A \to B$ является **би-обратимой**, если она имеет как левый обратный, так и правый обратный, и при этом:

$$\mathsf{biinv}(f) :\equiv \mathsf{linv}(f) \times \mathsf{rinv}(f) .$$

В §2.4 мы доказали, что $\mathsf{qinv}(f) \to \mathsf{biinv}(f)$ и $\mathsf{biinv}(f) \to \mathsf{qinv}(f)$. Остается следующее:

Теорема 4.3.2. *Для любой $f : A \to B$ тип $\mathsf{biinv}(f)$ является простым высказыванием.*

Доказательство. Мы можем предположить, что f является би-обратимой и показать, что $\mathsf{biinv}(f)$ стягиваем. Но так как $\mathsf{biinv}(f) \to \mathsf{qinv}(f)$, в силу леммы 4.2.9, в этом случае оба, $\mathsf{linv}(f)$ и $\mathsf{rinv}(f)$, являются стягиваемыми, а произведение стягиваемых типов стягиваемо. □

Обратите внимание, что это также соответствует предложению, сделанному в начале §4.2: мы объединяем g и h в стягиваемый тип и добавляем элемент данных, который объединяется с ϵ в стягиваемый тип. Разница заключается в том, что вместо добавления элемента данных *более высокого* порядка (двумерного пути) для объединения с ϵ, мы добавляем элемент *пониженного* порядока (правый обратный, который отделен от левого обратного).

Следствие 4.3.3. *Для любой $f : A \to B$ имеет место $\mathsf{biinv}(f) \simeq \mathsf{ishae}(f)$.*

Доказательство. Мы имеем $\mathsf{biinv}(f) \to \mathsf{qinv}(f) \to \mathsf{ishae}(f)$ и $\mathsf{ishae}(f) \to \mathsf{qinv}(f) \to \mathsf{biinv}(f)$. Поскольку, и $\mathsf{ishae}(f)$, и $\mathsf{biinv}(f)$, являются простыми высказываниями, эквивалентность следует из леммы 3.3.3. □

4.4 Стягиваемые слои

Заметим, что наши доказательства, касающиеся $\mathsf{ishae}(f)$ и $\mathsf{biinv}(f)$, существенно использовали тот факт, что слои эквивалентности являются стягиваемыми. Фактически, оказывается, что это свойство само является достаточным определением эквивалентности.

Определение 4.4.1 (Стягиваемые отображения)**.** Отображение $f : A \to B$ называется **стягиваемым**, если для любого $y : B$ слой $\mathsf{fib}_f(y)$ является стягиваемым.

Таким образом, тип $\mathsf{isContr}(f)$ определяется как

$$isContr(f) :\equiv \prod_{y:B} \mathsf{isContr}(\mathsf{fib}_f(y))\,. \tag{4.4.2}$$

Заметим, что в §3.11 мы определили, что означает то, что *тип* является стягиваемым. Здесь мы определяем, что означает то, что *отображение* является стягиваемым. Наша терминология следует общей теоретико-гомотопической практике, заключающейся в том, что отображение имеет определенное свойство, если все его (гомотопические) слои обладают этим свойством. Таким образом, тип A является стягиваемым только тогда, когда отображение $A \to \mathbf{1}$ является стягиваемым. Начиная с главы 7, мы также будем называть стягиваемые отображения и типы (-2)-*усеченными*.

Мы уже показали в теореме 4.2.6, что $\mathsf{ishae}(f) \to \mathsf{isContr}(f)$. Наоборот:

Теорема 4.4.3. *Для любых $f : A \to B$ имеет место* $\mathsf{isContr}(f) \to \mathsf{ishae}(f)$.

Доказательство. Пусть $P : \mathsf{isContr}(f)$. Определим обратное отображение $g : B \to A$, направив каждый $y : B$ в центр стягивания слоя в точке y:

$$g(y) :\equiv \mathsf{pr}_1(\mathsf{pr}_1(Py))\,.$$

Таким образом, мы можем определить гомотопию e путем отображения y в свидетельство, что $g(y)$ действительно принадлежит слою в точке y:

$$\epsilon(y) :\equiv \mathsf{pr}_2(\mathsf{pr}_1(Py))\,.$$

Остается определить η и τ, что, конечно, означает предоставление элемента из $\mathsf{rcoh}_f(g, \epsilon)$. По лемме 4.2.11 это то же самое, что и для каждого $x : A$ — путь от $(gfx, \epsilon(fx))$ к (x, refl_{fx}) в слое f над fx. Но это легко: для любого $x : A$ тип $\mathsf{fib}_f(fx)$ является стягиваемым по предположению, поэтому такой путь должен существовать. Мы можем построить его явно как

$$\big(\mathsf{pr}_2(P(fx))(gfx, \epsilon(fx))\big)^{-1} \cdot \big(\mathsf{pr}_2(P(fx))(x, \mathsf{refl}_{fx})\big)\ .$$

□

Также легко понять:

Лемма 4.4.4. *Для любого f тип* $\mathsf{isContr}(f)$ *— простое высказывание.*

Доказательство. По лемме 3.11.4 каждый тип $\mathsf{isContr}(\mathsf{fib}_f(y))$ является простым высказыванием. Таким образом, согласно примеру 3.6.2, таким же является (4.4.2). □

Теорема 4.4.5. *Для любого $f : A \to B$ имеет место* $\mathsf{isContr}(f) \simeq \mathsf{ishae}(f)$.

Доказательство. Мы уже установили логическую эквивалентность $\mathsf{isContr}(f) \Leftrightarrow \mathsf{ishae}(f)$, оба компонента которой являются простыми высказываниями (лемма 4.4.4 и теорема 4.2.13). Таким образом, применяем лемму 3.3.3. □

Обычно мы доказываем, что функция является эквивалентностью, предъявляя квазиобратную, но иногда такое определение является более удобным в других ситуациях. Например, оно означает, что при доказательстве того, что функция является эквивалентностью, мы можем предположить, что ее кообласть обитаема.

Следствие 4.4.6. *Если $f : A \to B$ таково, что $B \to \mathsf{isequiv}(f)$, то f является эквивалентностью.*

Доказательство. Чтобы показать, что f является эквивалентностью, достаточно показать, что $\mathsf{fib}_f(y)$ стягиваемо для любого $y : B$. Но если $e : B \to \mathsf{isequiv}(f)$, то для любого такого y имеем $e(y) : \mathsf{isequiv}(f)$, так что f является эквивалентностью, и, следовательно, $\mathsf{fib}_f(y)$ стягиваемо, что и требовалось. □

4.5 Об определении эквивалентностей

Мы показали, что все три определения эквивалентности удовлетворяют трем желательным свойствам и попарно эквивалентны:

$$\mathsf{isContr}(f) \simeq \mathsf{ishae}(f) \simeq \mathsf{biinv}(f)$$

(существуют еще возможные определения эквивалентности, но мы остановимся на этих трех; см. упражнение 3.8 и упражнения в этой главе для некоторых из них). Таким образом, мы можем считать любой из них «определением» для $\mathsf{isequiv}(f)$. Для определенности мы решили определить

$$\mathsf{isequiv}(f) :\equiv \mathsf{ishae}(f)\,.$$

Этот выбор выгоден для формализации, поскольку $\mathsf{ishae}(f)$ содержит наиболее непосредственно полезные данные. С другой стороны, для определенных целей с $\mathsf{biinv}(f)$ часто проще иметь дело, поскольку в ней нет двухмерных путей, и две симметричные половины могут обрабатываться независимо. Однако для целей этой книги конкретный выбор не будет иметь большого значения.

В оставшейся части этой главы мы изучим некоторые другие свойства и характеристики эквивалентностей.

4.6 Сюръекции и вложения

Когда A и B являются множествами, а $f : A \to B$ — эквивалентностью, последнее также называют **изоморфизмом** или **биекцией** (мы избегаем употребления этих терминов для типов, которые не являются множествами, так как в теории гомотопий и теории высших категорий они часто обозначают более строгое понятие «одинаковости», а не гомотопическую эквивалентность). В теории множеств функция является биекцией, когда она является одновременно инъективной и сюръективной. То же самое будет верно и в теории типов, если мы сформулируем эти условия соответствующим образом. Для ясности, при работе с типами, которые не являются множествами, мы будем говорить о *вложениях* вместо инъекций.

Определение 4.6.1. Пусть $f : A \to B$.

(i) Мы говорим, что f является **сюръективной** (или **сюръекцией**), если для каждого $b : B$ имеем $\|\mathsf{fib}_f(b)\|$.

(ii) Скажем, что f является **вложением**, если для любых $x, y : A$ функция $\mathsf{ap}_f : (x =_A y) \to (f(x) =_B f(y))$ является некоторой эквивалентностью.

Другими словами, f сюръективна, если каждый слой из f просто обитаем, или эквивалентно, если для всех $b : B$ просто существует $a : A$ такое, что $f(a) = b$. В традиционных логических обозначениях f является сюръективной, если $\forall(b : B).\exists(a : A).(f(a) = b)$. Это следует отличать от более сильного утверждения, что $\prod_{(b:B)} \sum_{(a:A)}(f(a) = b)$; если это выполняется, мы говорим, что f является **расщепленной сюръекцией** (так как этот последний тип эквивалентен $\sum_{(g:B\to A)} \prod_{(b:B)}(f(g(b)) = b)$; по смыслу, расщепленная сюръекция совпадает с *ретракцией*, определенной в §3.11).

Аксиома выбора из §3.8 именно говорит о том, что каждая сюръекция *между множествами* является расщепленной. Однако при наличии аксиомы унивалентности просто неверно, что *все* сюръекции являются расщепленными. В лемме 3.8.5 построено семейство типов $Y : X \to \mathcal{U}$ такое, что $\prod_{(x:X)} \|Y(x)\|$, но: $\neg \prod_{(x:X)} Y(x)$; для любого такого семейства первая проекция $(\sum_{(x:X)} Y(x)) \to X$ является сюръекцией, но не расщепленной.

Если A и B — множества, то по лемме 3.3.3 f является вложением, когда

$$\prod_{x,y,:A} (f(x) =_B f(y)) \to (x =_A y)\,. \tag{4.6.2}$$

В этом случае мы говорим, что f является **инъективной** или **инъекцией**. Мы не используем этих терминов для типов, которые не являются множествами, потому что они могут быть интерпретированы как (4.6.2), что является неверным понятием для не-множеств. Истинно к тому же, что любая функция между множествами сюръективна тогда и только тогда, когда она является *эпиморфизмом* в подходящем смысле, но это также терпит неудачу для более общих типов, а сюръективность обычно является более важным понятием.

Теорема 4.6.3. *Функция $f : A \to B$ является эквивалентностью тогда и только тогда, когда она одновременно сюръективная и является вложением.*

Доказательство. Если f — эквивалентность, то каждый $\mathsf{fib}_f(b)$ является стягиваемым, следовательно, так же, как и $\mathsf{fib}_f(b)$, f является сюръективной. А в теореме 2.11.1 мы показали, что любая эквивалентность является вложением.

Обратно, пусть f — сюръективное вложение. Пусть $b : B$; мы покажем, что $\sum_{(x:A)}(f(x) = b)$ является стягиваемым. Так как f сюръективна, то просто существует такое $a : A$, что $f(a) = b$. Таким образом, слой из f над b обитаем; остается показать, что это простое высказывание. Для этого предположим, что заданы $x, y : A$ с $p : f(x) = b$ и $q : f(y) = b$. Тогда, поскольку ap_f является эквивалентностью, существует $r : x = y$ с $\mathsf{ap}_f(r) = p \cdot q^{-1}$. Однако, используя характеризацию путей в $\sum$-типах, последнее равенство перестраивается на $r_*(p) = q$. Таким образом, вместе с r это показывает, что $(x, p) = (y, q)$ в слое из f над b. $\square$

Следствие 4.6.4. *Для любой $f : A \to B$ имеет место*

$$\mathsf{isequiv}(f) \simeq (\mathsf{isEmbedding}(f) \times \mathsf{isSurjective}(f))\,.$$

Доказательство. Являясь сюръекцией и вложением, обе части эквивалентности — это простые высказывания; далее применяем лемму 3.3.3. □

Конечно, это следствие не может быть использовано как определение «эквивалентности», поскольку определение вложений относится к эквивалентности. Однако такая характеризация может, тем не менее, быть полезной; см. §8.8. Мы обобщим ее в главе 7.

4.7 Свойства замкнутости эквивалентностей

В лемме 2.4.12 мы уже установили замкнутость эквивалентностей относительно композиции. Кроме того:

Теорема 4.7.1 (свойство 2-из-3)**.** *Пусть $f : A \to B$ и $g : B \to C$. Если любые две из функций f, g и $g \circ f$ являются эквивалентностями, то таковой является и третья.*

Доказательство. Если $g \circ f$ и g — эквивалентности, то $(g \circ f)^{-1} \circ g$ является квазиобратной к f. С одной стороны, мы имеем $(g \circ f)^{-1} \circ g \circ f \sim \mathsf{id}_A$, а с другой —

$$\begin{aligned}(g \circ f)^{-1} \circ g &\sim g^{-1} \circ g \circ f \circ (g \circ f)^{-1} \circ g \\ &\sim g^{-1} \circ g \\ &\sim \mathsf{id}_B\,.\end{aligned}$$

Аналогично, если $g \circ f$ и f — эквивалентности, то $f \circ (g \circ f)^{-1}$ является квазиобратной к g. □

Это представляет собой стандартное условие замыкания на эквивалентностях из гомотопической теории. Также хорошо известно, что они замкнуты под ретрактами в следующем смысле.

Определение 4.7.2. Функция $g : A \to B$ называется **ретрактом** функции $f : X \to Y$, если существут диаграмма

$$\begin{tikzcd} A \arrow[r, "s"] \arrow[d, "g"'] & X \arrow[r, "r"] \arrow[d, "f"'] & A \arrow[d, "g"] \\ B \arrow[r, "s'"'] & Y \arrow[r, "r'"'] & B \end{tikzcd}$$

для которой существуют

(i) гомотопия $R : r \circ s \sim \mathsf{id}_A$.

(ii) гомотопия $R' : r' \circ s' \sim \mathsf{id}_B$.

(iii) гомотопия $L : f \circ s \sim s' \circ g$.

(iv) гомотопия $K : g \circ r \sim r' \circ f$.

(v) для каждого $a : A$ путь $H(a)$, свидетельствующий о коммутативности квадрата

$$\begin{tikzcd} g(r(s(a))) \arrow[r, equal, "K(s(a))"] \arrow[d, equal, "g(R(a))"'] & r'(f(s(a))) \arrow[d, equal, "r'(L(a))"] \\ g(a) \arrow[r, equal, "R'(g(a))^{-1}"'] & r'(s'(g(a))) \end{tikzcd}$$

Напомним, что в §3.11 мы определили, что означает для типа быть ретрактом другого. Это частный случай вышеприведенного определения, где B и Y равны $\mathbf{1}$. Напротив, так же как и со стягиваемостью, ретракции отображений индуцируют ретракции их слоев.

Лемма 4.7.3. *Если функция $g : A \to B$ является ретрактом функции $f : X \to Y$, то $\mathsf{fib}_g(b)$ является ретрактом $\mathsf{fib}_f(s'(b))$ для любого $b : B$, где $s' : B \to Y$ такой же, как в определении 4.7.2.*

Доказательство. Предположим, что $g : A \to B$ является ретрактом $f : X \to Y$. Тогда для любого $b : B$ имеются функции

$$\begin{aligned}
&\varphi_b : \mathsf{fib}_g(b) \to \mathsf{fib}_f(s'(b))\,, && \varphi_b(a,p) :\equiv (s(a), L(a) \cdot s'(p))\,,\\
&\psi_b : \mathsf{fib}_f(s'(b)) \to \mathsf{fib}_g(b)\,, && \psi_b(x,q) :\equiv (r(x), K(x) \cdot r'(q) \cdot R'(b))\,.
\end{aligned}$$

Тогда $\psi_b(\varphi_b(a,p)) \equiv (r(s(a)), K(s(a)) \cdot r'(L(a) \cdot s'(p)) \cdot R'(b))$. Мы утверждаем, что ψ_b является ретракцией с сечением φ_b для всех $b : B$, т.е. для всех $(a,p) : \mathsf{fib}_g(b)$ имеем $\psi_b(\varphi_b(a,p)) = (a,p)$. Другими словами, мы хотим показать, что

$$\prod_{(b:B)} \prod_{(a:A)} \prod_{(p:g(a)=b)} \psi_b(\varphi_b(a,p)) = (a,p)\,.$$

Переупорядочивая первые два $\prod$ и применяя версию леммы 3.11.9, получаем эквивалентное

$$\prod_{a:A} \psi_{g(a)}(\varphi_{g(a)}(a, \mathsf{refl}_{g(a)})) = (a, \mathsf{refl}_{g(a)})\,.$$

Для любого a по теореме 2.7.2 это равенство пар эквивалентно паре равенств. Первые компоненты равны $R(a) : r(s(a)) = a$, поэтому надо только показать, что

$$R(a)_* \left(K(s(a)) \cdot r'(L(a)) \cdot R'(g(a))\right) = \mathsf{refl}_{g(a)}\,.$$

Но это транспортирование вычисляется как $g(R(a))^{-1} \cdot K(s(a)) \cdot r'(L(a)) \cdot R'(g(a))$, поэтому требуемый путь задается согласно $H(a)$. □

Теорема 4.7.4. *Если g является ретрактом эквивалентности f, то и g является эквивалентностью.*

Доказательство. По лемме 4.7.3 каждый слой g является ретрактом слоя f. Таким образом, согласно лемме 3.11.7, если последний является стягиваемым, то и первый — тоже. □

Наконец, покажем, что послойные эквивалентности можно охарактеризовать в терминах эквивалентностей пространств расслоений. Чтобы объяснить терминологию, напомним из §2.3, что семейство типов $P : A \to \mathcal{U}$ можно рассматривать как расслоение над A с пространством расслоений $\sum_{(x:A)} P(x)$, причем расслоение является проекцией $\mathsf{pr}_1 : \sum_{(x:A)} P(x) \to A$. С этой точки зрения, для двух семейств типов $P, Q : A \to \mathcal{U}$, мы можем относиться к функции $f : \prod_{(x:A)} P(x) \to Q(x))$ как к **послойному отображению** или **послойному преобразованию**. Такое отображение индуцирует функцию на пространствах расслоений:

Определение 4.7.5. Для семейств типов $P, Q : A \to \mathcal{U}$ и отображения $f : \prod_{(x:A)} P(x) \to Q(x))$ определим

$$\mathsf{total}(f) :\equiv \lambda w.\, (\mathsf{pr}_1 w, f(\mathsf{pr}_1 w, \mathsf{pr}_2 w)) : \sum_{x:A} P(x) \to \sum_{x:A} Q(x)\,.$$

Теорема 4.7.6. *Предположим, что f является послойным преобразованием между семействами P и Q над типом A и пусть $x : A$, а $v : Q(x)$. Тогда имеем эквивалентность*

$$\mathsf{fib}_{\mathsf{total}(f)}((x, v)) \simeq \mathsf{fib}_{f(x)}(v)\,.$$

Доказательство. Произведем вычисление:

$$\begin{aligned}
\mathsf{fib}_{\mathsf{total}(f)}((x, v)) &\equiv \sum_{w:\sum_{(x:A)} P(x)} (\mathsf{pr}_1 w, f(\mathsf{pr}_1 w, \mathsf{pr}_2 w)) = (x, v) \\
&\simeq \sum_{(a:A)} \sum_{(u:P(a))} (a, f(a, u)) = (x, v) && \text{(по упражнению 2.10)} \\
&\simeq \sum_{(a:A)} \sum_{(u:P(a))} \sum_{(p:a=x)} p_*(f(a, u)) = v && \text{(по теореме 2.7.2)} \\
&\simeq \sum_{(a:A)} \sum_{(p:a=x)} \sum_{(u:P(a))} p_*(f(a, u)) = v \\
&\simeq \sum_{u:P(x)} f(x, u) = v && (*) \\
&\equiv \mathsf{fib}_{f(x)}(v)\,.
\end{aligned}$$

Эквивалентность $(*)$ следует из лемм 3.11.8 и 3.11.9 и упражнения 2.10. □

Будем говорить, что послойное преобразование $f : \prod_{(x:A)} P(x) \to Q(x)$ является **послойной эквивалентностью**, если каждое $f(x) : P(x) \to Q(x)$ является эквивалентностью.

Теорема 4.7.7. *Предположим, что f является послойным преобразованием между семействами P и Q над типом A. Тогда f является послойной эквивалентностью тогда и только тогда, когда* $\mathsf{total}(f)$ *является эквивалентностью.*

Доказательство. Пусть f, P, Q и A удовлетворяют формулировке данной теоремы. По теореме 4.7.6 для всех $x : A$ и $v : Q(x)$ следует, что $\mathsf{fib}_{\mathsf{total}(f)}((x, v))$ является стягиваемым тогда и только тогда, когда $\mathsf{fib}_{f(x)}(v)$ стягиваемо. Таким образом, $\mathsf{fib}_{\mathsf{total}(f)}(w)$ является стягиваемым для всех $w : \sum_{(x:A)} Q(x)$ тогда и только тогда, когда $\mathsf{fib}_{f(x)}(v)$ стягиваемо для всех $x : A$ и $v : Q(x)$. □

4.8 Классификатор объектов

В теории типов имеется основное понятие *семейства типов*, а именно функция $B : A \to \mathcal{U}$. Мы видели, что такие семейства ведут себя как *расслоения* в гомотопической теории, причем расслоением является проекция $\mathsf{pr}_1 : \sum_{(a:A)} B(a) \to A$. Основным фактом в теории гомотопий является то, что каждое отображение эквивалентно расслоению. Имея в своем распоряжении унивалентность, мы можем доказать то же самое в теории типов.

Лемма 4.8.1. *Для любого семейства типов $B : A \to \mathcal{U}$ слой $\mathsf{pr}_1 : \sum_{(x:A)} B(x) \to A$ над A эквивалентен $B(a)$:*

$$\mathsf{fib}_{\mathsf{pr}_1}(a) \simeq B(a)\,.$$

Доказательство. Имеем

$$\begin{aligned}\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} \mathsf{pr}_1(u) = a \\ &\simeq \sum_{(x:A)} \sum_{(b:B(x))} (x = a) \\ &\simeq \sum_{(x:A)} \sum_{(p:x=a)} B(x) \\ &\simeq B(a)\end{aligned}$$

используя левую часть универсального свойства типов тождественности. □

Лемма 4.8.2. *Для любой функции $f : A \to B$ имеет место $A \simeq \sum_{(b:B)} \mathsf{fib}_f(b)$.*

Доказательство. Имеем

$$\begin{aligned}\sum_{b:B} \mathsf{fib}_f(b) &:\equiv \sum_{(b:B)} \sum_{(a:A)} (f(a) = b) \\ &\simeq \sum_{(a:A)} \sum_{(b:B)} (f(a) = b) \\ &\simeq A\end{aligned}$$

используя тот факт, что $\sum_{(b:B)}(f(a) = b)$ является стягиваемым. □

Теорема 4.8.3. *Для любого типа B существует эквивалентность*

$$\chi : \left(\sum_{A:\mathcal{U}} (A \to B)\right) \simeq (B \to \mathcal{U}) .$$

Доказательство. Мы должны построить квазиобратные

$$\begin{aligned}\chi &: \left(\sum_{A:\mathcal{U}} (A \to B)\right) \to B \to \mathcal{U} \\ \psi &: (B \to \mathcal{U}) \to \left(\sum_{A:\mathcal{U}} (A \to B)\right) .\end{aligned}$$

Определим χ согласно $\chi((A, f), b) :\equiv \mathsf{fib}_f(b)$ и ψ согласно $\psi(P) :\equiv \left((\sum_{(b:B)} P(b)), \mathsf{pr}_1\right)$. Теперь мы должны проверить, что $\chi \circ \psi \sim \mathsf{id}$ и $\psi \circ \chi \sim \mathsf{id}$.

(i) Пусть $P : B \to \mathcal{U}$. По лемме 4.8.1, $\mathsf{fib}_{\mathsf{pr}_1}(b) \simeq P(b)$ для любого $b : B$, поэтому отсюда сразу следует, что $P \sim \chi(\psi(P))$.

(ii) Пусть $f : A \to B$ — функция. Мы должны найти путь

$$\left(\sum_{(b:B)} \mathsf{fib}_f(b), \mathsf{pr}_1\right) = (A, f) .$$

Прежде всего заметим, что по лемме 4.8.2 имеем $e : \sum_{(b:B)} \mathsf{fib}_f(b) \simeq A$ с $e(b, a, p) :\equiv a$ и $e^{-1}(a) :\equiv (\mathsf{e(a)}, \mathsf{a}, \mathsf{refl}_{f(a)})$. По теореме 2.7.2 остается показать, что $(\mathsf{ua}(e))_*(\mathsf{pr}_1) = f$. Но по правилу вычисления для унивалентности и (2.9.4) имеем $(\mathsf{ua}(e))_*(\mathsf{pr}_1) = \mathsf{pr}_1 \circ e^{-1}$, а определение e^{-1} сразу дает $\mathsf{pr}_1 \circ e^{-1} \equiv f$.

□

В частности, это означает, что мы имеем *классификатор объектов* в смысле теории высших топосов. Напомним из определения 2.1.7, что $\mathcal{U}_\bullet$ обозначает тип $\sum_{(A:U)} A$ точечных типов.

Теорема 4.8.4. *Пусть $f : A \to B$ — функция. Тогда диаграмма*

$$\begin{tikzcd} A \arrow[r, "\theta_f"] \arrow[d, "f"'] & \mathcal{U}_\bullet \arrow[d, "\mathsf{pr}_1"] \\ B \arrow[r, "\chi_f"'] & \mathcal{U} \end{tikzcd}$$

является квадратом обратного образа (см. упражнение 2.11). Здесь функция θ_f определяется как

$$\lambda a.\, (\mathsf{fib}_f(f(a)), (a, \mathsf{refl}_{f(a)}))\,.$$

Доказательство. Заметим, что имеются эквивалентности

$$\begin{aligned} A &\simeq \sum_{b:B} \mathsf{fib}_f(b) \\ &\simeq \sum_{(b:B)} \sum_{(X:\mathcal{U})} \sum_{(p:\mathsf{fib}_f(b)=X)} X \\ &\simeq \sum_{(b:B)} \sum_{(X:\mathcal{U})} \sum_{(x:X)} \mathsf{fib}_f(b) = X \\ &\simeq \sum_{(b:B)} \sum_{(Y:\mathcal{U}_\bullet)} \mathsf{fib}_f(b) = \mathsf{pr}_1 Y \\ &\equiv B \times_{\mathcal{U}} \mathcal{U}_\bullet\,. \end{aligned}$$

которые дают нам составную эквивалентность $e : A \simeq B \times_{\mathcal{U}} \mathcal{U}_\bullet$. Можно показать действие этой сложной эквивалентности шаг за шагом:

$$\begin{aligned} a &\mapsto (f(a), (a, \mathsf{refl}_{f(a)})) \\ &\mapsto (f(a), \mathsf{fib}_f(f(a)), \mathsf{refl}_{\mathsf{fib}_f(f(a))}, (a, \mathsf{refl}_{f(a)})) \\ &\mapsto (f(a), \mathsf{fib}_f(f(a)), (a, \mathsf{refl}_{f(a)}), \mathsf{refl}_{\mathsf{fib}_f(f(a))})\,. \end{aligned}$$

Поэтому мы получаем гомотопии $f \sim \mathsf{pr}_1 \circ e$ и $\theta_f \sim \mathsf{pr}_2 \circ e$. □

4.9 Унивалентность означает функциональную экстенсиональность

В последнем разделе этой главы мы приведем доказательство того, что аксиома унивалентности подразумевает функциональную экстенсиональность. Таким образом, в этом разделе мы работаем без аксиомы функциональной экстенсиональности. Доказательство состоит из двух шагов. Сначала в теореме 4.9.4 покажем, что аксиома унивалентности подразумевает слабую форму функциональной экстенсиональности, определенную в определении 4.9.1 ниже. Принцип слабой функциональной экстенсиональности, в свою очередь, влечет за собой обычную функциональную экстенсиональность, и это делается без аксиомы унивалентности (теорема 4.9.5).

Пусть $\mathcal{U}$ — универсум; мы будем явно указывать, когда мы предполагаем, что он унивалентен.

Определение 4.9.1. Принцип слабой функциональной экстенсиональности утверждает, что существует функция

$$\left(\prod_{x:A}\mathsf{isContr}(P(x))\right) \to \mathsf{isContr}\left(\prod_{x:A}P(x)\right)$$

для любого семейства $P : A \to \mathcal{U}$ типов над любым типом A.

Следующую лемму легко доказать, используя функциональную экстенсиональность; дело здесь в том, что независимо от этого, она также следует из унивалентности без предположения о функциональной экстенсиональности.

Лемма 4.9.2. *Предполагая, что $\mathcal{U}$ унивалентен, для любых $A, B, X : \mathcal{U}$ и любой $e : A \simeq B$ существует эквивалентность*

$$(X \to A) \simeq (X \to B)$$

у которой отображение, лежащее в ее основе, задается пост-композицией с функцией, лежащей в основе e.

Доказательство. Как и при доказательстве леммы 4.1.1, можно считать, что $e = \mathsf{idtoeqv}(p)$ для некоторого $p : A = B$. Тогда по индукции пути можно считать, что p есть refl_A, так что $e = \mathsf{id}_A$. Но в этом случае пост-композиция с e есть тождество, а следовательно, эквивалентность. □

Следствие 4.9.3. *Пусть $P : A \to \mathcal{U}$ является семейством стягиваемых типов, то есть $\prod_{(x:A)} \mathsf{isContr}(P(x))$. Тогда проекция $\mathsf{pr}_1 : (\sum_{(x:A)} P(x)) \to A$ является эквивалентностью. Из предположения унивалентности $\mathcal{U}$ следует, что пост-композиция с pr_1 дает эквивалентность*

$$\alpha : \left(A \to \sum_{x:A} P(x)\right) \simeq (A \to A)\,.$$

Доказательство. По лемме 4.8.1 для $\mathsf{pr}_1 : (\sum_{(x:A)} P(X)) \to A$ и $x : A$ имеем эквивалентность

$$\mathsf{fib}_{\mathsf{pr}_1}(x) \simeq P(x)\,.$$

Следовательно, pr_1 является эквивалентностью всякий раз, когда каждый $P(x)$ является стягиваемым. Утверждение теперь является следствием леммы 4.9.2. □

В частности, гомотопический слой вышеуказанной эквивалентности при id_A является стягиваемым. Таким образом, мы можем показать, что унивалентность подразумевает слабую функциональную экстенсиональность, показывая, что зависимый функциональный тип $\prod_{(\mathrm{x:A})} \mathrm{P(x)}$ является ретрактом $\mathsf{fib}_\alpha(\mathsf{id}_A)$.

Теорема 4.9.4. *В унивалентном универсуме $\mathcal{U}$ предположим, что $P : A \to \mathcal{U}$ — семейство стягиваемых типов, и пусть α — функция, упоминаемая в следствии 4.9.3. Тогда $\prod_{(x:A)} P(x)$ является ретрактом $\mathsf{fib}_\alpha(\mathsf{id}_A)$. Как следствие, $\prod_{(x:A)} P(x)$ стягиваемо. Другими словами, аксиома унивалентности подразумевает принцип слабой функциональной экстенсиональности.*

Доказательство. Определим функции

$$\varphi : (\textstyle\prod_{(x:A)} P(x)) \to \mathsf{fib}_\alpha(\mathsf{id}_A)\,,$$
$$\varphi(f) :\equiv (\lambda x.\,(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\,,$$

и

$$\psi : \mathsf{fib}_\alpha(\mathsf{id}_A) \to \textstyle\prod_{(x:A)} P(x)\,,$$
$$\psi(g,p) :\equiv \lambda x. \mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))\,.$$

Тогда $\psi(\varphi(f)) = \lambda x.\,f(x)$, что есть f, по принципу единственности для зависимых функциональных типов. □

Покажем теперь, что слабая функциональная экстенсиональность подразумевает обычную функциональную экстенсиональность. Напомним из (2.9.2) функцию $\mathsf{happly}(f,g) : (f = g) \to (f \sim g)$, которая преобразует равенство функций в гомотопию. В следующем доказательстве аксиома унивалентности не используется.

Теорема 4.9.5. *Из слабой функциональной экстенсиональности следует аксиома функциональной экстенсиональности 2.9.3.*

Доказательство. Мы хотим показать, что

$$\prod_{(A:\mathcal{U})} \prod_{(P:A\to\mathcal{U})} \prod_{(f,g:\prod_{(x:A)} P(x))} \mathsf{isequiv}(\mathsf{happly}(f,g))\,.$$

Так как расслоенное отображение индуцирует эквивалентность на пространствах расслоений тогда и только тогда, когда оно является расслоенной эквивалентностью по теореме 4.7.7, достаточно показать, что функция типа

$$\left(\sum_{(g:\prod_{(x:A)} P(x))} (f = g)\right) \to \sum_{g:\prod_{(x:A)} P(x)} (f \sim g)$$

индуцированная согласно $\lambda(g : \prod_{(x:A)} P(x)).\,\mathsf{happly}(f,g)$, является эквивалентностью. Так как тип слева является стягиваемым по лемме 3.11.8, достаточно показать, что тип справа:

$$\sum_{(g:\prod_{(x:A)} P(x))} \prod_{(x:A)} f(x) = g(x) \tag{4.9.6}$$

является стягиваемым. Теперь, теорема 2.15.7 утверждает, что это эквивалентно

$$\prod_{(x:A)} \sum_{(u:P(x))} f(x) = u\,. \tag{4.9.7}$$

Доказательство теоремы 2.15.7 использует функциональную экстенсиональность, но только для одного из составляющих элементов. Таким образом, не предполагая функциональной экстенсиональности, можно заключить, что (4.9.6) является ретрактом (4.9.7). А (4.9.7) является произведением стягиваемых типов, которые стягиваются по принципу слабой функциональной экстенсиональности; следовательно, (4.9.6) также стягиваемо. □

Примечания

Тот факт, что пространство непрерывных отображений, снабженных квазиобратными, имеет неправильный гомотопический тип — «пространство гомотопических эквивалентностей», хорошо известно в алгебраической топологии. В этом контексте «пространство гомотопических эквивалентностей» $(A' \simeq B)$ обычно определяется просто как подпространство функционального пространства $(A \to B)$, состоящее из функций, которые являются гомотопическими эквивалентностями. В теории типов это будет наиболее близко соответствовать $\sum_{(f:A\to B)} \|\mathsf{qinv}(f)\|$; см. упражнение 3.8.

Первое определение эквивалентности, данное в гомотопической теории типов, было тем, что мы назвали $\mathsf{isContr}(f)$, которое было связано с Воеводским. Возможность других определений впоследствии исследовалась различными людьми. Основные теоремы о сопряженных эквивалентностях, такие как лемма 4.2.2 и теорема 4.2.3, — это адаптация стандартных фактов в теории высших категорий и гомотопической теории. Использование биобратимости в качестве определения эквивалентности было предложено Андрэ Джойалем (André Joyal).

Свойства эквивалентностей, обсуждаемые в §§ 4.6 и 4.7, хорошо известны в теории гомотопий. Большинство из них были впервые доказаны Воеводским в теории типов.

Тот факт, что каждая функция эквивалентна расслоению, является стандартным фактом в теории гомотопий. Понятие классификатора объектов в теории $(\infty, 1)$-категорий (категорный аналог теоремы 4.8.3) принадлежит Резку (Rezk, см. [Rez05, Lur09]).

Наконец, тот факт, что унивалентность означает расширение функции (§4.9), принадлежит Воеводскому. Наше доказательство — это его упрощение.

Упражнения

Упражнение 4.1. Рассмотрим тип «двухсторонних данных сопряженной эквивалентности » для $f : A \to B$,

$$\sum_{(g:B\to A)} \sum_{(\eta:g\circ f\sim \mathsf{id}_A)} \sum_{(\epsilon:f\circ g\sim \mathsf{id}_B)} \left(\prod_{x:A} f(\eta x) = \epsilon(fx)\right) \times \left(\prod_{y:B} g(\epsilon y) = \eta(gy)\right).$$

По лемме 4.2.2 мы знаем, что если f является эквивалентностью, то этот тип заселен. Дайте характеризацию этого типа, аналогичную лемме 4.1.1.
Можете ли вы привести пример, показывающий, что этот тип вообще-то не является простым высказыванием? (на этот вопрос будет легче ответить после чтения главы 6).

Упражнение 4.2. Покажите, что для любых $A, B : \mathcal{U}$ следующий тип эквивалентен $A \simeq B$.

$$\sum_{R:A\to B\to\mathcal{U}} \left(\prod_{a:A} \mathsf{isContr}\left(\sum_{b:B} R(a,b)\right)\right) \times \left(\prod_{b:B} \mathsf{isContr}\left(\sum_{a:A} R(a,b)\right)\right).$$

Можете ли вы извлечь из этого определение типа, удовлетворяющего трем вариантам $\mathsf{isequiv}(f)$?

Упражнение 4.3. Докажите лемму 4.1.1 без использования унивалентности.

Упражнение 4.4. (Нестабильная октаэдрическая аксиома) Предположим, что $f : A \to B$, $g : B \to C$ и $b : B$.

(i) Покажите, что существует естественное отображение $\mathsf{fib}_{g\circ f}(g(b)) \to \mathsf{fib}_g(g(b))$, слой которого над $(b, \mathsf{refl}_g(b))$ является эквивалентностью к $\mathsf{fib}_f(b)$.

(ii) Покажите, что $\mathsf{fib}_{g\circ f}(c) \simeq \sum_{(w:\mathsf{fib}_g(c))} \mathsf{fib}_f(\mathsf{pr}_1 w)$.

Упражнение 4.5. Докажите, что эквивалентности удовлетворяют свойству «2-из-6»: для $f : A \to B$, $g : B \to C$ и $h : C \to D$, если $g \circ f$ и $h \circ g$ — эквивалентности, то такими же являются f, g, h и $h \circ g \circ f$. Используйте это, чтобы дать высоко-уровневое доказательство теоремы 2.11.1.

Упражнение 4.6. Для $A, B : \mathcal{U}$ определим

$$\mathsf{idtoqinv}_{A,B} : (A = B) \to \sum_{f:A\to B} \mathsf{qinv}(f)$$

индукцией пути очевидным образом. Пусть **qinv-унивалентность** обозначает модифицированную форму аксиомы унивалентности, которая утверждает, что для всех $A, B : \mathcal{U}$ функция $\mathsf{idtoqinv}_{A,B}$ имеет квазиобратную.

(i) Покажите, что qinv-унивалентность может использоваться вместо унивалентности при доказательстве функциональной экстенсиональности в §4.9.

(ii) Покажите, что qinv-унивалентность может быть использована вместо унивалентности при доказательстве теоремы 4.1.3.

(iii) Покажите, что qinv-унивалентность несогласована (т.е. позволяет построить обитателя в $\mathbf{0}$). Таким образом, использование «хорошей» версии isequiv имеет важное значение в формулировке унивалентности.

Упражнение 4.7. Покажите, что функция $f : A \to B$ является вложением тогда и только тогда, когда выполнены следующие два условия:

(i) f является *сократимой слева*, т.е. для любых $x, y : A$, если $f(x) = f(y)$, то $x = y$.

(ii) Для любого $x : A$ отображение $\mathsf{ap}_f : \Omega(A, x) \to \Omega(B, f(x))$ является эквивалентностью.

(в частности, если A — множество, то f является вложением тогда и только тогда, когда оно сократимо слева и $\Omega(B, f(x))$ является стягиваемым для всех $x : A$). Приведите примеры, чтобы показать, что никакой из (i) или (ii) не подразумевает другого.

Упражнение 4.8. Покажите, что тип функций сократимых слева $\mathbf{2} \to B$ (см. упражнение 4.7) эквивалентен $\sum_{(x,y:B)}(x \neq y)$. Дайте аналогичную явную характеризацию типа вложений $\mathbf{2} \to B$.

Упражнение 4.9. **Наивная аксиома независимой функциональной экстенсиональности** гласит, что для $A, B : \mathcal{U}$ и $f, g : A \to B$ существует функция $(\prod_{(x:A)} f(x) = g(x)) \to (f = g)$. Измените аргумент из §4.9 так, чтобы показать, что эта аксиома влечет аксиому полной функциональной экстенсиональности (аксиома 2.9.3).

Глава 5

Индукция

В главе 1 мы рассмотрели разные способы формирования новых типов из уже используемых. За исключением (зависимых) функциональных типов и универсумов, все эти правила являются особыми случаями общего понятия *индуктивного определения*. В этой главе мы изучим индуктивные определения более подробно.

5.1 Введение в индуктивные типы

Индуктивный тип X можно интуитивно понимать как тип, «свободно порожденный» некоторой конечной совокупностью *конструкторов*, каждый из которых является функцией (некоторого числа аргументов) с кообластью X. В число конструкторов входят и нульарные функции, которые являются просто элементами X.

При описании конкретного индуктивного типа мы приводим перечисляемый список конструкторов. Например, тип $\mathbf{2}$ из §1.8 индуктивно генерируется следующими конструкторами:

- $0_{\mathbf{2}} : \mathbf{2}$
- $1_{\mathbf{2}} : \mathbf{2}$

Аналогично, $\mathbf{1}$ индуктивно генерируется конструктором:

- $\star : \mathbf{1}$

а $\mathbf{0}$ индуктивно порождается вообще без конструкторов. Примером, когда функции-конструкторы принимают аргументы, является копроизведение $A + B$, которое генерируется двумя конструкторами

- $\mathsf{inl} : A \to A + B$
- $\mathsf{inr} : B \to A + B$.

Примером конструктора, принимающим несколько аргументов, является декартово произведение $A \times B$, которое генерируется одним конструктором

- $(\_,\_) : A \to B \to A \times B$.

Реально, мы также допускаем конструкторы индуктивных типов, которые принимают аргументы от определяемого индуктивного типа. Например, тип $\mathbb{N}$ натуральных чисел имеет конструкторы

- $0 : \mathbb{N}$
- $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

Еще одним полезным примером является тип $\mathsf{List}(A)$ конечных списков элементов некоторого типа A, который имеет конструкторы

- $\mathsf{nil} : \mathsf{List}(A)$
- $\mathsf{cons} : A \to \mathsf{List}(A) \to \mathsf{List}(A)$.

Интуитивно, мы должны понимать индуктивный тип как тип, который *свободно генерируется* своими конструкторами. То есть элементы индуктивного типа — это именно то, что можно получить из ничего, неоднократно применяя конструкторы (мы увидим в §5.8 и главе 6, что эту концепцию нужно немного модифицировать для более общих видов индуктивных определений, но на данный момент она удовлетворительна). Например, в случае $\mathbf{2}$ мы должны ожидать, что единственными элементами являются $0_{\mathbf{2}}$ и $1_{\mathbf{2}}$. Аналогично, в случае $\mathbb{N}$ мы должны ожидать, что каждый элемент равен либо 0 или получен путем применения succ к некоторому «предварительно построенному» натуральному числу.

Однако вместо того, чтобы декларировать свойства, подобные этому, непосредственно, мы выражаем их с помощью *принципа индукции*, также называемого (*зависимым*) *правилом исключения*. Мы уже встречались с этими принципами в главе 1. Например, принцип индукции для $\mathbf{2}$ звучит так:

- При доказательстве утверждения $E : \mathbf{2} \to \mathcal{U}$ о *всех* обитателях $\mathbf{2}$, достаточно доказать его для $0_{\mathbf{2}}$ и $1_{\mathbf{2}}$, т.е. привести доказательства $e_0 : E(0_{\mathbf{2}})$ и $e_1 : E(1_{\mathbf{2}})$.

Кроме того, полученное доказательство $\mathsf{ind}_{\mathbf{2}}(E, e_0, e_1) : \prod_{(b:\mathbf{2})} E(b)$ ведет себя как и ожидалось при применении к конструкторам $0_{\mathbf{2}}$ и $1_{\mathbf{2}}$; этот принцип выражается *правилами вычислений*:

- $\mathsf{ind}_{\mathbf{2}}(E, e_0, e_1, 0_{\mathbf{2}}) \equiv e_0$.
- $\mathsf{ind}_{\mathbf{2}}(E, e_0, e_1, 1_{\mathbf{2}}) \equiv e_1$.

Таким образом, принцип индукции для типа $\mathbf{2}$ булевых значений позволяет нам рассуждать по отдельности. Поскольку ни один из двух конструкторов не принимает никаких аргументов, это все, что нужно для этого типа.

Однако, для натуральных чисел анализ случаев обычно недостаточен: в случае, соответствующем индуктивному шагу $\mathsf{succ}(n)$, мы к тому же хотим предположить, что доказанное утверждение уже было предъявлено для n. Это дает нам следующий принцип индукции:

- При доказательстве утверждения $E : N \to \mathcal{U}$ о *всех* натуральных числах, достаточно доказать это для 0 и для $\mathsf{succ}(n)$, считая это справедливым для n, т.е. мы строим $e_z : E(0)$ и $e_s : \prod_{(n:\mathcal{N})} E(n) \to E(\mathsf{succ}(n))$.

Как и в случае с булевыми значениями, имеем соответствующие правила вычислений для функции $\mathsf{ind}_{\mathbb{N}}(E, e_z, e_s) : \prod_{(x:\mathbb{N})} E(x)$:

- $\mathsf{ind}_{\mathbb{N}}(E, e_z, e_s, 0) \equiv e_z$.
- $\mathsf{ind}_{\mathbb{N}}(E, e_z, e_s, \mathsf{succ}(n)) \equiv e_s(n, \mathsf{ind}_{\mathbb{N}}(E, e_z, e_s, n))$ для любых $n : \mathbb{N}$.

Таким образом, зависимая функция $\mathsf{ind}_{\mathbb{N}}(E, e_z, e_s)$ может быть понята как рекурсивно определяемая по аргументу $x : \mathbb{N}$ через функции e_z и e_s, которые мы называем **рекурентностями**. Когда x равно нулю, функция просто возвращает e_z. Когда x является преемником другого натурального числа n, результат получается путем применения рекурентности e_s и подстановки конкретного предшественника n и рекурсивного значения вызова $\mathsf{ind}_{\mathbb{N}}(E, e_z, e_s, n)$.

Принципы индукции для всех приведенных выше примеров разделяют это семейное сходство. В §5.6 мы обсудим общее понятие «индуктивное определение» и как получить для него соответствующий *принцип индукции*, но сначала исследуем различные общности между индуктивными определениями.

Например, в каждом случае в главе 1 мы заметили, что по принципу индукции мы можем получить *принцип рекурсии*, в котором кообласть представляет собой простой тип (а не семейство). Принципы индукции и рекурсии могут казаться разделенными, поскольку они гарантируют только *существование* функции, что, по-видимому, не характеризует ее однозначно. Однако на самом деле принцип индукции достаточно силен и предлагает свой собственный *принцип единственности*, как в следующей теореме.

Теорема 5.1.1. *Пусть $f, g : \prod_{(x:\mathbb{N})} E(x)$ — две функции, удовлетворяющие рекуррентностям*

$$e_z : E(0) \quad \textit{и} \quad e_s : \prod_{n:\mathbb{N}} E(n) \to E(\mathsf{succ}(n))$$

с точностью до пропозиционального равенства, т.е. такие, что

$$f(0) = e_z \quad \textit{и} \quad g(0) = e_z$$

так же как

$$\prod_{n:\mathbb{N}} f(\mathsf{succ}(n)) = e_s(n, f(n)),$$
$$\prod_{n:\mathbb{N}} g(\mathsf{succ}(n)) = e_s(n, g(n)).$$

Тогда f и g равны.

Доказательство. Мы используем индукцию по семейству типов $D(x) :\equiv f(x) = g(x)$. Для базового случая имеем

$$f(0) = e_z = g(0).$$

Для индуктивного случая предположим, что $n : N$ такое, что $f(n) = g(n)$. Тогда

$$f(\mathsf{succ}(n)) = e_s(n, f(n)) = e_s(n, g(n)) = g(\mathsf{succ}(n)).$$

Первое и последнее равенство вытекают из предположений о f и g. Среднее равенство следует из индуктивной гипотезы и того факта, что применение сохраняет равенство. Это дает нам поточечное равенство между f и g; применение функциональной экстенсиональности завершает доказательство. □

Заметим, что принцип единственности применим даже к функциям, которые только удовлетворяют рекурентностям с точностью до *пропозиционального равенства*, т.е. пути. Конечно, частичная функция, полученная по принципу индукции, удовлетворяет этим рекурентностям дефинициально; мы вернемся к этому в §5.5. С другой стороны, сама теорема утверждает только пропозициональное равенство между функциями (см. также упражнение 5.2). С гомотопической точки зрения естественно спросить, является ли этот путь *когерентным*, т.е. является ли равенство $f = g$ единственным с высшими путями; в §5.4 мы увидим, что это действительно так.

Конечно, аналогичные теоремы единственности для функций обычно можно сформулировать и продемонстрироватьь для других индуктивных типов. В следующем разделе мы покажем, как свойство единственности вместе с унивалентностью подразумевает, что индуктивный тип, такой как натуральные числа, полностью характеризуется правилами его введения, исключения и вычисления.

5.2 Единственность индуктивных типов

Мы определили «натуральные числа» как особый тип $\mathbb{N}$ со специфическими индуктивными образующими 0 и succ. Однако по общему принципу индуктивных определений в теории типов, описанных в предыдущем разделе, нет ничего, что помешало бы нам определить *другой* тип идентичным образом. То есть, предположим, что $\mathbb{N}'$ — индуктивный тип, генерируемый конструкторами

- $0' : \mathbb{N}'$
- $\mathsf{succ}' : \mathbb{N}' \to \mathbb{N}$'.

Тогда $\mathbb{N}'$ и $\mathbb{N}$ будет иметь одинаково выглядящие принципы индукции и рекурсии. При доказательстве утверждения $E : \mathbb{N}' \to \mathcal{U}$ для всех «новых» натуральных чисел достаточно дать доказательства $e_z : E(0')$ и $e_s : \prod_{(n:N')} E(n) \to E(\mathsf{succ}'(n))$. И функция $\mathsf{rec}_{\mathbb{N}'}(E, e_z, e_s) : \prod_{(n:N')} E(n)$ имеет следующие правила вычислений:

- $\mathsf{rec}_{\mathbb{N}'}(E, e_z, e_s, 0') \equiv e_z$,
- $\mathsf{rec}_{\mathbb{N}'}(E, e_z, e_s, \mathsf{succ}'(n)) \equiv e_s(n, \mathsf{rec}_{\mathbb{N}'}(E, e_z, e_s, n)$ для любых $n : \mathbb{N}'$.

Но какова связь между $\mathbb{N}$ и $\mathbb{N}'$?

Это не просто академический вопрос, так как структуры, которые «похожи» на натуральные числа, можно найти во многих других местах. Например, мы можем отождествлять натуральные числа со списками над типом одного элемента (это похоже на, возможно, самый древний вид, найденный на стенах пещер), с целыми неотрицательными числами, с подмножествами рациональных и вещественных чисел и т.д. И с точки зрения программирования «унарное» представление наших натуральных чисел очень неэффективно, поэтому мы иногда предпочитаем использовать двоичный код. Мы хотели бы иметь возможность идентифицировать все эти версии «натуральных чисел» друг с другом, чтобы передавать конструкции и результаты от одного к другому.

Конечно, если две версии натуральных чисел удовлетворяют одинаковым принципам индукции, то они имеют идентичную индуцированную структуру. Например, напомним функцию double, определенную в §1.9. Аналогичная функция для наших новых натуральных чисел легко определяется дублированием функции double и добавлением пометок в виде штриха:

$$\mathsf{double}' :\equiv \mathsf{rec}_{\mathbb{N}'}(\mathbb{N}', 0', \lambda n.\, \lambda m.\, \mathsf{succ}'(\mathsf{succ}'(m)))\,.$$

Все просто, как может показаться, но имеется явный недостаток, приводящий к росту копий. Необходимо дублировать не только функции, но и все леммы и их доказательства. Например, простой результат, такой как $\prod_{(n:\mathbb{N})} \mathsf{double}(\mathsf{succ}(n)) = \mathsf{succ}(\mathsf{succ}(\mathsf{double}(n)))$, а также его доказательство по индукции, также должны быть «подвергнуты штриховке».

В традиционной математике только провозглашается, что $\mathbb{N}$ и $\mathbb{N}'$, очевидно, «одинаковы» и могут быть заменены друг на друга всякий раз, когда возникнет необходимость. И это, как правило, непроблематично, но подметает под ковер много чего, увеличивая разрыв между неформальной математикой и ее точным описанием. В гомотопической теории типов мы можем сделать лучше.

Прежде всего заметим, что имеются следующие определяемые отображения:

- $f :\equiv \mathsf{rec}_{\mathbb{N}}(\mathbb{N}', 0', \lambda n.\, \mathsf{succ}') : \mathbb{N} \to \mathbb{N}'$,
- $g :\equiv \mathsf{rec}_{\mathbb{N}'}(\mathbb{N}, 0, \lambda n.\, \mathsf{succ}) : \mathbb{N}' \to \mathbb{N}$.

Так как композиция g и f удовлетворяет тем же рекуррентностям, что и тождественная функция на $\mathbb{N}$, то по теорема 5.1.1 $\prod_{(n:\mathbb{N})} g(f(n)) = n$, а «заштрихованная» версия той же теоремы дает $\prod_{(n:\mathbb{N}')} f(g(n)) = n$. Таким образом, f и g являются квазиобратными, так что $\mathbb{N} \simeq \mathbb{N}'$. Теперь мы можем переносить функции на $\mathbb{N}$ непосредственно к функциям на $\mathbb{N}'$ (и наоборот) вдоль этой эквивалентности, например:

$$\mathsf{double}' :\equiv \lambda n.\, f(\mathsf{double}(g(n)))\,.$$

В качестве простого упражнения, можно показать, что эта версия double' равна предыдущей.

Конечно, в этом нет ничего удивительного; такой изоморфизм — это именно то, как математик представляет себе «отождествление» $\mathbb{N}$ с $\mathbb{N}'$. Однако механизм «переноса» изоморфизмом зависит от передаваемой вещи; это не всегда так просто, как пре- и пост-компоновка одиночной функции с f и g. Рассмотрим, например, простую лемму:

$$\prod_{n:\mathbb{N}'} \mathsf{double}'(\mathsf{succ}'(n)) = \mathsf{succ}'(\mathsf{succ}'(\mathsf{double}'(n)))\,.$$

Вставка правильных f и g немного проще, чем повторное подтверждение этого индукцией по $n : \mathbb{N}'$ непосредственно.

Здесь вступает в действие аксиома унивалентности: поскольку $\mathbb{N} \simeq \mathbb{N}'$, то также имеем $\mathbb{N} =_{\mathcal{U}} \mathbb{N}'$, то есть $\mathbb{N}$ и $\mathbb{N}'$ *равны* как типы. Теперь принцип индукции для тождества гарантирует, что любая конструкция или доказательство, связанное с $\mathbb{N}$, могут автоматически быть перенесены на $\mathbb{N}'$ таким же образом. Мы просто рассматриваем тип функции или теоремы как типизированное по типу семейство типов $P : \mathcal{U} \to \mathcal{U}$, причем данный объект является элементом $P(\mathbb{N})$ и перемещается по пути $\mathbb{N} = \mathbb{N}'$. Все это связано со значительно меньшими накладными расходами.

Для простоты мы описали этот метод для случая двух типов $\mathbb{N}$ и $\mathbb{N}'$ с *идентично* выглядящими определениями. Однако более распространенная ситуация на практике заключается в том, что определения не являются буквально идентичными, но, тем не менее, один принцип индукции подразумевает другой. Рассмотрим, например, тип списков от одноэлементного типа, $\mathsf{List}(\mathbf{1})$, который порождается

- элементом $\mathsf{nil} : \mathsf{List}(\mathbf{1})$ и
- функцией $\mathsf{cons} : \mathbf{1} \times \mathsf{List}(\mathbf{1}) \to \mathsf{List}(\mathbf{1})$.

Это не тождественно определению $\mathbb{N}$, и это не приводит к идентичному принципу индукции. Принцип индукции для $\mathsf{List}(\mathbf{1})$ говорит, что для любого $E : \mathsf{List}(1) \to \mathcal{U}$ вместе с рекуррентными данными $e_{\mathsf{nil}} : E(\mathsf{nil})$ и $e_{cons} : \prod_{(u:\mathbf{1})} \prod_{(\ell:\mathsf{List}(\mathbf{1}))} E(\ell) \to E(\mathsf{cons}(u, \ell))$, существует $f : \prod_{(\ell:\mathsf{List}(\mathbf{1}))} E(\ell)$ такая, что $f(\mathsf{nil}) \equiv e_{\mathsf{nil}}$ и $f(\mathsf{cons}(u, \ell)) \equiv e_{\mathsf{cons}}(u, \ell, f(\ell))$ (мы рассмотрим, как получить индуктивный принцип индуктивного определения в §5.6).

Предположим теперь, что мы определяем $0'' :\equiv \mathsf{nil} : \mathsf{List}(1)$ и $\mathsf{succ}'' : \mathsf{List}(\mathbf{1}) \to \mathsf{List}(\mathbf{1})$ как $\mathsf{succ}''(\ell) :\equiv \mathsf{cons}(\star, \ell)$. Тогда для любого $E : \mathsf{List}(\mathbf{1}) \to \mathcal{U}$ вместе с $e_0 : E(0'')$ и $e_s : \prod_{(\ell:\mathsf{List}(\mathbf{1}))} E(\ell) \to E(\mathsf{succ}''(\ell))$, мы можем определить

$$\begin{aligned} e_{\mathsf{nil}} &:\equiv e_0 \\ e_{\mathsf{cons}}(\star, \ell, x) &:\equiv e_s(\ell, x) \end{aligned}$$

(в определении e_{cons} мы используем принцип индукции для $\mathbf{1}$, чтобы предположить, что u есть $\star$). Теперь мы можем применить принцип индукции для $\mathsf{List}(\mathbf{1})$, получая $f : \prod_{(\ell:\mathsf{List}(\mathbf{1}))} E(\ell)$ такую, что

$$\begin{aligned} f(0'') \equiv f(\mathsf{nil}) \equiv e_{\mathsf{nil}} \equiv e_0 \\ f(\mathsf{succ}''(\ell)) \equiv f(\mathsf{cons}(\star, \ell)) \equiv e_{\mathsf{cons}}(\star, \ell, x) \equiv e_s(\ell, f(\ell)) \,. \end{aligned}$$

Таким образом, $\mathsf{List}(\mathbf{1})$ удовлетворяет тому же принципу индукции, что и $\mathbb{N}$, и, следовательно, (по той же вышеприведенной аргументации) совпадает с ним.

Наконец, эти выводы не ограничиваются натуральными числами: они применимы к любому индуктивному типу. Если мы имеем индуктивно определенный тип, скажем W, и некоторый другой тип W', который удовлетворяет тому же принципу индукции, что и W, то $W \simeq W'$ и, следовательно, $W = W'$. Мы используем полученные принципы рекурсии для W и W' для построения отображений $W \to W'$ и $W' \to W$, соответственно, а затем принципы индукции для каждого отображения, чтобы утверждать, что обе комбинации равны тождествам. Например, в главе 1 мы видели, что копроизведение $A + B$ также можно было определить как $\sum_{(x:\mathbf{2})} \mathsf{rec}_{\mathbf{2}}(\mathcal{U}, A, B, x)$. Последний тип удовлетворяет тому же принципу индукции, что и первый; следовательно, они канонически эквивалентны.

Это, конечно, очень похоже на известный факт из теории категорий, что если два объекта имеют одно и то же *универсальное свойство*, то они эквивалентны. В §5.4 мы увидим, что индуктивные типы действительно имеют универсальное свойство, что является проявлением этого общего принципа.

5.3 W-типы

Индуктивные типы очень общие, что отлично подходит для проявления их полезности и применимости, но затрудняет их изучение в целом. К счастью, все они могут быть формально сведены к нескольким особым случаям. Обсуждение такой редукции выходит за рамки данной книги и в любом случае не имеет отношения к математику, использующему теорию типов на практике, но мы потратим немного времени, чтобы обсудить один из основных особых случаев, который мы еще не встречали. Это W-*типы* Мартина-Лефа, также известные как типы *вполне фундированных деревьев*. W-типы являются обобщением таких типов, как натуральные числа, списки и двоичные деревья, и которые являются достаточно общими для инкапсуляции «рекурсивного» аспекта *любого* индуктивного типа.

Определенный W-тип задается заданием двух параметров $A : \mathcal{U}$ и $B : A \to \mathcal{U}$, и в этом случае результирующий W-тип записывается как $\mathsf{W}_{(a:A)}B(a)$. Тип A представляет собой тип меток для $\mathsf{W}_{(a:A)}B(a)$, которые выполняют функции конструкторов (однако мы оставляем это слово для фактических функций, которые возникают в индуктивных определениях). Например, при определении натуральных чисел как W-типа тип A будет типом $\mathbf{2}$, обитаемым двумя элементами $0_{\mathbf{2}}$ и $1_{\mathbf{2}}$, так как существует ровно два способа получить натуральное число — либо оно будет равно нулю, либо преемником другого натурального числа.

Семейство типов $B : A \to \mathcal{U}$ используется для записи арности ярлыков: метка $a : A$ охватывает семейство индуктивных аргументов, проиндексированных над $B(a)$. Поэтому мы можем думать о «$B(a)$-многих» аргументах для a. Эти аргументы представлены функцией $f : B(a) \to \mathsf{W}_{(a:A)}B(a)$, при том понимании, что для любого $b : B(a)$, $f(b)$ является «b-м» аргументом для метки a. Таким образом, W-тип $\mathsf{W}_{(a:A)}B(a)$ можно рассматривать как тип обоснованных деревьев, где узлы обозначены элементами из A и каждый узел, помеченный как $a : A$, имеет $B(a)$-много ветвей.

В случае натуральных чисел метка $0_{\mathbf{2}}$ имеет арность 0, так как она конструирует константу zero; метка $1_{\mathbf{2}}$ имеет арность 1, так как она конструирует преемника своего аргумента. Мы можем зафиксировать это, используя простое исключение на $\mathbf{2}$, чтобы определить функцию $\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, \mathbf{0}, \mathbf{1})$ в универсуме типов; эта функция возвращает пустой тип $\mathbf{0}$ для $0_{\mathbf{2}}$ и тип единицы $\mathbf{1}$ для $1_{\mathbf{2}}$. Таким образом, мы можем определить

$$\mathbf{N}^{\mathbf{w}} :\equiv \mathsf{W}_{(b:\mathbf{2})}\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, \mathbf{0}, \mathbf{1}, b)$$

где верхний индекс $\mathbf{w}$ служит для выделения этой версии натуральных чисел от ранее использованной. Точно так же мы можем определить тип списков над A как W-тип с $\mathbf{1} + A$ многими метками: одна нульарная метка для пустого списка, плюс одна унарная метка для каждого $a : A$, соответствующая добавлению a к заголовку списка:

$$\mathsf{List}(A) :\equiv \mathsf{W}_{(x:\mathbf{1}+A)}\mathsf{rec}_{\mathbf{1}+A}(\mathcal{U}, \mathbf{0}, \lambda a.\, \mathbf{1}, x)\,.$$

В общем случае W-тип $\mathsf{W}_{(a:A)}B(a)$, заданный посредством $A : \mathcal{U}$ и $B : A \to \mathcal{U}$, является индуктивным типом, сгенерированным следующим конструктором:

- $\mathsf{sup} : \prod_{(a:A)} \big(B(a) \to \mathsf{W}_{(x:A)}B(x)\big) \to \mathsf{W}_{(x:A)}B(x)$.

Конструктор sup (сокращение от supremum) принимает метку $a : A$ и функцию $f : B(a) \to \mathsf{W}_{(x:A)}B(x)$, представляющий аргументы для a, и конструирует новый элемент $\mathsf{W}_{(x:A)}B(x)$. Используя нашу предыдущую кодировку натуральных чисел как W-типов, мы можем, например, определить

$$0^{\mathbf{w}} :\equiv \mathsf{sup}(0_{\mathbf{2}}, \lambda x.\, \mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x))\,.$$

По-другому, предположим, что мы используем метку $0_{\mathbf{2}}$ для построения $0^{\mathbf{w}}$. Тогда $\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, \mathbf{0}, \mathbf{1}, 0_{\mathbf{2}})$ оценивается как $\mathbf{0}$, как и должно быть, поскольку $0_{\mathbf{2}}$ является нульарной меткой. Таким образом, нам нужно сконструировать функцию $f : \mathbf{0} \to \mathbf{N}^{\mathbf{w}}$, представляющей (нулевые) аргументы, заданные для $0_{\mathbf{2}}$. Это, конечно, тривиально, используя простое исключение $\mathbf{0}$, как и показано. Аналогично, мы можем определить $1^{\mathbf{w}}$ и преемственную функцию $\mathsf{succ}^{\mathbf{w}}$

$$1^{\mathbf{w}} :\equiv \mathsf{sup}(1_{\mathbf{2}}, \lambda x.\, 0^{\mathbf{w}})$$
$$\mathsf{succ}^{\mathbf{w}} :\equiv \lambda n.\, \mathsf{sup}(1_{\mathbf{2}}, \lambda x.\, n)\,.$$

Имеем следующий принцип индукции для W-типов:

- При доказательстве утверждения $E : \big(\mathsf{W}_{(x:A)}B(x)\big) \to \mathcal{U}$ обо всех элементах W-типа $\mathsf{W}_{(x:A)}B(x)$ достаточно доказать его для $\mathsf{sup}(a, f)$, считая, что оно выполняется для всех $f(b)$ с $b : B(a)$. Другими словами, достаточно привести доказательство

$$e : \prod_{(a:A)} \prod_{(f:B(a)\to \mathsf{W}_{(x:A)}B(x))} \prod_{(g:\prod_{(b:B(a))} E(f(b)))} E(\mathsf{sup}(a, f)) \,.$$

Переменная g представляет собой нашу индуктивную гипотезу, а именно, что все аргументы из a удовлетворяют E. Чтобы сформулировать это, мы проводим количественную оценку по всем элементам типа $B(a)$, так как каждому $b : B(a)$ соответствует один аргумент $f(b)$ из а.

Как бы мы определили функцию double на натуральных числах, закодированных как W-тип? Мы хотели бы использовать принцип рекурсии для $\mathbf{N^w}$ с кообластью самого $\mathbf{N^w}$. Поэтому нам нужно построить подходящую функцию

$$e : \prod_{(a:\mathbf{2})} \prod_{(f:B(a)\to \mathbf{N^w})} \prod_{g:B(a)\to \mathbf{N^w}} \mathbf{N^w}$$

которая будет представлять собой повтор функции double; для простоты обозначим семейство типов $\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, \mathbf{0}, \mathbf{1})$ через B.

Ясно, что e будет функцией, принимающей $a : \mathbf{2}$ в качестве первого аргумента. Следующий шаг — выполнение анализа значения a и продолжение, исходя из того, является ли оно $0_{\mathbf{2}}$ или $1_{\mathbf{2}}$. Это предполагает следующую форму

$$e :\equiv \lambda a.\, \mathsf{rec}_{\mathbf{2}}(C, e_0, e_1, a)$$

где

$$C :\equiv \prod_{(f:B(a)\to \mathbf{N^w})} \prod_{g:B(a)\to \mathbf{N^w}} \mathbf{N^w} \,.$$

Если a есть $0_{\mathbf{2}}$, тип $B(a)$ становится $\mathbf{0}$. Таким образом, при $f : \mathbf{0} \to \mathbf{N^w}$ и $g : \mathbf{0} \to \mathbf{N^w}$, мы хотим построить элемент из $\mathbf{N^w}$. Так как метка $0_{\mathbf{2}}$ представляет $\mathbf{0}$, ей нужны нулевые индуктивные аргументы, а переменные f и g не имеют никакого отношения к этому. В качестве результате мы возвращаем $0^{\mathbf{w}}$:

$$e_0 :\equiv \lambda f.\, \lambda g.\, 0^{\mathbf{w}} \,.$$

Аналогично, если a есть $1_{\mathbf{2}}$, тип $B(a)$ становится $\mathbf{1}$. Так как метка $1_{\mathbf{2}}$ представляет собой оператор-преемник, то необходим один индуктивный аргумент — предшественник — который представляется переменной $f : \mathbf{1} \to \mathbf{N^w}$. Значение рекурсивного вызова для предшественника представляется переменной $g : \mathbf{1} \to \mathbf{N^w}$. Таким образом, взяв это значение (а именно $g(\star)$) и применяя функцию-преемник дважды, получим желаемый результат:

$$e_1 :\equiv \lambda f.\lambda g.\mathsf{succ}^{\mathbf{w}}(\mathsf{succ}^{\mathbf{w}}(g(\star))) \,.$$

Собирая все вместе, таким образом, имеем

$$\mathsf{double} :\equiv \mathsf{rec}_{\mathbf{N^w}}(\mathbf{N^w}, e)$$

с e, определенным выше.

Соответствующим правилом вычисления для функции $\mathsf{rec}_{\mathsf{W}_{(x:A)}B(x)}(E, e) : \prod_{(w:\mathsf{W}_{(x:A)}B(x))} E(w)$ является следующее.

- Для любых $a : A$ и $f : B(a) \to \mathsf{W}_{(x:A)}B(x)$ имеет место

$$\mathsf{rec}_{\mathsf{W}_{(x:A)}B(x)}(E, e, \mathsf{sup}(a, f)) \equiv e(a, f, (\lambda b.\, \mathsf{rec}_{\mathsf{W}_{(x:A)}B(x)}(E, e, f(b)))) \,.$$

Другими словами, функция $\mathsf{rec}_{\mathsf{W}_{(x:A)}B(x)}(E, e)$ удовлетворяет рекуррентности e.

В соответствии с вышеприведенным правилом вычисления функция double ведет себя так, как ожидалось:

$$\begin{aligned}
\mathsf{double}(0^{\mathbf{w}}) &\equiv \mathsf{rec}_{\mathbf{N}^{\mathbf{w}}}(\mathbf{N}^{\mathbf{w}}, e, \mathsf{sup}(0_{\mathbf{2}}, \lambda x.\, \mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x))) \\
&\equiv e(0_{\mathbf{2}}, (\lambda x.\, \mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x)), (\lambda x.\, \mathsf{double}(\mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x)))) \\
&\equiv e_0((\lambda x.\, \mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x)), (\lambda x.\, \mathsf{double}(\mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x)))) \\
&\equiv 0^{\mathbf{w}}
\end{aligned}$$

и

$$\begin{aligned}
\mathsf{double}(1^{\mathbf{w}}) &\equiv \mathsf{rec}_{\mathbf{N}^{\mathbf{w}}}(\mathbf{N}^{\mathbf{w}}, e, \mathsf{sup}(1_{\mathbf{2}}, \lambda x.\, 0^{\mathbf{w}})) \\
&\equiv e(1_{\mathbf{2}}, (\lambda x.\, 0^{\mathbf{w}}), (\lambda x.\mathsf{double}(0^{\mathbf{w}}))) \\
&\equiv e_1((\lambda x.\, 0^{\mathbf{w}}), (\lambda x.\mathsf{double}(0^{\mathbf{w}}))) \\
&\equiv \mathsf{succ}^{\mathbf{w}}(\mathsf{succ}^{\mathbf{w}}((\lambda x.\mathsf{double}(0^{\mathbf{w}}))(\star))) \\
&\equiv \mathsf{succ}^{\mathbf{w}}(\mathsf{succ}^{\mathbf{w}}(0^{\mathbf{w}}))
\end{aligned}$$

и так далее.

Как и для натуральных чисел, мы можем доказать теорему единственности для W-типов:

Теорема 5.3.1. *Пусть $g, h : \prod_{w:\mathsf{W}_{(x:A)}B(x)} E(w)$ — две функции, которые удовлетворяют рекуррентности*

$$e : \prod_{a,f} \left(\prod_{b:B(a)} E(f(b)) \right) \to E(\mathsf{sup}(a, f)) \,,$$

пропозиционально, т.е. такие, что

$$\prod_{a,f} g(\mathsf{sup}(a, f)) = e(a, f, \lambda b.\, g(f(b))) \,,$$
$$\prod_{a,f} h(\mathsf{sup}(a, f)) = e(a, f, \lambda b.\, h(f(b))) \,.$$

Тогда g и h равны.

5.4 Индуктивные типы являются инициальными алгебрами

Как уже упоминалось ранее, индуктивные типы также имеют теоретико-категорное универсальное свойство. Это *гомотопически-инициальные алгебры*: начальные объекты (с точностью до когерентной гомотопии) в категории «алгебр», определяемые специфицированными конструкторами. В качестве простого примера рассмотрим натуральные числа. Соответствующий сорт «алгебры» здесь — это тип, снабженный той же структурой, что и конструкторы для $\mathbb{N}$.

Определение 5.4.1. **$\mathbb{N}$-алгебра** — это тип C с двумя элементами $c_0 : C$, $c_s : C \to C$. Типом таких алгебры является

$$\mathbb{N}\mathsf{Alg} :\equiv \sum_{C:\mathcal{U}} C \times (C \to C)\,.$$

Определение 5.4.2. **$\mathbb{N}$-гомоморфизм** между $\mathbb{N}$-алгебрами (C, c_0, c_s) и (D, d_0, d_s) — это функция $h : C \to D$ такая, что $h(c_0) = d_0$ и $h(c_s(c)) = d_s(h(c))$ для всех $c : C$. Типом таких гомоморфизмов является

$$\mathbb{N}\mathsf{Hom}((C, c_0, c_s), (D, d_0, d_s)) :\equiv \sum_{(h:C\to D)} (h(c_0) = d_0) \times \textstyle\prod_{(c:C)} h(c_s(c)) = d_s(h(c))\,.$$

Таким образом, мы имеем категорию $\mathbb{N}$-алгебр и $\mathbb{N}$-гомоморфизмов, и утверждение состоит в том, что $\mathbb{N}$ является инициальным объектом этой категории. Категорный теоретик сразу узнает в этом определение *объекта натуральных чисел* в категории.

Конечно, поскольку наши типы ведут себя как ∞-группоиды,, то фактически имеем $(\infty, 1)$-категорию $\mathbb{N}$-алгебр, и хотели бы видеть $\mathbb{N}$ в качестве инициального в соответствующем $(\infty, 1)$-категорном смысле. К счастью, это можно сформулировать без необходимости определения $(\infty, 1)$-категорий.

Определение 5.4.3. $\mathbb{N}$-алгебра I называется **гомотопически-инициальной** или, кратко, **h-инициальной**, если для любой другой $\mathbb{N}$-алгебры C, тип $\mathbb{N}$-гомоморфизмов от I к C является стягиваемым. Таким образом,

$$\mathsf{isHinit}_{\mathbb{N}}(I) :\equiv \prod_{C:\mathbb{N}\mathsf{Alg}} \mathsf{isContr}(\mathbb{N}\mathsf{Hom}(I, C))\,.$$

Когда они существуют, h-инициальные алгебры уникальны — не только с точностью до изоморфизма, как обычно формулируется в теории категорий, но и с точностью до равенства, по аксиоме унивалентности.

Теорема 5.4.4. *Любые две h-инициальные $\mathbb{N}$-алгебры равны. Таким образом, тип h-инициальных $\mathbb{N}$-алгебр является простым высказыванием.*

Доказательство. Пусть I и J — h-инициальные $\mathbb{N}$-алгебры. Тогда $\mathbb{N}\mathsf{Hom}(I, J)$ стягиваемо, поэтому населено некоторым $\mathbb{N}$-гомоморфизмом $f : I \to J$, а также имеется $\mathbb{N}$-гомоморфизм $g : J \to I$. Теперь, композиция $g \circ f$ является $\mathbb{N}$-гомоморфизм от I к I, как и id_I; но $\mathbb{N}\mathsf{Hom}(I, I)$ является стягиваемым, так что $g \circ f = \mathsf{id}_I$. Аналогично, $f \circ g = \mathsf{id}_J$. Следовательно, $I \simeq J$, и поэтому $I = J$. Поскольку стягиваемость — это простое высказывание, а зависимые произведения сохраняют простые высказывания, то отсюда следует, что быть h-инициальным само по себе является простым высказыванием. Таким образом, любые два доказательства, что I (или J) является h-инициальным, обязательно равны, что завершает доказательство. □

Теперь можно привести следующую теорему.

Теорема 5.4.5. *$\mathbb{N}$-алгебра $(\mathbb{N}, \mathbf{0}, \mathsf{succ})$ является гомотопической инициальной.*

Набросок доказательства. Зафиксируем произвольную $\mathbb{N}$-алгебру (C, c_0, c_s). Принцип рекурсии для $\mathbb{N}$ дает функцию $f : \mathbb{N} \to C$:

$$f(0) :\equiv c_0$$
$$f(\mathsf{succ}(n)) :\equiv c_s(f(n))\,.$$

Эти равенства делают f $\mathbb{N}$-гомоморфизмом, который можно принять за центр стягивания для $\mathbb{N}\mathsf{Hom}(\mathbb{N}, C)$. Из теоремы единственности (теорема 5.1.1) вытекает, что любой другой $\mathbb{N}$-гомоморфизм совпадает с f. $\square$

Чтобы поместить это в более общий контекст, полезно рассмотреть понятие *алгебры для эндофунктора*. Заметим, что создание типа C в $\mathbb{N}$-алгебре — это то же самое, что и предоставление функции $c : C + \mathbf{1} \to C$, а функция $f : C \to D$ есть $\mathbb{N}$-гомоморфизм, как раз, когда $f \circ c \sim d \circ (f + \mathbf{1})$. На категорном языке это означает, что $\mathbb{N}$-алгебры являются алгебрами для эндофунктора $F(X) :\equiv X + 1$ в категории типов.

Для более общего случая рассмотрим W-тип, ассоциированный с $A : \mathcal{U}$ и $B : A \to U$. Тогда имеем ассоциированный **полиномиальный функтор**:

$$P(X) = \sum_{x:A} (B(x) \to X)\,. \tag{5.4.6}$$

На самом деле это назначение функториально только с точностью до гомотопии, но это не имеет никакого значения для дальнейшего. По определению, P**-алгебра** — это тип C, снабженный функцией $s_C : PC \to C$. По универсальному свойству $\sum$-типов это эквивалентно заданию функции $\prod_{(a:A)}(B(a) \to C) \to C$. Будем называть такие объекты W**-алгебрами** для A и B и записывать

$$\mathsf{WAlg}(A, B) :\equiv \sum_{(C:\mathcal{U})} \prod_{(a:A)} (B(a) \to C) \to C.$$

Аналогично, для P-алгебр (C, s_C) и (D, s_D) **гомоморфизм** между ними $(f, s_f) : (C, s_C) \to (D, s_D)$ состоит из функции $f : C \to D$ и гомотопии между отображениями $PC \to D$

$$s_f : f \circ s_C = s_D \circ Pf\,,$$

где $Pf : PC \to PD$ является результатом просто определяемого действия P на $f : C \to D$. Такой гомоморфизм алгебры можно представить в виде:

$$\begin{tikzcd}
PC \arrow[r, "Pf"] \arrow[d, "s_C"'] \arrow[dr, phantom, "s_f"] & PD \arrow[d, "s_D"] \\
C \arrow[r, "f"'] & D
\end{tikzcd}$$

В терминах элементов f является P-гомоморфизмом (или W**-гомоморфизмом**), если

$$f(s_C(a, h)) = s_D(a, f \circ h)\,.$$

Мы имеем тип W-гомоморфизмов:

$$\mathsf{WHom}_{A,B}((C, s_C), (D, s_D)) :\equiv \sum_{(f:C\to D)} \prod_{(a:A)} \prod_{(h:B(a)\to C)} f(s_C(a, h)) = s_D(a, f \circ h)\,.$$

Наконец, P-алгебра (C, s_C) называется **гомотопически-инициальной**, если для любой P-алгебры (D, s_D), тип всех гомоморфизмов алгебр $(C, s_C) \to (D, s_D)$ является стягиваемым. То есть,

$$\mathsf{isHinit}_{\mathsf{W}}(A, B, I) :\equiv \prod_{C:\mathsf{WAlg}(A,B)} \mathsf{isContr}(\mathsf{WHom}_{A,B}(I, C)) .$$

Теперь можно привести аналог теоремы 5.4.5.

Теорема 5.4.7. *W-алгебра $(\mathsf{W}_{(x:A)}B(x), \mathsf{sup})$ является h-инициальной для любого типа $A : U$ и семейства типов $B : A \to \mathcal{U}$.*

Набросок доказательства. Предположим, что $A : \mathcal{U}$, $B : A \to \mathcal{U}$, и рассмотрим ассоциированный полиномиальный функтор $P(X) :\equiv \sum_{(x:A)}(B(x) \to X)$. Пусть $\mathsf{W} : \mathsf{W}_{(\mathrm{x:A})}\mathrm{B(x)}$. Тогда, используя правило W-введения из §5.3, имеем структурное отображение $s_W :\equiv \mathsf{sup} : P\mathsf{W} \to \mathsf{W}$. Мы хотим показать, что алгебра $(\mathsf{W}, s_{\mathsf{W}})$ является h-инициальной. Так что, рассмотрим другую алгебру (C, s_C) и покажем, что тип $T :\equiv \mathsf{WHom}_{A,B}((\mathsf{W}, s_{\mathsf{W}}), (C, s_C))$ W-гомоморфизмов из $(\mathsf{W}, s_{\mathsf{W}})$ в (C, s_C) является стягиваемым. Для этого заметим, что правило W-исключения и правило W-вычисления позволяют определить W-гомоморфизм $(f, s_f) : (\mathsf{W}, s_{\mathsf{W}}) \to (C, s_C)$, тем самым показывая, что T обитаем. Кроме того, необходимо показать, что для каждого W-гомоморфизма $(g, s_g) : (\mathsf{W}, s_{\mathsf{W}}) \to (C, s_C)$ существует доказательство тождественности

$$p : (f, s_f) = (g, s_g) . \tag{5.4.8}$$

Здесь используется тот факт, что тип формы $(f, s_f) = (g, s_g)$ в общем случае эквивалентен типу того, что мы называем **алгебрами 2-клеток** от f к g, канонические элементы которых являются парами формы (e, s_e), где $e : f = g$ и s_e — доказательство высшей тождественности между доказательствами тождественностей, представленными склеивающими диаграммами:

$$\begin{tikzcd}
PW \arrow[rr, bend left, "Pg"] \arrow[rr, bend right, "Pf"'] \arrow[rr, phantom, "Pe"] \arrow[d, "s_W"'] & & PC \arrow[d, "s_C"] \\
W \arrow[rr, bend right, "f"'] \arrow[urr, phantom, "s_f"] & & C
\end{tikzcd}
\qquad
\begin{tikzcd}
PW \arrow[rr, bend left, "Pg"] \arrow[d, "s_W"'] & & PC \arrow[d, "s_C"] \\
W \arrow[rr, bend left, "g"] \arrow[rr, bend right, "f"'] \arrow[rr, phantom, "e"] \arrow[urr, phantom, "s_g"] & & C
\end{tikzcd}$$

В свете этого факта, чтобы доказать, что существует элемент, как в (5.4.8), достаточно показать, что существует алгебра 2-клетки (e, s_e) от f к g. Теперь доказательство тождественности $e : f = g$ строится с помощью функциональной экстенсиональности и W-исключения, чтобы гарантировать существование требуемого доказательства тождественности s_e. □

5.5 Гомотопически-индуктивные типы

В §5.3 приведена кодировка натуральных чисел W-типами:

$$\begin{aligned}
\mathbf{N}^{\mathbf{w}} &:\equiv \mathsf{W}_{(b:\mathbf{2})}\mathsf{rec}_{\mathbf{2}}(\mathcal{U}, \mathbf{0}, \mathbf{1}, b) , \\
0^{\mathbf{w}} &:\equiv \mathsf{sup}(0_{\mathbf{2}}, (\lambda x.\, \mathsf{rec}_{\mathbf{0}}(\mathbf{N}^{\mathbf{w}}, x))) , \\
\mathsf{succ}^{\mathbf{w}} &:\equiv \lambda n.\, \mathsf{sup}(1_{\mathbf{2}}, (\lambda x.\, n)) .
\end{aligned}$$

Мы также показали, как можно определить функцию double на $\mathbf{N^w}$, используя принцип рекурсии. Однако, когда дело доходит до принципа индукции, это кодирование уже не является удовлетворительным: для $E : \mathbf{N^w} \to \mathcal{U}$ и рекуррентностей $e_z : E(0^{\mathbf{w}})$ и $e_s : \prod_{(n:\mathbf{N^w})} E(n) \to E(\mathsf{succ}^{\mathbf{w}}(n))$, мы можем построить только зависимую функцию $r(E, e_z, e_s) : \prod_{(n:\mathbf{N^w})} E(n)$, удовлетворяющую заданным рекурентностям *пропозиционально*, т.е. с точностью до пути. Это означает, что правила вычислений для натуральных чисел, которые дают дефиницальные равенства, не могут быть получены из правил для W-типов любым явным способом.

Эта проблема уходит, если вместо обычных индуктивных типов мы рассматриваем *гомотопически-индуктивные типы*, где все правила вычислений указываются с точностью до пути, т.е. символ $\equiv$ заменяется на $=$. Например, правилом вычисления для гомотопической версии W-типов W^h становится следующее:

- Для любых $a : A$ и $f : B(a) \to \mathsf{W}^h_{(x:A)} B(x)$ имеет место

$$\mathsf{rec}_{\mathsf{W}^h_{(x:A)} B(x)}(E, e, \mathsf{sup}(a, f)) = e\left(a, f, \left(\lambda b.\, \mathsf{rec}_{\mathsf{W}^h_{(x:A)} B(x)}(E, f(b))\right)\right).$$

Гомотопически-индуктивные типы имеют очевидный недостаток, когда речь идет о вычислительных свойствах — поведение любой функции, построенной с использованием принципа индукции, теперь можно охарактеризовать только пропозиционально. Но многие другие соображения побуждают нас также рассматривать гомотопически-индуктивные типы. Например, хотя в §5.4 мы показали, что индуктивные типы являются гомотопически-инициальными алгебрами, не всякая гомотопически-инициальная алгебра является индуктивным типом (т.е. удовлетворяет соответствующему принципу индукции), но всякая гомотопически-инициальная алгебра *является* гомотопически-индуктивным типом. Аналогичным образом, мы могли бы применить аргумент уникальности из §5.2, когда один или оба участвующих типов является только гомотопически-индуктивным типом — например, чтобы показать, что кодирование W-типа $\mathbb{N}$ эквивалентно обычному $\mathbb{N}$.

Кроме того, понятие гомотопически-индуктивного типа теперь является внутренним по отношению к теории типов. Например, это означает, что мы можем сформировать тип всех объектов натуральных чисел и сделать утверждения об этом. В случае W-типов мы можем охарактеризовать гомотопический W-тип $\mathsf{W}_{(x:A)} B(x)$ как какой-либо тип, наделенный функцией супремума и принципом индукции, удовлетворяющий правилу (пропозиционального) вычисления:

$$\begin{aligned}\mathsf{W}_d(A, B) :\equiv \sum_{(\mathsf{W}:\mathcal{U})} \; \sum_{(\mathsf{sup}:\prod_{(a)}(B(a)\to W)\to W)} \; \prod_{(E:\mathsf{W}\to\mathcal{U})} \\ \prod_{(e:\prod_{(a,f)}(\prod_{(b:B(a))} E(f(b)))\to E(\mathsf{sup}(a,f)))} \; \sum_{(\mathsf{ind}:\prod_{(w:\mathsf{W})} E(w))} \; \prod_{(a,f)} \\ \mathsf{ind}(\mathsf{sup}(a, f)) = e(a, \lambda b.\, \mathsf{ind}(f(b))).\end{aligned}$$

В главе 6 мы увидим некоторые другие причины, по которым заслуживают рассмотрения правила пропозициональных вычислений.

В этом разделе мы укажем некоторые основные факты о гомотопически-индуктивных типах. Мы опускаем большинство доказательств, которые носят несколько технический характер.

Теорема 5.5.1. *Для любых $A : \mathcal{U}$ и $B : A \to \mathcal{U}$ тип $\mathsf{W}_d(A, B)$ является простым высказыванием.*

Оказывается, что существует эквивалентная характеризация W-типов, используя принцип рекурсии, а также некоторые законы единственности и связности. Сначала представим принцип рекурсии:

- При построении функции из W-типа $\mathsf{W}^h_{(x:A)}B(x)$ в тип C достаточно указать ее для $\mathsf{sup}(a, f)$, считая, что нам даны значения всех $f(b)$ с $b : B(a)$. Другими словами, достаточно построить функцию

$$c : \prod_{a:A}(B(a) \to C) \to C\,.$$

Соответствующее правило вычисления для $\mathsf{rec}_{\mathsf{W}^h_{(x:A)}B(x)}(C, c) : (\mathsf{W}_{(x:A)}B(x)) \to C$ выглядит следующим образом:

- Для любых $a : A$ и $f : B(a) \to \mathsf{W}^h_{(x:A)}B(x)$ имеется свидетельство $\beta(C, c, a, f)$ для равенства

$$\mathsf{rec}_{\mathsf{W}^h_{(x:A)}B(x)}(C, c, \mathsf{sup}(a, f)) = c(a, \lambda b.\, \mathsf{rec}_{\mathsf{W}^h_{(x:A)}B(x)}(C, c, f(b)))\,.$$

Кроме того, мы заявляем следующий принцип единственности, говоря, что любые две функции, определяемые одной и той же рекурентностью, равны:

- Пусть заданы $C : \mathcal{U}$ и $c : \prod_{(a:A)}(B(a) \to C) \to C$. Пусть $g, h : (\mathsf{W}^h_{(x:A)}B(x)) \to C$ — функции, которые удовлетворяют рекуррентности c с точностью до пропозиционального равенства, т.е. такие, что имеет место

$$\beta_g \prod_{a,f} g(\mathsf{sup}(a, f)) = c(a, \lambda b.\, g(f(b)))\,,$$
$$\beta_h \prod_{a,f} h(\mathsf{sup}(a, f)) = c(a, \lambda b.\, h(f(b)))\,.$$

  Тогда g и h равны, т.е. существует $\alpha(C, c, f, g, \beta_g, \beta_h)$ типа $g = h$.

Напомним, что когда мы имеем принцип индукции в дополнение к принципу рекурсии, приведенный пропозициональный принцип единственности выводим (теорема 5.3.1). Но с одной рекурсией принцип единственности уже не выводится — и на самом деле последнее утверждение не является истинным (упражнение). Следовательно, мы постулируем его как аксиому. Мы также постулируем следующий закон связности, который говорит нам, как ведет себя доказательство единственности на канонических элементах:

- Для любых $a : A$ и $f : B(a) \to C$ следующая диаграмма пропозиционально коммутативна:

$$\begin{tikzcd}
g(\mathsf{sup}(a, f)) \arrow[r, "\beta_g"] \arrow[d, "\alpha(\mathsf{sup}(a,f))"'] & c(a, \lambda b.\, g(f(b))) \arrow[d, "c(a,_)(\mathsf{funext}(\lambda b.\, \alpha(f(b))))"] \\
h(\mathsf{sup}(a, f)) \arrow[r, "\beta_h"'] & c(a, \lambda b.\, h(f(b)))
\end{tikzcd}$$

  где α — аббревиатура пути $\alpha(C, c, f, g, \beta_g, \beta_h) : g = h$.

Объединение всех этих данных дает другую характеризацию $\mathsf{W}_{(x:A)}B(x)$, как типа с функцией супремума, удовлетворяющего правилам простого исключения, вычисления, единственности и связности:

$$\begin{aligned}\mathsf{W}_s(A,B) :\equiv \sum_{(\mathsf{W}:\mathcal{U})}\ \sum_{(\mathsf{sup}:\prod_{(a)}(B(a)\to\mathsf{W})\to\mathsf{W})}\ \prod_{(c:\prod_{(a)}(B(a)\to C)\to C}\ \sum_{(\mathsf{rec}:\mathsf{W}\to C)}\\ \sum_{(\beta:\prod_{(a,f)}\mathsf{rec}(\mathsf{sup}(a,f))=c(a,\lambda b.\,\mathsf{rec}(f(b))))}\ \prod_{(g:\mathsf{W}\to C)}\ \prod_{(h:\mathsf{W}\to C)}\ \prod_{(\beta_g:\prod_{(a,f)}g(\mathsf{sup}(a,f))=c(a,\lambda b.\,g(f(b))))}\\ \prod_{(\beta_h:\prod_{(a,f)}h(\mathsf{sup}(a,f))=c(a,\lambda b.\,h(f(b))))}\ \sum_{(\alpha:\prod_{(w:\mathsf{W})}g(w)=h(w))}\ \prod_{(a,f)}\\ \alpha(\mathsf{sup}(a,f))\cdot\beta_h = \beta_g\cdot c(a,-)(\mathsf{funext}\,\lambda b.\,\alpha(f(b)))\,.\end{aligned}$$

Теорема 5.5.2. *Для любых $A:\mathcal{U}$ и $B:A\to\mathcal{U}$ тип $\mathsf{W}_s(A,B)$ является простым высказыванием.*

Наконец, имеется третья, очень краткая характеризация $\mathsf{W}_{(x:A)}B(x)$ как h-инициальной W-алгебры:

$$\mathsf{W}_h(A,B) :\equiv \sum_{I:\mathsf{WAlg}(A,B)} \mathsf{isHinit}_{\mathsf{W}}(A,B,I)\,.$$

Теорема 5.5.3. *Для любых $A:\mathcal{U}$ и $B:A\to\mathcal{U}$ тип $\mathsf{W}_h(A,B)$ является простым высказыванием.*

Оказывается, что все три характеризации W-типов на самом деле эквивалентны:

Теорема 5.5.4. *Для любых $A:\mathcal{U}$ и $B:A\to\mathcal{U}$ имеет место*

$$\mathsf{W}_d(A,B)\simeq\mathsf{W}_s(A,B)\simeq\mathsf{W}_h(A,B)\,.$$

На самом деле, имеет место следующая теорема, являющаяся усовершенствованием теоремы 5.4.7:

Теорема 5.5.5. *Типы, удовлетворяющие правилам формирования, введения, исключения и пропозиционального вычисления для* W*-типов, являются в точности гомотопически-инициальными* W*-алгебрами.*

Набросок доказательства. Анализируя доказательство теоремы 5.4.7, мы видим, что для установления h-инициальности функции $\mathsf{W}_{(x:A)}B(x)$ требуется только правило пропозициального вычисления. Для обратной импликации предположим, что полиномиальный функтор, ассоциированный с $A:\mathcal{U}$ и $B:A\to\mathcal{U}$, имеет h-инициальную алгебру (W,s_W); мы покажем, что W удовлетворяет пропозициональным правилам W-типов. Правило W-введения простое; а именно, для $a:A$ и $t:B(a)\to W$ определим $sup(a,t):W$ в качестве результата применения структурного отображения $s_W:PW\to W$ к $(a,t):PW$. Для правила W-исключения допустим, что верны его посылки, в частности, $C':W\to\mathcal{U}$. Используя другие посылки, можно показать, что тип $C:\equiv\sum_{(w:W)}C'(w)$ может быть оснащен структурным отображением $s_C:PC\to C$. По h-инициальности W получаем гомоморфизм алгебры $(f,s_f):(W,s_W)\to(C,s_C)$. Кроме того, первая проекция $\mathsf{pr}_1 : C \to W$ может быть оснащена структурой гомоморфизма, так что получим диаграмму вида

$$\begin{tikzcd} PW \arrow[r, "Pf"] \arrow[d, "s_W"'] & PC \arrow[r, "P\mathsf{pr}_1"] \arrow[d, "s_C"'] & PW \arrow[d, "s_W"] \\ W \arrow[r, "f"'] & C \arrow[r, "\mathsf{pr}_1"'] & W\,. \end{tikzcd}$$

Но функция тождественности $1_W : W \to W$ имеет каноническую структуру гомоморфизма алгебры и, следовательно, по сократимости типа гомоморфизмов из (W, s_W) в себя, должно существовать доказательство тождественности между композицией (f, s_f) с $(\mathsf{pr}_1, s_{\mathsf{pr}_1})$ и $(1_W, s_{1_W})$. Это означает, в частности, что существует доказательство тождественности $p : \mathsf{pr}_1 \circ f = 1_W$.

Поскольку $(\mathsf{pr}_2 \circ f)w : C((\mathsf{pr}_1 \circ f)w)$, мы можем определить

$$\mathsf{rec}(w, c) :\equiv p_*((\mathsf{pr}_2 \circ f)w) : C(w)$$

где транспортировка p_* относится к семейству

$$\lambda u.\, C \circ u : (W \to W) \to W \to \mathcal{U}\,.$$

Проверка пропозиционального правила W-вычисления является вычислением, включающим свойства естественности операций вида p_*. $\square$

Наконец, при необходимости, можно кодировать гомотопически-натуральные числа как гомотопические W-типы:

Теорема 5.5.6. *Правила для натуральных чисел с правилами пропозициональных вычислений могут быть получены из правил для* W*-типов с пропозициональными правилами вычислений.*

5.6 Общий синтаксис индуктивных определений

До сих пор мы обсуждали только конкретные индуктивные типы: $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$, $\mathbb{N}$, копроизведения, произведения, $\sum$-типы, W-типы и т.д. Однако, важным аспектом теории типов является способность определять новые индуктивные типы, а не ограничиваться только определенным фиксированным списком из них. Чтобы иметь возможность сделать это, нужно знать, какие «индуктивные определения» являются обоснованными или допустимыми.

Чтобы убедиться, что не все, что «похоже на индуктивное определение» имеет смысл, рассмотрим следующий «конструктор» типа C:

- $g : (C \to \mathbb{N}) \to C$.

Принцип рекурсии для такого типа C должен указать, что для типа P, чтобы построить функцию $f : C \to P$, достаточно рассмотреть случай, когда вход $c : C$ имеет вид $g(\alpha)$ для некоторого $\alpha : C \to N$. Более того, мы ожидаем, что сможем каким-либо образом использовать «рекурсивные данные» f, примененные к α. Тем не менее, совсем не ясно, как «применить f к α», поскольку обе функции являются функциями с областью C.

Мы могли бы записать «принцип рекурсии» для C, просто предположив (необоснованно), что существует некоторый способ применения f к α и получить функцию $P \to \mathbb{N}$. Тогда вход в правило рекурсии запросит тип P вместе с функцией

$$h : (C \to \mathbb{N}) \to (P \to \mathbb{N}) \to P\,, \tag{5.6.1}$$

где два аргумента h — это α и «результат применения f к α». Однако, каково должно бы быть правило вычисления для результирующей функции $f : C \to P$? Рассматривая другие правила вычислений, мы ожидаем что-то вроде «$f(g(\alpha)) \equiv h(\alpha, f(\alpha))$» для $\alpha : C \to N$, но, как мы видели, «$f(\alpha)$» не имеет смысла. Принцип индукции для C еще более проблематичен; неясно, как записать гипотезы.

С другой стороны, мы могли бы записать другой «принцип рекурсии» для C, игнорируя «рекурсивное» присутствие C в области α, рассматривая его как просто тип индексации для семейства натуральных чисел. В этом случае вход будет запрашивать тип P вместе с функцией

$$h : (C \to \mathbb{N}) \to P\,,$$

поэтому тип принципа рекурсии должен быть $\mathsf{rec}_C : \prod_{(P:\mathcal{U})}((C \to \mathbb{N}) \to P) \to C \to P$, и аналогично для принципа индукции. Теперь можно записать правило вычисления, а именно $\mathsf{rec}_C(P, h, g(\alpha)) \equiv h(\alpha)$. Однако существование типа C с этим рекурсором и правилом вычисления оказывается непоследовательным. См. упражнения 5.7-5.10 для доказательства этого и других вариантов.

В этом примере предлагается одно ограничение на индуктивные определения: области всех конструкторов должны быть *ковариантными функторами* определяемого типа, поэтому мы можем «применить f к ним», чтобы получить результат «рекурсивного вызова». Другими словами, если мы заменим все вхождения типа, определяемого переменной $X : \mathcal{U}$, то каждая область конструктора должна быть выражением, которое может быть приготовлено в ковариантном функторе для X. Это справедливо для всех примеров, которые мы рассмотрели до сих пор. Например, с конструктором $\mathsf{inl} : A \to A + B$, соответствующий функтор является постоянным в A (т.е. $X \mapsto A$), а для конструктора $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$, функтором является тождественный функтор ($X \mapsto X$).

Однако, этого необходимого условия также недостаточно. Ковариантность предотвращает появление индуктивного типа слева от единичного типа функции, как в аргументе $C \to \mathbb{N}$ «конструктора» g, рассмотренного выше, так как это дает контравариантный функтор, а не ковариантный. Однако, поскольку композиция двух контравариантных функторов ковариантна, *двойные* функциональные типы, такие как $((X \to \mathbb{N}) \to \mathbb{N})$, ковариантны. Это позволяет нам воспроизводить парадоксы в стиле Кантора.

Например, рассмотрим «индуктивный тип» D со следующим конструктором:

- $k : ((D \to \mathsf{Prop}) \to \mathsf{Prop}) \to D$.

Предполагая, что такой тип существует, мы определяем функции

$$\begin{aligned} k &: D \to (D \to \mathsf{Prop}) \to \mathsf{Prop}\,, \\ f &: (D \to \mathsf{Prop}) \to D\,, \\ p &: (D \to \mathsf{Prop}) \to (D \to \mathsf{Prop}) \to \mathsf{Prop}\,, \end{aligned}$$

посредством

$$\begin{aligned} r(k(\theta)) &:\equiv \theta\,, \\ f(\delta) &:\equiv k(\lambda x.\,(x = \delta))\,, \\ p(\delta) &:\equiv \lambda x.\,\delta(f(x))\,. \end{aligned}$$

Здесь r определяется принципом рекурсии для D, а f и p определены явно. Тогда для любого $\delta : D \to \mathsf{Prop}$, имеем $r(f(\delta)) = \lambda x.\,(x = \delta)$.

В частности, поэтому, если $f(\delta) = f(\delta')$, то имеется путь $s : (\lambda x.\,(x = \delta)) = (\lambda x.\,(x = \delta'))$. Таким образом, $\mathsf{happly}(s, \delta) : (\delta = \delta) = (\delta = \delta')$, и поэтому в частности $\delta = \delta'$. Следовательно, f является «инъективным» (хотя априорно D не может быть множеством). Это уже звучит подозрительно — у нас есть «инъекция» «степенного множества» D в D — и небольшими усилиями мы можем растереть ее в противоречие.

Пусть задано $\theta : (D \to \mathsf{Prop}) \to \mathsf{Prop}$, определим $\delta : D \to \mathsf{Prop}$ по

$$\delta(d) :\equiv \exists(\gamma : D \to \mathsf{Prop}).\,(f(\gamma) = d) \times \theta(\gamma)\,. \tag{5.6.2}$$

Покажем, что $p(\delta) = \theta$. В силу функциональной экстенсиональности достаточно показать, что $p(\delta)(\gamma) =_{\mathsf{Prop}} \theta(\gamma)$ для любого $\gamma : D \to \mathsf{Prop}$. Основываясь на унивалентности, для этого достаточно показать, что каждый из них подразумевает другого. Теперь, по определению p имеем

$$\begin{aligned} p(\delta)(\gamma) &\equiv \delta(f(\gamma)) \\ &\equiv \exists(\gamma' : D \to \mathsf{Prop}).\,(f(\gamma') = f(\gamma)) \times \theta(\gamma')\,. \end{aligned}$$

Ясно, что это имеет место, если оно верно для $\theta(\gamma)$, так как мы можем положить $\gamma' :\equiv \gamma$. С другой стороны, если имеется γ' с $f(\gamma') = f(\gamma)$ и $\theta(\gamma')$, то $\gamma' = \gamma$, так как f инъективна, а значит, это относится и к $\theta(\gamma)$.

Это завершает доказательство того, что $p(\delta) = \theta$. Таким образом, каждый элемент $\theta : (D \to \mathsf{Prop}) \to \mathsf{Prop}$ — это образ под p некоторого элемента $\delta : D \to \mathsf{Prop}$. Однако, если мы определим θ классической диагонализацией:

$$\theta(\gamma) :\equiv \neg p(\gamma)(\gamma) \text{ для всех } \gamma : D \to \mathsf{Prop}$$

то из $\theta = p(\delta)$ выводим, что $p(\delta)(\delta) = \neg p(\delta)(\delta)$. Это противоречие: никакое предложение не может быть эквивалентно его отрицанию (предположим, что $P \Leftrightarrow \neg P$, если P, то $\neg P$ и, поэтому, $\mathbf{0}$; следовательно $\neg P$, но тогда P и, следовательно, $\mathbf{0}$).

Замечание 5.6.3. Возникает вопрос, касающийся размера универсума. В общем, индуктивный тип должен жить в универсуме, который уже содержит все типы, указанные в его определении. Таким образом, если в определении D двусмысленное обозначение Prop означает $\mathsf{Prop}_{\mathcal{U}}$, то мы не имеем $D : \mathcal{U}$, а только $D : \mathcal{U}'$ для некоторого большего универсума $\mathcal{U}'$ с $\mathcal{U} : \mathcal{U}'$. Поэтому, в предикативной теории правая часть (5.6.2) находится в $\mathsf{Prop}_{\mathcal{U}'}$, а не в $\mathsf{Prop}_{\mathcal{U}}$. Таким образом, это противоречие требует аксиомы пропозиционального изменения размера, упомянутой в §3.5.

Этот контрпример предполагает, что мы должны запретить индуктивный тип появляться слева от стрелки в области его конструкторов, даже если это появление вложено в другие стрелки, чтобы в итоге стать ковариантным (аналогично, ему также запрещается появляться в области зависимого типа функции). Это ограничение называется **строгой позитивностью** (обычная «позитивность» является по существу ковариацией), и этого оказывается достаточно.

В заключение, следовательно, допустимое индуктивное определение типа W состоит из списка *конструкторов*. Каждому конструктору присваивается тип, который является типом функции, принимающим некоторое число (возможно, нулевое) аргументов (возможно, зависящих друг от друга) и возвращающее элемент из W. Наконец, мы разрешаем W самому появляться во входных типах его конструкторов, но только строго позитивно. Это, по существу, означает, что каждый аргумент конструктора представляет собой либо тип, не содержащий W, либо некоторый итерированный тип функции с кообластью W. Например, следующее определение представляет собой допустимый тип конструктора:

$$c : (A \to W) \to (B \to C \to W) \to D \to W \to W\,. \tag{5.6.4}$$

Все эти типы функций также могут быть зависимыми функциями ($\prod$-типами)[1].

Заметим, мы требуем, чтобы индуктивное определение давалось *конечным* списком конструкторов. Это просто потому, что мы должны записать его на странице. Если же мы хотим, чтобы индуктивный тип вел себя так, как если бы он имел бесконечное число конструкторов, мы можем просто параметризовать один конструктор каким-то бесконечным типом. Например, конструктор, такой как $\mathbb{N} \to W \to W$ можно рассматривать как эквивалентный счетному числу конструкторов вида $W \to W$ (конечно, эта бесконечность теперь является *внутренней* по отношению к теории типов, как и для любой базовой системы). Точно так же, если нам нужен конструктор, который принимает «бесконечно много аргументов», мы можем позволить ему принять семейство аргументов, параметризованных некоторым бесконечным типом, например $(\mathbb{N} \to W) \to W$, который принимает бесконечную последовательность элементов из W.

Теперь, как только мы получим такое индуктивное определение, то что с ним можно делать? Во-первых, существует **принцип рекурсии**, который утверждает, что для определения функции $f : W \to P$, достаточно рассмотреть случай, когда вход $w : W$ возникает из одного из конструкторов, позволяя себе рекурсивно вызывать f на входах этого конструктора. Для примера конструктора (5.6.4) нам потребовалось бы, чтобы P был оснащен функцией типа

$$d : (A \to W) \to (A \to P) \to (B \to C \to W) \to (B \to C \to P) \to D \to W \to P \to P\,. \tag{5.6.5}$$

В соответствии с этими гипотезами принцип рекурсии дает $f : W \to P$, который, кроме того, «сохраняет данные конструктора» очевидным образом — это правило вычисления, в котором мы используем ковариацию входных данных. Например, в (5.6.4) правило вычисления гласит, что для любых $\alpha : A \to W$, $\beta : B \to C \to W$, $\delta : D$ и $\omega : W$, имеет место

$$f(c(\alpha, \beta, \delta, \omega)) \equiv d(\alpha, f \circ \alpha, \beta, f \circ \beta, \delta, \omega, f(\omega))\,. \tag{5.6.6}$$

Принцип индукции для общего индуктивного типа W лишь немного сложнее. Конечно, мы начинаем с семейства типов $P : W \to \mathcal{U}$, который требует наличия данных конструктора, «лежащих над» данными конструктора W. Это означает, что аргументы «рекурсивного вызова», такие как $A \to P$, упомянутые выше, должны быть заменены зависимыми функциями с такими типами, как $\prod_{(a:A)} P(\alpha(a))$. В полном примере (5.6.4) соответствующая гипотеза для принципа индукции потребует

$$\begin{multline} d : \prod_{\alpha : A \to W} \left(\prod_{a:A} P(\alpha(a)) \right) \to \prod_{\beta : B \to C \to W} \left(\prod_{(b:B)} \prod_{(c:C)} P(\beta(b,c)) \right) \to \\ \prod_{(\delta:D)} \prod_{(\omega:W)} P(\omega) \to P(c(\alpha, \beta, \delta, \omega)). \end{multline} \tag{5.6.7}$$

Соответствующее правило вычислений выглядит идентично (5.6.6). Конечно, принцип рекурсии — частный случай принципа индукции, с P, являющимся постоянным семейством. Как уже упоминалось ранее, принцип индукции также называется **выделителем**, а принцип рекурсии **— независимым выделителем**.

[1]На языке §5.4 условие строгой позитивности гарантирует, что соответствующий эндофунктор является полиномиальным. В теории категорий хорошо известно, что *не все* эндофункторы могут иметь инициальные алгебры; ограничение до полиномиальных функторов обеспечивает согласованность. Можно рассмотреть различные ослабления этого условия, но в этой книге мы ограничимся строгой позитивностью, определенной выше.

Как обсуждалось в §1.10, мы также позволим себе неявно ссылаться на принципы индукции и рекурсии, записывая дефинициальное уравнение с $:\equiv$ для каждого выражения, которое являлось бы гипотезой принципа индукции. Это называется предоставлением определения посредством (зависимого) **сопоставления с образцом**. В нашем текущем примере это означает, что мы можем определить $f : \prod_{(w:W)} P(w)$ через

$$f(c(\alpha, \beta, \delta, \omega)) :\equiv \cdots$$

где $\alpha : A \to W$, $\beta : B \to C \to W$, $\delta : D$ и $\omega : W$ — переменные, которые являются связанными в правой части. Более того, правая часть может включать в себя рекурсивные обращения к f вида $f(\alpha(a))$, $f(\beta(b, c))$ и $f(\omega)$. Когда это определение переупаковывается в терминах принципа индукции, мы заменяем такие рекурсивные вызовы на $\bar{\alpha}(a)$, $\bar{\beta}(b, c)$ и $\bar{\omega}$, соответственно, для новых переменных

$$\begin{aligned}
\bar{\alpha} &: \prod_{a:A} P(\alpha(a)),\\
\bar{\beta} &: \prod_{(b:B)} \prod_{(c:C)} P(\beta(b, c)),\\
\bar{\omega} &: P(\omega).
\end{aligned}$$

Тогда можно записать

$$f :\equiv \mathsf{ind}_W(P, \lambda\alpha.\, \lambda\bar{\alpha}.\, \lambda\beta.\, \lambda\bar{\beta}.\, \lambda\delta.\, \lambda\omega.\, \lambda\bar{\omega}.\, \cdots)$$

где второй аргумент ind_W имеет вид (5.6.7).

Мы не будем пытаться дать формальное представление грамматики правильного индуктивного определения, его результирующих принципов индукции и рекурсии и правил сопоставления с образцом. Это можно сделать (на самом деле, это необходимо сделать, если вы используете систему компьютерных доказательств), но это не дает дополнительной информации. Работая практически, каждый учится машинально выводить принципы индукции и рекурсии для любого индуктивного определения и использовать их, не задумываясь.

5.7 Обобщения индуктивных типов

Понятие индуктивного типа изучалось в теории типов на протяжении многих лет и допускает множество обобщений: семейства индуктивных типов, взаимные индуктивные типы, индуктивно-индуктивные типы, индуктивно-рекурсивные типы и т.д. В этом разделе приводится обзор некоторых из них, часть из которых будет использована в книге позже (в главе 6 мы будем более подробно изучать значительно отличающееся от других обобщение индуктивных типов, которое имеет особое отношение к *гомотопической* теории типов).

Большинство из этих обобщений включают в себя возможность одновременного определения более одного типа индукцией. Очень простым примером этого, с чем мы уже сталкивались, является копроизведение $A + B$. Было бы утомительно записывать отдельные индуктивные определения для $\mathbb{N} + \mathbb{N}$, $\mathbb{N} + \mathbf{2}$, $\mathbf{2} + \mathbf{2}$ и т.д., каждый раз, когда требуется рассмотреть копроизведение двух типов. Вместо этого можно создать одно определение, в котором A и B являются переменными, обозначающими типы; в теории типов они называются **параметрами**. Таким образом, технически говоря, результаты, полученные в результате определения, являются не одним типом, а семейством типов, например, как в случае $+ : \mathcal{U} \to \mathcal{U} \to \mathcal{U}$, когда используются два типа в качестве входных данных и производится их копроизведение. Аналогичным образом, тип $\mathsf{List}(A)$ списков является семейством $\mathsf{List}(\_) : \mathcal{U} \to \mathcal{U}$, в котором тип A является параметром.

В математике подобное настолько очевидно, что об этом и не стоит упоминать, но мы это делаем, в порядке контраста следующему примеру. Обратите внимание, что каждый тип $A + B$ определяется индуктивно *независимо* от других, как и каждый тип $\mathsf{List}(A)$. В отличие от этого, мы могли бы также рассмотреть определение целого семейства типов $B : A \to \mathcal{U}$ путем *совместной* индукции. Разница в том, что теперь конструкторы могут изменять индекс $a : A$, и, как следствие, мы не можем сказать, что индивидуальные типы $B(a)$ индуктивно определены, а только то, что индуктивно определено все семейство.

Стандартный пример — это тип *списков фиксированного размера*, традиционно называемых **векторами**. Фиксируем тип параметра A и определяем семейство типов $\mathsf{Vec}_n(A)$, для $n : \mathbb{N}$, сгенерированное следующими конструкторами:

- вектором $\mathsf{nil} : \mathsf{Vec}_0(A)$ нулевой длины,
- функцией $\mathsf{cons} : \prod_{(n:\mathbb{N})} A \to \mathsf{Vec}_n(A) \to \mathsf{Vec}_{\mathsf{succ}(n)}(A)$.

В отличие от списков векторы (с элементами фиксированного типа A) образуют семейство типов, индексированных по их длине. Пока A является параметром, мы говорим, что $n : \mathbb{N}$ — **индекс** индуктивного семейства. Индивидуальный тип, такой как $\mathsf{Vec}_3(A)$, не определяется индуктивно: конструкторы, которые строят элементы $\mathsf{Vec}_3(A)$, вводят из другого типа в семейство, например $\mathsf{cons} : A \to \mathsf{Vec}_2(A) \to \mathsf{Vec}_3(A)$.

В частности, принцип индукции также должен относиться и ко всему семейству типов; таким образом, гипотезы и заключение должны количественно оценивать индексы соответствующим образом. В случае векторов принцип индукции утверждает, что для семейства типов $C : \prod_{(n:\mathbb{N})} \mathsf{Vec}_n(A) \to \mathcal{U}$, вместе с

- элементом $c_{\mathsf{nil}} : C(0, \mathsf{nil})$ и
- функцией $c_{cons} : \prod_{(n:\mathbb{N})} \prod_{(a:A)} \prod_{(\ell:\mathsf{Vec}_n(A))} C(n, \ell) \to C(\mathsf{succ}(n), \mathsf{cons}(a, \ell))$

существует функция $f : \prod_{(n:\mathbb{N})} \prod_{(\ell:\mathsf{Vec}_n(A))} C(n, \ell)$ такая, что

$$f(0, \mathsf{nil}) \equiv c_{\mathsf{nil}}$$
$$f(\mathsf{succ}(n), \mathsf{cons}(a, \ell)) \equiv c_{cons}(n, a, \ell, f(\ell)) \,.$$

Одним из использований индуктивных семейств является индуктивное определение *предикатов*. Например, мы можем определить предикат $\mathsf{iseven} : \mathbb{N} \to \mathcal{U}$ как индуктивное семейство, индексированное посредством $\mathbb{N}$, со следующими конструкторами:

- элементом $\mathsf{even}_0 : \mathsf{iseven}(0)$,
- функцией $\mathsf{even}_{ss} : \prod_{(n:\mathbb{N})} \mathsf{iseven}(n) \to \mathsf{iseven}(\mathsf{succ}(\mathsf{succ}(n)))$.

Другими словами, мы оговариваем, что 0 четно, и что, если n четно, то таково будет и $\mathsf{succ}(\mathsf{succ}(n))$. Эти конструкторы «явно» не дают возможности построить элемент, скажем, $\mathsf{iseven}(1)$, и, поскольку iseven, как предполагается, свободно порождается введенными конструкторами, такого элемента не должно быть (однако, фактически доказано, что: $\neg\mathsf{iseven}(1)$ не является тривиальным). Принцип индукции для iseven утверждает, что для доказательства чего-либо о всех четных натуральных числах достаточно доказать его для 0 и проверить, что он сохраняется при добавлении 2.

Индуктивно определяемые предикаты широко используются в компьютерной формализации математики и проверки программного обеспечения. Но от них у нас будет мало пользы, с несколькими исключениями из §§ 10.3 и 11.5.

Другим важным частным случаем является тот, когда индексный тип индуктивного семейства конечен. В этом случае мы можем эквивалентно выразить индуктивное определение конечным набором типов, определяемых *взаимной индукцией*. Например, мы можем определить типы even и odd четных и нечетных натуральных чисел взаимной индукцией, где even порождается конструкторами

- $0 : \mathsf{even}$ и
- $\mathsf{esucc} : \mathsf{odd} \to \mathsf{even}$,

а odd генерируется одним конструктором

- $\mathsf{osucc} : \mathsf{even} \to \mathsf{odd}$.

Обратите внимание, что even и odd — это простые типы (не семейства типов), но их конструкторы могут ссылаться друг на друга. Если бы мы выразили это определение как индуктивный тип семейства $\mathsf{paritynat} : \mathbf{2} \to \mathcal{U}$, с $\mathsf{paritynat}(0_{\mathbf{2}})$ и $\mathsf{paritynat}(1_{\mathbf{2}})$, представляющими even и odd соответственно, то они имели бы конструкторы:

- $0 : \mathsf{paritynat}(0_{\mathbf{2}})$,
- $\mathsf{esucc} : \mathsf{paritynat}(1_{\mathbf{2}}) \to \mathsf{paritynat}(0_{\mathbf{2}})$,
- $\mathsf{osucc} : \mathsf{paritynat}(0_{\mathbf{2}}) \to \mathsf{paritynat}(1_{\mathbf{2}})$.

При явном выражении в качестве взаимного индуктивного определения, принцип индукции для even и odd говорит о том, что для $C : \mathsf{even} \to \mathcal{U}$ и $D : \mathsf{odd} \to \mathcal{U}$, вместе с

- $c_0 : C(0)$,
- $c_s : \prod_{(n:\mathsf{odd})} D(n) \to C(\mathsf{esucc}(n))$,
- $d_s : \prod_{(n:\mathsf{even})} C(n) \to D(\mathsf{osucc}(n))$,

существуют функции $f : \prod_{(n:\mathsf{even})} C(n)$ и $g : \prod_{(n:\mathsf{odd})} D(n)$ такие, что

$$\begin{aligned} f(0) &\equiv c_0 \\ f(\mathsf{esucc}(n)) &\equiv c_s(g(n)) \\ g(\mathsf{osucc}(n)) &\equiv d_s(f(n)) \,. \end{aligned}$$

В частности, так же, как мы можем действовать на индуктивное семейство «разом», мы должны действовать на even и odd одновременно. В этой книге мы не будем пользоваться большим количеством взаимных индуктивных определений.

Дальнейшее, более радикальное обобщение состоит в том, чтобы разрешить определение семейства типов $B : A \to \mathcal{U}$, в котором не только типы $B(a)$, но и сам тип A, определяются как часть одной большой индукции. Другими словами, мы не только указываем конструкторы для всех $B(a)$, которые могут принимать входные данные от других $B(a')$, как в индуктивных семействах, но и указываем конструкторы для самого A, которые могут принимать входы от разных $B(a)$. Это можно рассматривать как индуктивное семейство, в котором индексы индуктивно определяются одновременно с индексированными типами или как взаимное индуктивное определение, в котором один из типов может зависеть от остальных. Возможны также более сложные структуры зависимостей. В общем, они называются **индуктивно-индуктивными определениями**. По большей части мы не будем использовать их в этой книге, но их высшая форма (см. главу 6) появится в нескольких экспериментальных примерах в главе 11.

Последнее обобщение, о котором мы хотим упомянуть, — это **индуктивно-рекурсивные определения**, в которых тип определяется индуктивно одновременно с рекурсивной функцией на нем. То есть мы фиксируем известный тип P, задаем конструкторы индуктивного типа A и в то же время определяем функцию $f : A \to P$, используя принцип рекурсии для A, полученный из его конструкторов, — с той храктерной особенностью, что конструкторам для A разрешено ссылаться также на значения f. Мы еще не знаем, как выразить такие определения с гомотопической точки зрения, и мы не будем использовать их в этой книге.

5.8 Типы тождественности и системы тождественности

Теперь мы хотим акцентировать внимание на том, что *типы тождественности*, которые на самом деле играют центральную роль в гомотопической теории типов, также можно определять индуктивно. В частности, они являются «индуктивным семейством» с индексами в смысле §5.7. На самом деле, существует *два* способа описания типов тождественности как индуктивного семейства, в связи с чем в главе 1 описаны два принципа индукции, индукция пути и базированная индукция пути.

В обоих определениях тип A является параметром. Для первого определения мы индуктивно определяем семейство $=_A$: $A \to A \to \mathcal{U}$ с двумя индексами, принадлежащими A, следующим конструктором:

- для любого $a : A$, элементом $\mathsf{refl}_a : a =_A a$.

По аналогии с другими индуктивными семействами мы можем извлечь из этого определения принцип индукции. Он утверждает, что при любом $C : \prod_{(a,b:A)}(a =_A b) \to \mathcal{U}$ вместе с $d : \prod_{(a:A)} C(a, a, \mathsf{refl}_a)$, существует функция $f : \prod_{(a,b:A)} \prod_{(p:a=_A b)} C(a, b, p)$ такая, что $f(a, a, \mathsf{refl}_a) \equiv d(a)$. Это как раз принцип индукции пути для типов тождественности.

Для второго определения мы рассматриваем один элемент $a_0 : A$ как параметр, вместе с $A : \mathcal{U}$, и индуктивно определяем семейство $(a_0 =_A \_) : A \to \mathcal{U}$ с *одним* индексом, принадлежащим A, следующим конструктором:

- элементом $\mathsf{refl}_{a_0} : a_0 =_A a_0$.

Обратите внимание, что поскольку $a_0 : A$ был зафиксирован как параметр, конструктор refl_{a_0} появляется внутри индуктивного определения не как функция, а только как элемент. Принцип индукции для этого определения гласит, что для $C : \prod_{(b:A)}(a_0 =_A b) \to U$ вместе с элементом $d : C(a_0, \mathsf{refl}_{a_0})$, существует $f : \prod_{(b:A)} \prod_{(p:a_0=_A b)} C(b, p)$ с $f(a_0, \mathsf{refl}_{a_0}) \equiv d$. Это есть именно принцип базированной индукции пути для типов тождественности.

Точка зрения на типы тождественности как индуктивные типы исторически вызвал некоторую путаницу из-за интуиции, упомянутой в §5.1, что все элементы индуктивного типа должны быть получены путем многократного применения его конструкторов. Для обычных индуктивных типов, таких как $\mathbf{2}$ и $\mathbb{N}$, это так: мы видели в уравнении (1.8.1), что действительно каждый элемент из $\mathbf{2}$ является либо $0_{\mathbf{2}}$, либо $1_{\mathbf{2}}$, и аналогично можно доказать, что каждый элемент из $\mathbb{N}$ равен либо 0, либо преемнику.

Однако это *не* относится к типам тождественности: существует только один конструктор refl, но не каждый путь равен постоянному пути. Точнее, мы не можем доказать, используя только принцип индукции для типов тождественности (или единицы), что каждый обитатель из $a =_A a$ равен refl_a. Для того, чтобы фактически продемонстрировать контрпример, нам нужен какой-то дополнительный принцип, такой как аксиома унивалентности — напомним, что в примере 3.1.9 мы использовали унивалентность, чтобы указать конкретный путь $\mathbf{2} =_{\mathcal{U}} \mathbf{2}$, не равный $\mathsf{refl}_{\mathbf{2}}$.

Дело в том, что, как проверено изучением гомотопически-инициальных алгебр, индуктивное определение следует рассматривать как *свободно порожденное* его конструкторами. Разумеется, свободно сгенерированная структура может содержать элементы, отличные от своих образующих: например, свободная группа на двух символах x и y содержит не только x и y, но также такие слова, как xy, $yx^{-1}y$, и $\mathrm{x}^3y^2x^{-2}\mathrm{y}x$. В общем, элементы свободной структуры получаются путем применения не только образующих, но и операций внешней структуры, таких как групповые операции, если речь идет о свободных группах.

В случае индуктивных типов мы говорим о свободно сгенерированных *типах* — так что же из себя представляют «операции» структуры типа? Если типы рассматриваются как *множества*, как это традиционно имеет место в теории типов, то таких операций нет, и поэтому мы ожидаем, что в индуктивном типе нет элементов, кроме тех, которые возникают в его конструкторах. В гомотопической теории типов мы рассматриваем типы как *пространства* или ∞-группоиды, и в этом случае существует много операций на *путях* (конкатенация, инверсия и т.д.) — это будет важно в главе 6 — но, по-прежнему, здесь отсутствуют операции на *объектах* (элементах). Таким образом, для нас по-прежнему верно, что, например, каждый элемент из $\mathbf{2}$ является либо $0_{\mathbf{2}}$, либо $1_{\mathbf{2}}$, а каждый элемент из $\mathbb{N}$ является либо 0, либо преемником.

Однако, как мы видели в главе 2, рассмотрение типов как ∞-группоидов влечет за собой также рассмотрение функций как функторов, а это включает семейства типов $B : A \to \mathcal{U}$. Таким образом, тип тождественности $(a_0 =_A \_)$, рассматриваемый как семейство индуктивных типов, на самом деле является *свободно порожденным функтором* $A \to \mathcal{U}$. В частности, это функтор $F : A \to \mathcal{U}$, свободно порожденный одним элементом $\mathsf{refl}_{a_0} : F(a_0)$. А функтор действительно имеет операции над объектами — это действия морфизмов (путей) из A.

В теории категорий *лемма Йонеды* говорит, что для любой категории A и объекта a_0 функтор, свободно порожденный элементом из $F(a_0)$, является представимым $hom_A A(a_0, \_)$. Таким образом, мы должны ожидать, что тип тождественности $(a_0 =_A \_)$ будет этим представимым функтором, и это действительно так, поскольку $(a_0 =_A b)$ — пространство морфизмов (путей) в A от a_0 к b.

Одной из причин рассмотрения типов тождественности как индуктивных семейств является применение принципов уникальности из §5.2 и §5.5. В частности, мы можем охарактеризовать семейство типов тождественности типа A с точностью до эквивалентности, предоставив другое семейство типов над $A \times A$, удовлетворяющих одному и тому же принципу индукции. Это предполагает следующие определения и теорему.

Определение 5.8.1. Пусть A — тип, а $a_0 : A$ — элемент.

- **Точечный предикат** над (A, a_0) является семейством $R : A \to \mathcal{U}$, снабженный элементом $r_0 : R(a_0)$.

- Для точечных предикатов (R, r_0) и (S, s_0) семейство отображений $g : \prod_{(b:A)} R(b) \to S(b)$ является **точечным**, если $g(a_0, r_0) = s_0$. Имеет место

$$\mathsf{ppmap}(R, S) :\equiv \sum_{g:\prod_{(b:A)} R(b)\to S(b)} (g(a_0, r_0) = s_0)\,.$$

- **Система тождественности при** a_0 является точечным предикатом (R, r_0) таким, что для любого семейства типов $D : \prod_{(b:A)} R(b) \to \mathcal{U}$ и $d : D(a_0, r_0)$ существует функция $f : \prod_{(b:A)} \prod_{(r:R(b))} D(b, r)$ такая, что $f(a_0, r_0) = d$.

Теорема 5.8.2. *Для точечного предиката (R, r_0) над (A, a_0) следующие высказывания логически эквивалентны.*

(i) *(R, r_0) является системой тождественности при a_0.*

(ii) *Для любого точечного предиката (S, s_0) тип $\mathsf{ppmap}(R, S)$ является стягиваемым.*

(iii) *Для любого $b : A$ функция $\mathsf{transport}^R(\_, r_0) : (a_0 =_A b) \to R(b)$ является эквивалентностью.*

(iv) *Тип $\prod_{(b:A)} R(b)$ является стягиваемым.*

Заметим, что эквивалентности $(i) \Leftrightarrow (ii) \Leftrightarrow (iii)$ являются версией леммы 5.5.4 для типов тождественности $a_0 =_A \_$, рассматриваемых как индуктивные семейства, различающиеся при одном элементе из A. Конечно, (ii)-(iv) являются простыми высказываниями, так что логическая эквивалентность подразумевает фактическую эквивалентность (условие (i) также является простым высказыванием, но мы этого не будем доказывать). Отметим также, что в отличие от (i)-(iii) высказывание (iv) не ссылается на a_0 или r_0.

Доказательство. Во-первых, предположим выполнение (i) и пусть (S, s_0) — точечный предикат. Определим $D(b, r) :\equiv S(b)$ и $d :\equiv s_0 : S(a_0) \equiv D(a_0, r_0)$. Так как R — система тождественности, имеем $f : \prod_{(b:A)} R(b) \to S(b)$ с $f(a_0, r_0) = s_0$; следовательно, $\mathsf{ppmap}(R, S)$ обитаем. Теперь предположим, что $(f, f_r), (g, g_r) : \mathsf{ppmap}(R, S)$, определим $D(b, r) :\equiv (f(b, r) = g(b, r))$ и пусть $d : f_r \cdot g_r^{-1} : f(a_0, r_0) = s_0 = g(a_0, r_0)$. Тогда снова, так как R — система тождественности, имеем $h : \prod_{(b:A)} \prod_{(r:R(b))} D(b, r)$ такое, что $h(a_0, r_0) = f_r \cdot \mathrm{r}_r^{-1}$. По характеризации путей в $\sum$-типах и типах путей эти данные обеспечивают равенство $(f, f_r) = (g, g_r)$. Следовательно, $\mathsf{ppmap}(R, S)$ является обитаемым простым высказыванием и, следовательно, стягиваемым; так что выполняется (ii).

Предположим теперь выполняемость (ii) и определим $S(b) :\equiv (a_0 = b)$ с $s_0 :\equiv \mathsf{refl}_{a_0} : S(A_0)$. Тогда (S, s_0) является точечным предикатом, а $\lambda b.\, \lambda p.\, \mathsf{transport}^R(p, r) : \prod_{(b:A)} S(b) \to R(b)$ — точечное семейство отображений от S к R. По условию $\mathsf{ppmap}(R, S)$ является стягиваемым, следовательно, обитаемым, поэтому существует также точечное семейство отображений от R к S. И композиции в любом направлении представляют собой точечные семейства отображений от R к R и от S к S соответственно, следовательно, равны тождествам, так как $\mathsf{ppmap}(R, R)$ и $\mathsf{ppmap}(S, S)$ являются стягиваемыми. Таким образом, (iii) выполняется.

При предполажении выполнимости (iii), условие (iv) следует из леммы 3.11.8, используя то, что $\sum$-типы сохраняют эквивалентности (направление «если» теоремы 4.7.7).

Наконец, предположим (iv) и пусть $D : \prod_{(b:A)} R(b) \to \mathcal{U}$ и $d : D(a_0, r_0)$. Мы можем эквивалентно выразить D как семейство $D' : (\sum_{(b:A)} R(b)) \to \mathcal{U}$. Теперь, поскольку $\sum_{(b:A)} R(b)$ стягиваемо, имеем

$$p : \prod_{u:\sum_{(b:A)} R(b)} (a_0, r_0) = u\,.$$

Более того, поскольку типы путей стягиваемого типа являются опять стягиваемыми, имеем $p((a_0, r_0)) = \mathsf{refl}_{(a_0,r_0)}$. Определим $f(u) :\equiv \mathsf{transport}^{D'}(p(u), d)$, что дает $f : \prod_{(u:\sum_{(b:A)} R(b))} D'(u)$ или, что то же самое, $f : \prod_{(b:A)} \prod_{(r:R(b))} D(b, r)$. Наконец, имеем

$$f(a_0, r_0) \equiv \mathsf{transport}^{D'}(p((a_0, r_0)), d) = \mathsf{transport}^{D'}(\mathsf{refl}_{(a_0,r_0)}, d) = d\,.$$

Таким образом, выполняется (i). □

Мы можем вывести аналогичный результат для типов тождественности $=_A$, рассматриваемых как семейство, различающееся по двум элементам из A.

Определение 5.8.3. Система тождественности над типом A — это семейство $R : A \to A \to \mathcal{U}$ вместе с функцией $r_0 : \prod_{(a:A)} R(a, a)$, такая, что для любого семейства типов $D : \prod_{(a,b:A)} R(a, b) \to \mathcal{U}$ и $d : \prod_{(a:A)} D(a, a, r_0(a))$ существует функция $f : \prod_{(a,b:A)} \prod_{(r:R(a,b))} D(a, b, r)$ такая, что для всех $a : A$, $f(a, a, r_0(a)) = d(a)$.

Теорема 5.8.4. *Для $R : A \to A \to \mathcal{U}$ с $r_0 : \prod_{(a:A)} R(a, a)$, следующие высказывания логически эквивалентны.*

(i) *(R, r_0) является системой тождественности над A.*

(ii) *Для всех $a_0 : A$, точечный предикат $(R(a_0), r_0(a_0))$ является системой тождественности при a_0.*

(iii) *Для любого $S : A \to A \to \mathcal{U}$ и $s_0 : \prod_{(a:A)} S(a, a)$ тип*

$$\sum_{(g:\prod_{(a,b:A)} R(a,b)\to S(a,b))} \prod_{(a:A)} g(a, a, r_0(a)) = s_0(a)$$

является стягиваемым.

(iv) *Для любых $a, b : A$, отображение $\mathsf{transport}^{R(a)}(\_, r_0(a)) : (a =_A b) \to R(a, b)$ является эквивалентностью.*

(v) *Для любого $a : A$, тип $\sum_{(b:A)} R(a, b)$ является стягиваемым.*

Доказательство. Эквивалентность (i), (ii) точно соответствует доказательству эквивалентности между принципами индукции пути и базированной индукции пути для типов тождественности; см. §1.12. Тогда эквивалентность (iv) и (v) следует из теоремы 5.8.2, а (iii) прямолинейно. □

Одна из причин, по которой эта характеризация интересна, заключается в том, что она обеспечивает альтернативный способ подтверждения унивалентности и функциональной экстенсиональности. Аксиома унивалентности для универсума $\mathcal{U}$ как раз говорит о том, что семейство типов

$$(\_ \simeq \_) : \mathcal{U} \to \mathcal{U} \to \mathcal{U}$$

вместе с $\mathsf{id} : \prod_{(A:\mathcal{U})}(A \simeq A)$ удовлетворяет теореме 5.8.4(iv). Поэтому это эквивалентно соответствующей версии (i), которую мы можем сформулировать следующим образом.

Следствие 5.8.5 (Индукция эквивалентности)**.** *Для любого семейства типов* $D : \prod_{(A,B:\mathcal{U})}(A \simeq B)$ *и функции* $d : \prod_{(A:\mathcal{U})} D(A, A, \mathsf{id}_A)$ *существует функция* $f : \prod_{(A,B:U)} \prod_{(e:A\simeq B)} D(A, B, e)$ *такая, что* $f(A, A, \mathsf{id}_A) = d(A)$ *для всех* $A : \mathcal{U}$.

Другими словами, чтобы доказать что-либо обо всех эквивалентностях, достаточно доказать это об отображениях тождественности. Мы уже использовали этот принцип (не указывая его в общности) в лемме 4.1.1.

Аналогично, функциональная экстенсиональность означает, что для любого $B : A \to \mathcal{U}$, семейство типов

$$(\_ \simeq \_) : \left(\prod_{a:A} B(a)\right) \to \left(\prod_{a:A} B(a)\right) \to \mathcal{U}$$

вместе с $\lambda f.\, \lambda a.\, \mathsf{refl}_{f(a)}$ удовлетворяет теореме 5.8.4(iv). Таким образом, это также эквивалентно соответствующей версии (i).

Следствие 5.8.6 (Гомотопическая индукция)**.** *Для любого* $D : \prod_{(f,g:\prod_{(a:A)} B(a))}(f \sim g) \to \mathcal{U}$ и $d : \prod_{(f:\prod_{(a:A)} B(a))} D(f, f, \lambda x.\, \mathsf{refl}_{f(x)})$ *существует*

$$k : \prod_{(f,g:\prod_{(a:A)} B(a))} \prod_{(h:f\sim g)} D(f, g, h)$$

такая, что $k(f, f, \lambda x.\, \mathsf{refl}_{f(x)}) = d(f)$ *для всех* f.

Примечания

Индуктивные определения имеют длинную родословную в математике, возможно, начиная хотя бы с аксиом Фреге (Frege) и Пеано (Peano) для натуральных чисел. Более общие «индуктивные предикаты» не являются редкостью, но в теоретико-множественных основах они обычно строятся явно, либо как пересечение соответствующего класса подмножеств, либо с использованием трансфинитной итерации вдоль ординалов, а не рассматриваются как основное понятие.

В теории типов частные случаи индуктивных определений относятся к оригинальным работам Мартина-Лёфа (Martin-Löf): [ML71] представляет общее понятие индуктивно определенных предикатов и отношений; понятие индуктивного типа присутствовало (но только с примерами, а не как общее понятие) в первых статьях Мартина-Лёфа по теории типов [ML75]; а затем как общее понятие с W-типами в [ML82].

Общее понятие индуктивного типа было введено в 1985 году Констеблем (Constable) и Мендлером (Mendler) [CM85]. В [PPM90] была предложена общая схема индуктивных типов в теории интенсионального типа. Дальнейшие разработки включены в [CP90, Dyb91].

Понятие индуктивно-рекурсивного определения появляется в [Dyb00]. Важным теоретико-множественным понятием является понятие типов деревьев (общее выражение понятия Post-системы в теории типов), которое фигурирует в [PS89].

Универсальное свойство натуральных чисел как инициального объекта категории $\mathbb{N}$-алгебр принадлежит Ловеру (Lawvere) [Law06]. Позднее оно было обобщено на описание W-типов в качестве инициальных алгебр для полиномиальных эндофункторов согласно [MP00]. Теоретико-гомотопическая эквивалентность между такими универсальными свойствами и соответствующими принципами индукции (§§ 5.4 и 5.5) последовательно рассмотрена в [AGS12].

О реальных конструкциях индуктивных типов в теоретико-гомотопической семантике теории типов см. в [KLV12, MvdB13, LS17].

Упражнения

Упражнение 5.1. Вывести принцип индукции для типа $\mathsf{List}(A)$ списков из его определения как индуктивного типа в §5.1.

Упражнение 5.2. Постройте две функции на натуральных числах, которые удовлетворяют одной и той же рекуррентности (e_z, e_s) дефинициально, но не дефинициально равнозначно.

Упражнение 5.3. Постройте две разные рекуррентности (e_z, e_s) на одном и том же типе E, которые одновременно дефинициально соответствуют одной и той же функции $f : \mathbb{N} \to E$.

Упражнение 5.4. Покажите, что для любого семейства типов $E : \mathbf{2} \to \mathcal{U}$, оператор индукции

$$\mathsf{ind}_{\mathbf{2}}(E) : (E(0_{\mathbf{2}}) \times E(1_{\mathbf{2}})) \to \prod_{b:\mathbf{2}} E(b)$$

является эквивалентностью.

Упражнение 5.5. Покажите, что аналогичное утверждение упражнения 5.4 для $\mathbb{N}$ неверно.

Упражнение 5.6. Покажите, что если мы предполагаем простое вместо зависимого исключения для W-типов, то свойство уникальности (аналог теоремы 5.3.1) не выполняется. То есть, предъявите тип, удовлетворяющий принципу рекурсии W-типа, но для которого функции не определяются однозначно своей рекурентностью.

Упражнение 5.7. Предположим, что в «индуктивном определении» типа C в начале §5.6 тип $\mathbb{N}$ заменен на $\mathbf{0}$. Аналогично (5.6.1), можно рассмотреть принцип рекурсии для этого типа с гипотезой

$$h : (C \to \mathbf{0}) \to (P \to \mathbf{0}) \to P.$$

Покажите, что даже без правила вычисления этот принцип рекурсии противоречив, т.е. позволяет нам предъявить элемент из $\mathbf{0}$.

Упражнение 5.8. Рассмотрим «индуктивный тип» D с одним конструктором $\mathsf{scott} : (D \to D) \to D$. Второй рекурсор для C, предложенный в §5.6, приводит к следующему рекурсору для D:

$$\mathsf{rec}_D : \prod_{P:\mathcal{U}} ((D \to D) \to (D \to P) \to P) \to D \to P$$

с правилом вычисления $\mathsf{rec}_D(P, h, \mathsf{scott}(\alpha)) \equiv h(\alpha, (\lambda d.\, \mathsf{rec}_D(P, h, \alpha(d))))$. Показать, что это приводит к противоречию.

Упражнение 5.9. Пусть A — произвольный тип и в целом рассмотрим «индуктивное определение» типа L_A с конструктором $\mathsf{lawvere} : (L_A \to A) \to L_A$. Второй рекурсор для C, предложенный в §5.6, приводит к следующему рекурсору для L_A:

$$\mathsf{rec}_{L_A} : \prod_{P:\mathcal{U}} ((L_A \to A) \to P) \to L_A \to P$$

с правилом вычисления $\mathsf{rec}_{L_A}(P, h, \mathsf{lawvere}(\alpha)) h(\alpha)$. Используя это, покажите, что A имеет **свойство неподвижной точки**, т.е. для любой функции $f : A \to A$ существует такое $a : A$, что $f(a) = a$. В частности, L_A противоречив, если A является типом без свойства фиксированной точки, например $\mathbf{0}$, $\mathbf{2}$ или $\mathbb{N}$.

Упражнение 5.10. Продолжая упражнение 5.9, рассмотрим тип $L_{\mathbf{1}}$, который явно не противоречив, поскольку $\mathbf{1}$ имеет свойство с фиксированной точкой. Сформулируйте принцип индукции для $L_{\mathbf{1}}$ и его правило вычисления, аналогично — для его рекурсора, и используя это, докажите, что тип является стягиваемым.

Упражнение 5.11. В §5.1 определен тип $\mathsf{List}(A)$ конечных списков элементов некоторого типа A. Рассмотрим аналогичное индуктивное определение типа $\mathsf{Lost}(A)$, единственным конструктором которого является

$$\mathsf{cons} : A \to \mathsf{Lost}(A) \to \mathsf{Lost}(A)\,.$$

Покажите, что $\mathsf{Lost}(A)$ эквивалентен $\mathbf{0}$.

Упражнение 5.12. Предположим, что A является простым высказыванием и $B : A \to \mathcal{U}$.

(i) Пожите, что $\mathsf{W}_{(a:A)}B(a)$ является простым высказыванием.

(ii) Пожите, что $\mathsf{W}_{(a:A)}B(a)$ эквивалентно $\sum_{(a:A)} \neg B(a)$.

(iii) Без использования $\mathsf{W}_{(a:A)}B(a)$ покажите, что $\sum_{(a:A)} \neg B(a)$ есть гомотопический W-тип $\mathsf{W}^h_{(a:A)}B(a)$ в смысле §5.5.

Упражнение 5.13. Пусть $A : \mathcal{U}$ и $B : A \to \mathcal{U}$.

(i) Покажите, что $\Big(\sum_{(a:A)} \neg B(a)\Big) \to \big(\mathsf{W}_{(a:A)}B(a)\big)$.

(ii) Покажите, что $\big(\mathsf{W}_{(a:A)}B(a)\big) \to \Big(\neg \prod_{(a:A)} B(a)\Big)$.

Упражнение 5.14. Пусть $A : \mathcal{U}$ и предположим, что $B : A \to \mathcal{U}$ разрешим, т.е. $\prod_{(a:A)}(B(a) + \neg B(a))$ (см. определение 3.4.3). Покажите, что $\big(\mathsf{W}_{(a:A)}B(a)\big) \to \Big(\sum_{(a:A)} \neg B(a)\Big)$.

Упражнение 5.15. Покажите, что следующие свойства логически эквивалентны.

(i) $\big(\mathsf{W}_{(a:A)}B(a)\big) \to \Big\|\sum_{(a:A)} \neg B(a)\Big\|$ для любого $A : \mathsf{Set}$ и $B : A \to \mathsf{Prop}$.

(ii) $\Big(\neg \prod_{(a:A)} B(a)\Big) \to \big\|\mathsf{W}_{(a:A)}B(a)\big\|$ для любого $A : \mathsf{Set}$ и $B : A \to \mathsf{Prop}$.

(iii) Закон исключения третьего (как в §3.4).

Аналогично, используя следствие 3.2.7, покажите противоречивость предположения, что импликация либо в (i), либо в (ii) выполняется для всех $A : \mathcal{U}$ и $B : A \to \mathcal{U}$.

Упражнение 5.16. Для $A : \mathcal{U}$ и $B : A \to \mathcal{U}$ определим

$$W'_{A,B} :\equiv \prod_{R:\mathcal{U}} \left(\prod_{a:A} (B(a) \to R) \to R\right) \to R\,.$$

$W'_{A,B}$ называется **непредикативным кодированием** $\mathsf{W}_{(a:A)}B(a)$. Заметим, что в отличие от $\mathsf{W}_{(a:A)}B(a)$, он обитает в большем универсуме, чем A и B.

(i) Покажите, что $W'_{A,B}$ логически эквивалентен (как определено в §1.11) $\mathsf{W}_{(a:A)}B(a)$.

(ii) Покажите, что $W'_{A,B}$ означает $\neg\neg\sum_{(a:A)}\neg B(a)$.

(iii) Без использования $\mathsf{W}_{(a:A)}B(a)$, покажите, что $W'_{A,B}$ удовлетворяет тому же принципу рекурсии, что и $\mathsf{W}_{(a:A)}B(a)$ для определения функций в типы в универсуме $\mathcal{U}$ (к которой он сам не принадлежит).

(iv) Используя LEM, дайте пример с $A:\mathcal{U}$ и $B:A\to\mathcal{U}$, в котором $W'_{A,B}$ не эквивалентен $\mathsf{W}_{(a:A)}B(a)$.

Упражнение 5.17. Покажите, что для любого $A:\mathcal{U}$ и $B:A\to\mathcal{U}$ имеет место

$$\neg\left(\mathsf{W}_{(a:A)}B(a)\right)\simeq\neg\left(\sum_{(a:A)}\neg B(a)\right).$$

Другими словами, $\mathsf{W}_{(a:A)}B(a)$ пуст тогда и только тогда, когда он не имеет нульарного конструктора (сравните с упражнением 5.11).

Глава 6

Высшие индуктивные типы

6.1 Введение

Как и общие индуктивные типы, рассмотренные в главе 5, *высшие индуктивные типы* являются общей схемой для определения новых типов, порожденных некоторыми конструкторами. Но, в отличие от обычных индуктивных типов, при определении высшего индуктивного типа у нас могут быть «конструкторы», которые порождают не только *точки* этого типа, но также *пути* и высшие пути в этом типе. Например, можно рассмотреть высший индуктивный тип $\mathbb{S}^1$, порожденный

- точкой $\mathsf{base} : \mathbb{S}^1$ и
- путем $\mathsf{loop} : \mathsf{bas}e =_{\mathbb{S}^1} \mathsf{base}$.

Это следует рассматривать как полностью аналогичное определению, например, типу $\mathbf{2}$, порожденному

- точкой $0_{\mathbf{2}} : \mathbf{2}$ и
- точкой $1_{\mathbf{2}} : \mathbf{2}$,

или определению $\mathbb{N}$, порожденному

- точкой $0 : \mathbb{N}$ и
- функцией $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

Когда мы рассматриваем типы как высшие группоиды, более общее понятие «порождение» является очень естественным: поскольку высший группоид является «многосортным объектом» с путями и высшими путями, а также точками, мы должны предоставить «образующие» для всех размерностей.

Мы будем ссылаться на обычный тип конструкторов (таких как base) как на **конструкторы точек** или *обычные конструкторы*, а на другие (такие как loop) как на **конструкторы путей** или *высшие конструкторы*. Каждый конструктор пути должен указывать начальную и конечную точку пути, которые мы называем его **источником** и **целью**; для loop, как источником, так и целью, является base.

Обратите внимание, что конструктор пути, такой как loop, порождает *нового* обитателя типа тождественности, который не является (по крайней мере, не *априори*) равным любому ранее существующему такому обитателю. В частности, петля не является *априори* равной $\mathsf{refl}_{\mathsf{base}}$ (хотя доказательство того, что они определенно неравнозначны, заставляет слегка задуматься, см. лемму 6.4.1). Это то, что отличает $\mathbb{S}^1$ от обычного индуктивного типа $\mathbf{1}$.

В отношении изложенного обобщения есть несколько важных моментов.

Прежде всего, слово «порождение» следует воспринимать всерьез, в том же смысле, в каком группа может свободно порождаться некоторым множеством. В частности, поскольку высший группоид имеет *операции* над путями и высшими путями, то когда подобный объект «порожден» определенными конструкторами, операции создают дополнительные пути, которые не происходят непосредственно от самих конструкторов. Например, в высшем индуктивном типе $\mathbb{S}^1$ конструктор loop не является единственным нетривиальным путем от base к base; имеются также «$\mathsf{loop} \cdot \mathsf{loop}$», «$\mathsf{loop} \cdot \mathsf{loop} \cdot \mathsf{loop}$» и т.д., а также loop^{-1}, все различные. Это может показаться настолько очевидным, что и не требует упоминания, но это есть отклонение от поведения «обычных» индуктивных типов, когда не предполагается обнаружить в индуктивном типе дополнительных черт, кроме тех, что были «введены» непосредственно конструкторами.

Во-вторых, это порождение — действительно *свободное* порождение: высшие индуктивные типы технически не позволяют навязывать «аксиомы», такой как сведение «$\mathsf{loop} \cdot \mathsf{loop}$» к равному $\mathsf{refl}_{\mathsf{base}}$. Тем не менее, в мире ∞-группоидов существует небольшая разница между «свободным порождением» и «представлением», поскольку мы можем сделать два пути равными *с точностью до гомотопии*, добавляя к ним новую двумерную образующую (например, $\mathsf{loop} \cdot \mathsf{loop} = \mathsf{refl}_{\mathsf{base}}$ в $\mathsf{base} = \mathsf{base}$). Разумеется, нам приходится озаботиться тем, должна ли эта новая образующая удовлетворять своим собственным «аксиомам» и т.д., но в принципе любое «представление» может быть преобразовано в «свободное», путем внедрения аксиом в конструкторы. Как мы увидим, добавляя «конструкторы усечения», мы можем использовать высшие индуктивные типы для выражения классических понятий, таких как представления групп.

В-третьих, хотя высший индуктивный тип содержит «конструкторы», которые порождают *пути в* этом типе, он по-прежнему является индуктивным определением *одиночного* типа. В частности, как мы увидим, высший индуктивный тип сам по себе является универсальным свойством (выражаемое, как обычно, принципом индукции), а *не* его типы тождественности. Тип тождественности высшего индуктивного типа сохраняет обычный принцип индукции любого тождественного типа (т.е. индукции пути) и не приобретает никакого нового принципа индукции.

Таким образом, возможно нетривиально идентифицировать типы тождественности высшего индуктивного типа конкретным образом, в отличие от того, как в главе 2 мы смогли дать явные описания поведения типов тождественности под всеми традиционными операциями формирования типа. Например, существуют ли какие-либо пути от base к base в $\mathbb{S}^1$, которые не являются просто композициями копий loop и его обратного? Интуитивно кажется, что ответ должен быть нет (и это так), но доказательство этого не тривиально. Действительно, такие вопросы быстро приводят к таким проблемам, как вычисление гомотопических групп сфер, давней проблемы алгебраической топологии, для которой не известна простая формула. Гомотопическая теория типов привносит новую и мощную точку зрения на подобные вопросы, но также подразумевает, что теория типов должна быть такой же сложной, как ответы на эти вопросы.

В-четвертых, «размерность» конструкторов (т.е. порождают ли они точки, пути, пути между путями и т.д.) не имеет прямой связи с теми размерностями, в которых результирующий тип имеет нетривиальную гомотопию внутри. В качестве простого примера, если индуктивный тип B имеет конструктор типа $A \to B$, то любые пути и высшие пути в A приводят к путям и высшим путям в B, хотя конструктор вообще не является «высшим» конструктором. То же самое происходит и с высшими конструкторами: наличие конструктора типа $A \to (x =_B y)$ означает не только то, что точки из A производят пути от x к y в B, но и пути в A порождают пути между этими путями и т.д. Как мы увидим, эта возможность отвечает за большую часть мощи высших индуктивных типов.

С другой стороны, даже конструкторы, *не имеющие* высших типов на своих входах, могут создавать «неожиданные» высшие пути. Например, в двумерной сфере $\mathbb{S}^2$, порожденной

- точкой $\mathsf{base} : \mathbb{S}^2$ и

- 2-мерным путем $\mathsf{surf} : \mathsf{refl}_{\mathsf{base}} = \mathsf{refl}_{\mathsf{base}}$ в $\mathsf{base} = \mathsf{base}$,

существует нетривиальный *3-мерный путь* от $\mathsf{refl}_{\mathsf{refl}_{\mathsf{base}}}$ к себе. Топологи признают этот путь как воплощение *расслоения Хопфа*. С теоретико-категорной точки зрения это то же самое явление, что и упомянутый выше факт, что $\mathbb{S}^1$ содержит не только loop, но и $\mathsf{loop} \cdot \mathsf{loop}$ и т.д.: именно в *высшем* группоиде имеются *операции*, которые повышают размерность. В самом деле, мы видели многие из этих операций в §2.1: ассоциативность и единичные законы — это не просто свойства, а операции, входы которых являются 1-путями, а выходы — 2-путями.

6.2 Принципы индукции и зависимые пути

Когда мы описываем высший индуктивный тип, такой как окружность, который порождается определенными конструкторами, мы должны объяснить, что это означает, давая правила, аналогичные правилам для конструкторов базового типа из главы 1. Сами конструкторы дают правила *введения*, но это требует небольшого осмысления, чтобы объяснить правила *исключения*, т.е. принципы индукции и рекурсии. В этой книге мы не пытаемся дать общую формулировку того, что представляет собой «высшее индуктивное определение» и как извлечь правило исключения из такого определения — действительно, это тонкий вопрос и является предметом текущих исследований. Вместо этого мы будем опираться на некоторые общие неформальные обсуждения и многочисленные примеры.

Принцип рекурсии обычно легко описать: для любого типа, оснащенного той же структурой, которой конструкторы оборудуют высший индуктивный тип, существует функция, которая отображает конструкторы в эту структуру. Например, в случае $\mathbb{S}^1$ принцип рекурсии гласит, что для любого типа B, с точкой $b : B$ и путем $\ell : b = b$, существует функция $f : \mathbb{S}^1 \to B$ такая, что $f(\mathsf{base}) = b$ и $\mathsf{ap}_f(\mathsf{loop}) = \ell$.

Последние два равенства — это *правила вычислений*. Однако существует вопрос о том, являются ли эти правила вычислений дефинициальными равенствами или пропозициональными равенствами (путями). Для обычных индуктивных типов мы не испытывали никаких сомнений в том, чтобы сделать их дефинициальными, хотя мы видели в главе 5, что их пропозициональность все равно будет давать один и тот же тип с точностью до эквивалентности. В обычном случае можно утверждать, что правила вычисления являются действительно дефинициальными равенствами в интуитивном смысле, описанном во введении.

Для высших индуктивных типов это менее понятно. Более того, поскольку операция ap_f на самом деле не является фундаментальной частью теории типов, и которую мы *определили* с использованием принципа индукции типов тождественности (и могли бы определить каким-то другим, эквивалентным образом), представляется неуместным ссылаться к ней явно в *дефинициальном* равенстве. Дефинициальные равенства являются частью дедуктивной системы, которая не должна зависеть от конкретного выбора определений, которые мы можем производить *в рамках* этой системы. Также есть вопросы, связанные с семантикой и реализацией; см. примечания.

Кажется непростым делом создавать вычисляемые правила для конструкторов точек высшего индуктивного типа дефинициальными. В приведенном выше примере это означает, что мы имеем $f(\mathsf{base}) \equiv b$, дефинициально. Этот выбор облегчает вычислительный взгляд на высшие индуктивные типы. Более того, это также значительно упрощает нашу жизнь, поскольку в противном случае второе правило вычисления $\mathsf{ap}_f(\mathsf{loop}) = \ell$ даже не было бы хорошо типизировано как пропозициональное равенство; мы должны были бы скомпоновать одну или другую сторону, указанного идентификатором $f(\mathsf{base})$, с b (такие проблемы действительно возникают, в конце концов, когда мы начинаем говорить о путях высших размерностей, но нас здесь это не очень беспокоит. См. также §6.7). Таким образом, мы считаем правила вычислений для конструкторов точек дефинициальными, а также такими, которые имеют пути и высшие пути, чтобы быть пропозициональными[1].

Замечание 6.2.1. Напомним, что для обычных индуктивных типов мы рассматриваем правила вычислений для рекурсивно определенной функции как не просто субъективные, а как *определительные* равенства, и поэтому можем использовать для них обозначение $:\equiv$. Например, усеченная функция предшествования $p : \mathbb{N} \to \mathbb{N}$ определяется как $p(0) :\equiv 0$ и $p(\mathsf{succ}(n)) :\equiv n$. В случае высших индуктивных типов этот вид обозначений является разумным для конструкторов точек (например, $f(\mathsf{base}) :\equiv b$), но для конструкторов путей он может вводить в заблуждение, поскольку равенства, такие как $f(\mathsf{loop}) = \ell$, не являются определительными. Таким образом, мы совмещаем обозначения, записывая вместо этого $f(\mathsf{loop}) := \ell$ для такого рода «пропозиционального равенства по определению».

Теперь, как насчет принципа индукции (зависимого выделителя)? Напомним, что для обычного индуктивного типа W, чтобы доказать по индукции, что $\prod_{(x:W)} P(x)$, мы должны указать для каждого конструктора W операцию на P, действующую на «слои» над этим конструктором в W. Например, если W — натуральные числа $\mathbb{N}$, то для доказательства по индукции, что $\prod_{(x:\mathbb{N})} P(x)$, необходимо указать

- элемент $b : P(0)$ в слое над конструктором $0 : \mathbb{N}$ и
- для каждого $n : \mathbb{N}$ функцию $P(n) \to P(\mathsf{succ}(n))$.

Второе требование можно рассматривать как функцию «$P \to P$», лежащую над конструктором $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$, обобщающую первое требование.

По аналогии, поэтому, чтобы доказать $\prod_{(x:\mathbb{S}^1)} P(x)$, необходимо указать

- элемент $b : P(\mathsf{base})$ в слое над конструктором $\mathsf{base} : \mathbb{S}^1$ и
- путь от b к b, «лежащий над конструктором $\mathsf{loop} : \mathsf{base} = \mathsf{base}$».

[1]В частности, в терминологии §1.1 это означает, что наши высшие индуктивные типы представляют собой сочетание *правил* (с указанием того, как мы можем вводить такие типы и их элементы, их принцип индукции и их правила вычислений для конструкторов точек) и *аксиом* (правил вычисления для конструкторов путей, которые утверждают, что определенные типы тождественности заселены иначе неопределенными членами). Мы можем надеяться, что в конечном итоге будет создана более совершенная теория типов, в которой высшие индуктивные типы, такие как унивалентность, будут представлены с использованием только правил, без аксиом.

Обратите внимание, что хотя $\mathbb{S}^1$ содержит пути, отличные от loop (такие как $\mathsf{refl}_{\mathsf{base}}$ и $\mathsf{loop} \cdot \mathsf{loop}$), нам нужно только указать путь, лежащий над самим конструктором. Это выражает интуицию, что $\mathbb{S}^1$ «свободно порождается» своими конструкторами.

Вопрос, однако, заключается в том, что означает то, что путь «лежит над» другим путем. Это определенно *не* означает просто путь $b = b$, так как это будет путь в слое $P(\mathsf{base})$ (топологически, путь, лежащий над *постоянным* путем при base). На самом деле, однако, мы уже ответили на этот вопрос в главе 2: в обсуждении, предшествующем лемме 2.3.4, мы пришли к выводу, что путь от $u : P(x)$ к $v : P(y)$, лежащий над $p : x = y$, может быть представлен путем $p_*(u) = v$ в слое $P(y)$. Поскольку в этой главе мы будем многое использовать для таких **зависимых путей**, мы введем для них специальное обозначение:

$$(u =^P_p v) :\equiv (\mathsf{transport}^P(p, u) = v)\,. \tag{6.2.2}$$

Замечание 6.2.3. Существуют и другие возможные способы определения зависимых путей. Например, вместо $p_*(u) = v$ мы могли бы рассмотреть $u = (p_*^{-1}(v)$. Мы могли бы также получить их в качестве частного случая более общего «гетерогенного равенства» или прямым определением, как семейство индуктивных типов. Все эти определения приводят к эквивалентным типам, поэтому в этом смысле не имеет большого значения то, что мы используем. Однако, выбор $p_*(u) = v$ в качестве определения облегчает выявление других качеств зависимых путей, таких как то, что apd_f производит их, или что мы можем вычислить их в определенных семействах типов, используя леммы транспортировки из §2.5.

С понятием зависимых путей мы можем теперь более точно сформулировать принцип индукции для $\mathbb{S}^1$: для $P : \mathbb{S}^1 \to \mathcal{U}$,

- элемента $b : P(\mathsf{base})$ и
- пути $\ell : b =^P_{\mathsf{loop}} b$

существует функция $f : \prod_{(x:\mathbb{S}^1)} P(x)$ такая, что $f(\mathsf{base}) \equiv b$ и $\mathsf{apd}_f(\mathsf{loop}) = \ell$. Как и в случае независимых путей, мы говорим об определении f через $f(\mathsf{base}) :\equiv b$ и $\mathsf{apd}_f(\mathsf{loop}) := \ell$.

Замечание 6.2.4. При описании применения этого принципа индукции неформально, мы рассматриваем его как разделение цели «$P(x)$ для всех $x : \mathbb{S}^1$» на два случая, которые мы иногда будем вводить фразами, такими как «когда x является base» и «когда x изменяется вдоль loop», соответственно. Не существует специального математического смысла, обозначаемого «изменяющийся вдоль пути»: это просто удобный способ указать начало соответствующего раздела доказательства; см. лемму 6.4.2 для примера.

Топологически принцип индукции для $\mathbb{S}^1$ можно визуализировать, как показано на рис.6.1. Для расслоения над окружностью (которое на рисунке является тором), определить сечение этого расслоения — это то же, что предоставить точку b в слое над base вместе с путем от b к b, лежащим над loop. То, как мы интерпретируем это теоретико-типовым образом, используя наше определение зависимых путей, показано на рис.6.2: путь от b к b над loop представлен путем от $\mathsf{loop}_*(b)$ к b в слое над base.

Конечно, мы ожидаем, что будем иметь возможность доказать принцип рекурсии по принципу индукции, полагая, что P является семейством с постоянными типами. Это на самом деле

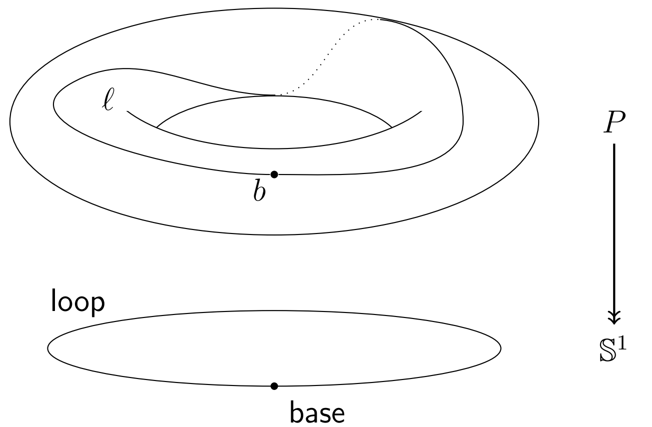


Рис. 6.1: Принцип топологической индукции для $\mathbb{S}^1$

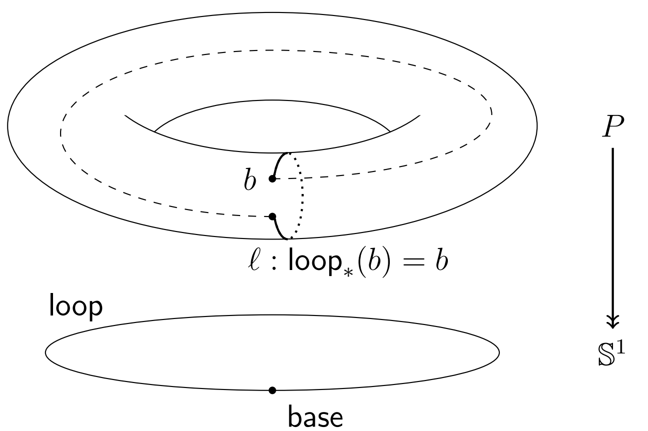


Рис. 6.2: Принцип теоретико-типовой индукции для $\mathbb{S}^1$

так, хотя получение независимого правила вычисления для loop (которое относится к ap_f) от зависимого (которое относится к apd_f), на удивление, немногим сложнее.

Лемма 6.2.5. *Если A — тип вместе с $a : A$ и $p : a =_A a$, то существует функция $f : \mathbb{S}^1 \to A$ с*

$$f(\mathsf{base}) :\equiv a\,,$$
$$\mathsf{ap}_f(\mathsf{loop}) := p\,.$$

Доказательство. Мы хотели бы применить принцип индукции для $\mathbb{S}^1$ к семейству с постоянными типами, $(\lambda x.\, A) : \mathbb{S}^1 \to \mathcal{U}$. Требуемые гипотезы для этого — точка из $(\lambda x.\, A)(\mathsf{base}) \equiv A$, которая имеется (именно, $a : A$), и зависимый путь в $a =^{x \mapsto A}_{\mathsf{loop}} a$, или эквивалентно $\mathsf{transport}^{x \mapsto A}(\mathsf{loop}, \mathsf{a}) = \mathsf{a}$. Этот последний тип не совпадает с типом $a =_A a$, где p обитает, но он эквивалентен ему, потому что по лемме 2.3.5 имеем $\mathsf{transportconst}^A_{\mathsf{loop}}(a) : \mathsf{transport}^{x \mapsto A}(\mathsf{loop}, \mathsf{a}) = \mathsf{a}$. Таким образом, для $a : A$ и $p : a = a$, мы можем рассмотреть композицию

$$\mathsf{transportconst}^A_{\mathsf{loop}}(a) \cdot p : (a =^{x \mapsto A}_{\mathsf{loop}} a)\,.$$

Применяя принцип индукции, получаем $f : \mathbb{S}^1 \to A$ такую, что

$$f(\mathsf{base}) \equiv a \qquad \text{и} \tag{6.2.6}$$

$$\mathsf{apd}_f(\mathsf{loop}) = \mathsf{transportconst}^A_{\mathsf{loop}}(a) \cdot p\,. \tag{6.2.7}$$

Остается вывести равенство $\mathsf{ap}_f(\mathsf{loop}) = p$. Однако, по лемме 2.3.8 имеем

$$\mathsf{apd}_f(\mathsf{loop}) = \mathsf{transportconst}^A_{\mathsf{loop}}(f(\mathsf{base})) \cdot \mathsf{ap}_f(\mathsf{loop})\,.$$

Объединяя это с (6.2.7) и удаляя вхождения $\mathsf{transportconst}$ (одинаковые в силу (6.2.6)), получим $\mathsf{ap}_f(\mathsf{loop}) = p$. □

Также имеется соответствующий принцип единственности.

Лемма 6.2.8. *Если A — тип, а $f, g : \mathbb{S}^1 \to A$ — два отображения вместе с равенствами*:

$$p : f(\mathsf{base}) =_A g(\mathsf{base})\,,$$
$$q : f(\mathsf{loop}) =^{\lambda x.\, x =_A x}_p g(\mathsf{loop})\,.$$

то для всех $x : \mathbb{S}^1$ имеем $f(x) = g(x)$.

Доказательство. Применим принцип индукции для $\mathbb{S}^1$ к семейству типов $P(x) :\equiv (f(x) = g(x))$. Когда x есть base, p — это именно то, что нам нужно. А когда x меняется вдоль loop, нам требуется $p =^{\lambda x.\, f(x) = g(x)}_{\mathsf{loop}} p$, что по теоремам 2.11.3 и 2.11.5 сводится к q. □

Из этих двух лемм вытекает ожидаемое универсальное свойство окружности:

Лемма 6.2.9. *Для любого типа A имеется естественная эквивалентность*

$$(\mathbb{S}^1 \to A) \simeq \sum_{x:A} (x = x)\,.$$

Доказательство. У нас есть каноническая функция $f : (\mathbb{S}^1 \to A) \to \sum_{(x:A)}(x = x)$, определяемая как $f(g) :\equiv (g(\mathsf{base}), g(\mathsf{loop}))$. Принцип индукции указывает, что слои f обитаемы, а принцип единственности показывает, что они являются простыми высказываниями. Следовательно, они стягиваемы, поэтому f является эквивалентностью. □

Как и в §5.5, мы можем показать, что заключение леммы 6.2.9 эквивалентно наличию индукционного принципа с пропозициональными правилами вычислений. Другие высшие индуктивные типы также удовлетворяют леммам, аналогичным леммам 6.2.5 и 6.2.9; мы, как правило, оставляем их доказательства читателю. Перейдем теперь к рассмотрению примеров.

6.3 Интервал

Интервал, обозначаемый I, является, возможно, еще более простым индуктивным типом, чем окружность. Он порождается:

- точкой $0_I : I$,
- точкой $1_I : I$ и

- путем $\mathsf{seg} : 0_I =_I 1_I$.

Принцип рекурсии для интервала утверждает, что для типа B с

- точкой $b_0 : B$,
- точкой $b_1 : B$ и
- путем $s : b_0 = b_1$

существует функция $f : I \to B$ такая, что $f(0_I) \equiv b_0$, $f(1_I) \equiv b_1$ и $f(\mathsf{seg}) = s$.

Принцип индукции гласит, что для $P : I \to \mathcal{U}$ вместе с

- точкой $b_0 : P(0_I)$,
- точкой $b_1 : P(1_I)$ и
- путем $s : b_0 =^P_{\mathsf{seg}} b_1$

существует функция $f : \prod_{(x:I)} P(x)$ такая, что $f(0_I) \equiv b_0$, $f(1_I) \equiv b_1$ и $\mathsf{apd}_f(\mathsf{seg}) = s$.

Рассматриваемый исключительно с точностью до гомотопии, интервал не очень интересен:

Лемма 6.3.1. *Тип I является стягиваемым.*

Доказательство. Докажем, что для всех $x : I$ имеем $x =_I 1_I$. Другими словами, нам нужна функция f типа $\prod_{(x:I)}(x =_I 1_I)$. Мы начнем определять f следующим образом:

$$f(0_I) :\equiv \mathsf{seg} : 0_I =_I 1_I \,,$$
$$f(1_I) :\equiv \mathsf{refl}_{1_I} : 1_I =_I 1_I \,.$$

Осталось определить $\mathsf{apd}_f(\mathsf{seg})$, который должен иметь тип $\mathsf{seg} =^{\lambda x.\, x =_I 1_I}_{\mathsf{seg}} \mathsf{refl}_{1_I}$. По определению этот тип является $\mathsf{seg}_*(\mathsf{seg}) =_{1_I =_I 1_I} \mathsf{refl}_{1_I}$, который, в свою очередь, эквивалентен $\mathsf{seg}^{-1} \cdot \mathsf{seg} = \mathsf{refl}_{1_I}$. Но имеется канонический элемент этого типа, а именно доказательство того, что обратные путей являются фактически обратными. □

Однако с теоретико-типовой точки зрения интервал все еще имеет некоторые интересные особенности, как и топологический интервал в классической теории гомотопий. Например, он позволяет нам дать простое доказательство функциональной экстенсиональности (конечно, как и в §4.9, на протяжении следующего доказательства мы приостанавливаем наше общее предположение об аксиоме функциональной экстенсиональности).

Лемма 6.3.2. *Если $f, g : A \to B$ — две функции такие, что $f(x) = g(x)$ для каждого $x : A$, то $f = g$ в типе $A \to B$.*

Доказательство. Представим доказательство, которое мы имеем, $p : \prod_{(x:A)}(f(x) = g(x))$. Для каждого $x : A$ определим функцию $\tilde{p}_x : I \to B$ как

$$\tilde{p}_x(0_I) :\equiv f(x) \,,$$
$$\tilde{p}_x(1_I) :\equiv g(x) \,,$$
$$\tilde{p}_x(\mathsf{seg}) :\equiv p(x) \,.$$

Теперь определим $q : I \to (A \to B)$ посредством

$$q(i) :\equiv (\lambda x.\, \tilde{p}_x(i)).$$

Тогда $q(0_I)$ — функция $\lambda x.\, \tilde{p}_x(0_I)$, которая равна f, потому что $\tilde{p}_x(0_I)$ определена как $f(x)$. Аналогично, $q(1_I) = g$ и, следовательно,

$$q(\mathsf{seg}) : f =_{(A\to B)} g\,.$$

$\square$

В упражнении 6.10 мы попросим читателя завершить доказательство аксиомы функциональной экстенсиональности из леммы 6.3.2.

6.4 Окружности и сферы

Мы уже обсуждали окружность $\mathbb{S}^1$ как высший индуктивный путь, порожденный

- точкой $\mathsf{base} : \mathbb{S}^1$ и
- путем $\mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}$.

Ее принцип индукции гласит, что для $P : \mathbb{S}^1 \to \mathcal{U}$ вместе с $b : P(\mathsf{base})$ и $\ell : b =^P_{\mathsf{loop}} b$ имеет место $f : \prod_{(x:\mathbb{S}^1)} P(x)$ с $f(\mathsf{base}) \equiv b$ и $\mathsf{apd}_f(\mathsf{loop}) = \ell$. Ее независимый вариант принципа рекурсии утверждает, что для B с $b : B$ и $\ell : b = b$ имеем $f : \mathbb{S}^1 \to B$ с $f(\mathsf{base}) \equiv b$ и $f(\mathsf{loop}) = \ell$.

Заметим, что окружность не является тривиальной.

Лемма 6.4.1. $\mathsf{loop} \neq \mathsf{refl}_{\mathsf{base}}$.

Доказательство. Предположим, что $\mathsf{loop} = \mathsf{refl}_{\mathsf{base}}$. Тогда, поскольку для любого типа A с $x : A$ и $p : x = x$ существует функция $f : \mathbb{S}^1 \to A$, определямая как $f(\mathsf{base}) :\equiv x$ и $f(\mathsf{loop}) := p$, имеет место

$$p = f(\mathsf{loop}) = f(\mathsf{refl}_{\mathsf{base}}) = \mathsf{refl}_x\,.$$

Но это означает, что каждый тип является множеством, что, как мы видели, не так (см. пример 3.1.9). $\square$

Окружность также обладает следующим интересным свойством, которое полезно как источник контрпримеров.

Лемма 6.4.2. *Существует элемент из* $\prod_{(x:\mathbb{S}^1)}(x = x)$*, не равный* $x \mapsto \mathsf{refl}_x$.

Доказательство. Определим элемент $f : \prod_{(x:\mathbb{S}^1)}(x = x)$ с помощью $\mathbb{S}^1$-индукции. Когда x есть base, положим $f(\mathsf{base}) :\equiv \mathsf{loop}$. Теперь, когда x меняется вдоль loop (см. замечание 6.2.4), мы должны показать, что $\mathsf{transport}^{x\mapsto x=x}(\mathsf{loop}, \mathsf{loop}) = \mathsf{loop}$. Однако в §2.11 мы заметили, что $\mathsf{transport}^{x\mapsto x=x}(p, q) = p^{-1} \cdot q \cdot p$, поэтому надо показать, что $\mathsf{loop}^{-1} \cdot \mathsf{loop} \cdot \mathsf{loop} = \mathsf{loop}$. Но это очевидно.

Чтобы показать, что $f \neq x \mapsto \mathsf{refl}_x$, достаточно показать, что $f(\mathsf{base}) \neq \mathsf{refl}_{\mathsf{base}}$. Но $f(\mathsf{base}) = \mathsf{loop}$, так что это просто предыдущая лемма. $\square$

В качестве применения, это позволяет нам расширить пример 3.1.9, показывая, что любой универсум, который содержит окружность, не может быть 1-типом.

Следствие 6.4.3. *Если тип $\mathbb{S}^1$ принадлежит некоторому универсуму $\mathcal{U}$, то $\mathcal{U}$ не является 1-типом.*

Доказательство. Тип $\mathbb{S}^1 = \mathbb{S}^1$ в $\mathcal{U}$, по унивалентности, эквивалентен типу $\mathbb{S}^1 \simeq \mathbb{S}^1$ автоэквивалентностей из $\mathbb{S}^1$, поэтому достаточно показать, что $\mathbb{S}^1 \simeq \mathbb{S}^1$ не является множеством. Для этого достаточно показать, что его тип равенства $\mathsf{id}_{\mathbb{S}^1} =_{(\mathbb{S}^1 \simeq \mathbb{S}^1)} \mathsf{id}_{\mathbb{S}^1}$ не является простым высказыванием. Поскольку эквивалентность является простым высказыванием, этот тип эквивалентен $\mathsf{id}_{\mathbb{S}^1} =_{(\mathbb{S}^1 \to \mathbb{S}^1)} \mathsf{id}_{\mathbb{S}^1}$. Но по функциональной экстенсиональности это эквивалентно $\prod_{(x:\mathbb{S}^1)}(x = x)$, которое, как мы видели в лемме 6.4.2, содержит два неравных элемента. $\square$

Мы также отметили, что 2-сфера $\mathbb{S}^2$ должна быть высшим индуктивным типом, порожденным

- точкой $\mathsf{base} : \mathbb{S}^2$ и
- 2-мерным путем $\mathsf{surf} : \mathsf{refl}_{\mathsf{base}} = \mathsf{refl}_{\mathsf{base}}$ в $\mathsf{base} = \mathsf{base}$.

Принцип рекурсии для $\mathbb{S}^2$ не сложен: он заявляет, что для B с $b : B$ и $s : \mathsf{refl}_b = \mathsf{refl}_b$ имеем $f : \mathbb{S}^2 \to B$ с $f(\mathsf{base}) \equiv b$ и $\mathsf{ap}^2_f(\mathsf{surf}) = s$. Здесь под «$\mathsf{ap}^2_f(\mathsf{surf})$» мы подразумеваем расширение функториального действия f на двумерные пути, которое можно сформулировать строго следующим образом.

Лемма 6.4.4. *Для $f : A \to B$, $x, y : A$, $p, q : x = y$ и $r : p = q$ имеется путь $\mathsf{ap}^2_f(r) : f(p) = f(q)$.*

Доказательство. По индукции пути можно считать $p \equiv q$, а r — рефлексивностью. Но тогда мы можем определить $\mathsf{ap}^2_f(\mathsf{refl}_p) :\equiv \mathsf{refl}_{f(p)}$. $\square$

Чтобы сформулировать общий принцип индукции, нам нужна версия этой леммы для зависимых функций, которая, в свою очередь, требует понятия зависимых двумерных путей. Как и прежде, существует много способов их определения; один из них представляет собой двумерную версию транспортирования.

Лемма 6.4.5. *Для $P : A \to \mathcal{U}$, $x, y : A$, $p, q : x = y$ и $r : p = q$, для любого $u : P(x)$ имеет место $\mathsf{transport}^2(r, u) : p_*(u) = q_*(u)$.*

Доказательство. Используя индукцию пути. $\square$

Предположим теперь, что даны $x, y : A$, $p, q : x = y$, $r : p = q$, а также точки $u : P(x)$ и $v : P(y)$ и зависимые пути $h : u =^P_p v$ и $k :$ и $=^P_q v$. По нашему определению зависимых путей это означает, что $h : p_*(u) = v$ и $k : q_*(u) = v$. Таким образом, разумно определить тип зависимых 2-путей над r, как

$$(h =^P_r k) :\equiv (h = \mathsf{transport}^2(r, u) \cdot k)\,.$$

Теперь мы можем сформулировать зависимую версию леммы 6.4.4.

Лемма 6.4.6. *Для $P : A \to \mathcal{U}$, $x, y : A$, $p, q : x = y$, $r : p = q$ и функции $f : \prod_{(x:A)} P(x)$ имеет место $\mathsf{apd}^2_f(r) : \mathsf{apd}_f(p) =^P_r \mathsf{apd}_f(q)$.*

Доказательство. Индукция пути. $\square$

Теперь мы можем сформулировать принцип индукции для $\mathbb{S}^2$: пусть дано $P : \mathbb{S}^2 \to \mathcal{U}$ с $b : P(\mathsf{base})$ и $s : \mathsf{refl}_b =^Q_{\mathsf{surf}} \mathsf{refl}_b$, где $Q :\equiv \lambda p.\, b =^P_p b$. Тогда существует функция $f : \prod_{(x:\mathbb{S}^2)} P(x)$ такая, что $f(\mathsf{base}) \equiv b$ и $\mathsf{apd}^2_f(\mathsf{surf}) = s$.

Конечно же, этот явный подход становится все более и более сложным, когда мы наращиваем измерение. Так что, если мы хотим определить n-сферы для всех n, нам нужна более систематическая идея. Один из подходов — работать напрямую с n-мерными петлями, а не с общими n-мерными путями.

Напомним из §2.1 определения *точечных типов* $\mathcal{U}_*$ и *пространства n-кратных петель* $\Omega^n : \mathcal{U}_* \to \mathcal{U}_*$ (определения 2.1.7 и 2.1.8). Теперь мы можем определить n-сферу $\mathbb{S}^n$ как высший индуктивный тип, порожденный

- точкой $\mathsf{base} : \mathbb{S}^n$ и
- n-петлей $\mathsf{loop}_n : \Omega^n(\mathbb{S}^n, \mathsf{base})$.

Чтобы записать принцип индукции для этого представления, нам нужно было бы определить понятие «зависимой n-петли » наряду с действием зависимых функций на n-петлях. Мы оставляем это читателю (см. упражнение 6.4); в следующем разделе мы обсудим другой способ определения сфер, с которым зачастую более удобно работать.

6.5 Надстройки

Надстройка типа A является универсальным способом превращения точек из A в пути (и, следовательно, путей из A в 2-пути и т.д.). Это тип ΣA, определяемый следующими образующими[2]:

- точкой $\mathsf{N} : \Sigma A$,
- точкой $\mathsf{S} : \Sigma A$ и
- функцией $\mathsf{merid} : A \to (\mathsf{N} =_{\Sigma A} \mathsf{S})$.

Имена образующих призваны предложить «глобус» сортов, с северным полюсом (N), южным полюсом (S) и A-оцениванием меридианов (merid) от одного полюса к другому. Действительно, как мы увидим, если $A = \mathbb{S}^1$, то его надстройка эквивалентна поверхности обычной сферы, $\mathbb{S}^2$.

Принцип рекурсии для ΣA говорит, что для типа B вместе с

- точками $n, s : B$ и
- функцией $m : A \to (n = s)$

имеется функция $f : \Sigma A \to B$ такая, что $f(\mathsf{N}) \equiv n$ и $f(\mathsf{S}) \equiv$s, и для всех $a : A$ имеет место $f(\mathsf{merid}(a)) = m(a)$. Аналогично, принцип индукции говорит, что для $P : \Sigma A \to \mathcal{U}$ вместе с

- точкой $n : P(\mathsf{N})$,
- точкой $s : P(\mathsf{S})$ и

[2]Очевиден конфликт введенных обозначений, записанных с использованием Σ, с зависимыми типами пар. Однако контекст обычно устраняет эту неоднозначность.

- путем $m(a) : n =^P_{\mathsf{merid}(a)} s$, для всех $a : A$,

существует функция $f : \prod_{(x:\Sigma A)} P(x)$ такая, что $f(\mathsf{N}) \equiv n$ и $f(\mathsf{S}) \equiv s$, и для каждого $a : A$ имеет место $\mathsf{apd}_f(\mathsf{merid}(a)) = m(a)$.

Наше первое замечание о надстройке — это то, что она дает другой способ определить окружность.

Лемма 6.5.1. $\Sigma\mathbf{2} \simeq \mathbb{S}^1$.

Доказательство. Определим $f : \Sigma\mathbf{2} \to \mathbb{S}^1$ рекурсией так, что $f(\mathsf{N}) :\equiv \mathsf{base}$ и $f(\mathsf{S}) :\equiv \mathsf{base}$, а $f(\mathsf{merid}(0_{\mathbf{2}})) := \mathsf{loop}$, но $f(\mathsf{merid}(1_{\mathbf{2}})) := \mathsf{refl}_{\mathsf{base}}$. Определим $g : \mathbb{S}^1 \to \Sigma\mathbf{2}$ рекурсией, так что $g(\mathsf{base}) :\equiv \mathsf{N}$ и $g(\mathsf{loop}) := \mathsf{merid}(0_{\mathbf{2}}) \cdot \mathsf{merid}(1_{\mathbf{2}})^{-1}$. Покажем теперь, что f и g являются квазиобратными.

Сначала покажем индукцией, что $\mathrm{g}(f(x)) = x$ для всех $x : \Sigma\mathbf{2}$. Если $x \equiv \mathsf{N}$, то $g(f(\mathsf{N})) \equiv g(\mathsf{base}) \equiv \mathsf{N}$, поэтому имеем $\mathsf{refl}_{\mathsf{N}} : g(f(\mathsf{N})) = \mathsf{N}$. Если $x \equiv \mathsf{S}$, то $g(f(\mathsf{S})) \equiv g(\mathsf{base}) \equiv \mathsf{N}$, и мы выбираем равенство $\mathsf{merid}(1_{\mathbf{2}}) : g(f(\mathsf{S})) = \mathsf{S}$. Остается показать, что при любом $\mathrm{y} : \mathbf{2}$ эти равенства сохраняются при изменении x вдоль $\mathsf{merid}(\mathrm{y})$, т.е. когда $\mathsf{refl}_{\mathsf{N}}$ транспортируется вдоль $\mathsf{merid}(\mathrm{y})$, он дает $\mathsf{merid}(1_{\mathbf{2}})$. Используя транспортировку в пространствах путей и отбрасывание расслоений, это означает, что мы должны показать, что

$$g(f(\mathsf{merid}(y)))^{-1} \cdot \mathsf{refl}_{\mathsf{N}} \cdot \mathsf{merid}(y) = \mathsf{merid}(1_{\mathbf{2}})\,.$$

Конечно, можно убрать $\mathsf{refl}_{\mathsf{N}}$. Теперь, используя 2-индукцию, мы можем предположить, что либо $y \equiv 0_{\mathbf{2}}$ или $y \equiv 1_{\mathbf{2}}$. Если $y \equiv 0_{\mathbf{2}}$, то имеем

$$\begin{aligned}
g(f(\mathsf{merid}(0_{\mathbf{2}})))^{-1} \cdot \mathsf{merid}(0_{\mathbf{2}}) &= g(\mathsf{loop})^{-1} \cdot \mathsf{merid}(0_{\mathbf{2}}) \\
&= (\mathsf{merid}(0_{\mathbf{2}}) \cdot \mathsf{merid}(1_{\mathbf{2}})^{-1})^{-1} \cdot \mathsf{merid}(0_{\mathbf{2}}) \\
&= \mathsf{merid}(1_{\mathbf{2}}) \cdot \mathsf{merid}(0_{\mathbf{2}})^{-1} \cdot \mathsf{merid}(0_{\mathbf{2}}) \\
&= \mathsf{merid}(1_{\mathbf{2}})\,,
\end{aligned}$$

а если $y \equiv 1_{\mathbf{2}}$, то

$$\begin{aligned}
g(f(\mathsf{merid}(1_{\mathbf{2}})))^{-1} \cdot \mathsf{merid}(1_{\mathbf{2}}) &= g(\mathsf{refl}_{\mathsf{base}})^{-1} \cdot \mathsf{merid}(1_{\mathbf{2}}) \\
&= {\mathsf{refl}_{\mathsf{N}}}^{-1} \cdot \mathsf{merid}(1_{\mathbf{2}}) \\
&= \mathsf{merid}(1_{\mathbf{2}})\,.
\end{aligned}$$

Таким образом, для всех $x : \Sigma\mathbf{2}$ имеем $g(f(x)) = x$.

Теперь индукцией покажем, что $f(g(x)) = x$ для всех $x : \mathbb{S}^1$. Если $x \equiv \mathsf{base}$, то $f(g(\mathsf{base})) \equiv f(\mathsf{N}) \equiv \mathsf{base}$, поэтому имеем $\mathsf{refl}_{\mathsf{base}} : f(g(\mathsf{base})) = \mathsf{base}$. Остается показать, что это равенство сохраняется при x меняющемся вдоль loop, т.е. транспорируется вдоль loop в себя. Опять же, транспортировкой в пространствах путей и отбрасыванием расслоений, это означает показать, что

$$f(g(\mathsf{loop}))^{-1} \cdot \mathsf{refl}_{\mathsf{base}} \cdot \mathsf{loop} = \mathsf{refl}_{\mathsf{base}}\,.$$

Однако, мы имеем

$$\begin{aligned}
f(g(\mathsf{loop})) &= f\left(\mathsf{merid}(0_{\mathbf{2}}) \cdot \mathsf{merid}(1_{\mathbf{2}})^{-1}\right) \\
&= f(\mathsf{merid}(0_{\mathbf{2}})) \cdot f(\mathsf{merid}(1_{\mathbf{2}}))^{-1} \\
&= \mathsf{loop} \cdot \mathsf{refl}_{\mathsf{base}}
\end{aligned}$$

и далее все ясно. □

Топологически двухточечное пространство $\mathbf{2}$ также известно как 0-мерная сфера $\mathbb{S}^0$ (например, это пространство точек на расстоянии 1 от начала координат в $\mathbb{R}^1$, так же как топологическая 1-сфера является пространством точек на расстоянии 1 от начала координат в $\mathbb{R}^2$). Таким образом, возникает соблазн формулировку леммы 6.5.1 перефразировать как $\Sigma\mathbb{S}^0 \simeq \mathbb{S}^1$. На самом деле, такая характеристика имеет развитие: можно определить все сферы индуктивно

$$\mathbb{S}^0 \equiv \mathbf{2} \quad \text{и} \quad \mathbb{S}^{n+1} :\equiv \Sigma\mathbb{S}^n\,. \tag{6.5.2}$$

Мы можем даже начать одной размерностью меньше, определяя $\mathbb{S}^{-1} :\equiv \mathbf{0}$ и замечая, что $\Sigma\mathbf{0} \simeq \mathbf{2}$.

Чтобы доказать, что это согласуется с определением $\mathbb{S}^n$ из предыдущего раздела, потребовалось бы сделать последнее более явным. Однако можно показать, что рекурсивное определение имеет то же универсальное свойство, которое мы ожидаем от другого, подобного ему. Если (A, a_0) и (B, b_0) являются точечными типами (с отмеченными точками, которые часто остаются неявными), пусть $\mathsf{Map}_*(A, B)$ обозначает тип, базирующийся на отображениях:

$$\mathsf{Map}_*(A, B) :\equiv \sum_{f:A\to B} (f(a_0) = b_0)\,.$$

Заметим, что любой тип A лежит в основе точечного типа $A_+ :\equiv A + \mathbf{1}$ с отмеченной точкой $\mathsf{inr}(\star)$; это называется *примыканием к дизъюнктной отмеченной точке*.

Лемма 6.5.3. *Для типа A и точечного типа (B, b_0) имеет место*

$$\mathsf{Map}_*(A_+, B) \simeq (A \to B)\,.$$

Заметим, что справа мы имеем обычный тип несвязанных функций от A к B.

Доказательство. Слева направо, для $f : A_+ \to B$ с $p : f(\mathsf{inr}(\star)) = b_0$, имеем $f \circ \mathsf{inl} : A \to B$. А справа налево, для $g : A \to B$ определим $g' : A_+ \to B$ через $g'(\mathsf{inl}(a)) :\equiv g(a)$ и $g'(\mathsf{inr}(u)) :\equiv b_0$. Мы оставляем читателю показать, что эти операции — квазиобратные. □

В частности, отметим, что $\mathbf{2} \simeq \mathbf{1}_+$. Таким образом, для любого точечного типа B имеем

$$\mathsf{Map}_*(\mathbf{2}, B) \simeq (\mathbf{1} \to B) \simeq B\,.$$

Напомним, что операция пространства петель Ω действует на точечные типы с определением $\Omega(A, a_0) :\equiv (a_0 =_A a_0, \mathsf{refl}_{a_0})$. Мы также можем реализовать действие надстройки Σ на точечные типы посредством $\Sigma(A, a_0) :\equiv (\Sigma A, \mathsf{N})$.

Лемма 6.5.4. *Для точеченых типов (A, a_0) и (B, b_0) имеет место*

$$\mathsf{Map}_*(\Sigma A, B) \simeq \mathsf{Map}_*(A, \Omega B).$$

Доказательство. Для начала, имеется следующая цепочка эквивалентностей:

$$\begin{aligned}
\mathsf{Map}_*(\Sigma A, B) &:\equiv \sum_{f:\Sigma A \to B} (f(\mathsf{N}) = b_0) \\
&\simeq \sum_{f:\sum_{(b_n:B)} \sum_{(b_s:B)} (A \to (b_n = b_s))} (\mathsf{pr}_1(f) = b_0) \\
&\simeq \sum_{(b_n:B)} \sum_{(b_s:B)} A \to (b_n = b_s) \times (b_n = b_0) \\
&\simeq \sum_{(p:\sum_{(b_n:B)} (b_n = b_0))} \sum_{(b_s:B)} (A \to (\mathsf{pr}_1(p) = b_s)) \\
&\simeq \sum_{(b_s:B)} (A \to (b_0 = b_s)) \,.
\end{aligned}$$

Первая эквивалентность есть универсальное свойство надстроек, в котором показывается, что

$$(\Sigma A \to B) \simeq \left(\sum_{(b_n:B)} \sum_{(b_s:B)} (A \to (b_n = b_s)) \right)$$

с функцией, действующей справа налево, заданной рекурсором (см. упражнение 6.11). Вторая и третья эквивалентности построены согласно упражнению 2.10, с переупорядочением компонентов. Наконец, последняя эквивалентность следует из леммы 3.11.9, так как по лемме 3.11.8 $\sum_{(b_n:B)}(b_n = b_0)$ является стягиваемой с центром $(b_0, \mathsf{refl}_{b_0})$.

Доказательство теперь завершается следующей цепочкой эквивалентностей:

$$\begin{aligned}
\sum_{(b_s:B)} (A \to (b_0 = b_s)) &\simeq \sum_{(b_s:B)} \sum_{(g:A \to (b_0 = b_s))} \sum_{(q:b_0 = b_s)} (g(a_0) = q) \\
&\simeq \sum_{(r:\sum_{(b_s:B)} (b_0 = b_s))} \sum_{(g:A \to (b_0 = \mathsf{pr}_1(r)))} (g(a_0) = \mathsf{pr}_2(r)) \\
&\simeq \sum_{(g:A \to (b_0 = b_s))} (g(a_0) = \mathsf{refl}_{b_0}) \\
&\equiv \mathsf{Map}_*(A, \Omega B) \,.
\end{aligned}$$

Как и выше, первая и последняя эквивалентности представлены леммами 3.11.8 и 3.11.9, а вторая — упражнением 2.10 и переупорядочением компонентов. □

В частности, для сфер, определенных в (6.5.2), имеем

$$\mathsf{Map}_*(\mathbb{S}^n, B) \simeq \mathsf{Map}_*(\mathbb{S}^{n-1}, \Omega B) \simeq \cdots \simeq \mathsf{Map}_*(\mathbf{2}, \Omega^n B) \simeq \Omega^n B \,.$$

Таким образом, сферы $\mathbb{S}^n$ обладают универсальным свойством, которое мы ожидаем от сфер, определенных непосредственно через n-кратные пространства петель, как в §6.4.

6.6 Клеточные комплексы

В классической топологии *клеточный комплекс* представляет собой пространство, полученное путем последовательного прикрепления дисков вдоль своих границ. Он называется *CW-комплексом*, если граница n-мерного диска ограничена в дисках размерности, строго меньшей n ($(n-1)$-скелетон).

Любой конечный CW-комплекс может быть представлен как высший индуктивный тип, путем превращения n-мерных дисков в n-мерные пути и разбиения образа прикрепляющего отображения на источник и цель, причем каждый из них записывается как совокупность путей более низкой размерности. Наши явные определения $\mathbb{S}^1$ и $\mathbb{S}^2$ в §6.4 как раз и имели такую форму.

Другим примером является тор T^2, который порождается:

- точкой $b : T^2$,
- путем $p : b = b$,
- другим путем $q : b = b$ и
- 2-путем $t : p \cdot q = q \cdot p$.

Возможно, проще всего понять, что это тор, — это начать с прямоугольника с четырьмя углами a, b, c, d, четырьмя ребрами p, q, r, s и содержимым, которое явно является 2-путем t от $p \cdot q$ к $r \cdot s$:

$$\begin{tikzcd}
a \arrow[r, equal, "p"] \arrow[d, equal, "r"'] \arrow[dr, phantom, "\Downarrow t"] & b \arrow[d, equal, "q"] \\
c \arrow[r, equal, "s"'] & d
\end{tikzcd}$$

Теперь отождествим ребро r с ребром q и ребро s с ребром p, в результате получим также отождествление всех четырех углов. Топологически это отождествление может быть использовано для получения тора.

Принцип индукции для тора является самым сложным из всех ранее сформулированных. Для $P : T^2 \to \mathcal{U}$ и сечения $\prod_{(x:T^2)} P(x)$ требуются:

- точка $b' : P(b)$,
- путь $p' : b' =^P_p b'$,
- путь $q' : b' =^P_p b'$ и
- 2-путь t' между «совокупностями» $p' \cdot q'$ и $q' \cdot p'$, лежащими над t.

Чтобы понять последнее требование, нам требуется операция композиции для зависимых путей, но ее не сложно определить. Тогда принцип индукции дает функцию $f : \prod_{(x:T^2)} P(x)$ такую, что $f(b) \equiv b'$ с $\mathsf{apd}_f(p) = p'$ и $\mathsf{apd}_f(q) = q'$, а также что-то вроде «$\mathsf{apd}^2_f(t) = t$'». Однако это не так хорошо типизировано, как хотелось бы, во-первых, потому что равенства $\mathsf{apd}_f(p) = p'$ и $\mathsf{apd}_f(q) = q'$ не являются дефинициальными, а во-вторых, потому что apd_f сохраняет конкатенацию путей только с точностью до гомотопии. Мы оставляем детали читателю (см. упражнение 6.1).

Разумеется, другое определение тора есть $T^2 :\equiv \mathbb{S}^1 \times \mathbb{S}^1$ (в упражнении 6.3 мы просим читателя проверить эквивалентность двух определений тора). Однако определение клеточного комплекса легко обобщается на другие пространства без использования таких описаний, так как бутылка Клейна, проективная плоскость и т.д. Но становится все труднее формулировать принципы индукции, требующие определения понятий зависимых n-путей и apd, действующих на n-пути. К счастью, когда у нас в распоряжении имеются сферы, есть способ обойти эти сложности.

6.7 Концентраторы и спицы

В топологии обычно говорят о построении CW-комплексов путем присоединения n-мерных дисков вдоль их $(n-1)$-мерных граничных сфер. Однако, другой способ выразить это — склеивание в *конусе* на $(n-1)$-мерной сфере. То есть, мы рассматриваем диск как состоящий из конусной точки (или «концентратора») с меридианами (или «спицами»), соединяющими эту точку с каждой точкой на границе, непрерывно, как показано на рис.6.3.

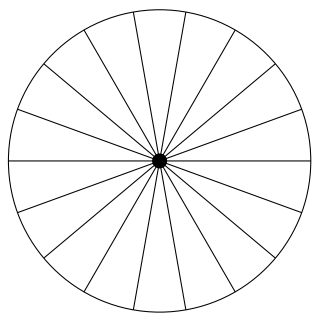


Рис. 6.3: 2-диск, составленный из концентратора и спиц

Мы можем использовать эту идею для выражения высших индуктивных типов, содержащих конструкторы n-мерных путей для $n > 1$ в терминах единиц, содержащих только конструкторы одномерных путей. Дело в том, что мы можем получить n-мерный путь как непрерывное семейство одномерных путей, параметризованных $(n-1)$-мерным объектом. Простейшим $(n-1)$-мерным объектом для использования является $(n-1)$-сфера, хотя в некоторых случаях может быть предпочтительным другой объект (напомним, что мы смогли индуктивно определить сферы в §6.5, используя надстройки, которые включают только конструкторы одномерных путей; действительно, надстройка также может рассматриваться как пример этой идеи, поскольку она включает в себя семейство одномерных путей, параметризованные типом, который является надстроечным).

Например, в отличие от определения из предыдущего раздела, тор T^2 может быть порожден:

- точкой $b : T^2$,
- путем $p : b = b$,
- другим путем $q : b = b$ и
- для каждого $x : \mathbb{S}^1$, путем $s(x) : f(x) = h$, где $f : \mathbb{S}^1 \to T^2$ определяется посредством $f(\mathsf{base}) :\equiv b$ и $f(\mathsf{loop}) := p \cdot q \cdot p^{-1} \cdot q^{-1}$.

Принцип индукции этой версии тора гласит, что для $P : T^2 \to \mathcal{U}$ и сечения $\prod_{(x:T^2)} P(x)$ требуются:

- точка $b' : P(b)$,

- путь $p' : b' =^P_p b'$,
- путь $q' : b' =^P_q b'$,
- точка $h' : P(h)$ и
- для каждого $x : \mathbb{S}^1$, путь $g(x) =^P_{s(x)} h'$, где $g : \prod_{(x:\mathbb{S}^1)} P(f(x))$ определяется через $g(\mathsf{base}) :\equiv b'$ и $\mathsf{apd}_g(\mathsf{loop}) := p' \cdot q' \cdot (p')^{-1} \cdot (q')^{-1}$.

Заметим, что нет необходимости в зависимых 2-путях или apd^2. Мы оставляем читателю выписывание соответствующих правил вычислений.

Замечание 6.7.1. Можно задать вопрос о необходимости введения концентратора h; нельзя ли вместо этого просто добавлять пути, непрерывно связывающие границу диска с точкой на этой границе, как показано на рисунке 6.4? Однако это не работает без дополнительных изменений. Если для некоторой $f : \mathbb{S}^1 \to X$ мы предоставляем конструктор путей, соединяющий каждую $f(x)$ с $f(\mathsf{base})$, то, что мы получаем, больше похоже на изображение на рисунке 6.5 конуса, вершина которого скручивается и приклеивается к какой-то точке на ее основании. Проблема в том, что указанный путь от $f(\mathsf{base})$ к себе может не быть рефлексивностью. Мы могли бы устранить эту проблему, добавив конструктор двумерного пути, чтобы обеспечить это, но использование отдельного концентратора позволяет избежать необходимости в каких-либо конструкторах путей размерностью выше 1.

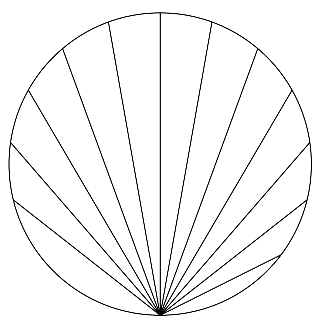


Рис. 6.4: Спицы без концентратора

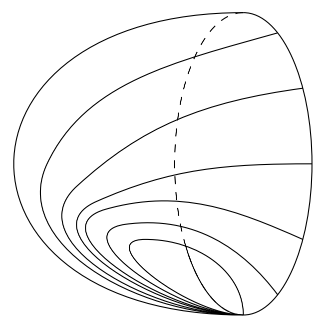


Рис. 6.5: Спицы без концентратора, II

Замечание 6.7.2. Обратите также внимание на то, что «перевод» высших путей в 1-пути не сохраняет дефинициальные правила вычислений для этих путей, хотя сохраняет пропозициональные правила.

6.8 Амальгамы

С теоретико-категорной точки зрения, одним из важных аспектов любой основополагающей системы является способность создавать пределы и копределы. В теоретико-множественных основаниях — это пределы и копределы множеств, тогда как в нашем случае они являются пределами и копределами *типов*. В §2.15 мы видели, что типы декартовых произведений имеют корректное универсальное свойство категорного произведения типов, а в упражнении 2.9, — что типы копроизведений также имеют ожидаемое универсальное свойство.

Как отмечалось в §2.15, более общие пределы могут быть построены с использованием типов тождественности и $\sum$-типов, например, обратный образ $f : A \to C$ и $g : B \to C$ —

это $\sum_{(a:A)} \sum_{(b:B)} (f(a) = g(b))$ (см. упражнение 2.11). Однако более общие *копределы* требуют идентификации элементов, поступающих из разных типов, для которых более высшие индуктивности хорошо адаптированы. Так как все наши конструкции гомотопчески инвариантны, все наши копределы обязательно являются *гомотопическими копределами*, но мы опускаем вездесущее прилагательное в интересах конкретизации.

В этом разделе мы обсудим *амальгамы* (другие названия: *расслоенные копроизведения*, *кодекартовы квадраты*, – *прим.перев.*), как, возможно, самый простой и один из самых полезных копределов. Действительно, можно ожидать, что все конечные копределы (для подходящего гомотопического определения «конечный») будут построены из амальгам и конечных копроизведений. Также возможно предложить прямую конструкцию более общих копределов с использованием высших индуктивных типов, но это отчасти технический аспект, к тому же не вполне удовлетворительный, так как пока отсутствует хорошее общее представление гомотопических когерентных диаграмм.

Предположим, что задан *пролет* типов и функций:

$$\mathscr{D} = \begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] & B \\ A & \end{tikzcd}$$

Амальгама этого пролета является высшим индуктивным типом $A \sqcup^C B$, представленным

- функцией $\mathsf{inl} : A \to A \sqcup^C B$,
- функцией $\mathsf{inr} : B \to A \sqcup^C B$ и
- для каждого $c : C$, путем $\mathsf{glue}(c) : (\mathsf{inl}(f(c)) = \mathsf{inr}(g(c)))$.

Другими словами, $A \sqcup^C B$ является несвязным объединением A и B, а также, для каждого $c : C$, свидетельством того, что $f(c)$ и $g(c)$ равны. Принцип рекурсии гласит, что если D — другой тип, то можно определить отображение $s : A \sqcup^C B \to D$ через

- значение $s(\mathsf{inl}(a)) : D$ для каждого $a : A$,
- значение $s(\mathsf{inr}(b)) : D$ для каждого $b : B$ и
- значение $\mathsf{ap}_s(\mathsf{glue}(c)) : s(\mathsf{inl}(f(c))) = s(\mathsf{inr}(g(c)))$ для каждого $c : C$.

Мы оставляем читателю формулировку принципа индукции. Также подразумевается выполнение принципа уникальности: если $s, s' : A \sqcup^C B \to D$ — две отображения такие, что

$$\begin{aligned} s(\mathsf{inl}(a)) &= s'(\mathsf{inl}(a)) \\ s(\mathsf{inr}(b)) &= s'(\mathsf{inr}(b)) \\ \mathsf{ap}_s(\mathsf{glue}(c)) &= \mathsf{ap}_{s'}(\mathsf{glue}(c)) \qquad \text{(по модулю предыдущих двух равенств)} \end{aligned}$$

для любых a, b, c, то $s = s'$.

Чтобы сформулировать универсальное свойство амальгамы, введем следующее

Определение 6.8.1. Для пролета $\mathscr{D} = (A \xleftarrow{f} C \xrightarrow{g} B)$ и типа D **коконус под** $\mathscr{D}$ **с вершиной** D состоит из функций $i : A \to D$ и $j : B \to D$, и гомотопии $h : \prod_{(c:C)}(i(f(c)) = j(g(c)))$:

$$\begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "\nearrow h"] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] & D \end{tikzcd}$$

Обозначим через $\mathsf{cocone}_{\mathscr{D}}(D)$ тип всех таких коконусов, т.е

$$\mathsf{cocone}_{\mathscr{D}}(D) :\equiv \sum_{(i:A\to D)} \sum_{(j:B\to D)} \prod_{(c:C)} (i(f(c)) = j(g(c)))\,.$$

Конечно, существует канонический коконус под $\mathscr{D}$ с вершиной $A \sqcup^C B$, состоящий из inl, inr и glue.

$$\begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "\nearrow \mathsf{glue}"] & B \arrow[d, "\mathsf{inr}"] \\ A \arrow[r, "\mathsf{inl}"'] & A \sqcup^C B \end{tikzcd}$$

Следующая лемма утверждает, что такой коконус является универсальным.

Лемма 6.8.2. *Для любого типа E существует эквивалентность*

$$(A \sqcup^C B \to E) \simeq \mathsf{cocone}_{\mathscr{D}}(E)\,.$$

Доказательство. Рассмотрим произвольный тип $E : \mathcal{U}$. Существует каноническая функция $c_\sqcup$, определяемая формулами

$$\begin{aligned} (A \sqcup^C B \to E) &\longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ t &\longmapsto (t \circ \mathsf{inl}, t \circ \mathsf{inr}, \mathsf{ap}_t \circ \mathsf{glue}) \end{aligned}$$

Мы записываем неформально $t \mapsto t \circ c_\sqcup$ для этой функции. Покажем, что она является эквивалентностью.

В первую очередь, для $c = (i, j, h) : \mathsf{cocone}_{\mathscr{D}}(E)$ нам нужно построить отображение $\mathsf{s}(c)$ от $A \sqcup^C B$ к E.

$$\begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] \arrow[ur, phantom, "\overset{h}{\nearrow}"] & E \end{tikzcd}$$

Отображение $s(c)$ определяется следующим образом

$$\begin{aligned} \mathsf{s}(c)(\mathsf{inl}(a)) &:\equiv i(a)\,, \\ \mathsf{s}(c)(\mathsf{inr}(b)) &:\equiv j(b)\,, \\ \mathsf{ap}_{\mathsf{s}(c)}(\mathsf{glue}(x)) &:= h(x)\,. \end{aligned}$$

Определим отображение

$$\begin{aligned} \mathsf{cocone}_{\mathscr{D}}(E) &\longrightarrow (A \sqcup^C B \to E) \\ c &\longmapsto \mathsf{s}(c) \end{aligned}$$

и нам нужно доказать, что это отображение является обратным к $t \mapsto t \circ c_\sqcup$. С одной стороны, если $c = (i, j, h) : \mathsf{cocone}_{\mathscr{D}}(E)$, то имеем

$$\begin{aligned} \mathsf{s}(c) \circ c_\sqcup &= (\mathsf{s}(c) \circ \mathsf{inl}, \mathsf{s}(c) \circ \mathsf{inr}, \mathsf{ap}_{\mathsf{s}(c)} \circ \mathsf{glue}) \\ &= (\lambda a.\, \mathsf{s}(c)(\mathsf{inl}(a)), \lambda b.\, \mathsf{s}(c)(\mathsf{inr}(b)), \lambda x.\, \mathsf{ap}_{\mathsf{s}(c)}(\mathsf{glue}(x))) \\ &= (\lambda a.\, i(a), \lambda b.\, j(b), \lambda x.\, h(x)) \\ &\equiv (i, j, h) \\ &= c\,. \end{aligned}$$

С другой стороны, если $A \sqcup^C B \to E$, то мы хотим доказать, что $\mathsf{s}(t \circ c_\sqcup) = t$. Для $a : A$ имеем

$$\mathsf{s}(t \circ c_\sqcup)(\mathsf{inl}(a)) = t(\mathsf{inl}(a))\,,$$

потому что первая компонента $t \circ c_\sqcup$ есть $t \circ \mathsf{inl}$. Точно так же для $b : B$ имеем

$$\mathsf{s}(t \circ c_\sqcup)(\mathsf{inr}(b)) = t(\mathsf{inr}(b))\,,$$

и для $x : C$ имеем

$$\mathsf{ap}_{\mathsf{s}(t \circ c_\sqcup)}(\mathsf{glue}(x)) = \mathsf{ap}_t(\mathsf{glue}(x))\,,$$

следовательно, $\mathsf{s}(t \circ c_\sqcup) = t$.

Это доказывает, что отображение $c \mapsto \mathsf{s}(c)$ является квази-обратным к $t \mapsto t \circ c_\sqcup$, что и требовалось. □

Некоторые стандартные теоретико-гомотопические конструкции могут быть выражены как (гомотопические) амальгамы.

- Амальгама пролета $\mathbf{1} \leftarrow A \rightarrow \mathbf{1}$ есть **надстройка** $\sum A$ (см. §6.5).

- Амальгама для $A \xleftarrow{\mathsf{pr}_1} A \times B \xrightarrow{\mathsf{pr}_2} B$ называется **соединением** A и B, записываемое как $A * B$.

- Амальгама для $\mathbf{1} \leftarrow A \xrightarrow{f} B$ есть **конус** или **ко-слой** для f.

- Если A и B оснащены отмеченными точками $a_0 : A$ и $b_0 : B$, то амальгама для $A \xleftarrow{a_0} \mathbf{1} \xrightarrow{b_0}$ B является **клином** $A \vee B$.

- Если A и B такие же, как в предыдущем пункте, определим $f : A \vee B \to A \times B$ посредством $f(\mathsf{inl}(a)) :\equiv (a, b_0)$ и $f(\mathsf{inr}(b)) :\equiv (a_0, b)$ с $f(\mathsf{glue}) := \mathsf{refl}_{(a_0,b_0)}$. Тогда конус для f называется **скрещенным произведением** $A \wedge B$.

Мы обсудим амальгамы далее в главах 7 и 8.

Замечание 6.8.3. Как отмечалось в §3.7, обозначения $\wedge$ и $\vee$ для скрещенного произведения и клина точечных пространств также используются в логике для «и» и «или» соответственно. Поскольку типы в гомотопической теории типов могут вести себя как пространства или как высказывания, технически существует возможность для конфликта, но поскольку они редко используются совместно, контекст в целом устраняет неоднозначность. Кроме того, скрещенное произведение и клин применяются только к *точеченым* пространствам, в то время как единственное точеченое простое высказывание есть $\top \equiv \mathbf{1}$ — и мы имеем $\mathbf{1} \wedge \mathbf{1} = \mathbf{1}$ и $\mathbf{1} \vee \mathbf{1} = \mathbf{1}$ для обоих значений $\wedge$ и $\vee$.

Замечание 6.8.4. Обратите внимание, что копределы вообще не сохраняют усеченности. Например, $\mathbb{S}^0$ и $\mathbf{1}$ — оба множества, но амальгама для $\mathbf{1} \leftarrow \mathbb{S}^0 \rightarrow \mathbf{1}$ есть $\mathbb{S}^1$, не являющееся множеством. Если нас интересуют копределы в категории n-типов, поэтому (и, в частности, в категории множеств), нам нужно как-то «усечь» копредел. Мы вернемся к этому в §6.9 и главах 7 и 10.

6.9 Усечения

В §3.7 мы ввели пропозициональное усечение как операцию формирования нового типа; теперь же отметим, что ее можно получить как частный случай высших индуктивных типов. Это сводит проблему понимания усечений к проблеме понимания высших индуктивностей, которые, по крайней мере, поддаются систематической трактовке. Это к тому же интересно, поскольку дает наш первый пример высшого индуктивного типа, который действительно *рекурсивен*, по той причине, что его конструкторы принимают входные данные от определенного типа (как и преемник $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$).

Пусть A — тип; определим его пропозициональное усечение $\|A\|$ как высший индуктивный тип, порожденный:

- функцией $|\_| : A \to \|A\|$ и

- для любых $x, y : \|A\|$, путем $x = y$.

Заметим, что второй конструктор по определению является утверждением, что $\|A\|$ — простое высказывание. Таким образом, определение $\|A\|$ можно интерпретировать, как выражающее, что $\|A\|$ свободно порождается функцией $A \to \|A\|$, и тем, что это простое высказывание.

Принцип рекурсии для этого высшего индуктивного определения легко описать: в нем говорится, что для любого типа B вместе с

- функцией $g : A \to B$ и
- путем $x =_B y$ для любых $x, y : B$,

существует функция $f : \|A\| \to B$ такая, что

- $f(|a|) \equiv g(a)$ для всех $a : A$ и
- для любых $x, y : \|A\|$ функция ap_f применяет определенный путь $x = y$ из A к определенному пути $f(x) = f(y)$ из B (пропозиционально).

Это именно те гипотезы, которые были сформулированы в §3.7 для принципа рекурсии пропозиционального усечения — функция $A \to B$ такая, что B — простое высказывание, а первая часть заключения — это именно то, что мы там изложили. Вторая часть (действие ap_f) ранее не упоминалась, но в данном случае она оказывается пустой, так как B — простое высказывание, поэтому любые два пути в ней автоматически равны.

Существует также принцип индукции для $\|A\|$, который гласит, что для любого $B : \|A\| \to \mathcal{U}$ вместе с

- функцией $g : \prod_{(a:A)} B(|a|)$ и
- зависимым путем $q : u =^B_{p(x,y)} v$, где $p(x, y)$ — путь, идущий от второго конструктора $\|A\|$, для любых $x, y : \|A\|$, $u : B(x)$ и $v : B(y)$,

существует $f : \prod_{(x:\|A\|)} B(x)$ такая, что $f(|a|) \equiv g(a)$ для $a : A$, а также другое правило вычисления. Однако, поскольку между любыми двумя простыми высказываниями (с точностью до гомотопии) может находиться не более одной функции, этот принцип индукции не очень полезен (см. также упражнение 3.17).

Однако, можно распространить эту идею на построение подобных усечений, размещением их в n-типах для любого n. Например, мы могли бы определить 0-*усечение* $\|A\|_0$, которое должно быть порождено

- функцией $|\_|_0 : A \to \|A\|_0$ и,
- для любых $x, y : \|A\|_0$ и любых $p, q : x = y$, путем $p = q$.

Тогда $\|A\|_0$ будет свободно порождаться функцией $A \to \|A\|_0$ вместе с утверждением, что $\|A\|_0$ является множеством. Естественный принцип индукции для него будет утверждать, что для $B : \|A\|_0 \to \mathcal{U}$ вместе с

- функцией $g : \prod_{(a:A)} B(|a|_0)$ и,
- для любых $x, y : \|A\|_0$ с $z : B(x)$ и $w : B(y)$, и любых $p, q : x = y$ с $r : z =^B_p w$ и $s : z =^B_q w$, 2-путем $v : r =^{z =^B_- w}_{u(x,y,p,q)} s$, где $u(x, y, p, q) : p = q$ получен из второго конструктора $\|A\|_0$,

существует функция $f : \prod_{(x:\|A\|_0)} B(x)$ такая, что $f(|a|_0) \equiv g(a)$ для всех $a : A$, а также $\mathsf{apd}^2_f(u(x, y, p, q))$ — 2-путь, указанный выше (как и в пропозициональном случае, последнее условие оказывается неинтересным). Однако из этого можно доказать более полезный принцип индукции.

Лемма 6.9.1. *Пусть задано* $B : \|A\|_0 \to \mathcal{U}$ *вместе с* $g : \prod_{(a:A)} B(|a|_0)$ *и предположим, что каждое* $B(x)$ *является множеством. Тогда существует* $f : \prod_{(x:\|A\|_0)} B(x)$ *такая, что* $f(|a|_0) \equiv g(a)$ *для всех* $a : A$.

Доказательство. Достаточно построить для любых x, y, z, w, p, q, r, s, как указано выше, 2-путь $v : r =^B_{u(x,y,p,q)} s$. Однако, по определению зависимых 2-путей, это обычный 2-путь в слое $B(y)$. Поскольку $B(y)$ — множество, то существует 2-путь между любыми двумя параллельными путями. □

Это подразумевает ожидаемое универсальное свойство.

Лемма 6.9.2. *Для любого множества* B *и любого типа* A *композиция с* $|\_|_0 : A \to \|A\|_0$ *определяет эквивалентность*

$$(\|A\|_0 \to B) \simeq (A \to B)\,.$$

Доказательство. Частный случай леммы 6.9.1, когда B — семейство констант, дает отображение справа налево, которое является правым обратным к функции «скомпоновать с $|\_|_0$» слева направо. Чтобы показать, что оно также является левым обратным, пусть $h : \|A\|_0 \to B$ и определим $h' : \|A\|_0 \to B$, применяя лемму 6.9.1 к совокупности $a \mapsto h(|a|_0)$. Таким образом, $h'(|a|_0) = h(|a|_0)$.

Однако, так как B — множество, то для любого $x : \|A\|_0$ тип $h(x) = h'(x)$ является простым высказыванием, а, следовательно, также и множеством. Поэтому по лемме 6.9.1 наблюдение, что $h'(\|\, a_0) = h(|a|_0)$ для любого $a : A$, влечет $h(x) = h'(x)$ для любого $x : \|A\|_0$ и, следовательно, $h = h'$. □

К примеру, это позволяет нам строить копределы множеств. Мы видели, что если $A \xleftarrow{f} C \xrightarrow{g} B$ является пролетом множеств, то амальгама $A \sqcup^C B$ больше не может быть множеством (например, если A и B являются $\mathbf{1}$, а C — $\mathbf{2}$, то амальгама есть $\mathbb{S}^1$). Однако, мы можем построить амальгаму, которая является множеством, и имеет ожидаемое универсальное свойство по отношению к другим множествам, путем усечения,

Лемма 6.9.3. *Пусть* $A \xleftarrow{f} C \xrightarrow{g} B$ *— пролет множеств. Тогда для любого множества* E *существует каноническая эквивалентность*

$$\left(\left\|A \sqcup^C B\right\|_0 \to E\right) \simeq \mathsf{cocone}_{\mathscr{D}}(E)\,.$$

Доказательство. Компоновка эквивалентностей из лемм 6.8.2 и 6.9.2. □

Мы относимся к $\left\|A \sqcup^C B\right\|_0$ как к теоретико-множественной амальгаме для f и g, чтобы отличить его от (гомотопической) амальгамы $A \sqcup^C B$. В качестве альтернативы можно изменить определение амальгамы в §6.8, чтобы напрямую включить конструктор 0-усечения, избегая необходимости усечения впоследствии. Подобные замечания применимы к любому типу копределов множеств; мы рассмотрим это далее в главе 10.

Однако, хотя вышеописанное определение 0-усечения работает — оно дает то, что мы хотим, и оно прямолинейное — у него имеется пара проблем. Во-первых, оно не так хорошо вписывается в общую теорию высших индуктивных типов. В общем, сложно взаимодействовать с конструкторами, такими как второй, который мы ввели для $\|A\|_0$, *входные данные* которого включают не только элементы определенного типа, но и пути в нем.

Тем не менее, этого довольно легко добиться. В §5.1 мы упоминали, что можем разрешить конструктору индуктивного типа W принимать «бесконечно много аргументов» типа W, если он принимает один аргумент типа $\mathbb{N} \to W$. Существует общий принцип: для моделирования конструктора с причудливыми входными данными использовать вспомогательный индуктивный тип (например, $\mathbb{N}$) для их параметризации, сводя входные данные к простой функции с индуктивной областью.

Для 0-усечения можно рассмотреть вспомогательный *высший* индуктивный тип S, порожденный двумя точками $a, b : S$ и двумя путями $p, q : a = b$. Тогда невыразительный конструктор для $\|A\|_0$ может быть заменен на приемлемый

- Для каждой $f : S \to \|A\|_0$ путь $\mathsf{ap}_f(p) = \mathsf{ap}_f(q)$.

Так как построение отображения от S эквивалентно взятию двух точек и двух параллельных путей между ними, это приводит к тому же принципу индукции.

Однако более серьезная проблема с текущим определением 0-усечения заключается в том, что оно не очень хорошо обобщается. Если мы хотим построить понятие определения «n-усечения» в n-типах равномерно для всех $n : N$, то этот подход невыполним, так как второму конструктору понадобится число аргументов, увеличивающееся с ростом n. Поэтому в §7.3 мы будем использовать другую идею для их построения, основываясь на наблюдении, что введенный выше тип S эквивалентен окружности $\mathbb{S}^1$. Это включает в себя 0-усечение как частный случай и удовлетворяет обобщенным версиям лемм 6.9.1 и 6.9.2.

6.10 Частные

Особенно важным видом копредела множеств является *частное* по отношению. То есть, пусть A — множество и $R : A \times A \to \mathsf{Prop}$ — семейство простых высказываний (**простое отношение**). Его частным должно быть множество-коуравнитель двух проекций

$$\textstyle\sum_{(a,b:A)} R(a,b) \rightrightarrows A\,.$$

Мы можем также описать это непосредственно, как высший индуктивный тип A/R, порожденный

- функцией $q : A \to A/R$;
- для любых $a, b : A$ таких, что $R(a,b)$, равенством $q(a) = q(b)$; и
- конструктором 0-усечения: для всех $x, y : A/R$ и $r, s : x = y$ имеем $r = s$.

Мы будем иногда ссылаться на этот высший индуктивный тип A/R как на множество-частное A по R, чтобы подчеркнуть, что он порождает множество по определению (в теории гомотопий имеются более общие понятия «частного», но они в основном выходят за рамки данной книги; однако в §9.9 мы рассмотрим «частное» типа по 1-группоиду, которое появляется на следующем уровне от множеств-частных).

Замечание 6.10.1. На самом деле нет необходимости в определении множеств-частных и большинства их свойств, поскольку A является множеством. Однако, как правило, это и область наибольшего интереса.

Лемма 6.10.2. *Функция $q : A \to A/R$ сюръективна.*

Доказательство. Мы должны показать, что для любого $x : A/R$ просто существует $a : A$ с $q(a) = x$. Будем использовать принцип индукции для A/R. Начальный случай тривиален: если x есть $q(a)$, то, конечно, существует такое a, что $q(a) = q(a)$. И поскольку целью является простое высказывание, оно автоматически удовлетворяется всеми конструкторами путей, что и завершает доказательство. □

Теперь мы можем доказать, что множество-частное имеет ожидаемое универсальное свойство (множества-) коуравнителя.

Лемма 6.10.3. *Для любого множества B предварительная компоновка с q порождает эквивалентность*

$$(A/R \to B) \simeq \left(\sum_{(f:A\to B)} \prod_{(a,b:A)} R(a,b) \to (f(a) = f(b)) \right).$$

Доказательство. Квазиобратный для $\_ \circ q$, действующий справа налево, является просто принципом рекурсии для A/R. То есть, для $f : A \to B$ такой, что $\prod_{(a,b:A)} R(a,b) \to (f(a) = f(b))$, определим $\bar{f} : A/R \to B$ через $\bar{f}(q(a)) :\equiv f(a)$. Это определяющее уравнение показывает, что $(f \mapsto \bar{f})$ является правым обратным к $(\_ \circ q)$.

Чтобы он также был левым обратным, необходимо показать, что для любого $g : A/R \to B$ и $x : A/R$ имеет место $g(x) = \overline{g \circ q}(x)$. Однако по лемме 6.10.2 существует такое a, что $q(a) = x$. Так как искомое равенство есть простое высказывание, можно считать, что такое a существует, и в этом случае $g(x) = g(q(a)) = \overline{g \circ q}(q(a)) = \overline{g \circ q}(x)$. □

Конечно, в классическом случае рассмотрения, R является **отношением эквивалентности**, т.е. имеем

- **рефлексивность** — $\prod_{(a:A)} R(a,a)$,
- **симметричность** — $\prod_{(a,b:A)} R(a,b) \to R(b,a)$ и
- **транзитивность** — $\prod_{(a,b,c:A)} R(a,b) \times R(b,c) \to R(a,c)$.

В этом случае множество-частное A/R обладает дополнительными хорошими свойствами, как мы увидим в §10.1: например, $R(a,b) \simeq (q(a) =_{A/R} q(b))$. Мы часто будем записывать отношение эквивалентности $R(a,b)$ в инфиксной форме $a \sim b$.

Частное по отношению эквивалентности также может быть построено другими способами. Теоретико-множественный подход состоит в том, чтобы рассматривать множество классов эквивалентности как подмножество степенного множества A. Мы можем подражать этой «непредикативной» конструкции и в теории типов.

Определение 6.10.4. Предикат $P : A \to \mathsf{Prop}$ является **классом эквивалентности** отношения $R : A \times A \to \mathsf{Prop}$, если существует такое $a : A$, что для всех $b : A$ имеем $R(a,b) \simeq P(b)$.

Поскольку R и P — простые высказывания, эквивалентность $R(a,b) \simeq P(b)$ — это то же самое, что и импликации $R(a,b) \to P(b)$ и $P(b) \to R(a,b)$. И, конечно, для любого $a : A$ мы имеем класс канонической эквивалентности $P_a(b) :\equiv R(a,b)$.

Определение 6.10.5. Обозначим

$$A /\!\!/ R :\equiv \{ P : A \to \mathsf{Prop} \mid P \text{ является классом эквивалентности для } R \}.$$

Функция $q' : A \to A /\!\!/ R$ определяется посредством $q'(a) :\equiv P_a$.

Теорема 6.10.6. *Для любого отношения эквивалентности R на A тип $A /\!\!/ R$ эквивалентен множеству-частному A/R.*

Доказательство. Во-первых, заметим, что если $R(a,b)$, то, поскольку R является отношением эквивалентности, имеем $R(a,c) \Leftrightarrow R(b,c)$ для любого $c : A$. Таким образом, $R(a,c) = R(b,c)$ по унивалентности, следовательно, $P_a = P_b$ по функциональной экстенсиональности, т.е. $q'(a) = q'(b)$. Поэтому по лемме 6.10.3 имеем индуцированное отображение $f : A/R \to A /\!\!/ R$ такое, что $f \circ q = q'$.

Покажем, что f инъективна и сюръективна, следовательно, является эквивалентностью. Сюръективность непосредственно вытекает из того факта, что q' сюръективна, что, в свою очередь, по существу, является истиной, по определению $A /\!\!/ R$. Для инъективности, если $f(x) = f(y)$, то, чтобы показать простоту высказывания $x = y$, в связи с сюръективностью q можно считать $x = q(a)$ и $y = q(b)$ для некоторых $a, b : A$. Тогда $R(a,c) = f(q(a))(c) = f(q(b))(c) = R(b,c)$ для любого $c : A$ и, в частности, $R(a,b) = R(b,b)$. Но $R(b,b)$ обитаемо, так как R является отношением эквивалентности, следовательно, $R(a,b)$. Таким образом, $q(a) = q(b)$ и, следовательно, $x = y$. □

В §10.1.3 мы дадим альтернативное доказательство этой теоремы. Заметим, что в отличие от A/R конструкция $A /\!\!/ R$ повышает уровень универсума: если $A : \mathcal{U}_i$ и $R : A \to A \to \mathsf{Prop}_{\mathcal{U}_i}$, то в определении $A /\!\!/ R$ мы также должны использовать $\mathsf{Prop}_{\mathcal{U}_i}$ для включения всех классов эквивалентности, так что $A /\!\!/ R : \mathcal{U}_{i+1}$. Конечно, можно избежать этого, если предположить выполнение аксиомы пропозиционального изменения размера из §3.5.

Замечание 6.10.7. Предыдущие две конструкции предоставляют частные в общем случае, но в отдельных случаях могут быть более легкие конструкции. Например, можно определить целые числа $\mathbb{Z}$ как множество-частное

$$\mathbb{Z} :\equiv (\mathbb{N} \times \mathbb{N})/\sim,$$

где $\sim$является отношением эквивалентности, определяемым как

$$(a,b) \sim (c,d) :\equiv (a + d = b + c).$$

Другими словами, пара (a,b) представляет целое число $a - b$. В этом случае, однако, существуют *канонические представители* классов эквивалентности: в форме $(n,0)$ или $(0,n)$.

В следующей лемме говорится, что когда такое происходит, не требуется общая конструкция частных (функция $r : A \to A$ называется **идемпотентом**, если $r \circ r = r$).

Лемма 6.10.8. *Предположим, что $\sim$ является отношением на множестве A, и существует такой идемпотент $r : A \to A$, что $(r(x) = r(y)) \simeq (x \sim y)$ для всех $x, y : A$ (это означает, что это отношение эквивалентности). Тогда тип*

$$(A/\sim) :\equiv \left(\sum_{x:A} r(x) = x\right)$$

удовлетворяет универсальному свойству множества-степени A по $\sim$ и, следовательно, эквивалентен ему. Другими словами, существует отображение $q : A \to (A/\sim)$ такое, что для любого множества B предварительная композиция с q индуцирует эквивалентность

$$\Big((A/\sim) \to B\Big) \simeq \left(\sum_{(g:A \to B)} \prod_{(x,y:A)} (x \sim y) \to (g(x) = g(y))\right). \tag{6.10.9}$$

Доказательство. Пусть $i : \prod_{(x:A)} r(r(x)) = r(x)$ — идемпотентность r. Отображение $q : A \to (A/\sim)$ определяется посредством $q(x) :\equiv (r(x), i(x))$. Заметим, что поскольку A — множество, то $q(x) = q(y)$ тогда и только тогда, когда $r(x) = r(y)$, следовательно (по предположению) тогда и только тогда, когда $x \sim y$. Определим отображение e слева направо в (6.10.9) как

$$e(f) :\equiv (f \circ q, \_)\,,$$

где знак подчеркивания _ означает следующее доказательство: если $x, y : A$ и $x \sim y$, то $q(x) = q(y)$, как отмечалось выше, следовательно, $f(q(x)) = f(q(y))$. Чтобы убедиться, что e является эквивалентностью, рассмотрим отображение e' в противоположном направлении, определяемом формулой

$$e'(g, s)(x, p) :\equiv g(x)\,.$$

Для любой $f : (A/\sim) \to B$

$$e'(e(f))(x, p) \equiv f(q(x)) \equiv f(r(x), i(x)) = f(x, p)\,,$$

где последнее равенство имеет место, потому что $p : r(x) = x$, а также $(x, p) = (r(x), i(x))$, потому что A — множество. Аналогично вычисляем

$$e(e'(g, s)) \equiv e(g \circ \mathsf{pr}_1) \equiv (g \circ \mathsf{pr}_1 \circ q, \_)\,.$$

Поскольку B — это множество, не нужно беспокоиться о части _, тогда как для первого компонента имеем

$$g(\mathsf{pr}_1(q(x))) \equiv g(r(x)) = g(x)\,,$$

где последнее равенство выполняется, поскольку $r(x) \sim x$, а g не нарушает $\sim$ по предположению s. $\square$

Следствие 6.10.10. *Пусть $p : A \to B$ — стягивающее отображение между множествами. Тогда B — частное для A по отношению эквивалентности $\sim$, определяемое посредством*

$$(a_1 \sim a_2) :\equiv (p(a_1) = p(a_2))\,.$$

Доказательство. Предположим, что $s : B \to A$ — сечение для p. Тогда $s \circ p : A \to$ A — идемпотент, удовлетворяющий условию леммы 6.10.8 для $\sim$, а s индуцирует изоморфизм из B в его множество неподвижных точек. $\square$

Замечание 6.10.11. Лемма 6.10.8 применима к $\mathbb{Z}$ с идемпотентом $r : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \times \mathbb{N}$, определяемым как

$$r(a, b) = \begin{cases} (a - b, 0) & \text{если } a \geqslant b, \\ (0, b - a) & \text{иначе} \end{cases}$$

(это корректное определение даже конструктивно, так как отношение $\sim$ на $\mathbb{N}$ разрешимо). Таким образом, неотрицательное целое число канонически представлено как $(k, 0)$ и неположительное — как $(0, m)$ для $k, m : \mathbb{N}$. Это деление на случаи подразумевает следующий «принцип индукции» для целых чисел, который будет полезен в главе 8 (как обычно, мы определяем натуральное число n соответствующим неотрицательным целым числом, то есть, образом $(n, 0) : \mathbb{N} \times \mathbb{N}$ в $\mathbb{Z}$):

Лемма 6.10.12. *Пусть $P : \mathbb{Z} \to \mathcal{U}$ — семейство типов и*

- $d_0 : P(0)$,
- $d_+ : \prod_{(n:\mathbb{N})} P(n) \to P(\mathsf{succ}(n))$ *и*
- $d_- : \prod_{(n:\mathbb{N})} P(-n) \to P(-\mathsf{succ}(n))$.

Тогда существует функция $f : \prod_{(z:\mathbb{Z})} P(z)$ *такая, что*

- $f(0) = d_0$,
- $f(\mathsf{succ}(n)) = d_+(n, f(n))$*для всех* $n : \mathbb{N}$ *и*
- $f(-\mathsf{succ}(n)) = d_-(n, f(-n))$*для всех* $n : \mathbb{N}$.

Доказательство. Для целей этого доказательства пусть $\mathbb{Z}$ обозначает $\sum_{(x:\mathbb{N}\times\mathbb{N})}(r(x) = x)$, где r — введенный выше идемпотент (мы можем тогда переносить результат на любое эквивалентное определение $\mathbb{Z}$). Пусть $q : \mathbb{N} \times \mathbb{N} \to \mathbb{Z}$ — отображение частных, определяемое через $q(x) = (r(x), i(x))$, как в лемме 6.10.8. Теперь определим $Q :\equiv P \circ q : \mathbb{N} \times \mathbb{N} \to \mathcal{U}$. Транспортируя данные по соответствующим равенствам, получим

$$\begin{aligned}
d'_0 &: Q(0,0)\,,\\
d'_+ &: \prod_{n:\mathbb{N}} Q(n,0) \to Q(\mathsf{succ}(n),0)\,,\\
d'_- &: \prod_{n:\mathbb{N}} Q(0,n) \to Q(0,\mathsf{succ}(n))\,.
\end{aligned}$$

Заметим также, что поскольку $q(n,m) = q(\mathsf{succ}(n), \mathsf{succ}(m))$, то имеем индуцированную эквивалентность

$$e_{n,m} : Q(n,m) \simeq Q(\mathsf{succ}(n), \mathsf{succ}(m))\,.$$

Тогда мы можем построить $g : \prod_{(x:\mathbb{N}\times\mathbb{N})} Q(x)$ двойной индукцией по x:

$$\begin{aligned}
g(0,0) &:\equiv d'_0\,,\\
g(\mathsf{succ}(n),0) &:\equiv d'_+(n, g(n,0))\,,\\
g(0,\mathsf{succ}(m)) &:\equiv d'_-(m, g(0,m))\,,\\
g(\mathsf{succ}(n),\mathsf{succ}(m)) &:\equiv e_{n,m}(g(n,m))\,.
\end{aligned}$$

Теперь имеем $\mathsf{pr}_1 : \mathbb{Z} \to \mathbb{N} \times \mathbb{N}$, с тем свойством, что $q \circ \mathsf{pr}_1 = \mathsf{id}$. В частности, поэтому, имеем $Q \circ \mathsf{pr}_1 = P$ и, следовательно, семейство эквивалентностей $s : \prod_{(z:\mathbb{Z})} Q(\mathsf{pr}_1(z)) \simeq P(z)$. Таким образом, можно определить $f(z) = s(z, g(\mathsf{pr}_1(z)))$, чтобы получить $f : \prod_{(z:\mathbb{Z})} P(z)$ и проверить требуемые равенства. □

Мы будем иногда обозначать функцию $f : \prod_{(z:\mathbb{Z})} P(z)$, полученную из леммы 6.10.12, используя синтаксис сопоставления с образцом, включающим три случая, d_0, d_+ и d_-:

$$\begin{aligned}
f(0) &:= d_0\,,\\
f(\mathsf{succ}(n)) &:= d_+(n, f(n))\,,\\
f(-\mathsf{succ}(n)) &:= d_-(n, f(-n))\,.
\end{aligned}$$

Мы используем $:=$, а не $:\equiv$, как это было сделано для конструкторов путей высших индуктивных типов, чтобы указать, что правила вычисления, указанные в лемме 6.10.12, являются только пропозициональными. Например, таким образом мы можем определить n-кратную конкатенацию петли для любого целого числа n.

Следствие 6.10.13. *Пусть A — тип с $a : A$ и $p : a = a$. Существует функция $\prod_{(n:\mathbb{Z})}(a = a)$, обозначаемая $n \mapsto p^n$, и определяемая следующим образом*

$$\begin{aligned} p^0 &:= \mathsf{refl}_a \\ p^{n+1} &:= p^n \cdot p && \text{для } n \geqslant 0 \\ p^{n-1} &:= p^n \cdot p^{-1} && \text{для } n \leqslant 0\,. \end{aligned}$$

Мы обсудим целые числа далее в §§ 6.11 и 11.1.

6.11 Алгебра

В дополнение к построению объектов высших размерностей, таких как сферы и клеточные комплексы, высшие индуктивные типы также очень полезны даже при работе только с множествами. Мы уже видели один пример в лемме 6.9.3: они позволили нам построить копредел для любой диаграммы множеств, что невозможно в базовой теории типов. Высшие индуктивные типы также очень полезны при изучении множеств, наделенных алгебраической структурой.

В качестве начального примера в этом разделе мы рассмотрим *группы*, которые знакомы большинству математиков и демонстрируют весьма важные феномены (и будут необходимы в последующих главах). Однако большинство из того, о чем мы будем говорить, одинаково хорошо применимо к любому сорту алгебраической структуры.

Определение 6.11.1. Моноид — это множество G вместе с

- функцией *умножения* $G \times G \to G$, записываемой в инфиксной форме как $(x, y) \mapsto x \cdot y$,
- *единичным* элементом $e : G$ таким, что
- $x \cdot e = x$ и $e \cdot x = x$, для любого $x : G$,
- $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, для любых $x, y, z : G$.

Группа — это моноид G вместе с

- *обратной* функцией $i : G \to G$, записываемой в виде $x \mapsto x^{-1}$, такой, что
- $x \cdot x^{-1} = e$ и $x^{-1} \cdot x = e$ для любого $x : G$.

Замечание 6.11.2. Обратите внимание, мы требуем, чтобы группа была множеством. Мы могли бы рассмотреть более общее понятие «∞-группы», которое не является множеством, но это увело бы нас дальше, чем это необходимо на данный момент. В соответствии с данным определением мы можем ожидать, что полученная «теория групп» будет вести себя аналогично тому, как это происходит в теоретико-множественной математике (с оговоркой, что, если мы не примем LEM, это будет «конструктивная» теория групп).

Пример 6.11.3. Натуральные числа $\mathbb{N}$ являются моноидом по сложению с единицей 0, а также — по умножению с единицей 1. Если мы определяем арифметические операции над целыми числами $\mathbb{Z}$ очевидным образом, то, как обычно, они являются группой по сложению и моноидом по умножению (и, конечно, кольцом). Например, если $u, v \in \mathbb{Z}$ представлены, соответственно, как (a, b) и (c, d), то $u + v$ представляется $(a + c, b + d)$, $-u$ представляется как (b, a), а uv представляется как $(ac + bd, ad + bc)$.

Пример 6.11.4. По существу, в §2.1 мы заметили, что если (A, a) — точечный тип, то его пространство петель $\Omega(A, a) :\equiv (a =_A a)$ имеет структуру группы, за исключением того, что оно вообще не является множеством. Оно должно быть «∞-группой» в смысле, указанном в замечании 6.11.2, но мы также можем сделать его группой с помощью усечения. В частности, определим **фундаментальную группу** для A, основанную на $a : A$, как

$$\pi_1(A, a) :\equiv \left\| \Omega(A.a) \right\|_0 .$$

Она наследует структуру группы; например, умножение $\pi_1(A, a) \times \pi_1(A, a) \to \pi_1(A, a)$ определяется двойной индукцией по усечению из конкатенации путей.

В более общем смысле n-ая **гомотопическая группа** для (A, a) есть $\pi_n(A, a) :\equiv \left\| \Omega^n(A.a) \right\|_0$. Тогда $\pi_n(A, a) := \pi_1(\Omega^{n-1}(A, a))$ для $n \geqslant 1$, так что это также группа (когда $n = 0$, мы имеем $\pi_0(A) \equiv \|A\|_0$, что не является группой). Более того, из аргумента Экманна-Хилтона (теорема 2.1.6) следует, что если $n \geqslant 2$, то $\pi_n(A, a)$ является *абелевой* группой, т.е. $x \cdot y = y \cdot x$ для всех x, y. Более углубленно эти группы будут изучаться в главе 8.

Одним из важных понятий в теории групп является понятие *свободной группы*, порожденной множеством или, более общо, группы, *представленной* образующими и соотношениями. В теории типов хорошо известно, что *некоторые* свободные алгебраические объекты могут быть определены с использованием *обычных* индуктивных типов. Например, свободный моноид на множестве A можно отождествить с типом $\mathsf{List}(A)$ *конечных списков* элементов из A, который индуктивно порождается

- конструктором $\mathsf{nil} : \mathsf{List}(A)$ и
- элементом $\mathsf{cons}(a, \ell) : \mathsf{List}(A)$ для всех $\ell : \mathsf{List}(A)$ и $a : A$.

Имеется очевидное включение $\eta : A \to \mathsf{List}(A)$, определяемое как $a \mapsto \mathsf{cons}(a, \mathsf{nil})$. Моноидная операция на $\mathsf{List}(A)$ является операцией конкатенации, определяемой рекурсивно:

$$\mathsf{nil} \cdot \ell :\equiv \ell$$
$$\mathsf{cons}(a, \ell_1) \cdot \ell_2 :\equiv \mathsf{cons}(a, \ell_1 \cdot \ell_2) .$$

Нетрудно доказать, используя принцип индукции для $\mathsf{List}(A)$, что $\mathsf{List}(A)$ является множеством, а конкатенация списков ассоциативна, с nil в качестве единицы. Таким образом, $\mathsf{List}(A)$ является моноидом.

Лемма 6.11.5. *Для любого множества A тип $\mathsf{List}(A)$ является свободным моноидом на A. Другими словами, для любого моноида G композиция с η является эквивалентностью*

$$\mathsf{hom}_{\mathsf{Monoid}}(\mathsf{List}(A), G) \simeq (A \to G) ,$$

где $\mathsf{hom}_{\mathsf{Monoid}}(\_, \_)$ обозначает множество моноидных гомоморфизмов (функций, которые сохраняют умножение и единицу).

Доказательство. Для $f : A \to G$ определим $\bar{f} : \mathsf{List}(A) \to G$ рекурсией

$$\begin{aligned}\bar{f}(\mathsf{nil}) &:\equiv e\\ \bar{f}(\mathsf{cons}(a, \ell)) &:\equiv f(a) \cdot \bar{f}(\ell)\,.\end{aligned}$$

По индукции легко доказать, что $\bar{f}$ является моноидным гомоморфизмом, а $f \mapsto \bar{f}$ является квазиобратным к $(\_ \circ \eta)$; см. упражнение 6.8. □

Эта конструкция свободного моноида возможна, по существу, потому, что элементы свободного моноида имеют вычислимые канонические формы (а именно, конечные списки). Однако элементы других свободных (и представимых) алгебраических структур, таких как группы, вообще не имеют *вычислимых* канонических форм. Например, равенство слов в групповых представлениях алгоритмически неразрешимо. Однако мы все еще можем описать свободные алгебраические объекты как *высшие* индуктивные типы, просто заявляя все аксиоматические уравнения в качестве конструкторов путей.

Например, пусть A — множество, определим высший индуктивный тип $F(A)$ следующими конструкторами:

- функцией $\eta : A \to F(A)$;
- функцией $m : F(A) \times F(A) \to F(A)$;
- элементом $e : F(A)$;
- функцией $i : F(A) \to F(A)$;
- равенством $m(x, m(y, z)) = m(m(x, y), z)$, для любых $x, y, z : F(A)$;
- равентствами $m(x, e) = x$ и $m(e, x) = x$, для каждого $x : F(A)$;
- равентствами $m(x, i(x)) = e$ и $m(i(x), x) = e$, для каждого $x : F(A)$;
- конструктором 0-усечения: для любых $x, y : F(A)$ и $p, q : x = y$ имеет место $p = q$.

Первый конструктор указывает, что A отображается на $F(A)$. Следующие три наделяют $F(A)$ операциями группы: умножением, единичным элементом и инверсией. Далее, три конструктора предъявляют аксиомы группы: ассоциативности, единицы и обратных. Наконец, последний конструктор утверждает, что $F(A)$ является множеством.

Следовательно, $F(A)$ является группой. Также легко доказывается следующая

Теорема 6.11.6. *$F(A)$ является свободной группой на A. Другими словами, для любой (заданной) группы G композиция с $\eta : A \to F(A)$ определяет эквивалентность*

$$\mathsf{hom}_{\mathsf{Group}}(F(A), G) \simeq (A \to G)\,,$$

где $\mathsf{hom}_{\mathsf{Group}}(\_, \_)$ обозначает множество групповых гомоморфизмов между двумя группами.

Доказательство. Принцип рекурсии высшего индуктивного типа $F(A)$ *в точности* говорит о том, что если G — группа и $f : A \to G$, то и $\bar{f} : F(A) \to G$. Ее правила вычисления говорят, что $\bar{f} \circ \eta \equiv f$, и что f — групповой гомоморфизм. Таким образом, $(\_ \circ \eta) : \mathsf{hom}_{\mathsf{Group}}(F(A), G) \to (A \to G)$ имеет правый обратный, который, как показывает прямое использование принципа индукции для $F(A)$, к тому же является и левым обратным. □

Стоит вернуться на шаг, чтобы рассмотреть то, что мы только что сделали. Было доказано, что свободная группа на любом множестве существует *без* явного ее построения. По сути, все, что нам нужно было сделать, это записать универсальное свойство, которому она должна удовлетворять. В теории множеств можно было бы добиться аналогичного результата, обратившись к черным ящикам, таким как теорема сопряженного функтора; теория типов закладывает такие конструкции в основания математики.

Конечно, иногда полезно также иметь конкретное описание свободных алгебраических структур. В случае свободных групп мы можем предоставить таковое, используя частные. Рассмотрим $\mathsf{List}(A+A)$, где в $A+A$ будем записывать $\mathsf{inl}(a)$ как a, а $\mathsf{inr}(a)$ — как $\hat{a}$ (для обозначения формально обратного к a). Элементы из $\mathsf{List}(A+A)$ являются *словами* для свободной группы на A.

Теорема 6.11.7. *Пусть A — множество и пусть $F'(A)$ — множество-частное для $\mathsf{List}(A+A)$ согласно следующим соотношениям:*

$$(\ldots, a_1, a_2, \hat{a_2}, a_3, \ldots) = (\ldots, a_1, a_3, \ldots)\,,$$
$$(\ldots, a_1, \hat{a_2}, a_2, a_3, \ldots) = (\ldots, a_1, a_3, \ldots)\,.$$

Тогда $F'(A)$ также является свободной группой на A.

Доказательство. Сначала покажем, что $F'(A)$ — группа. Мы выяснили, что $List(A+A)$ является моноидом; мы утверждаем, что моноидная структура сводится к частному. Определим $F'(A) \times F'(A) \to F'(A)$ двойной рекурсией по частному; достаточно проверить, что отношение эквивалентности, порожденное заданными соотношениями, сохраняется при конкатенации списков. Аналогично доказываются законы ассоциативности и единицы, с помощью индукции частных.

В порядке определения обратных в $F'(A)$ в первую очередь определим $\mathsf{reverse} : \mathsf{List}(B) \to \mathsf{List}(B)$ рекурсией на списках:

$$\begin{aligned}
\mathsf{reverse}(\mathsf{nil}) &:\equiv \mathsf{nil}\,,\\
\mathsf{reverse}(\mathsf{cons}(b,\ell)) &:\equiv \mathsf{reverse}(\ell) \cdot \mathsf{cons}(b,\mathsf{nil})\,.
\end{aligned}$$

Теперь определим $i : F'(A) \to F'(A)$ рекурсией по частным, действуя на список $\ell : \mathsf{List}(A+A)$ перестановкой двух копий A и обращением списка. Это сохраняет соотношения, следовательно, сводится к частному. И мы можем доказать, что $i(x)x = e$ для $x : F'(A)$ по индукции. Во-первых, индукция по частным позволяет предположить, что x происходит из $\ell : \mathsf{List}(A+A)$, а затем мы можем выполнить индукцию списка; если записать отображение частных как $q : \mathsf{List}(A+A) \to F'(A)$, возможными случаями будут

$$\begin{aligned}
i(q(\mathsf{nil})) \cdot q(\mathsf{nil}) &= q(\mathsf{nil}) \cdot q(\mathsf{nil})\\
&= q(\mathsf{nil})\,,\\
i(q(\mathsf{cons}(a,\ell))) \cdot q(\mathsf{cons}(a,\ell)) &= i(q(\ell)) \cdot q(\mathsf{cons}(\hat{a},\mathsf{nil})) \cdot q(\mathsf{cons}(a,\ell))\\
&= i(q(\ell)) \cdot q(\mathsf{cons}(\hat{a},\mathsf{cons}(a,\ell)))\\
&= i(q(\ell)) \cdot q(\ell)\\
&= q(\mathsf{nil})\,. \qquad \text{(по индуктивному предположению)}
\end{aligned}$$

(мы опустили ряд довольно очевидных лемм, о поведении конкатенации списков и т.д.).

Это завершает доказательство того, что $F'(A)$ — группа. Теперь, если G — любая группа с функцией $f : A \to G$, мы можем определить $A + A \to G$, в качестве f на первой копии A, и f, скомпонованной с обратным отображением G, — на второй копии A. Теперь тот факт, что G является моноидом, дает моноидный гомоморфизм $\mathsf{List}(A+A) \to G$. Так как G — группа, то это отображение учитывает заданные соотношения, поэтому сводится к отображению $F'(A) \to G$. Нетрудно доказать, что это групповой гомоморфизм и он единственный, ограничивающий f на A. □

Если A имеет разрешимое равенство (например, если мы предполагаем исключение третьего), то определяющее частное $F'(A)$ можно получить из идемпотента, как в лемме 6.10.8. Мы определяем слово, которое, как мы помним, является просто элементом из $\mathsf{List}(A + A)$, которое будет **редуцированным**, если оно не содержит смежных пар вида $(a, \hat{a})$ или $(\hat{a}, a)$. Когда A имеет разрешимое равенство, нетрудно определить **редукцию** слова, которое является идемпотентом, порождающим соответствующее частное; мы оставляем детали читателю.

Если $A :\equiv \mathbf{1}$, которое имеет разрешимое равенство, то редуцированное слово должно состоять либо полностью из $\star$, либо полностью из $\star'$. Таким образом, свободная группа на $\mathbf{1}$ эквивалентна целым числам $\mathbb{Z}$, причем 0 соответствует nil, положительное целое n соответствует редуцированному слову из n элементов $\star$, а отрицательное целое число $(-n)$, соответствует редуцированному слову из n элементов $\star'$. Разумеется, можно было бы прямо показать, что $\mathbb{Z}$ обладает универсальным свойством для $F(\mathbf{1})$.

Замечание 6.11.8. Нигде в построении $F(A)$ и $F'(A)$ и доказательстве их универсальных свойств не было использовано предположение, что A — множество. Таким образом, фактически можно построить свободную группу на произвольном типе. Сравнивая универсальные свойства, заключаем, что $F(A) \simeq F(\|A\|_0)$.

Можно также использовать высшие индуктивные типы для построения копределов алгебраических объектов. Например, пусть $f : G \to H$ и $g : G \to K$ — групповые гомоморфизмы. Их амальгама в категории групп, называемая **объединенным** (или, **амальгамированным**) **свободным произведением** $H *_G K$, может быть построено как высший индуктивный тип, порожденный

- функциями $h : H \to H *_G K$ и $k : K \to H *_G K$;
- операциями и аксиомами группы, в качестве определения $F(A)$;
- аксиомами, утверждающими, что h и k — групповые гомоморфизмы;
- равенством $h(f(x)) = k(g(x))$ для любого $x : G$;
- конструктором 0-усечения.

С другой стороны, его можно также построить явно, как множество-частное $\mathsf{List}(H + K)$, с помощью следующих соотношений:

$$\begin{aligned}
(\dots, x_1, x_2, \dots) &= (\dots, x_1 \cdot x_2, \dots) && \text{для } x_1, x_2 : H \\
(\dots, y_1, y_2, \dots) &= (\dots, y_1 \cdot y_2, \dots) && \text{для } y_1, y_2 : K \\
(\dots, 1_G, \dots) &= (\dots, \dots) \\
(\dots, 1_H, \dots) &= (\dots, \dots) \\
(\dots, f(x), \dots) &= (\dots, g(x), \dots) && \text{для } x : G\,.
\end{aligned}$$

Мы оставляем доказательства читателю. В частном случае, когда G является тривиальной группой, последнее соотношение не нужно, и мы получаем **свободное произведение** $H * K$, копроизведение в категории групп (это обозначение, к сожалению, совпадает с введенным для *соединения* типов, как в §6.8, но контекст обычно устраняет неоднозначность).

Обратите внимание, что группы, определяемые *представлениями*, могут рассматриваться как частный случай копределов. Предположим, что задано множество (или, более обще, тип) A, и пара функций $R \rightrightarrows F(A)$. Мы рассматриваем R как тип «отношения», причем две функции присваивают каждому отношению два слова, которые этот тип устанавливает равными. Например, в представлении $\langle a \mid a^2 = e \rangle$ мы имели бы $A :\equiv \mathbf{1}$ и $R :\equiv \mathbf{1}$, с двумя морфизмами $R \rightrightarrows F(A)$, выбирающими список (a, a) и пустой список nil, соответственно. Тогда, по универсальному свойству свободных групп, получим пару групповых гомоморфизмов $F(R) \rightrightarrows F(A)$. Их коуравнитель в категории групп, который может быть построен так же, как и амальгама, является группой, *рассматриваемой* через это представление.

Обратите внимание, что все эти виды построений применимы только к *алгебраическим* теориям, аксиомы которых являются (универсально квантифицированными) уравнениями, относящимися к переменным, константам и операциям из заданной сигнатуры. Их можно модифицировать, чтобы применить также к тем, которые именуются *эссенциально алгебраическими теориями*: тем, чьи операции частично определены в области, заданной равенствами между предшествующими операциями. Они не применяются, например, к теории полей, в которой операция инверсии частично определена в области $\{x \mid x \,\#\, 0\}$, заданной *расстоянием* $\#$ между предшествующими операциями, см. теорему 11.2.4. И действительно, общеизвестно, что категория полей не имеет инициального объекта.

С другой стороны, эти конструкции действительно применимы также к *бесконечным* алгебраическим теориям, чьи «операции» могут принимать бесконечно много входных значений. В таких случаях не может быть никакого представления свободных алгебр или копределов алгебр в виде простого частного, если мы не примем аксиому выбора. Это означает, что высшие индуктивные типы представляют собой значительное усиление конструктивной теории типов (не обязательно с теоретико-доказательной, но с практической позиции), которая действительно, в какой-то мере, является более мощной, чем теория множеств Цермело-Френкеля (без аксиомы выбора).

6.12 Лемма сглаживания

Как мы увидим в главе 8, появляются удивительные эффекты, когда мы объединяем высшие индуктивные типы с унивалентностью. Основной способ заключается в том, что если W — высший индуктивный тип, а $\mathcal{U}$ — универсум типов, то мы можем определить семейство типов $P : W \to \mathcal{U}$, используя принцип рекурсии для W. Когда мы переходим к положениям принципа рекурсии, относящимся к конструкторам пути W, нам нужно будет решать вопросы представления путей в $\mathcal{U}$, и именно здесь приходит на помощь унивалентность.

Например, предположим, что имеется тип X и самоэквивалентность $e : X \simeq X$. Тогда можно определить семейство типов $P : \mathbb{S}^1 \to U$, используя $\mathbb{S}^1$-рекурсию:

$$P(\mathsf{base}) :\equiv X \qquad \text{и} \qquad P(\mathsf{loop}) := \mathsf{ua}(e)\,.$$

Таким образом, тип X появляется как слой $P(\mathsf{base})$ для P над отмеченной точкой. Самоэквивалентность e заувалирована в P, но следующая лемма говорит, что ее можно извлечь, используя транспортировку вдоль loop.

Лемма 6.12.1. *Для $B : A \to \mathcal{U}$ и $x, y : A$ с путем $p : x = y$ и эквивалентностью $e : B(x) \simeq B(y)$ такой, что $B(p) = \mathsf{ua}(e)$ для любого $u : B(x)$, имеем*

$$\mathsf{transport}^B(p, u) = e(u)\,.$$

Доказательство. Применяя лемму 2.10.5, имеем

$$\begin{aligned}\mathsf{transport}^B(p, u) &= \mathsf{idtoeqv}(B(p))(u)\\ &= \mathsf{idtoeqv}(ua(e))(u)\\ &= e(u)\,.\end{aligned}$$

□

Мы уже сталкивались с семейством типов, определяемых рекурсией, до такого определения: в §§ 2.12 и 2.13 мы использовали их для характеризации типов тождественности (обычных) индуктивных типов. В главе 8 мы будем использовать аналогичные идеи для вычисления гомотопических групп высших индуктивных типов.

В этом разделе мы представим общую лемму о семействах типов такого рода, которая будет полезна позже. Мы называем ее **леммой сглаживания**: она выражает то, что если $P : W \to \mathcal{U}$ определяется, как выше, рекурсивно, то его пространство расслоений $\sum_{(x:W)} P(x)$ эквивалентно «приглаженному» высшему индуктивному типу, конструкторы которого могут быть выведены из тех, что определяют W, и определения P (с теоретико-категорной точки зрения $\sum_{(x:W)} P(x)$ является «конструкцией Гротендика» для P, а лемма сглаживания выражает его универсальное свойство как «слабого копредела»; хотя из-за того, что типы в гомотопической теории типов (например, W) категорно соответствуют ∞-группоидам (поскольку все пути обратимы), то в этом случае слабый копредел является таким же, как и псевдо копредел).

Мы докажем здесь один общий случай леммы сглаживания, который непосредственно подразумевает множество частных случаев и предлагает метод доказательства для других случаев. Предположим, что $A, B : \mathcal{U}$ и $f, g : B \to A$, а высший индуктивный тип W порожден

- $\mathsf{c} : A \to W$ и
- $\mathsf{p} : \prod_{(b:B)} (\mathsf{c}(f(b)) =_W \mathsf{c}(g(b)))$.

Таким образом, W является **(гомотопическим) коуравнителем** для f и g. Используя бинарные суммы (копроизведения) и зависимые суммы ($\sum$-типы), в этой форме может быть представлено много интересных нерекурсивных высших индуктивных типов. Все точечные конструкторы должны быть связаны в типе A, а все конструкторы путей — в типе B. Например:

- Окружность $\mathbb{S}^1$ может быть представлена, если положить $A :\equiv \mathbf{1}$ и $B :\equiv \mathbf{1}$, с тождественными f и g.
- Амальгама для $j : X \to Y$ и $k : X \to Z$ может быть представлена, если $A :\equiv Y + Z$ и $B :\equiv X$ с $f :\equiv \mathsf{inl} \circ j$ и $g :\equiv \mathsf{inr} \circ k$.

Предположим теперь, что

- $C : A \to \mathcal{U}$ — семейство типов над A, а
- $D : \prod_{(b:B)} C(f(b)) \simeq C(g(b))$ — семейство эквивалентностей над B.

Определим семейство типов $P : W \to \mathcal{U}$ рекурсивно по

$$\begin{aligned} P(\mathsf{c}(a)) &:\equiv C(a) \\ P(\mathsf{p}(b)) &:= \mathsf{ua}(D(b)) \,. \end{aligned}$$

Пусть $\widetilde{W}$ — высший индуктивный тип, порожденный

- $\widetilde{\mathsf{c}} : \prod_{(a:A)} C(a) \to \widetilde{W}$ и
- $\widetilde{\mathsf{p}} : \prod_{(b:B)} \prod_{(y:C(f(b)))} (\widetilde{\mathsf{c}}(f(b), y) =_{\widetilde{W}} \widetilde{\mathsf{c}}(g(b), D(b)(y)))$.

Лемма 6.12.2 (Лемма сглаживания). *В приведенной выше ситуации имеет место*

$$\left(\sum_{x:W} P(x)\right) \simeq \widetilde{W} \,.$$

Как отмечалось выше, эту эквивалентность можно рассматривать как выражение универсального свойства $\sum_{(x:W)} P(x)$ в качестве «слабого копредела» P над W. Ее можно также рассматривать как часть устойчивости и свойство спуска копределов, которое характеризует высшие топосы.

Доказательство леммы 6.12.2 занимает оставшуюся часть этого раздела. Оно содержит ряд технических деталей и может быть пропущено при первом чтении. Но оно является также хорошим примером «доказательной математики», поэтому мы рекомендуем его при повторном прочтении книги.

Идея доказательства заключается в демонстрации того, что $\sum_{(x:W)} P(x)$ обладает тем же универсальным свойством, что и $\widetilde{W}$. Начнем с аналогов конструкторов $\widetilde{\mathsf{c}}$ и $\widetilde{\mathsf{p}}$.

Лемма 6.12.3. *Существуют функции*

- $\widetilde{\mathsf{c}}' : \prod_{(a:A)} C(a) \to \sum_{(x:W)} P(x)$ *и*
- $\widetilde{\mathsf{p}}' : \prod_{(b:B)} \prod_{(y:C(f(b)))} \Big(\widetilde{\mathsf{c}}'(f(b), y) =_{\sum_{(x:W)} P(x)} \widetilde{\mathsf{c}}'(g(b), D(b)(y))\Big)$.

Доказательство. Первая функция получается просто; определим $\widetilde{\mathsf{c}}'(a, x) :\equiv (\mathsf{c}(a), x)$ и заметим, что по определению, $P(\mathsf{c}(a)) \equiv C(a)$. Для второй функции, предположим, что $b : B$ и $y : C(f(b))$; надо убедиться в справедливости равенства

$$(\mathsf{c}(f(b)), y) = (\mathsf{c}(g(b)), D(b)(y)) \,.$$

Поскольку $\mathsf{p}(b) : \mathsf{c}(f(b)) = \mathsf{c}(g(b))$, то в связи с равенствами в $\sum$-типах достаточно выполнения $p(b)_*(y) = D(b)(\mathrm{y})$. Но оно следует из леммы 6.12.1, используя определение P. □

Теперь, следующая лемма утверждает, что для определения сечения семейства типов над $\sum_{(w:W)} P(w)$ достаточно предоставить данные, аналогичные случаю для $\widetilde{W}$.

Лемма 6.12.4. *Пусть* $Q : (\sum_{(x:W)} P(x)) \to \mathcal{U}$ *— семейство типов и имеются*

- $c : \prod_{(a:A)} \prod_{(x:C(a))} Q(\widetilde{\mathsf{c}}'(a, x))$ *и*

- $p : \prod_{(b:B)} \prod_{(y:C(f(b)))} (\widetilde{\mathsf{p}}'(b,y)_*(c(f(b),y)) = c(g(b), D(b)(y)))$.

Тогда существует $k : \prod_{(z:\sum_{(w:W)} P(w))} Q(z)$ *такое, что* $k(\widetilde{\mathsf{c}}'(a,x)) \equiv c(a,x)$.

Доказательство. Пусть заданы $w : W$ и $x : P(w)$; мы должны построить элемент $k(w,x) : Q(w,x)$. Индукцией по w достаточно рассмотреть два случая. Когда $w \equiv \mathsf{c}(a)$, то имеем $x : C(a)$, и поэтому $c(a,x) : Q(\mathsf{c}(a),x)$ что и требовалось (эта часть первого определения также гарантирует соблюдение указанного правила вычисления).

Теперь мы должны показать, что это определение сохраняется путем транспортировки вдоль $\mathsf{p}(b)$ для любого $b : B$. Поскольку то, что мы определяем, для всех $w : W$ является функцией типа $\prod_{(x:P(w))} Q(w,x)$, то по лемме 2.9.7 достаточно показать, что для любого $y : C(f(b))$, имеет место

$$\mathsf{transport}^Q(\mathsf{pair}^=(p(b), \mathsf{refl}_{p(b)_*(y)}), c(f(b),y)) = c(g(b), p(b)_*(y)) .$$

Пусть $q : \mathsf{p}(b)_*(y) = D(b)(y)$ — путь, полученный из леммы 6.12.1. Тогда

$$\begin{aligned} c(g(b), \mathsf{p}(b)_*(y)) &= \mathsf{transport}^{x \mapsto Q(\widetilde{\mathsf{c}}'(g(b),x))}(q^{-1}, c(g(b), D(b)(y))) && (\text{по } \mathsf{apd}_{x \mapsto c(g(b),x)}(q^{-1})^{-1}) \\ &= \mathsf{transport}^Q(\mathsf{ap}_{x \mapsto \widetilde{\mathsf{c}}'(g(b),x)}(q^{-1}), c(g(b), D(b)(y))) . && (\text{по лемме } 2.3.10) \end{aligned}$$

Таким образом, достаточно показать, что

$$\mathsf{transport}^Q\left(\mathsf{pair}^=(\mathsf{p}(b), \mathsf{refl}_{\mathsf{p}(b)_*(y)}), c(f(b),y)\right) = \\ \mathsf{transport}^Q\left(\mathsf{ap}_{x \mapsto \widetilde{\mathsf{c}}'(g(b),x)}(q^{-1}), c(g(b), D(b)(y))\right) .$$

Перемещение правосторонней транспортировки на другую сторону и компоновка двух транспортировок, дает, эквивалентно,

$$\mathsf{transport}^Q\left(\mathsf{pair}^=(\mathsf{p}(b), \mathsf{refl}_{\mathsf{p}(b)_*(y)}) \cdot \mathsf{ap}_{x \mapsto \widetilde{\mathsf{c}}'(g(b),x)}(q), c(f(b),y)\right) = c(g(b), D(b)(y)) .$$

Однако,

$$\mathsf{pair}^=(\mathsf{p}(b), \mathsf{refl}_{\mathsf{p}(b)_*(y)}) \cdot \mathsf{ap}_{x \mapsto \widetilde{\mathsf{c}}'(g(b),x)}(q) = \\ \mathsf{pair}^=(\mathsf{p}(b), \mathsf{refl}_{\mathsf{p}(b)_*(y)}) \cdot \mathsf{pair}^=(\mathsf{refl}_{\mathsf{c}(g(b))}, q) = \mathsf{pair}^=(\mathsf{p}(b), q) = \widetilde{\mathsf{p}}'(b,y)$$

поэтому построение завершается предположением $p(b,y)$ типа

$$\mathsf{transport}^Q(\widetilde{\mathsf{p}}'(b,y), c(f(b),y)) = c(g(b), D(b)(y)) .$$

□

Лемма 6.12.4 *почти* дает $\sum_{(w:W)} P(w)$ тот же принцип индукции, что и $\widetilde{W}$. Недостающее — это равенство $\mathsf{apd}_k(\widetilde{\mathsf{p}}'(b,y)) = p(b,y)$. Чтобы это доказать, необходимо проанализировать доказательство леммы 6.12.4, которое, конечно же, является определением k.

Это возможно сделать, но, оказывается, нам нужно только правило вычисления для принципа независимой рекурсии. Таким образом, теперь мы приведем несколько более простое прямое построение рекурсора и доказательство его правила вычисления.

Лемма 6.12.5. *Пусть* Q *— тип и имеются*

- $c : \prod_{(a:A)} C(a) \to Q$ *и*

- $p : \prod_{(b:B)} \prod_{(y:C(f(b)))} (c(f(b), y) =_Q c(g(b), D(b)(y)))$.

Тогда существует $k : (\sum_{(w:W)} P(w)) \to Q$ *такое, что* $k(\widetilde{\mathsf{c}}'(a,x)) \equiv c(a,x)$.

Доказательство. Как и в лемме 6.12.4, определим $k(w,x)$ индукцией по $w : W$. Когда $w \equiv \mathsf{c}(a)$, определим $k(\mathsf{c}(a),x) :\equiv c(a,x)$. Теперь, по лемме 2.9.6 достаточно рассмотреть, для $b : B$ и $y : C(f(b))$, составной путь

$$\mathsf{transport}^{x \mapsto Q}(\mathsf{p}(b), c(f(b),y)) = c(g(b), \mathsf{transport}^{P}(\mathsf{p}(b), y)), \tag{6.12.6}$$

определенный согласно композиции

$$\begin{aligned}
\mathsf{transport}^{x \mapsto Q}(\mathsf{p}(b), c(f(b),y)) &= c(f(b),y) && \text{(по лемме 2.3.5)}\\
&= c(g(b), D(b)(y)) && \text{(по } p(b,y))\\
&= c(g(b), \mathsf{transport}^{P}(\mathsf{p}(b), y)). && \text{(по лемме 6.12.1)}
\end{aligned}$$

Правило вычисления $k(\widetilde{\mathsf{c}}'(a,x)) \equiv c(a,x)$ следует из определения, как и ранее. $\square$

Для второго правила вычисления нам необходима следующая лемма.

Лемма 6.12.7. *Пусть* $Y : X \to \mathcal{U}$ *— семейство типов и пусть* $k : (\sum_{(x:X)} Y(x)) \to Z$ *определено покомпонентно посредством* $k(x,y) :\equiv d(x)(y)$ *для каррированной функции* $d : \prod_{(x:X)} Y(x) \to Z$. *Тогда для любого* $s : x_1 =_X x_2$ *и любых* $y_1 : Y(x_1)$ *и* $y_2 : Y(x_2)$ *с путем* $r : s_*(y_1) = y_2$, *путь*

$$\mathsf{ap}_k(\mathsf{pair}^{=}(s,r)) : k(x_1,y_1) = k(x_2,y_2)$$

эквивалентен составному

$$\begin{aligned}
k(x_1,y_1) &\equiv d(x_1)(y_1)\\
&= \mathsf{transport}^{x \mapsto Z}(s, d(x_1)(y_1)) && \text{(по (лемме 2.3.5)}^{-1})\\
&= \mathsf{transport}^{x \mapsto Z}(s, d(x_1)(s_*^{-1}(s_*(y_1))))\\
&= (\mathsf{transport}^{x \mapsto (Y(x) \to Z)}(s, d(x_1)))(s_*(y_1)) && \text{(по (2.9.4))}\\
&= d(x_2)(s_*(y_1)) && (\text{по } \mathsf{happly}(\mathsf{apd}_d(s))(s_*(y_1)))\\
&= d(x_2)(y_2) && (\text{по } \mathsf{ap}_{d(x_2)}(r))\\
&\equiv k(x_2,y_2).
\end{aligned}$$

Доказательство. После индукции пути по s и r оба равенства сводятся к рефлексивности. $\square$

Вначале может показаться удивительным, что в лемме 6.12.7 такое сложное утверждение можно доказать так просто. Причина усложнения формулировки заключается в том, чтобы убедиться, что утверждение хорошо типизировано, $\mathsf{ap}_k(\mathsf{pair}^{=}(s,r))$, и оба эквивалентных пути должны иметь одинаковые начальные и конечные точки. Как только это показано, дальнейшее доказательство проводится просто, по индукции пути.

Лемма 6.12.8. *В условиях леммы 6.12.5 имеет место* $\mathsf{ap}_k(\widetilde{\mathsf{p}}'(b,y)) = p(b,y)$.

Доказательство. Напомним, что $\widetilde{\mathsf{p}}'(b,y) :\equiv \mathsf{pair}^{=}(p(b),q)$, где $q : p(b)_*(y) = D(b)(y)$ происходит из леммы 6.12.1. Таким образом, поскольку k определяется покомпонентно, мы можем вычислить $\mathsf{ap}_k(\widetilde{\mathsf{p}}'(b,y))$ по лемме 6.12.7 с

$$
\begin{aligned}
x_1 &:\equiv \mathsf{c}(f(b)) & y_1 &:\equiv y \\
x_2 &:\equiv \mathsf{c}(g(b)) & y_2 &:\equiv D(b)(y) \\
s &:\equiv \mathsf{p}(b) & r &:\equiv q\,.
\end{aligned}
$$

Каррированная функция $d : \prod_{(w:W)} P(w) \to Q$ была определена индукцией по $w : W$; для применения леммы 6.12.7 нам нужно осмыслить $\mathsf{ap}_{d(x_2)}(r)$ и $\mathsf{happly}(\mathsf{apd}_d(s), s_*(y_1))$.

Для первого, поскольку $d(\mathsf{c}(a),x) \equiv c(a,x)$, имеем

$$\mathsf{ap}_{d(x_2)}(r) \equiv \mathsf{ap}_{c(g(b),\_)}(q)\,.$$

Для второго, правило вычисления для принципа индукции W указывает, что $\mathsf{apd}_d(\mathsf{p}(b))$ равно совокупности (6.12.6), пропущенной через эквивалентность из леммы 2.9.6. Таким образом, из правила вычисления, приведенного в лемме 2.9.6, следует, что $\mathsf{happly}(\mathsf{apd}_d(\mathsf{p}(b)), \mathsf{p}(b)_*(y))$ равно совокупности

$$
\begin{aligned}
(\mathsf{p}(b)_*(c(f(b),\_)))\,(\mathsf{p}(b)_*(y)) &= \mathsf{p}(b)_*\left(c(f(b),\mathsf{p}(b)_*^{-1}(\mathsf{p}(b)_*(y)))\right) && \text{(по (2.9.4))}\\
&= \mathsf{p}(b)_*(c(f(b),y)) \\
&= c(f(b),y) && \text{(по лемме 2.3.5)}\\
&= c(f(b),D(b)(y)) && \text{(по } p(b,y))\\
&= c(f(b),\mathsf{p}(b)_*(y))\,. && \text{(по } \mathsf{ap}_{c(g(b),\_)}(q)^{-1})
\end{aligned}
$$

Наконец, подставив в лемму 6.12.7 эти значения $\mathsf{ap}_{d(x_2)}(r)$ и $\mathsf{happly}(\mathsf{apd}_d(s), s_*(y_1))$, получаем, что все пути сворачиваются попарно, оставляя только $p(b,\mathrm{y})$. □

Теперь мы, наконец, готовы доказать лемму сглаживания.

Доказательство леммы 6.12.2. Определим $h : \widetilde{W} \to \sum_{(w:W)} P(w)$, используя принцип рекурсии для $\widetilde{W}$, с $\widetilde{\mathsf{c}}'$ и $\widetilde{\mathsf{p}}'$ в качестве входных данных. Аналогично, определим $k : (\sum_{(w:W)} P(w)) \to \widetilde{W}$, используя принцип рекурсии из леммы 6.12.5, с $\widetilde{\mathsf{c}}$ и $\widetilde{\mathsf{p}}$ в качестве входных данных.

С одной стороны, мы должны показать, что для любого $z : \widetilde{W}$ имеем $k(h(z)) = z$. Индукцией по z достаточно рассмотреть два конструктора $\widetilde{W}$. Но

$$k(h(\widetilde{\mathsf{c}}(a,x))) \equiv k(\widetilde{\mathsf{c}}'(a,x)) \equiv \widetilde{\mathsf{c}}(a,x)$$

по определению, и аналогично

$$k(h(\widetilde{\mathsf{p}}(b,y))) = k(\widetilde{\mathsf{p}}'(b,y)) = \widetilde{\mathsf{p}}(b,y)$$

используя правило пропозиционального вычисления для $\widetilde{W}$ и лемму 6.12.8.

С другой стороны, мы должны показать, что для любого $z : \sum_{(w:W)} P(w)$ имеем $h(k(z)) = z$. Но это по существу тождественно, используя лемму 6.12.4 для «индукции по $\sum_{(w:W)} P(w)$» и те же правила вычисления. □

6.13 Общий синтаксис высших индуктивных определений

В §5.6 мы обсудили условия предполагаемого «индуктивного определения», которые делают его приемлемым, а именно, что все индуктивные вхождения типа в его конструкторах должны быть «строго позитивны». В этом разделе мы поговорим о дополнительных условиях, необходимых для *высших* индуктивных определений. Нахождение общего синтаксического описания обоснованных высших индуктивных определений является областью текущих исследований, и все предлагаемые на сегодняшний день решения носят несколько технический характер; поэтому мы приводим только общее описание, а не точное определение. К счастью, тупиковые случаи никогда не возникают на практике.

Как и обычное индуктивное определение, высшее индуктивное определение задается списком *конструкторов*, каждый из которых является (зависимой) функцией. Для простоты мы можем потребовать, чтобы входы каждого конструктора удовлетворяли тому же условию, что и входы для конструкторов обычных индуктивных типов. В частности, они могут содержать тип, который определяется только строго положительно. Обратите внимание, что это исключает такие определения, как 0-усечение, представленое в §6.9, где вход конструктора содержит не только определяемый индуктивный тип, но и его тип тождественности. Возможно, можно расширить синтаксис, чтобы допускать такие определения; но также в §7.3 мы дадим другую конструкцию 0-усечения, конструкторы которой удовлетворяют более ограничительному условию.

Единственное различие между обычным индуктивным определением и высшим состоит в том, что тип *выходных данных* конструктора может быть не определяемым типом (скажем, W), а некоторым его типом тождественности, таким как $u =_W v$, или, в более общем смысле, итерированным типом тождественности, таким как $p =_{(u =_W v)} q$. Таким образом, когда мы формулируем высшее индуктивное определение, мы должны указывать не только входы каждого конструктора, но и выражения u и v (или u, v, p, q и т.д.), которые определяют источник и цель построенного пути.

Важно отметить, что эти выражения могут относиться к *другим* конструкторам W. Например, в определении $\mathbb{S}^1$ конструктор loop имеет и u и v, являющиеся базой, предыдущим конструктором. Чтобы понять это, мы требуем, чтобы конструкторы высшего индуктивного типа указывались *по порядку*, и мы разрешаем исходным и целевым выражениям u и v каждого конструктора ссылаться на предыдущие конструкторы, но не на более поздние (конечно, на практике конструкторы любого индуктивного определения записываются в некотором порядке, но для обычных индуктивных типов этот порядок не имеет значения).

Обратите внимание, что этот порядок не обязательно является порядком «размерности»: в принципе, одномерный конструктор путей может ссылаться на двумерный и, следовательно, должен следовать за ним. Тем не менее, мы не предоставили 0-мерным конструкторам (точечным конструкторам) какого-либо способа ссылаться на предыдущие конструкторы, поэтому они могли бы быть все на первом месте. И если мы используем конструкцию концентратор-спица (§6.7), чтобы свести все конструкторы к точкам и 1-путям, то мы могли бы предположить, что все точечные конструкторы находятся на первом месте, за которыми следуют все конструкторы 1-путей, но порядок среди конструкторов 1-путей продолжает иметь значение.

Остается вопрос, какого сорта могут быть выражения u и v? Мы можем надеяться, что они могут быть любыми выражениями, связанными с предыдущими конструкторами. Однако следующий пример показывает, что наивный подход к этой идее не работает.

Пример 6.13.1. Рассмотрим семейство функций $f : \prod_{(X:\mathcal{U})}(X \to X)$. Конечно, f_X может быть просто id_X для всех X, но другие подобные f также могут существовать. Например, ничто не мешает $f_{\mathbf{2}} : \mathbf{2} \to \mathbf{2}$ быть от нетождественного автоморфизма (см. упражнение 6.9).

Предположим теперь, что мы пытаемся определить высший индуктивный тип K, порожденный:

- двумя элементами $a, b : K$ и
- путем $\sigma : f_K(a) = f_K(b)$.

Что представлял бы собой принцип индукции для K? Предположим, что имеется семейство типов $P : K \to \mathcal{U}$, и, конечно, нам понадобятся $x : P(a)$ и $y : P(b)$. Остальные данные должны быть зависимым путем в P, обитающем над σ, который должен связывать некоторый элемент из $P(f_K(a))$ с некоторым элементом из $P(f_K(b))$. Но какими могут быть эти элементы? Мы знаем, что $P(a)$ и $P(b)$ заселены элементами x и y, соответственно, но это ничего не говорит о $P(f_K(a))$ и $P(f_K(b))$.

Ясно, что для того, чтобы определение было осмысленным, требуется некоторое условие на u и v. Кажется, что так же, как область каждого конструктора необходима (среди прочего) для *ковариантного функтора*, соответствующее условие для выражений u и v заключается в том, что они определяют *естественные преобразования*. Точное определение этого требования выходит за рамки этой книги, но неформально это означает, что u и v должны включать только те операции, которые сохраняются всеми функциями между типами.

Например, для u и v допустимо ссылаться на конкатенацию путей, как в случае заключительного конструктора тора в x6.6, так как все функции теории типов сохраняют конкатенацию путей (с точностью до гомотопии). Однако им не разрешено ссылаться на операцию, такую как функция f в примере 6.13.1, что не обязательно является естественным: может существовать некоторая функция $g : X \to Y$ такая, что $f_Y \circ g \neq g \circ f_X$ (из унивалентности следует, что f_X должна быть естественной относительно всех *эквивалентностей*, но не обязательно относительно функций, которые не являются эквивалентностями).

Интуиция по естественности дает лишь приблизительное руководство, когда допустимо высшее индуктивное определение. Даже если бы можно было дать точную спецификацию допустимых форм таких определений в этой книге, такая спецификация, вероятно, была бы устаревшей, поскольку постоянно исследуются новые расширения теории. Например, представление n-сфер в терминах «зависимых n-петель» , упомянутое в §6.4, и «высшие индуктивно-рекурсивные определения», используемые в главе 11, были нововведениями, которые появивились во время написания этой книги. Мы рекомендуем читателю экспериментировать, но с осторожностью.

Примечания

Общая идея высших индуктивных типов была сформулирована в ходе дискуссий между Андреем Бауэром (Andrej Bauer), Питером Лемсдейном (Peter Lumsdaine), Майком Шульманом (Mike Shulman) и Майклом Уорреном (Michael Warren) на встрече в Оберволфахе в 2011 году, хотя имелись некоторые предложения о некоторых особых случаях в более ранних работах. Впоследствии Гийом Брунери (Guillaume Brunerie) и Дэн Ликата (Dan Licata) внесли существенный вклад в общую теорию, особенно найдя удобные способы их представления в системах (помощниках) по компьютерным доказательствам и занимаясь созданием теории гомотопий с их помощью (см. главу 8).

Общее обсуждение синтаксиса высших индуктивных типов и их семантика в моделях высших категорий представлены в [LS17]. Как и в случае обычных индуктивных типов, модели высших индуктивных типов могут быть построены посредством трансфинитных итерационных процессов; особенность состоит в том, что обычные индуктивные типы описывают *свободные* монады, а высшие индуктивные типы описывают *представления* монад. Введение конструкторов путей также включает теоретико-модельно-категорную эквивалентность между «правыми гомотопиями» (определяемыми использованием пространств путей) и «левыми гомотопиями» (определяемыми использованием цилиндров) — тот факт, что эта эквивалентность определена, как правило, только с точностью до гомотопии, обеспечивает семантическую причину предпочтения правил пропозициональных вычислений для конструкторов путей.

Другая (временная) причина для этого предпочтения исходит из ограничений существующих компьютерных реализаций. Помощники по компьютерным доказательствам, такие как Coq и Agda, имеют встроенные обычные индуктивные типы, но еще не оснащены высшими индуктивными типами. Разумеется, мы можем вводить их, предполагая наличие множество аксиом, но это приводит только к правилам пропозициональных вычислений. Тем не менее, существует трюк от Дэн Ликата (Dan Licata), в котором высшие индуктивные типы реализуются использованием пользовательских типов данных; это обеспечивает дефиницsmåциальные правила для конструкторов точек, но не для конструкторов путей.

Теоретико-типовое описание высших сфер с использованием пространств петель и надстроек в §§ 6.4 и 6.5 во многом связано с Брунери (Brunerie) и Ликата (Licata); Хоу (Hou) дал теоретическую версию альтернативного описания, в котором используются n-мерные пути. Сведение высших путей к одномерным путям с концентраторами и спицами (§6.7) связано с Люмсдайном (Lumsdaine) и Шульманом (Shulman). Описание усечения как высшего индуктивного типа связано с Люмсдайном (Lumsdaine); (-1)-усечение тесно связано с «типами скобок» [AB04]. Лемма сглаживания в общем виде впервые была сформулирована Брунери (Brunerie).

Типы частных являются беспроблемными в теории экстенсионального типа, такой как NuPRL [CAB+86]]. Их часто добавляют путем перехода к расширенной системе сетоидовsetoid (типов с понятием эквивалентности – *прим.перев.*). Тем не менее, частные являются более сложной проблемой в теории интенсионального типа (отправной точкой для гомотопической теории типов), поскольку нельзя просто добавлять новые пропозициональные равенства, не указывая, как они должны себя вести. Были предложены некоторые решения этой проблемы [Hof95, Alt99, AMS07] и рассмотрено несколько различных понятий типов частных. Построение множеств-частных с использованием высших индуктивностей дает аргумент для нашего конкретного подхода (который аналогичен тому, который ранее рассматривался), потому что он возникает как пример общего механизма. Наша конструкция еще не дает нового решения для всех вычислительных проблем, связанных с частными, поскольку у нас по-прежнему отсутствует хорошее вычислительное понимание высших индуктивных типов в целом, но это означает, что текущая работа по вычислительной интерпретации высших индуктивностей применяется также и к частным. Построение частных в терминах классов эквивалентности — это, конечно, стандартная теоретико-множественная идея и хорошо известный аспект элементарной теории топосов; его использование в теории типов (которое зависит от аксиомы унивалентности, по крайней мере для простых высказываний) было предложено Воеводским. Тот факт, что типы частных в интенсиональной теории типов подразумевают расширение функциональности, доказано в [Hof95], вдохновленный работой [Car95] по строгим расширениям; лемма 6.3.2 является адаптацией таких аргументов.

Упражнения

Упражнение 6.1. Определите конкатенацию зависимых путей, докажите, что применение зависимых функций сохраняет конкатенацию, и подробно сформулируйте принцип индукции для тора T^2 с его правилами вычислений.

Упражнение 6.2. Докажите, что $\sum \mathbb{S}^1 \simeq \mathbb{S}^2$, используя явное определение $\mathbb{S}^2$ в терминах base и surf, введенных в §6.4.

Упражнение 6.3. Докажите, что тор T^2, определенный в §6.6, эквивалентен $\mathbb{S}^1 \times \mathbb{S}^1$ (предупреждение: алгебра путей для этого довольно сложна).

Упражнение 6.4. Определите зависимые n-петлиloop!dependent n-@dependent n- и действие зависимых функций на n-петлях, запишите принцип индукции для n-сфер, определенных в конце §6.4.

Упражнение 6.5. Докажите, что $\sum \mathbb{S}^n \simeq \mathbb{S}^{n+1}$, используя определение $\mathbb{S}^n$ в обозначениях Ω^n из §6.4.

Упражнение 6.6. Докажите, что если тип $\mathbb{S}^2$ принадлежит некоторому универсуму $\mathcal{U}$, то $\mathcal{U}$ не является 2-типом.

Упражнение 6.7. Докажите, что если G — моноид и $x : G$, то $\sum_{(y:G)}((x \cdot y = e) \times (y \cdot x = e))$ является простым высказыванием. Выведите, используя принцип единственности выбора (следствие 3.9.2), что оно будет эквивалентно определению группы как моноида такого, что для любого $x : G$ просто существует $y : G$ такое, что $x \cdot y = e$ и $y \cdot x = e$.

Упражнение 6.8. Докажите, что если A — множество, то $\mathsf{List}(A)$ — моноид. Затем завершите доказательство леммы 6.11.5.

Упражнение 6.9. Предполагая LEM, постройте семейство $f : \prod_{(X:\mathcal{U})}(X \to X)$ такое, что $f_{\mathbf{2}} : \mathbf{2} \to \mathbf{2}$ — нетождественный автоморфизм.

Упражнение 6.10. Покажите, что отображение, построенное в лемме 6.3.2, фактически является квазиобратным к happly, так что тип интервала подразумевает аксиому функциональной экстенсиональности (возможно, придется использовать упражнение 2.16).

Упражнение 6.11. Докажите универсальное свойство надстройки

$$\left(\sum A \to B\right) \simeq \left(\sum_{(b_n:B)} \sum_{(b_s:B)} (A \to (b_n = b_s))\right).$$

Упражнение 6.12. Покажите, что $\mathbb{Z} \simeq \mathbb{N}+\mathbf{1}+\mathbb{N}$. Покажите, что если определить $\mathbb{Z}$ как $\mathbb{N}+\mathbf{1}+\mathbb{N}$, то можно получить лемму 6.10.12 с правилами дефинициального вычисления.

Упражнение 6.13. Покажите, что можно доказать лемму 6.3.2, используя $\|\mathbf{2}\|$ вместо I.

Глава 7

Гомотопические n-типы

Одним из основных понятий гомотопической теории является *гомотопический n-тип*: пространство, не содержащее никакой интересной гомотопии над размерностью n. Например, гомотопический 0-тип является по существу множеством, не содержащим нетривиальных путей, тогда как гомотопический 1-тип может содержать нетривиальные пути, но нет нетривиальных путей между путями. Гомотопические n-типы также называются *n-усеченными пространствами*. Мы уже упоминали это понятие в §3.1; наша первая цель в этой главе — дать этому понятию точное определение в теории гомотопических типов.

Двойственное понятие усечения — это связность: пространство *n-связно*, если оно не имеет никакой интересной гомотопии в размерностях n и *ниже*. Например, пространство является 0-связным (также называемым просто «связным»), если оно имеет только одну связную компоненту, и 1-связным (также называемое «односвязным»), если оно также не имеет нетривиальных петель (хотя оно может иметь нетривиальные высшие петли между петлями).

Двойственность между усечением и связностью наиболее легко осознать, распространяя оба понятия на отображения. Назовем отображение *n-усеченным* или *n-связным*, если все его слои таковы. Тогда n-связные и n-усеченные отображения образуют два класса отображений в *ортогональной системе факторизации*, т.е. каждый фактор отображения однозначно является n-связным отображением, за которым следует n-усеченный.

В случае $n = -1$ n-усеченные отображения являются вложениями, а n-связные отображения являются сюръекциями, как определено в §4.6. Таким образом, n-связная система факторизации представляет собой солидное обобщение стандартной факторизации образа функции между множествами в сюръекцию с последующей инъекцией. В конце этой главы мы кратко набросаем еще более общую теорию: любая теоретико-типовая *модальность* порождает аналогичную систему факторизации.

7.1 Определение n-типов

Как упоминалось в §§ 3.1 и 3.11, оказывается удобным определить n-типы, начинающиеся двумя уровнями ниже нуля, причем (-1)-типы являются простыми высказываниями, а (-2)-типы — стягиваемыми высказываниями.

Определение 7.1.1. Определим предикат $\mathsf{is}\text{-}n\text{-}\mathsf{type} : \mathcal{U} \to \mathcal{U}$, для $n \geqslant -2$, рекурсией

$$\mathsf{is}\text{-}n\text{-}\mathsf{type}(X) :\equiv \begin{cases} \mathsf{isContr}(X) & \text{если } n = -2\,, \\ \prod_{(x,y:X)} \mathsf{is}\text{-}n'\text{-}\mathsf{type}(x =_X y) & \text{если } n = n' + 1\,. \end{cases}$$

X является **n-типом** или, иногда, *n-усеченным*, если is-n-type(X) обитаем.

Замечание 7.1.2. Число n в определении 7.1.1 охватывает все целые числа, большие или равные -2. Мы могли бы понять это формально, задав тип $\mathbb{Z}_{\geqslant -2}$ таких целых чисел (тип, принцип индукции которого идентичен принципу индукции для $\mathbb{N}$), или вместо этого, определяя предикат is-$(k-2)$-type для $k : \mathbb{N}$. В любом случае, мы можем доказать теоремы о n-типах индукцией по n, с $n = -2$ в качестве базового случая.

Пример 7.1.3. В лемме 3.11.10 мы видели, что X является (-1)-типом тогда и только тогда, когда он является простым высказыванием. Следовательно, X является 0-типом тогда и только тогда, когда он является множеством.

Мы также знаем, что существуют типы, которые не являются множествами (пример 3.1.9). Однако до сих пор мы не показывали ни для какого $n > 0$, что существуют типы, которые не являются n-типами. Но в главе 8 мы покажем, что $(n+1)$-сфера $\mathbb{S}^{n+1}$ не является n-типом (Краус также показал, без использования каких-либо высших индуктивных типов, что n-й вложенный унивалентный универсум также не является n-типом). Кроме того, в §8.8 приведен пример типа, который не является n-типом для любого (конечного) числа n.

Мы начнем излагать общую теорию n-типов, показывая, что они замкнуты при определенных операциях и конструкторах.

Теорема 7.1.4. *Пусть $p : X \to Y$ — стягивание и предположим, что X — n-тип для всех $n \geqslant -2$.Тогда Y также является n-типом.*

Доказательство. Проведем индукцией по n. Базовый случай $n = 2$ обрабатывается леммой 3.11.7.

Для индуктивного шага предположим, что любое стягивание n-типа является n-типом, а X — $(n+1)$-типом. Пусть $y, y' : Y$; мы должны показать, что $y = y'$ является n-типом. Пусть s — сечение для p, а ϵ — гомотопия $\epsilon : p \circ s \sim 1$. Поскольку X является $(n+1)$-типом, $s(y) =_X s(y')$ является n-типом. Покажем, что $y = y'$ является стягиванием для $s(y) =_X s(y')$. Для сечения возьмем

$$\mathsf{ap}_s : (y = y') \to (s(y) = s(y'))\,.$$

Для стягивания определим $t : (s(y) = s(y')) \to (y = y')$ посредством

$$t(q) :\equiv \epsilon_y^{-1} \cdot p(q) \cdot \epsilon_{y'}\,.$$

Чтобы показать, что t — стягивание для ap_s, надо показать, что

$$\epsilon_y^{-1} \cdot p(s(r)) \cdot \epsilon_{y'} = r$$

для любого $r : y = y'$. Но это следует из леммы 2.4.3. □

В качестве непосредственного следствия мы получаем устойчивость n-типов при эквивалентности (что также является неотъемлемой частью унивалентности):

Следствие 7.1.5. *Если $X \simeq Y$ и X является n-типом, то Y также является n-типом.*

Напомним также понятие вложения из §4.6.

Теорема 7.1.6. *Если $f : X \to Y$ — вложение, а Y — n-тип для некоторого $n \geqslant -1$, то и X является n-типом.*

Доказательство. Пусть $x, x' : X$; надо показать, что $x =_X x'$ является $(n-1)$-типом. Но, поскольку f — вложение, то $(x =_X x') \simeq (f(x) =_Y f(x'))$, а последнее является $(n-1)$-типом по предположению. $\square$

Заметим, что эта теорема не верна, когда $n = 2$: отображение $\mathbf{0} \to \mathbf{1}$ является вложением, но $\mathbf{1}$ является (2)-типом, а $\mathbf{0}$ — нет.

Теорема 7.1.7. *Иерархия n-типов является кумулятивной в следующем смысле: для $n \geqslant -2$, если X является n-типом, то он также является и $(n+1)$-типом.*

Доказательство. Проведем индукцией по n.

Для $n = -2$ нужно показать, что стягиваемый тип, скажем, A, имеет стягиваемые пространства путей. Пусть $a_0 : A$ — центр стягивания A и $x, y : A$. Покажем, что $x =_A y$ стягиваемо. По стягиваемости A имеем путь $\mathsf{contr}_x \cdot \mathsf{contr}_y^{-1} : x = y$, который мы выбираем как центр стягивания для $x = y$. Для любого $p : x = y$ нужно показать, что $p = \mathsf{contr}_x \cdot \mathsf{contr}_y^{-1}$. По индукции пути достаточно показать, что $\mathsf{refl}_x = \mathsf{contr}_x \cdot \mathsf{contr}_x^{-1}$, что тривиально.

Для индуктивного шага нужно показать, что $x =_X y$ является $(n+1)$-типом, если X является $(n+1)$-типом. Применение индуктивного предположения к $x =_X y$ дает желаемый результат. $\square$

Покажем теперь, что n-типы сохраняются большей частью операций формирования типа.

Теорема 7.1.8. *Пусть $n \geqslant -2$, $A : \mathcal{U}$ и $B : A \to \mathcal{U}$. Если A — n-тип и для любого $a : A$, $B(a)$ является n-типом, то $\sum_{(x:A)} B(x)$ также является n-типом.*

Доказательство. Проведем индукцией по n.

Для $n = -2$, выберем центр стягивания для $\sum_{(x:A)} B(x)$ как пару (a_0, b_0), где $a_0 : A$ — центр стягивания A и $b_0 : B(a_0)$ — центр стягивания $B(a_0)$. Для любого другого элемента (a, b) из $\sum_{(x:A)} B(x)$ мы предоставляем путь $(a, b) = (a_0, b_0)$ согласно стягиваемости A и $B(a_0)$, соответственно.

Для индуктивного шага, предположим, что A является $(n+1)$-типом и для любого $a : A$, $B(a)$ является $(n+1)$-типом. Покажем, что $\sum_{(x:A)} B(x)$ является $(n+1)$-типом: зафиксируем (a_1, b_1) и (a_2, b_2) в $\sum_{(x:A)} B(x)$ и покажем, что $(a_1, b_1) = (a_2, b_2)$ является n-типом. По теореме 2.7.2 имеем

$$((a_1, b_1) = (a_2, b_2)) \simeq \sum_{p : a_1 = a_2} (p_*(b_1) =_{B(a_2)} b_2)$$

и, при сохранении n-типов при эквивалентности (следствие 7.1.5), достаточно доказать, что последнее является n-типом. Но это следует из индуктивного предположения. $\square$

В качестве частного случая, если A и B являются n-типами, то и $A \times B$ является n-типом. Отметим также, что из теоремы 7.1.7 следует, что если A — n-тип, то и $x =_A y$ — n-тип для любых $x, y : A$. Объединяя это с теоремой 7.1.8, видим, что для любых функций $f : A \to C$ и $g : B \to C$ между n-типами, их обратный образ

$$A \times_C B :\equiv \sum_{(x:A)} \sum_{(y:B)} (f(x) = g(y))$$

(см. упражнение 2.11) также является n-типом. Более обще, n-типы замкнуты под всеми пределами.

Теорема 7.1.9. *Пусть $n \geqslant -2$, $A : \mathcal{U}$ и $B : A \to \mathcal{U}$. Если для любого $a : A$, $B(a)$ — n-тип, то $\prod_{(x:A)} B(x)$ также является n-типом.*

Доказательство. Проведем индукцией по n. Для $n = -2$, результат — просто лемма 3.11.6.

Для шага индукции предположим, что результат верен для n-типов и что каждый $B(a)$ является $(n+1)$-типом. Пусть $f, g : \prod_{(a:A)} B(a)$. Нужно показать, что $f = g$ является n-типом. По функциональной экстенсиональности и замкнутости n-типов при эквивалентности достаточно показать, что $\prod_{(a:A)}(f(a) =_{B(a)} g(a))$ является n-типом. Но это следует из индуктивного предположения. ☐

Как частный случай этой теоремы, функциональное пространство $A \to B$ является n-типом при условии, что B является n-типом. Теперь мы можем обобщить наблюдения в главе 2, что $\mathsf{isSet}(A)$ и $\mathsf{isProp}(A)$ являются простыми высказываниями.

Теорема 7.1.10. *Для любого $n \geqslant -2$ и любого типа X тип $\mathsf{is}\text{-}n\text{-}\mathsf{type}(X)$ является простым высказыванием.*

Доказательство. Проведем индукцией по n.

Для базового случая нужно показать, что, для любого X, тип $\mathsf{isContr}(X)$ является простым высказыванием. Это лемма 3.11.4.

Для шага индукции надо показать

$$\prod_{X:\mathcal{U}} \mathsf{isProp}(\mathsf{is}\text{-}n\text{-}\mathsf{type}(X)) \to \prod_{X:\mathcal{U}} \mathsf{isProp}(\mathsf{is}\text{-}(n+1)\text{-}\mathsf{type}(X)).$$

Чтобы показать заключение этой импликации, нужно показать, что для любого типа X, тип

$$\prod_{x,x':X} \mathsf{is}\text{-}n\text{-}\mathsf{type}(x = x')$$

является простым высказыванием. В силу примера 3.6.2 или теоремы 7.1.9 достаточно показать, что, для всех $x, x' : X$, тип $\mathsf{is}\text{-}n\text{-}\mathsf{type}(x =_X x')$ является простым высказыванием. Но это следует из индуктивного предположения, примененного к типу $(x =_X x')$. ☐

Наконец, мы покажем, что тип n-типов сам по себе является $(n+1)$-типом. Определим это как:

$$n\text{-}\mathsf{Type} :\equiv \sum_{X:\mathcal{U}} \mathsf{is}\text{-}n\text{-}\mathsf{type}(X).$$

При необходимости, можно специфицировать универсум $\mathcal{U}$, записав $n\text{-}\mathsf{Type}_{\mathcal{U}}$. В частности, имеем $\mathsf{Prop} :\equiv (-1)\text{-}\mathsf{Type}$ и $\mathsf{Set} :\equiv 0\text{-}\mathsf{Type}$, как определено в главе 2. Заметим, что так же, как и для Prop и Set, поскольку $\mathsf{is}\text{-}n\text{-}\mathsf{type}(X)$ является простым высказыванием, по лемме 3.5.1 для любых $(X, p), (X', p') : n\text{-}\mathsf{Type}$ имеем

$$\begin{aligned}((X,p) =_{n\text{-}\mathsf{Type}} (X',p')) &\simeq (X =_{\mathcal{U}} X') \\ &\simeq (X \simeq X').\end{aligned}$$

Теорема 7.1.11. *Для любого $n \geqslant -2$ тип $n\text{-}\mathsf{Type}$ является $(n+1)$-типом.*

Доказательство. Пусть $(X, p), (X', p') : n\text{-}\mathsf{Type}$; мы хотим показать, что $(X, p) = (X', p')$ является n-типом. Согласно вышеприведенному наблюдению, этот тип эквивалентен $X \simeq X'$. Далее заметим, что проекция

$$(X \simeq X') \to (X \to X')$$

является вложением, так что, если $n \geqslant -1$, то по теореме 7.1.6 достаточно показать, что $X \to X'$ является n-типом. Но так как n-типы сохраняются под стрелочным типом, это сводится к предположению, что X' является n-типом.

В случае $n = -2$ этот аргумент показывает, что $X \simeq X'$ является (-1)-типом, но он к тому же и обитаем, так как любые два стягиваемые типы эквивалентны $\mathbf{1}$ и, следовательно, друг другу. Таким образом, $X \simeq X'$ также является (-2)-типом. □

7.2 Доказательства единственности тождественности и теорема Хедберга

В §3.1 мы определили тип X как множество, если для всех $x, y : X$ и $p, q : x =_X y$ имеет место $p = q$. В традиционной теории типов это свойство имеет название **доказательств единственности тождественности (UIP)**. Мы также видели, что оно эквивалентен 0-типу в смысле предыдущего раздела. Вот еще одна эквивалентная характеристика, включающая «Аксиому K» Стрейчера (Streicher) [Str93b]:

Теорема 7.2.1. *Тип X является множеством тогда и только тогда, когда он удовлетворяет **Аксиоме K**: для всех $x : X$ и $p : (x =_A x)$ имеет место $p = \mathsf{refl}_x$.*

Доказательство. Ясно, что аксиома K является частным случаем UIP. Обратно, если X удовлетворяет аксиоме K, пусть $x, y : X$ и $p, q : (x = y)$; надо показать, что $p = q$. Но индукция по q сводит эту цель именно к аксиоме K. □

Подчеркнем, что *мы* не предполагаем UIP или принцип K в качестве аксиом! Это просто свойства, которым конкретный тип может или не может удовлетворять (что эквивалентно тому, чтобы быть множеством). Напомним пример 3.1.9 о том, что *не* все типы являются множествами.

Следующая теорема — еще один полезный способ показать, когда типы — это множества.

Теорема 7.2.2. *Предположим, что R — рефлексивное простое отношение на типе X, означающее тождественность. Тогда X является множеством, а $R(x, y)$ эквивалентно $x =_X y$ для всех $x, y : X$.*

Доказательство. Пусть $\rho : \prod_{(x:X)} R(x, x)$ свидетельство рефлексивности R и пусть $f : \prod_{(x,y:X)} R(x, y) \to$ $(x =_X y)$ — свдетельство того, что R подразумевает тождественность. Заметим сначала, что два утверждения в теореме эквивалентны. С одной стороны, если X — множество, то $x =_X y$ — простое высказывание, и поскольку оно логически эквивалентно простому высказыванию $R(x, y)$ по условию, оно также должно быть эквивалентно ему. С другой стороны, если $x =_X y$ эквивалентно $R(x, y)$, то, как и последнее, это простое высказывание для всех $x, y : X$ и, следовательно, X является множеством.

Приведем два доказательства этой теоремы. Первое показывает, что X — множество; второе — что $R(x, y) \simeq (x = y)$.

Первое доказательство: покажем, что X является множеством. Идея такая же, как и в лемме 3.3.4: функция f должна быть непрерывной по своим аргументам x и y. Тем не менее, это несколько более сложно на уровне обозначений, потому что нам приходится иметь дело с дополнительным аргументом типа $R(x, y)$.

Сначала, для любых $x : X$ и $p : x =_X x$ рассмотрим $\mathsf{apd}_{f(x)}(p)$. Это зависимый путь от $f(x, x)$ к себе. Так как $f(x, x)$ является функцией $R(x, x) \to (x =_X y)$, то по лемме 2.9.6 это дает для любого $r : R(x, x)$ путь

$$p_*(f(x, x, r)) = f(x, x, p_*(r))\,.$$

С левой стороны мы имеем транспортировку в единичном типе, которая является конкатенацией. А для правой части имеем $p_*(r) = r$, так как и $p_*(r)$ и r содержатся в простом высказывании $R(x, x)$. Таким образом, подставляя $r :\equiv \rho(x)$, получим

$$f(x, x, \rho(x)) \cdot p = f(x, x, \rho(x))\,,$$

откуда $p = \mathsf{refl}_x$. Таким образом, X удовлетворяет аксиоме K и, следовательно, является множеством.

Второе доказательство: покажем, что каждая $f(x, y) : R(x, y) \to x =_X y$ является эквивалентностью. По теореме 4.7.7 достаточно показать, что f индуцирует эквивалентность пространств расслоений:

$$\left(\sum_{y:X} R(x, y)\right) \simeq \left(\sum_{y:X} x =_X y\right)\,.$$

По лемме 3.11.8 тип справа является стягиваемым, поэтому достаточно показать, что тип слева также стягиваемый. В качестве центра стягивания возьмем пару $(x, \rho(x))$. Остается показать, что для любого y : X и любого $H : R(x, y)$

$$(x, \rho(x)) = (y, H)\,.$$

Но так как $R(x, y)$ — простое высказывание, то по теореме 2.7.2 достаточно показать, что $x =_X y$, которое мы получаем из $f(H)$. □

Следствие 7.2.3. *Если тип X обладает свойством $\neg\neg(x = y) \to (x = y)$ для всех , то X является множеством.*

Другим удобным способом показать, что тип является множеством, является следующий. Напомним из §3.4, что тип X называется *разрешимым равенством*, если для всех $x, y : X$ имеет место

$$(x =_X y) + \neg(x =_X y)\,.$$

Это очень сильное условие: в нем говорится, что путь $x = y$ может быть выбран, когда он существует, непрерывно (или вычислимо или функториально) по x и y. Это, как оказывается, означает, что X является множеством в силу теоремы 7.2.2 и следующей леммы.

Лемма 7.2.4. *Для любого типа A имеет место $(A + \neg A) \to (\neg\neg A \to A)$.*

Доказательство. Это, по существу, уже доказано в следствии 3.2.7, но мы повторим аргументацию. Предположим, что $x : A + \neg A$. Рассмотрим два случая. Если x является $\mathsf{inl}(a)$ для некоторого $a : A$, то имеем константную функцию $\neg\neg A \to A$, которая отображает все к a. Если x является $\mathsf{inr}(t)$ для некоторого $t : \neg A$, то $g(t) : \mathbf{0}$ для любого $g : \neg\neg A$. Следовательно, можно использовать «из ложного — что угодно» (ex falso quodlibet), что есть $\mathsf{rec}_{\mathbf{0}}$, чтобы получить элемент из A, для любого $g : \neg\neg A$. □

Теорема 7.2.5 (Хедберг)**.** *Если X содержит разрешимое равенство, то X является множеством.*

Доказательство. Если X содержит разрешимое равенство, то $\neg\neg(x = y) \to (x = y)$ для всех $x, y : X$. Поэтому теорема Хедберга следует из следствия 7.2.3. □

Существует, конечно, сильная связь между этой теоремой и следствием 3.2.7. Утверждение LEM_∞, которое отрицается по следствию 3.2.7, ясно означает, что каждый тип имеет разрешимое равенство и, следовательно, является множеством, что, как мы знаем, не так. Заметим, что согласованная аксиома LEM из §3.4 влечет за собой только то, что каждый тип имеет *просто разрешимое равенство*, т.е. что для любого A имеем

$$\prod_{a,b:A} (\|a = b\| + \neg \|a = b\|) .$$

В качестве примера применения теоремы 7.2.5 напомним, что в примере 3.1.4 мы заметили, что $\mathbb{N}$ является множеством, используя нашу характеризацию его типов равенства в §2.13. Более традиционное доказательство этой теоремы использует только (2.13.2) и (2.13.3), а не полную характеризацию из теоремы 2.13.1, с теоремой 7.2.5 для заполнения пробелов.

Теорема 7.2.6. *Тип $\mathbb{N}$ натуральных чисел содержит разрешимое равенсто, и поэтому является множеством.*

Доказательство. Пусть заданы $x, y : \mathbb{N}$; будем действовать индукцией по x и анализом случаев с y, чтобы доказать $(x = y) + \neg(x = y)$. Если $x \equiv 0$ и $y \equiv 0$, то положим $\mathsf{inl}(\mathsf{refl}_0)$. Если $x \equiv 0$, а $y \equiv \mathsf{succ}(n)$, то по (2.13.2) примем $\neg(0 = \mathsf{succ}(n))$.

Для шага индукции, пусть $x \equiv \mathsf{succ}(n)$. Если $y \equiv 0$, опять воспользуемся (2.13.2). Наконец, если $y \equiv \mathsf{succ}(m)$, предположение индукции дает $(m = n) + \neg(m = n)$. В первом случае, если $p : m = n$, то $\mathsf{succ}(p) : \mathsf{succ}(m) = \mathsf{succ}(n)$. А во втором случае, (2.13.3) дает $\neg(\mathsf{succ}(m) = \mathsf{succ}(n))$. □

Хотя теорема Хедберга представляется весьма специфичной для множеств (0-типов), «аксиома K» естественно обобщается на n-типы. Заметим, что обычная аксиома K (как свойство типа X) утверждает, что для всех $x : X$ пространство петель $\Omega(X, x)$ (см. определение 2.1.8) стягиваемо. Так как $\Omega(X, x)$ всегда обитаемо (посредством refl_x), это эквивалентно тому, что оно является простым высказыванием ((-1)-типом). Так как $0 = (-1) + 1$, это предполагает следующее обобщение.

Теорема 7.2.7. *Для любого $n \geqslant -1$ тип X является $(n+1)$-типом тогда и только тогда, когда для всех $x : X$ тип $\Omega(X, x)$ является n-типом.*

Перед тем, как это доказать, докажем вспомогательную лемму.

Лемма 7.2.8. *Дано $n \geqslant -1$ и $X : \mathcal{U}$. Если при любом обитателе X является n-типом, то безотносительно этого X является n-типом.*

Доказательство. Пусть имеется отображение $f : X \to \mathsf{is}\text{-}n\text{-}\mathsf{type}(X)$. Нужно показать, что для любых $x, x' : X$ тип $x = x'$ является $(n-1)$-типом. Но тогда $f(x)$ показывает, что X является n-типом, поэтому все его пространства путей являются $(n-1)$-типами. □

Доказательство теоремы 7.2.7. Предписание «только тогда» очевидно, так как $\Omega(X,x) :\equiv (x =_X x)$. Обратно, чтобы показать, что X является $(n+1)$-типом, нужно показать, что для любых $x, x' : X$ тип $x = x'$ является n-типом. Следуя лемме 7.2.8, достаточно предъявить отображение

$$(x = x') \to \mathsf{is}\text{-}n\text{-}\mathsf{type}(x = x')\,.$$

По индукции пути достаточно сделать это, когда $x \equiv \mathrm{x}'$, и тогда это следует из предположения, что $\Omega(X,x)$ является n-типом. □

Используя индукцию и чуть изощренного вискеринга, можно получить обобщение свойства K на $n > 0$.

Теорема 7.2.9. *Для любого $n \geqslant -1$ тип A является n-типом тогда и только тогда, когда $\Omega^{n+1}(A,a)$ является стягиваемым для всех $a : A$.*

Доказательство. Напомним, что $\Omega^0(A,a) = (A,a)$, при $n = -1$, упражнение 3.5. Если $n = 0$, то это — теорема 7.2.1. Далее используем индукцию; предположим, что утверждение выполнено для $n : \mathbb{N}$. По теореме 7.2.7, A является $(n+1)$-типом тогда и только тогда, когда $\Omega(A,a)$ является n-типом для всех $a : A$. По предположению индукции, последнее эквивалентно тому, что $\Omega^{n+1}(\Omega(A,a),p)$ является стягиваемым для всех $p : \Omega(A,a)$.

Поскольку $\Omega^{n+2}(A,a) :\equiv \Omega^{n+1}(\Omega(A,a),\mathsf{refl}_a)$ и $\Omega^{n+1} = \Omega^n \circ \Omega$, достаточно показать, что $\Omega(\Omega(A,a),p)$ равно $\Omega(\Omega(A,a),\mathsf{refl}_a)$ в типе $\mathcal{U}_\bullet$ точечных типов. Для этого достаточно предоставить эквивалентность

$$g : \Omega(\Omega(A,a),p) \simeq \Omega(\Omega(A,a),\mathsf{refl}_a)\,,$$

которая переносит отмеченную точку refl_p на отмеченную точку $\mathsf{refl}_{\mathsf{refl}_a}$. Для $q : p = p$ определим $g(q) : \mathsf{refl}_a = \mathsf{refl}_a$ в следующем составном виде:

$$\mathsf{refl}_a = p \cdot p^{-1} \overset{q}{=} p \cdot p^{-1} = \mathsf{refl}_a\,,$$

где путь, помеченный «q» на самом деле есть $\mathsf{ap}_{\lambda r.\, r \cdot p^{-1}}(q)$. Тогда g является эквивалентностью, потому что это композиция эквивалентностей

$$(p = p) \xrightarrow{\lambda r.\, r \cdot p^{-1}} (p \cdot p^{-1} = p \cdot p^{-1}) \xrightarrow{i \cdot \_ \cdot i^{-1}} (\mathsf{refl}_a = \mathsf{refl}_a)\,,$$

используя пример 2.4.8 и теорему 2.11.1, где $i : \mathsf{refl}_a = p \cdot p^{-1}$ — это каноническое равенство. И очевидно, что $g(\mathsf{refl}_p) = \mathsf{refl}_{\mathsf{refl}_a}$. □

7.3 Усечения

В §3.7 мы ввели пропозициональное усечение, которое дает «наилучшее приближение» типа, являющегося простым высказыванием, т.е. (1)-типом. В §6.9 мы построили это усечение как высший индуктивный тип и дали один из способов обобщить его на 0-усечение. Теперь мы объясним лучшее обобщение этого, которое усекает любой тип в n-тип для любого $n \geqslant -2$; в классической гомотопической теории это можно было бы назвать n-м **сечением Постникова**.

Идея состоит в том, чтобы использовать теорему 7.2.9, которая гласит, что A является n-типом, когда $\Omega^{n+1}(A,a)$ стягиваемо, для всех $a : A$ и леммы 6.5.4, откуда следует, что $\Omega^{n+1}(A,a) \simeq \mathsf{Map}_*(\mathbb{S}^{n+1},(A,a))$, где $\mathbb{S}^{n+1}$ снабжен некоторой отмеченной точкой, которую мы можем также обозначить base. Однако, стягиваемость $\mathsf{Map}_*(\mathbb{S}^{n+1},(A,a))$ — это то, что мы можем обеспечить непосредственно, предоставив конструкторы путей.

Мы будем использовать конструкцию «концентратор и спица», как в §6.7. Таким образом, для $n \geqslant -1$, возьмем $\|A\|_n$ в качестве высшего индуктивного типа, порожденного:

- функцией $|\_|_n : A \to \|A\|_n$,
- для каждого $r : \mathbb{S}^{n+1} \to \|A\|_n$, точкой *концентратора* $h(r) : \|A\|_n$, и
- для каждого $r : \mathbb{S}^{n+1} \to \|A\|_n$ и каждого $x : \mathbb{S}^{n+1}$, путем *спицы* $s_r(x) : r(x) = h(r)$.

Существование этих конструкторов достаточно, чтобы показать следующее:

Лемма 7.3.1. *$\|A\|_n$ является n-типом.*

Доказательство. В силу теоремы 7.2.9 достаточно показать, что $\Omega^{n+1}(\|A\|_n, b)$ стягиваемо для всех $b : \|A\|_n$, что по лемме 6.5.4 эквивалентно $\mathsf{Map}_*(\mathbb{S}^{n+1},(\|A\|_n, b))$. В качестве центра стягивания для последнего выберем функцию $c_b : \mathbb{S}^{n+1} \to \|A\|_n$, которая постоянна при b, вместе с $\mathsf{refl}_b : c_b(\mathsf{base}) = b$.

Теперь, произвольный элемент из $\mathsf{Map}_*(\mathbb{S}^{n+1},(\|A\|_n, b))$ состоит из отображения $r : \mathbb{S}^{n+1} \to \|A\|_n$ вместе с путем $p : r(\mathsf{base}) = b$. По функциональной экстенсиональности, чтобы показасть, что $r = c_b$, достаточно, для каждого $x : \mathbb{S}^{n+1}$, предъявить путь $r(x) = c_b(x) \equiv b$. Выберем его как композицию $s_r(x) \cdot s_r(\mathsf{base})^{-1} \cdot p$, где $s_r(x)$ — спица при x.

Наконец, мы должны показать, что при транспортировке вдоль равенства, $r = c_b$, путь p становится refl_b. При транспортировке по типам путей это означает, что нам необходимо выполнение равенства

$$(s_r(\mathsf{base}) \cdot s_r(\mathsf{base})^{-1} \cdot p)^{-1} \cdot p = \mathsf{refl}_b\,.$$

Но это следует непосредственно из операций пути. □

(Эта конструкция не выполняется при $n = -2$, но в этом случае мы можем просто определить $\|A\|_{-2} :\equiv \mathbf{1}$ для всех A. Поэтому, с этого момента предполагаем, что $n \geqslant -1$)

Чтобы вывести необходимое универсальное свойство n-усечения, нам нужен принцип индукции. Мы извлекаем его из конструкторов обычным способом; он говорит, что для $P : \|A\|_n \to \mathcal{U}$ вместе с

- элементом $g(a) : P(|a|_n)$ для каждого $a : A$,
- элементом $h'(r, r') : P(h(r))$ для любых $r : \mathbb{S}^{n+1} \to \|A\|_n$ и $r' : \prod_{(x:\mathbb{S}^{n+1})} P(r(x))$,
- зависимым путем $r'(x) =^P_{s_r(x)} h'(r, r')$ для любых $r : \mathbb{S}^{n+1} \to \|A\|_n$ и $r' : \prod_{(x:\mathbb{S}^{n+1})} P(r(x))$, и каждого $x : \mathbb{S}^{n+1}$,

существует сечение $f : \prod_{(x:\|A\|_n)} P(x)$ с $f(|a|_n) \equiv g(a)$, для всех $a : A$. Чтобы сделать принцип более полезным, переформулируем его следующим образом.

Теорема 7.3.2. *Для любого семейства типов $P : \|A\|_n \to \mathcal{U}$ такого, что каждый $P(x)$ есть n-тип, и любой функции $g : \prod_{(a:A)} P(|a|_n)$, существует сечение $f : \prod_{(x:\|A\|_n)} P(x)$ такое, что $f(|a|_n) :\equiv g(a)$ для всех $a : A$.*

Доказательство. Достаточно построить данные второго и третьего пунктов принципа, перечисленные выше, так как g имеет точно тот тип данных, который указан в первом пункте. Для $r : \mathbb{S}^{n+1} \to \|A\|_n$ и $r' : \prod_{(x:\mathbb{S}^{n+1})} P(r(x))$ имеем $h(r) : \|A\|_n$ и $s_r : \prod_{(x:\mathbb{S}^{n+1})}(r(x) = h(r))$. Определим $t : \mathbb{S}^{n+1} \to P(h(r))$ посредством $t(x) :\equiv s_r(x)_*(r'(x))$. Тогда, поскольку $P(h(r))$ является n-усеченным, существует точка $u : P(h(r))$ и стягивание $v : \prod_{(x:\mathbb{S}^{n+1})}(t(x) = u)$. Определим $h'(r, r') :\equiv u$, удовлетворяя выполнимость второго пункта принципа. Далее (ссылаясь на определение зависимых путей), v имеет именно тот тип, который требуется для третьего пункта принципа. □

В частности, если E — некоторый n-тип, то можно рассматривать постоянное семейство типов, равное E для каждой точки A. Таким образом, каждое отображение $f : A \to E$ можно продолжить до отображения $\mathsf{ext}(f) : \|A\|_n \to E$, определяемое как $\mathsf{ext}(f)(|a|_n) :\equiv f(a)$; это *принцип рекурсии* для $\|A\|_n$.

Принцип индукции также подразумевает принцип единственности для функций этого вида. Именно, если E — некоторый n-тип и $g, g' : \|A\|_n \to E$ такие, что $g(|a|_n) = g'(|a|_n)$ для любого $a : A$, то $g(x) = g'(x)$ для всех $x : \|A\|_n$, поскольку тип $g(x) = g'(x)$ является n-типом. Таким образом, $g = g'$ (на самом деле этот принцип единственности имеет более общий характер, когда E является $(n+1)$-типом). Это порождает следующее универсальное свойство.

Лемма 7.3.3 (Универсальное свойство усечений)**.** *Пусть $n \geqslant -2$, $A : \mathcal{U}$ и $B : n\text{-}\mathsf{Type}$. Следующее отображение является эквивалентностью:*

$$\begin{cases} (\|A\|_n \to B) & \longrightarrow (A \to B) \\ \qquad\qquad g & \longmapsto g \circ |\_|_n \end{cases}$$

Доказательство. Считая, что B — n-усеченный, любую $f : A \to B$ можно расширить до отображения $\mathsf{ext}(f) : \|A\|_n \to E$. Отображение $\mathsf{ext}(f) \circ |\_|_n$ равно f, потому что для любого $a : A$ имеем $\mathsf{ext}(f)(|a|_n) = f(a)$ по определению. А отображение $\mathsf{ext}(g) \circ |\_|_n$ равно g, потому что они оба пересылают $|a|_n$ к $g(|a|_n)$. □

Говоря категорным языком, n-типы образуют рефлективную подкатегорию категории типов (чтобы сформулировать это более точно, следует использовать язык $(\infty, 1)$-категорий). В частности, это означает, что n-усечение функториально: для $f : A \to B$ применение принципа рекурсии к композиции $A \xrightarrow{f} B \to \|B\|_n$ порождает отображение $\|f\|_n : \|A\|_n \to \|B\|_n$. По определению, имеется гомотопия

$$\mathsf{nat}^f_n : \prod_{a:A} \|f\|_n (|a|_n) = |f(a)|_n \,, \tag{7.3.4}$$

выражающую *естественность* отображений $|\_|_n$.

Единственность подразумевает наличие законов функториальности, таких как $\|g \circ f\|_n = \|g\|_n \circ \|f\|_n$ и $\|\mathsf{id}_A\|_n = \mathsf{id}_{\|A\|_n}$, с соответствующими законами когерентности (согласованности). Также, имеется высшая функториальность, например:

Лемма 7.3.5. *Для $f, g : A \to B$ и гомотопии $h : f \sim g$ существует индуцированная гомотопия $\|h\|_n : \|f\|_n \to \|g\|_n$ такая, что компоновка*

$$|f(a)|_n \xlongequal{\mathsf{nat}^f_n(a)^{-1}} \|f\|_n (|a|_n) \xlongequal{\|h\|_n(|a|_n)} \|g\|_n (|a|_n) \xlongequal{\mathsf{nat}^g_n(a)} |g(a)|_n \tag{7.3.6}$$

равна $\mathsf{ap}_{|\_|_n}(h(a))$.

Доказательство. Во-первых, фактически имеется гомотопия с компонентами $\mathsf{ap}_{|\_|_n}(h(a)) : |f(a)|_n = |g(a)|_n$. Проведя компоновку обеих сторон с путями $|f(a)|_n = \|f\|_n(|a|_n)$ и $|g(a)|_n = \|g\|_n(|a|_n)$, которые возникают из определений $\|f\|_n$ и $\|g\|_{n'}$, мы получаем гомотопию $(\|f\|_n \circ |\_|_n) \sim (\|g\|_n \circ |\_|_n)$ и, следовательно, равенство по функциональной экстенсиональности. Но, поскольку $(\_ \circ |\_|_n)$ является эквивалентностью, должен существовать путь $\|f\|_n = \|g\|_n$, индуцирующий ее, и из законов когерентности для функциональной экстенсиональности следует (7.3.6). □

Следующее наблюдение за рефлективными подкатегориями также является стандартным.

Следствие 7.3.7. *Тип A является n-типом, тогда и только тогда, когда $|\_|_n : A \to \|A\|_n$ является эквивалентностью.*

Доказательство. «если» следует из замкнутости n-типов под эквивалентностью. С другой стороны, если A является n-типом, мы можем определить $\mathsf{ext}(\mathsf{id}_A) : \|A\|_n \to A$. Тогда, по определению, имеем $\mathsf{ext}(\mathsf{id}_A) \circ |\_|_n = \mathsf{id}_A : A \to A$. Чтобы доказать, что $|\_|_n \circ \mathsf{ext}(\mathsf{id}_A) = \mathsf{id}_{\|A\|_n}$, достаточно только доказать, что $|\_|_n \circ \mathsf{ext}(\mathsf{id}_A) \circ |\_|_n = \mathsf{id}_{\|A\|_n} \circ |\_|_n$. Это снова верно:

$$\begin{tikzcd}
A \arrow[r, "|_|_n"] \arrow[dr, "\mathsf{id}_A"'] & \|A\|_n \arrow[d, "\mathsf{ext}(\mathsf{id}_A)"] \arrow[dd, bend left=60, "\mathsf{id}_{\|A\|_n}"] \\
& A \arrow[d, "|_|_n"'] \\
& \|A\|_n
\end{tikzcd}$$

□

Категория n-типов также имеет некоторые особые свойства, которыми не обладают все рефлективные подкатегории. Например, рефлектор $\|\_\|_n$ сохраняет конечные произведения.

Теорема 7.3.8. *Для любых типов A и B, индуцированное отображение $\|A \times B\|_n \to \|A\|_n \times \|B\|_n$ является эквивалентностью.*

Доказательство. Достаточно показать, что $\|A\|_n \times \|B\|_n$ имеет такое же универсальное свойство, что и $\|A \times B\|_n$. Таким образом, пусть C — n-тип; тогда

$$\begin{aligned}
(\|A\|_n \times \|B\|_n \to C) &= (\|A\|_n \to (\|B\|_n \to C)) \\
&= (\|A\|_n \to (B \to C)) \\
&= (A \to (B \to C)) \\
&= (A \times B \to C)
\end{aligned}$$

используя универсальные свойства для $\|B\|_n$ и $\|A\|_n$, а также то, что $B \to C$ является n-типом, так как таковым, по предположению, является C. Нетрудно проверить, что требуемая эквивалентность задается композицией с $|\_|_n \times |\_|_n$. □

Часто полезно использовать следующий, связанный с этим, факт о зависимых суммах.

Теорема 7.3.9. *Пусть $P : A \to \mathcal{U}$ — семейство типов. Тогда существует эквивалентность*

$$\left\| \sum_{x:A} \|P(x)\|_n \right\|_n \simeq \left\| \sum_{x:A} P(x) \right\|_n .$$

Доказательство. Мы несколько раз используем принцип индукции для n-усечения при построении функций

$$\varphi : \left\| \sum_{x:A} \|P(x)\|_n \right\|_n \to \left\| \sum_{x:A} P(x) \right\|_n$$
$$\psi : \left\| \sum_{x:A} P(x) \right\|_n \to \left\| \sum_{x:A} \|P(x)\|_n \right\|_n$$

и гомотопий $H : \varphi \circ \psi \sim \mathsf{id}$ и $K : \psi \circ \varphi \sim \mathsf{id}$, демонстрируя их как квазиобратные. Определим φ, полагая $\varphi(|(x, |u|_n)|_n) :\equiv |(x, u)|_n$, и ψ, полагая $\psi(|(x, u)|_n) :\equiv |(x, |u|_n)|_n$. Затем мы определяем $H(|(x, u)|_n) :\equiv \mathsf{refl}_{|(x,u)|_n}$ и $K(|(x, |u|_n)|_n) :\equiv \mathsf{refl}_{|(x,|u|_n)|_n}$. □

Следствие 7.3.10. *Если A — n-тип, а $P : A \to \mathcal{U}$ — любое семейство типов, то*

$$\sum_{a:A} \|P(a)\|_n \simeq \left\| \sum_{a:A} P(a) \right\|_n .$$

Доказательство. Если A является n-типом, то тип слева (заключения) уже является n-типом, что эквивалентно его n-усечению; поэтому само заключение следует из теоремы 7.3.9. □

Можно охарактеризовать пространства путей усечения с использованием того же метода, который был использован в §§ 2.2 и 2.3 для копроизведений и натуральных чисел (и который будет использован в главе 8 для вычисления гомотопических групп). Неудивительно, что пространства путей в $(n+1)$-усечении для A являются n-усечениями пространств путей. Действительно, для любых $x, y : A$ существует каноническое отображение

$$f : \|x =_A y\|_n \to \left(|x|_{n+1} =_{\|A\|_{n+1}} |y|_{n+1} \right) , \tag{7.3.11}$$

определяемое как

$$f(|p|_n) :\equiv \mathsf{ap}_{|\_|_{n+1}}(p) .$$

Это определение использует принцип рекурсии для $\|\_\|_n$, что верно, потому что $\|A\|_{n+1}$ является $(n+1)$-усеченным, так что кообласть f является n-усеченной.

Теорема 7.3.12. *Для любых A, $x, y : A$ и $n \geqslant 2$ отображение (7.3.11) является эквивалентностью, т.е.*

$$\|x =_A y\|_n \simeq \left(|x|_{n+1} =_{\|A\|_{n+1}} |y|_{n+1} \right) .$$

Доказательство. Доказательство представляет собой просто применение метода кодирования-декодирования: как и в предыдущих ситуациях, мы не можем напрямую определить квазиобратный к отображению (7.3.11), потому что нет способа ввести в действие равенство между $|x|_{n+1}$ и $|y|_{n+1}$. Таким образом, вместо этого мы обобщаем его тип, чтобы иметь общие элементы типа $\|A\|_{n+1}$ вместо $|x|_{n+1}$ и $|y|_{n+1}$. Определим $P : \|A\|_{n+1} \to \|A\|_{n+1} \to n\text{-}\mathsf{Type}$ как

$$P\left(|x|_{n+1}, |y|_{n+1}\right) :\equiv \|x =_A y\|_n .$$

Это определение корректно, потому что $\|x =_A y\|_n$ является n-усеченным, а n-Type является $(n+1)$-усеченным по теореме 7.1.11. Теперь, для любых $u, v : \|A\|_{n+1}$ существует отображение

$$\mathsf{decode} : P(u,v) \to \left(u =_{\|A\|_{n+1}} v\right)$$

определенное для $u = |x|_{n+1}$, $v = |y|_{n+1}$ и $p : x = y$ как

$$\mathsf{decode}(|p|_n) :\equiv \mathsf{ap}_{|{-}|_{n+1}}(p)\,.$$

Поскольку кообласть decode является n-усеченной, ее достаточно определить только для u и v этого вида, и тогда это будет то же определение, что и приведенное выше. Определим также функцию

$$r : \prod_{u:\|A\|_{n+1}} P(u,u)$$

индукцией по u, где $r(|x|_{n+1}) :\equiv |\mathsf{refl}_x|_n$.

Теперь мы можем определить обратное отображение

$$\mathsf{encode} : \left(u =_{\|A\|_{n+1}} v\right) \to P(u,v)$$

как

$$\mathsf{encode}(p) :\equiv \mathsf{transport}^{v \mapsto P(u,v)}(p, r(u))\,.$$

Чтобы показать, что компоновка

$$\left(u =_{\|A\|_{n+1}} v\right) \xrightarrow{encode} P(u,v) \xrightarrow{decode} \left(u =_{\|A\|_{n+1}} v\right)$$

является функцией тождественности, по индукции пути достаточно проверить ее для $\mathsf{refl}_u : u = u$, и в этом случае нам нужно показать, что $\mathsf{encode}(r(u)) = \mathsf{refl}_u$. Но, так как это n-тип, а следовательно, также и $(n+1)$-тип, можно считать, что $u \equiv |x|_{n+1}$, и в этом случае равенство получается верным по определению r и encode. Наконец, чтобы показать, что

$$P(u,v) \xrightarrow{decode} \left(u =_{\|A\|_{n+1}} v\right) \xrightarrow{encode} P(u,v)$$

является функцией тождественности, то так как цель снова является n-типом, мы можем предположить, что $u = |x|_{n+1}$ и $v = |y|_{n+1}$, и что мы рассматриваем $|p|_n : P(|x|_{n+1}, |y|_{n+1})$ для некоторого p: x = y. Тогда имеем

$$\begin{aligned}
\mathsf{encode}(\mathsf{decode}(|p|_n)) &= \mathsf{encode}(\mathsf{ap}_{|{-}|_{n+1}}(p)) \\
&= \mathsf{transport}^{v \mapsto P(|x|_{n+1}, v)}(\mathsf{ap}_{|{-}|_{n+1}}(p), |\mathsf{refl}_x|_n) \\
&= \mathsf{transport}^{v \mapsto \|u=v\|_n}(p, |\mathsf{refl}_x|_n) \\
&= \left|\mathsf{transport}^{v \mapsto (u=v)}(p, \mathsf{refl}_x)\right|_n \\
&= |p|_n\,.
\end{aligned}$$

Это завершает доказательство того, что декодирование и кодирование являются квазиобратными. Указанный результат — это особый случай, когда $u = |x|_{n+1}$ и $v = |y|_{n+1}$. □

Следствие 7.3.13. *Пусть $n \geqslant 2$ и (A, a) — точечный тип. Тогда*

$$\|\Omega(A,a)\|_n = \Omega\,\|(A,a)\|_{n+1}\,.$$

Доказательство. Это частный случай предыдущей леммы, когда $x = y = a$. □

Следствие 7.3.14. *Пусть $n \geqslant 2$, $k \geqslant 0$ и (A, a) — точечный тип. Тогда*

$$\left\|\Omega^k(A, a)\right\|_n = \Omega^k \left\|(A, a)\right\|_{n+k} .$$

Доказательство. Индукцией по k, используя рекурсивное определение Ω^k. □

Мы также отмечаем, что «усечения кумулятивны»: если мы усекаем что-то до n-типа, а затем до k-типа при $k \leqslant n$, то мы могли бы также провести одно усечение сразу до k-типа.

Лемма 7.3.15. *Пусть $k, n \geqslant -2$ с $k \leqslant n$ и $A : \mathcal{U}$. Тогда $\left\|\|A\|_n\right\|_k = \|A\|_k$.*

Доказательство. Определим два отображения $f : \left\|\|A\|_n\right\|_k \to \|A\|_k$ и $g : \|A\|_k \to \left\|\|A\|_n\right\|_k$ как

$$f(\left||a|_n\right|_k) :\equiv |a|_k \quad \text{и} \quad g(|a|_k) :\equiv \left||a|_n\right|_k .$$

Отображение f корректно определено, т.к. $\|A\|_k$ является k-усеченным, а также n-усеченным (поскольку $k \leqslant n$), и отображение g корректно определено, потому что $\left\|\|A\|_n\right\|_k$ является k-усеченным.

Композиция $f \circ g : \|A\|_k \to \|A\|_k$ удовлетворяет равенству $(f \circ g)(|a|_k) = |a|_k$, поэтому $f \circ g = \mathsf{id}_{\|A\|_k}$. Аналогично, имеем $(g \circ f)(\left||a|_n\right|_k) = \left||a|_n\right|_k$ и поэтому $g \circ f = \mathsf{id}_{\left\|\|A\|_n\right\|_k}$. □

7.4 Копределы n-типов

Напомним, что в §6.8 мы использовали высшие индуктивные типы для определения амальгам типов и доказали их универсальное свойство. В общем случае (гомотопический) копредел n-типов может не быть n-типом (контрпример см. в упражнении 7.2). Однако, если мы проводим для него n-усечение, то получаем n-тип, который удовлетворяет корректному универсальному свойству по отношению к другим n-типам.

В этом разделе мы докажем это для амальгам, которые являются наиболее важным и нетривиальным случаем копределов. Напомним следующие определения из §6.8.

Определение 7.4.1. Пролет — это 5-кортеж $\mathcal{D} = (A, B, C, f, g)$ с $f : C \to A$ и $g : C \to B$.

$$\mathscr{D} = \begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] & B \\ A & \end{tikzcd}$$

Определение 7.4.2. Для пролета $\mathscr{D} = (A \xleftarrow{f} C \xrightarrow{g} B)$ и типа D **коконус под $\mathscr{D}$ с базой D** — это тройка (i, j, h) с $i : A \to D$, $j : B \to D$, и $h : \prod_{(c:C)}(i(f(c)) = j(g(c)))$:

$$\begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] \arrow[dr, phantom, "\overset{h}{\Nearrow}"] & B \arrow[d, "j"] \\ A \arrow[r, "i"'] & D \end{tikzcd}$$

Обозначим через $\mathsf{cocone}_{\mathscr{D}}(D)$ тип всех таких коконусов.

Тип коконусов — (ковариантно) функториальный. Например, для D, E и отображения $t : D \to E$, существует отображение

$$\begin{cases} \mathsf{cocone}_{\mathscr{D}}(D) & \longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ c & \longmapsto t \circ c \end{cases}$$

определяемое как

$$t \circ (i, j, h) = (t \circ i, t \circ j, \mathsf{ap}_t \circ h) .$$

Также, для D, E, F, функций $t : D \to E$, $u : E \to F$ и $c : \mathsf{cocone}_{\mathscr{D}}(D)$ имеет место

$$\mathsf{id}_D \circ c = c , \tag{7.4.3}$$

$$(u \circ t) \circ c = u \circ (t \circ c) . \tag{7.4.4}$$

Определение 7.4.5. Для пролета $\mathscr{D}$ n-типов, n-типа D и коконуса $c : \mathsf{cocone}_{\mathscr{D}}(D)$, пара (D, c) называется **амальгамой $\mathscr{D}$ в n-типах**, если для каждого n-типа E, отображение

$$\begin{cases} (D \to E) & \longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ t & \longmapsto t \circ c \end{cases}$$

является эквивалентностью.

Чтобы конструировать амальгамы n-типов, объясним, как отображать пролеты и коконусы.

Определение 7.4.6. Пусть

$$\mathscr{D} = \begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] & B \\ A & \end{tikzcd}$$

— пролет. Обозначим через $\|\mathscr{D}\|_n$ следующий пролет n-типов:

$$\|\mathscr{D}\|_n :\equiv \begin{tikzcd} \|C\|_n \arrow[r, "\|g\|_n"] \arrow[d, "\|f\|_n"'] & \|B\|_n \\ \|A\|_n & \end{tikzcd}$$

Определение 7.4.7. Пусть $D : \mathcal{U}$ и $c = (i, j, h) : \mathsf{cocone}_{\mathscr{D}}(D)$. Определим

$$\|c\|_n = (\|i\|_n, \|j\|_n, \|h\|_n) : \mathsf{cocone}_{\|\mathscr{D}\|_n}(\|D\|_n) ,$$

где k есть составная гомотопия

$$\|h\|_n : \|i\|_n \circ \|f\|_n \sim \|j\|_n \circ \|g\|_n ,$$

используя лемму 7.3.5 и функториальность $\|-\|_n$.

Заметим теперь, что отображения от каждого типа к его n-усечению собираются в отображение пролетов в следующем смысле.

Определение 7.4.8. Пусть

$$\mathscr{D} = \begin{tikzcd} C \arrow[r, "g"] \arrow[d, "f"'] & B \\ A & \end{tikzcd} \quad \text{и} \quad \mathscr{D}' = \begin{tikzcd} C' \arrow[r, "g'"] \arrow[d, "f'"'] & B' \\ A' & \end{tikzcd}$$

— пролеты. **Отображение пролетов** $\mathscr{D} \to \mathscr{D}'$ состоит из функций $\alpha : A \to A'$, $\beta : B \to B'$,$\gamma : C \to C'$ и гомотопий $\phi : \alpha \circ f \sim f'$, $\psi : \beta \circ g \sim g' \circ \gamma$.

Таким образом, для любого пролета $\mathscr{D}$ имеется отображение пролетов $|\_|_n^{\mathscr{D}} : \mathscr{D} \to \|\mathscr{D}\|_n$, содержащее $|\_|_n^A$, $|\_|_n^B$, $|\_|_n^C$ и естественные гомотопии nat_n^f, nat_n^g, как в (7.3.4).

Нам также необходимо убедиться, что отображения пролетов ведут себя функториально. Именно, если $(\alpha, \beta, \gamma, \phi, \psi) : \mathscr{D} \to \mathscr{D}'$ есть отображение пролетов и D — произвольный тип, то имеет место

$$\begin{cases} \mathsf{cocone}_{\mathscr{D}'}(D) & \longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ \qquad (i, j, h) & \longmapsto (i \circ \alpha, j \circ \beta, k) \end{cases}$$

где $k : \prod_{(z:C)} i(\alpha(f(z))) = j(\beta(g(z)))$ — компоновка

$$i(\alpha(f(z))) \overset{\mathsf{ap}_i(\phi)}{=\!=\!=} i(f'(\gamma(z))) \overset{h(\gamma(z))}{=\!=\!=} i(g'(\gamma(z))) \overset{\mathsf{ap}_j(\psi)}{=\!=\!=} j(\beta(g(z))) \ . \tag{7.4.9}$$

Обозначим этот коконус через $(i, j, h) \circ (\alpha, \beta, \gamma, \phi, \psi)$. Более того, это функториальное действие коммутирует с другой функториальностью коконусов:

Лемма 7.4.10. *Для $(\alpha, \beta, \gamma, \phi, \psi) : \mathscr{D} \to \mathscr{D}'$ и $t : D \to E$ следующая диаграмма коммутативна*:

$$\begin{tikzcd} \mathsf{cocone}_{\mathscr{D}'}(D) \arrow[r, "t\circ\_"] \arrow[d] & \mathsf{cocone}_{\mathscr{D}'}(E) \arrow[d] \\ \mathsf{cocone}_{\mathscr{D}}(D) \arrow[r, "t\circ\_"'] & \mathsf{cocone}_{\mathscr{D}}(E) \end{tikzcd}$$

Доказательство. Для $(i, j, h) : \mathsf{cocone}_{\mathscr{D}'}(D)$ заметим, что обе компоновки (на диаграмме) дают коконус, первые два компонента которого равны $t \circ i \circ \alpha$ и $t \circ j \circ \beta$. Таким образом, остается проверить соответствие гомотопий. Для верхней правой компоновки диаграммы гомотопия есть (7.4.9) с заменой (i, j, h) на $(t \circ i, t \circ j, \mathsf{ap}_t \circ h)$:

$$t\, i\, \alpha\, f\, z \xlongequal{\mathsf{ap}_{t\circ i}(\phi)} t\, i\, f'\, \gamma\, z \xlongequal{\mathsf{ap}_t(h(\gamma(z)))} t\, j\, g'\, \gamma\, z \xlongequal{\mathsf{ap}_{t\circ j}(\psi)} t\, j\, \beta\, g\, z$$

(для краткости мы опускаем круглые скобки вокруг аргументов функций). С другой стороны, для левой нижней компоновки диаграммы гомотопия есть применение ap_t к (7.4.9). Так как ap не нарушает конкатенацию пути, то в этом случае имеем

$$t\, i\, \alpha\, f\, z \xlongequal{\mathsf{ap}_t(\mathsf{ap}_i(\phi))} t\, i\, f'\, \gamma\, z \xlongequal{\mathsf{ap}_t(h(\gamma(z)))} t\, j\, g'\, \gamma\, z \xlongequal{\mathsf{ap}_t(\mathsf{ap}_j(\psi))} t\, j\, \beta\, g\, z\ .$$

Но $\mathsf{ap}_t \circ \mathsf{ap}_i = \mathsf{ap}_{t\circ i}$ и аналогично для j, поэтому эти две гомотопии равны. □

Наконец, заметим, что, поскольку мы определили $\|c\|_n : \mathsf{cocone}_{\|\mathscr{D}\|_n}(\|D\|_n)$, используя лемму 7.3.5, из дополнительного условия (7.3.6) следует

$$|\_|_n^D \circ c = \|c\|_n \circ |\_|_n^D \tag{7.4.11}$$

для любых $c : \mathsf{cocone}_{\mathscr{D}}(D)$. Теперь мы можем доказать ожидаемую теорему.

Теорема 7.4.12. *Пусть $\mathscr{D}$ — пролет, а (D, c) — его амальгама. Тогда $(\|\mathscr{D}\|_n, \|c\|_n)$ является амальгамой для $\|\mathscr{D}\|_n$ в n-типах.*

Доказательство. Пусть E — n-тип; рассмотрим диаграмму:

$$\begin{tikzcd}
(\|D\|_n \to E) \arrow[r, "_\circ|_|_n^D"] \arrow[d, "_\circ\|c\|_n"'] & (D \to E) \arrow[d, "_\circ c"] \\
\mathsf{cocone}_{\|\mathscr{D}\|_n}(E) \arrow[r, "_\circ|_|_n^{\mathscr{D}}"] & \mathsf{cocone}_{\mathscr{D}}(E) \\
(\|A\|_n \to E) \times_{(\|C\|_n \to E)} (\|B\|_n \to E) \arrow[r] \arrow[u, "\ell_1"] & (A \to E) \times_{(C\to E)} (B \to E) \arrow[u, "\ell_2"']
\end{tikzcd}$$

Верхняя горизонтальная стрелка является эквивалентностью, так как E является n-типом, а $\_\circ c$ является эквивалентностью, так как c является амальгамой коконуса. Таким образом, по свойству 2-из-3, чтобы показать, что $\_ \circ \|c\|_n$ является эквивалентностью, достаточно показать, что верхний квадрат коммутативен, а средняя горизонтальная стрелка является эквивалентностью. Чтобы убедиться, что верхний квадрат коммутативен, пусть $t : \|D\|_n \to E$; тогда

$$\begin{aligned}
(t \circ \|c\|_n) \circ |_|_n^{\mathscr{D}} &= t \circ (\|c\|_n \circ |_|_n^{\mathscr{D}}) && \text{(по лемме 7.4.10)}\\
&= t \circ (|_|_n^{\mathscr{D}} \circ c) && \text{(по (7.4.11))}\\
&= (t \circ |_|_n^{\mathscr{D}}) \circ c. && \text{(по (7.4.4))}
\end{aligned}$$

Чтобы показать, что средняя горизонтальная стрелка является эквивалентностью, рассмотрим нижний квадрат. Две нижние вертикальные стрелки — это просто применения happly:

$$\ell_1(i, j, p) :\equiv (i, j, \mathsf{happly}(p))$$
$$\ell_2(i, j, p) :\equiv (i, j, \mathsf{happly}(p))$$

и, следовательно, являются эквивалентностями по функциональной экстенсиональности. Нижняя горизонтальная стрелка определяется формулой

$$(i, j, p) \longmapsto \left(i \circ |\_|_n^A, j \circ |\_|_n^B, q\right)$$

где q — компоновка

$$\begin{aligned} i \circ |\_|_n^A \circ f &= i \circ \|f\|_n \circ |\_|_n^C && (\text{по } \mathsf{funext}(\lambda z.\, \mathsf{ap}_i(\mathsf{nat}_n^f(z)))) \\ &= j \circ \|g\|_n \circ |\_|_n^C && (\text{по } \mathsf{ap}_{\_\circ|\_|_n^C}(p)) \\ &= j \circ |\_|_n^B \circ g\,. && (\text{по } \mathsf{funext}(\lambda z.\, \mathsf{ap}_j(\mathsf{nat}_n^g(z)))) \end{aligned}$$

Это эквивалентность, потому что она индуцирована эквивалентностью ко-пролетов. Таким образом, по 2-из-3, достаточно показать, что нижний квадрат коммутативен. Но две компоновки вокруг нижнего квадрата согласуются дефинициально на первых двух компонентах, поэтому достаточно показать, что для (i, j, p) в левом нижнем углу и $z : C$, путь

$$\mathsf{happly}(q, z) : i(|f(z)|_n) = j(|g(z)|_n)$$

(с q, определенном выше) равен компоновке

$$\begin{aligned} i(|f(z)|_n) &= i(\|f\|_n(|z|_n)) && (\text{по } \mathsf{ap}_i(\mathsf{nat}_n^f(z))) \\ &= j(\|g\|_n(|z|_n)) && (\text{по } \mathsf{happly}(p, |z|_n)) \\ &= j(|g(z)|_n)\,. && (\text{по } \mathsf{ap}_j(\mathsf{nat}_n^g(z))) \end{aligned}$$

Однако, поскольку happly является функториальным, достаточно проверить равенство для трехкомпонентных путей:

$$\begin{aligned} \mathsf{happly}(\mathsf{funext}(\lambda z.\, \mathsf{ap}_i(\mathsf{nat}_n^f(z))), z) &= \mathsf{ap}_i(\mathsf{nat}_n^f(z)) \\ \mathsf{happly}(\mathsf{ap}_{\_\circ|\_|_n^C}(p), z) &= \mathsf{happly}(p, |z|_n) \\ \mathsf{happly}(\mathsf{funext}(\lambda z.\, \mathsf{ap}_j(\mathsf{nat}_n^g(z))), z) &= \mathsf{ap}_j(\mathsf{nat}_n^g(z))\,. \end{aligned}$$

Первый и третий из них — это только тот факт, что happly является квазиобратным к funext, а второй — просто обычная лемма о happly и предкомпозиции. ☐

7.5 Связность

n-тип — это тот, у которого нет интересной информации при размерности выше n. Напротив, *n-связный тип* — это тот, у которого нет никакой интересной информации при размерности *ниже* n. Разумеется, естественно также исследовать более общее понятие для функций.

Определение 7.5.1. Функция — **n-связная**, если для всех $b : B$ тип $\|\mathsf{fib}_f(b)\|_n$ является стягиваемым:

$$\mathsf{conn}_n(f) :\equiv \prod_{b:B} \mathsf{isContr}\left(\|\mathsf{fib}_f(b)\|_n\right)\,.$$

Тип A — **n-связный**, если единственная функция $A \to \mathbf{1}$ является n-связной т.е. если $\|A\|_n$ является стягиваемым.

Таким образом, функция $f : A \to B$ является n-связной тогда и только тогда, когда $\mathsf{fib}_f(b)$ — n-связный для любого $b : B$. Конечно, каждая функция (-2)-связная. На следующем уровне мы имеем:

Лемма 7.5.2. *Функция f является (-1)-связной тогда и только тогда, когда она сюръективна в смысле §4.6.*

Доказательство. Мы определили f сюръективной, если $\|\mathsf{fib}_f(b)\|_{-1}$ обитаем для всех b. Но поскольку это простое высказывание, то обитаемость эквивалентна стягиваемости. $\square$

Таким образом, n-связность функции при $n \geqslant 0$ можно рассматривать как сильную форму сюръективности. С теоретико-категорной точки зрения (-1)-связность соответствует сущности сюръективности на объектах, а n-связность соответствует сущности сюръективности на k-морфизмах при $k \leqslant n+1$.

Из леммы 7.5.2 также следует, что тип A является (-1)-связным тогда и только тогда, когда он просто обитаем. Когда тип — 0-связный, мы можем просто сказать, что он **связный**, а когда он 1-связный, мы говорим, что он **односвязный**.

Замечание 7.5.3. Хотя введенное представление о n-связности для типов согласуется со стандартным понятием в теории гомотопий, наше понятие n-связности для *функций* отличается от одноименного понятия в классической гомотопической теории. В то время как мы говорим, что функция f является n-связной, если все ее слои являются n-связными, некоторые теоретики классической теории гомотопий называют такую функцию $(n+1)$-связной (это обусловлено исторически сложившейся фокусировкой на *кослои*, а не на слои).

Теперь мы приведем некоторые свойства замыканий связных отображений.

Лемма 7.5.4. *Предположим, что g является ретрактом n-связной функции f. Тогда g является n-связной.*

Доказательство. Это является прямым следствием леммы 4.7.3. $\square$

Следствие 7.5.5. *Если g гомотопна n-связной функции f, то g является n-связной.*

Лемма 7.5.6. *Предположим, что $f : A \to B$ — n-связная. Тогда $g : B \to C$ является n-связной тогда и только тогда, когда $g \circ f$ является n-связной.*

Доказательство. Для любого $c : C$ имеем

$$\begin{aligned}
\|\mathsf{fib}_{g\circ f}(c)\|_n &\simeq \Bigg\| \sum_{w:\mathsf{fib}_g(c)} \mathsf{fib}_f(\mathsf{pr}_1 w) \Bigg\|_n && \text{(по упражнению 4.4)}\\
&\simeq \Bigg\| \sum_{w:\mathsf{fib}_g(c)} \|\mathsf{fib}_f(\mathsf{pr}_1 w)\|_n \Bigg\|_n && \text{(по теореме 7.3.9)}\\
&\simeq \|\mathsf{fib}_g(c)\|_n\,. && \text{(поскольку } \|\mathsf{fib}_f(\mathsf{pr}_1 w)\|_n \text{ — стягиваемый)}
\end{aligned}$$

Отсюда следует, что $\|\mathsf{fib}_g(c)\|_n$ является стягиваемым тогда и только тогда, когда $\|\mathsf{fib}_{g\circ f}(c)\|_n$ является стягиваемым. $\square$

Важно отметить, что n-связные функции могут быть эквивалентно охарактеризованы как удовлетворяющие «принципу индукции» по отношению к n-типам. Эта идея приведет непосредственно к нашему доказательству теоремы о надстройке Фрейденталя в §8.6.

Лемма 7.5.7. *Для $f : A \to B$ и $P : B \to \mathcal{U}$ рассмотрим функцию:*

$$\lambda s.\, s \circ f : \left(\prod_{b:B} P(b)\right) \to \left(\prod_{a:A} P(f(a))\right).$$

При фиксированной f и $n \geqslant -2$, следующие высказывания эквивалентны.

(i) f является n-связной.

(ii) Для каждого $P : B \to n\text{-}\mathsf{Type}$ отображение $\lambda s.\, s \circ f$ является эквивалентностью.

(iii) Для каждого $P : B \to n\text{-}\mathsf{Type}$ отображение $\lambda s.\, s \circ f$ имеет сечение.

Доказательство. Предположим, что f является n-связной и пусть $P : B \to n\text{-}\mathsf{Type}$. Тогда имеем эквивалентности

$$\begin{aligned}
\prod_{b:B} P(b) &\simeq \prod_{b:B} \left(\left\|\mathsf{fib}_f(b)\right\|_n \to P(b)\right) && \text{(поскольку } \left\|\mathsf{fib}_f(b)\right\|_n \text{— стягиваемый)} \\
&\simeq \prod_{b:B} \left(\mathsf{fib}_f(b) \to P(b)\right) && \text{(поскольку } P(b) \text{ — } n\text{-тип)} \\
&\simeq \prod_{(b:B)} \prod_{(a:A)} \prod_{(p:f(a)=b)} P(b) && \text{(по свойству левой универсальности } \textstyle\sum \text{ типов)} \\
&\simeq \prod_{a:A} P(f(a)). && \text{(посвойствулевойуниверсальноститиповпутей)}
\end{aligned}$$

Мы опускаем доказательство того, что эта эквивалентность действительно задается выражением $\lambda s.\, s \circ f$. Таким образом, (i) $\Rightarrow$ (ii) и ясно, что (ii) $\Rightarrow$ (iii). Чтобы показать, что (iii) $\Rightarrow$ (i), рассмотрим семейство типов

$$P(b) :\equiv \left\|\mathsf{fib}_f(b)\right\|_n .$$

Тогда (iii) дает отображение $c : \prod_{(b:B)} \left\|\mathsf{fib}_f(b)\right\|_n$ с $c(f(a)) = \left|(a, \mathsf{refl}_{f(a)})\right|_n$. Чтобы показать, что каждый $\left\|\mathsf{fib}_f(b)\right\|_n$ является стягиваемым, построим функцию типа

$$\prod_{b:B} \prod_{(w:\left\|\mathsf{fib}_f(b)\right\|_n)} w = c(b).$$

В силу теоремы 7.3.2 для этого достаточно построить функцию типа

$$\prod_{(b:B)} \prod_{(a:A)} \prod_{(p:f(a)=b)} |(a,p)|_n = c(b).$$

Но, переставляя переменные и индукцию пути, это эквивалентно типу

$$\prod_{a:A} \left|\left(a, \mathsf{refl}_{f(a)}\right)\right|_n = c(f(a)).$$

А это свойство выполняется по нашему выбору $c(f(a))$. $\square$

Следствие 7.5.8. *Для любого A каноническая функция $|\_|_n : A \to \|A\|_n$ является n-связной.*

Доказательство. По теореме 7.3.2 и связанному принципу единственности, выполняется условие леммы 7.5.7. □

Например, когда $n = -1$, следствие 7.5.8 говорит, что отображение $A \to \|A\|$ от типа к его пропозициональному усечению является сюръективным.

Следствие 7.5.9. *Тип A является n-связным тогда и только тогда, когда*

$$\lambda b.\, \lambda a.\, b : B \to (A \to B)$$

является эквивалентностью для любого n-типа B. Другими словами, «каждое отображение от A к n-типу является постоянным».

Доказательство. По лемме 7.5.7 применяется к функции с кообластью $\mathbf{1}$. □

Лемма 7.5.10. *Пусть B — n-тип, $f : A \to B$ — функция. Тогда индуцированная функция $g : \|A\|_n \to B$ является эквивалентностью тогда и только тогда, когда f является n-связной.*

Доказательство. По следствию 7.5.8 $|\_|_n$ является n-связной. Таким образом, поскольку $f = g \circ |\_|_n$, по лемме 7.5.6 f является n-связной тогда и только тогда, когда g является n-связной. Но поскольку g является функцией между n-типами, ее слои также являются n-типами. Таким образом, g является n-связной тогда и только тогда, когда она является эквивалентностью. □

Мы также можем характеризовать связные точечные типы в терминах связности включения их отмеченных точек.

Лемма 7.5.11. *Пусть A — n-тип, а $a_0 : \mathbf{1} \to A$ — отмеченная точка, с $n \geqslant -1$. Тогда A является n-связным тогда и только тогда, когда отображение a_0 является $(n-1)$-связным.*

Доказательство. Сначала, предположим, что $a_0 : \mathbf{1} \to A$ является $(n-1)$-связной, и пусть B — некоторый n-тип; будем опираться на следствие 7.5.9. Отображение $\lambda b.\, \lambda a.\, b : B \to (A \to B)$ имеет стягивание, задаваемое посредством $f \mapsto f(a_0)$, поэтому достаточно показать, что оно также имеет сечение, т.е. что для любой $f : A \to B$ существует $b : B$ такое, что $f = \lambda a.\, b$. Выберем $b :\equiv f(a_0)$. Определим $P : A \to \mathcal{U}$ как $P(a) :\equiv (f(a) = f(a_0))$. Тогда P является семейством $(n-1)$-типов и мы имеем $P(a_0)$; следовательно, имеем $\prod_{(a:A)} P(a)$ поскольку $a_0 : \mathbf{1} \to A$ является $(n-1)$-связной. Таким образом, $f = \lambda a.\, f(a_0)$, что и требовалось.

Теперь предположим, что A является n-связным, и пусть заданы $P : A \to (n-1)\text{-}\mathsf{Type}$ и $u : P(a_0)$. По лемме 7.5.7 достаточно построить $f : \prod_{(a:A)} P(a)$ такую, что $f(a_0) = u$. Теперь $(n-1)\text{-}\mathsf{Type}$ есть n-тип, а A является n-связным, так что по следствию 7.5.9 существует n-тип B такой, что $P = \lambda a.\, B$. Следовательно, имеем семейство эквивалентностей $g : \prod_{(a:A)}(P(a) \simeq B)$. Определим $f(a) : g_a^{-1}(g_{a_0}(u))$; тогда $f : \prod_{(a:A)} P(a)$ и $f(a_0) = u$, что и требовалось. □

В частности, точечный тип (A, a_0) является 0-связным тогда и только тогда, когда $a_0 : \mathbf{1} \to A$ сюръективно, то есть, например, когда $\prod_{(x:A)} \|x = a_0\|$. Аналогичный результат, в необязательно точечном случае, см. в упражнении 7.6.

Полезным вариантом леммы 7.5.6 является:

Лемма 7.5.12. *Пусть $f : A \to B$ — функция, а $P : A \to \mathcal{U}$ и $Q : B \to \mathcal{U}$ — семейства типов. Предположим, что $g : \prod_{(a:A)} P(a) \to Q(f(a))$ — послойно n-связное семейство функций, т.е. каждая функция $g_a : P(a) \to Q(f(a))$ является n-связной. Если также и f является n-связной, то таковой является и функция*

$$\varphi : \left(\sum_{a:A} P(a)\right) \to \left(\sum_{b:B} Q(b)\right)$$
$$\varphi(a,u) :\equiv (f(a), g_a(u)).$$

Обратно, если φ и каждый g_a являются n-связными, и, кроме того, Q послойно просто обитаем (т.е. имеем $\|Q(b)\|$ для каждого $b : B$), то f является n-связной.

Доказательство. Для каждого $b : B$ и $v : Q(b)$ имеем

$$\begin{aligned}
\left\|\mathsf{fib}_\varphi((b,v))\right\|_n &\simeq \left\| \sum_{(a:A)} \sum_{(u:P(a))} \sum_{(p:f(a)=b)} p_*(g_a(u)) = v \right\|_n \\
&\simeq \left\| \sum_{(w:\mathsf{fib}_f(b))} \sum_{(u:P(\mathsf{pr}_1(w)))} g_{\mathsf{pr}_1 w}(u) = \mathsf{pr}_2(w)_*^{-1}(v) \right\|_n \\
&\simeq \left\| \sum_{w:\mathsf{fib}_f(b)} \mathsf{fib}_{g(\mathsf{pr}_1 w)}(\mathsf{pr}_2(w)_*^{-1}(v)) \right\|_n \\
&\simeq \left\| \sum_{w:\mathsf{fib}_f(b)} \left\|\mathsf{fib}_{g(\mathsf{pr}_1 w)}(\mathsf{pr}_2(w)_*^{-1}(v))\right\|_n \right\|_n \\
&\simeq \left\|\mathsf{fib}_f(b)\right\|_n
\end{aligned}$$

где транспортировки вдоль $f(p)$ и $f(p)^{-1}$ относятся к Q. Следовательно, если один из этих двух типов является стягиваемым, то таков и другой.

В частности, если f является n-связной, то $\|\mathsf{fib}_f(b)\|_n$ является стягиваемым для всех $b : B$, и следовательно, таким же является и $\|\mathsf{fib}_\varphi((b,v))\|_n$ для всех $(b,v) : \sum_{(b:B)} Q(b)$. С другой стороны, если φ является n-связной, то $\|\mathsf{fib}_\varphi((b,v))\|_n$ является стягиваемым для всех (b,v), следовательно, таким же является и $\|\mathsf{fib}_f(b)\|_n$ для всех $b : B$, так что существует некоторое $v : Q(b)$. Наконец, поскольку стягиваемость является простым высказыванием, достаточно просто иметь такое v. □

Обращение леммы 7.5.12 может потерпеть неудачу, если Q не является послойно просто обитаемым. Например, если P и Q являются постоянными при $\mathbf{0}$, то φ и каждое g_a являются эквивалентностями, а f может быть произвольным.

Обращение леммы имеет следующую формулировку:

Лемма 7.5.13. *Пусть $P, Q : A \to \mathcal{U}$ — семейства типов и имеется послойное преобразование*

$$f : \prod_{a:A} (P(a) \to Q(a))$$

от P к Q. Тогда индуцированное отображение $\mathsf{total}(f) : \sum_{(a:A)} P(a) \to \sum_{(a:A)} Q(a)$ является n-связным, тогда и только тогда, когда каждая $f(a)$ является n-связной.

Конечно, направление «только тогда» также является частным случаем леммы 7.5.12.

Доказательство. По теореме 4.7.6 имеем $\mathsf{fib}_{\mathsf{total}(f)}((x,v)) \simeq \mathsf{fib}_{f(x)}(v)$ для любых $x : A$ и $v : Q(x)$. Следовательно, $\left\|\mathsf{fib}_{\mathsf{total}(f)}((x,v))\right\|_n$ является стягиваемым, тогда и только тогда, когда $\left\|\mathsf{fib}_{f(x)}(v)\right\|_n$ является стягиваемым. □

Другим полезным фактом о связных отображениях является то, что они индуцируют эквивалентность на n-усечениях:

Лемма 7.5.14. *Если $f : A \to B$ является n-связным, то оно индуцирует эквивалентность $\|A\|_n \simeq \|B\|_n$.*

Доказательство. Пусть c — доказательство n-связности. Слева-направо используем отображение $\|f\|_n : \|A\|_n \to \|B\|_n$. Чтобы определить отображение справа-налево, с помощью универсального свойства усечений, достаточно предоставить отображение $\mathsf{back} : B \to \|A\|_n$. Мы можем определить это отображение следующим образом:

$$\mathsf{back}(y) :\equiv \|\mathsf{pr}_1\|_n \left(\mathsf{pr}_1(c(y))\right).$$

По определению, $c(y)$ имеет тип $\mathsf{isContr}(\|\mathsf{fib}_f(y)\|_n)$, так что его первая компонента имеет тип $\|\mathsf{fib}_f(y)\|_n$, и мы можем получить элемент из $\|A\|_n$ проекцией.

Затем мы покажем, что компоновки являются тождественными. В обоих направлениях, поскольку целью является путь в n-усеченном типе, достаточно рассмотреть случай конструктора $\|\_\|_n$.

В одном направлении мы должны показать, что для всех $x : A$,

$$\|\mathsf{pr}_1\|_n \left(\mathsf{pr}_1(c(f(x)))\right) = |x|_n\ .$$

Но $\left|(x, \mathsf{refl}_{f(x)})\right|_n : \|\mathsf{fib}_f(f(x))\|_n$, а $c(f(x))$ указывает, что этот тип является стягиваемым, так что

$$\mathsf{pr}_1(c(f(x))) = |(x, \mathsf{refl})|_n\ .$$

Применяя $\|\mathsf{pr}_1\|_n$ к обеим частям этого уравнения, получаем требуемый результат.

В другом направлении мы должны показать, что для всех $y : B$,

$$\|f\|_n \left(\|\mathsf{pr}_1\|_n \left(\mathsf{pr}_1(c(y))\right)\right) = |y|_n\ .$$

$\mathsf{pr}_1(c(y))$ имеет тип $\|\mathsf{fib}_f(y)\|_n$, и путь, который нам нужен, по существу, является второй составляющей от $\mathsf{fib}_f(y)$, но мы должны убедиться, что усечения работают.

В общем случае предположим, что нам дано $p : \left\|\sum_{(x:A)} B(x)\right\|_n$ и мы хотим доказать $P(\|\mathsf{pr}_1\|_n (p))$. Индукцией усечения достаточно доказать $P(|a|_n)$ для всех $a : A$ и $b : B(a)$. Применяя этот принцип в данном случае, достаточно доказать

$$\|f\|_n \left(|a|_n\right) = |y|_n$$

для $a : A$ и $b : f(a) = y$. Но левая часть равна $|f(a)|_n$, так что применение $|\_|_n$ к обоим частям b дает требуемый результат. □

Можно догадаться, что этот факт характеризует n-связные отображения, но на самом деле n-связность немного сильнее этого. Например, включение $0_2 : \mathbf{1} \to \mathbf{2}$ индуцирует эквивалентность на (-1)-усечениях, но не является сюръективным (т.е. (-1)-связным). В §8.4 мы увидим, что разница в общем случае является аналогичной небольшому добавлению к сюръективности.

7.6 Ортогональная факторизация

В теории множеств сюръекции и инъекции образуют уникальную систему факторизации: каждая функция существенно факторизуется как сюръекция с последующей инъекцией. Мы видели, что сюръекции естественным образом обобщаются на n-связные отображения, поэтому естественно выяснить, участвуют ли они и в системе факторизации. Имеется соответствующее обобщение инъекций.

Определение 7.6.1. Функция $f : A \to B$ является n**-усеченной**, если слой $\mathsf{fib}_f(b)$ является n-типом для всех $b : B$.

В частности, f является (-2)-усеченной тогда и только тогда, когда она является эквивалентностью. И, конечно, A является n-типом тогда и только тогда, когда $A \to \mathbf{1}$ n-усечен. Более того, n-усеченные отображения могут быть эквивалентно определены рекурсивно, как и n-типы.

Лемма 7.6.2. *Для любого $n \geqslant -2$ функция $f : A \to B$ является $(n+1)$-усеченной тогда и только тогда, когда для всех $x, y : A$ отображение $\mathsf{ap}_f : (x = y) \to (f(x) = f(y))$ является n-усеченным. В частности, f является (-1)-усеченной тогда и только тогда, когда она является вложением в смысле §4.6.*

Доказательство. Заметим, что для любых $(x, p), (y, q) : \mathsf{fib}_f(b)$ имеет место

$$\begin{aligned}((x,p) = (y,q)) &= \sum_{r:x=y} (p = \mathsf{ap}_f(r) \cdot q) \\ &= \sum_{r:x=y} (\mathsf{ap}_f(r) = p \cdot q^{-1}) \\ &= \mathsf{fib}_{\mathsf{ap}_f}(p \cdot q^{-1})\,.\end{aligned}$$

Таким образом, любое пространство путей в любом слое от f является слоем от ap_f. С другой стороны, выбирая $b :\equiv f(y)$ и $q :\equiv \mathsf{refl}_{f(y)}$ мы получаем, что любой слой от ap_f является пространством путей в слое от f. Результат следует из того, что f является $(n+1)$-усеченным, если все пространства пути его слоев являются n-типами. □

Теперь мы можем построить факторизацию довольно очевидным образом.

Определение 7.6.3. Пусть $f : A \to B$ — функция. n**-образ** f определим как

$$\mathsf{im}_n(f) :\equiv \sum_{b:B} \left\| \mathsf{fib}_f(b) \right\|_n\,.$$

Если $n = -1$, то будем записывать просто как $\mathsf{im}(f)$ и называть **образом** f.

Лемма 7.6.4. *Для любой функции $f : A \to B$ каноническая функция $\tilde{f} : A \to \mathsf{im}_n(f)$ является n-связной. Следовательно, любая функция факторизуется как n-связная функция, за которой следует n-усеченная функция.*

Доказательство. Отметим, что $A \simeq \sum_{(b:B)} \mathsf{fib}_f(b)$. Функция $\tilde{f}$ является функцией на пространствах расслоений, индуцированной каноническим расслоенным преобразованием

$$\prod_{b:B} \left(\mathsf{fib}_f(b) \to \left\| \mathsf{fib}_f(b) \right\|_n\right)\,.$$

Поскольку каждое отображение $\mathsf{fib}_f(b) \to \|\mathsf{fib}_f(b)\|_n$ является n-связным, по следствию 7.5.8, $\tilde{f}$ является n-связной по лемме 7.5.13. Наконец, проекция $\mathsf{pr}_1 : \mathsf{im}_n(f) \to B$ является n-усеченной, так как ее слои эквивалентны n-усечениям слоев f. □

В следующей лемме мы представляем некоторую технику, используемую при доказательстве уникальной теоремы факторизации.

Лемма 7.6.5. *Пусть имеется коммутативная диаграмма функций*

$$\begin{tikzcd} A \arrow[r, "g_1"] \arrow[d, "g_2"'] & X_1 \arrow[d, "h_1"] \\ X_2 \arrow[r, "h_2"'] & B \end{tikzcd}$$

с $H : h_1 \circ g_1 \sim h_2 \circ g_2$, где g_1 и g_2 являются n-связными, а h_1 и h_2 являются n-усеченными. Тогда существует эквивалентность

$$E(H, b) : \mathsf{fib}_{h_1}(b) \simeq \mathsf{fib}_{h_2}(b)$$

для всех $b : B$, такая, что для всех $a : A$ имеется идентификация

$$\overline{E}(H, a) : E(H, h_1(g_1(a)))(g_1(a), \mathsf{refl}_{h_1(g_1(a))}) = (g_2(a), H(a)^{-1}) .$$

Доказательство. Пусть $b : B$. Тогда имеем следующие эквивалентности:

$$\begin{aligned} \mathsf{fib}_{h_1}(b) &\simeq \sum_{w:\mathsf{fib}_{h_1}(b)} \|\mathsf{fib}_{g_1}(\mathsf{pr}_1 w)\|_n && \text{(поскольку } g_1 \text{ является } n\text{-связной)} \\ &\simeq \Big\| \sum_{w:\mathsf{fib}_{h_1}(b)} \mathsf{fib}_{g_1}(\mathsf{pr}_1 w) \Big\|_n && \text{(по следствию 7.3.10, поскольку } h_1 \text{ — } n\text{-усеченная)} \\ &\simeq \|\mathsf{fib}_{h_1 \circ g_1}(b)\|_n && \text{(по упражнению 4.4)} \end{aligned}$$

и, подобно этому, для h_2 и g_2. Также, поскольку имеется гомотопия $H : h_1 \circ g_1 \sim h_2 \circ g_2$, существует очевидная эквивалентность $\mathsf{fib}_{h_1 \circ g_1}(b) \simeq \mathsf{fib}_{h_2 \circ g_2}(b)$. Отсюда получаем

$$\mathsf{fib}_{h_1}(b) \simeq \mathsf{fib}_{h_2}(b)$$

для всех $b : B$. Анализируя основные функции, мы получаем следующее представление о том, что происходит с элементом $(g_1(a), \mathsf{refl}_{h_1(g_1(a))})$ после применения каждой из эквивалентностей, из которых состоит E. Некоторые из идентификаций являются определениями, но другие (помеченные знаком $=$) являются только пропозициональными; собирая их вместе, получим $\overline{E}(H, a)$.

$$\begin{aligned} (g_1(a), \mathsf{refl}_{h_1(g_1(a))}) &\overset{=}{\longmapsto} \big((g_1(a), \mathsf{refl}_{h_1(g_1(a))}), \big|(a, \mathsf{refl}_{g_1(a)})\big|_n\big) \\ &\longmapsto \big|\big((g_1(a), \mathsf{refl}_{h_1(g_1(a))}), (a, \mathsf{refl}_{g_1(a)})\big)\big|_n \\ &\longmapsto \big|(a, \mathsf{refl}_{h_1(g_1(a))})\big|_n \\ &\overset{=}{\longmapsto} \big|(a, H(a)^{-1})\big|_n \\ &\longmapsto \big|\big((g_2(a), H(a)^{-1}), (a, \mathsf{refl}_{g_2(a)})\big)\big|_n \\ &\longmapsto \big((g_2(a), H(a)^{-1}), \big|(a, \mathsf{refl}_{g_2(a)})\big|_n\big) \\ &\longmapsto (g_2(a), H(a)^{-1}) \end{aligned}$$

Первое равенство выражает, что для общего b отображение $\mathsf{fib}_{h_1}(b) \to \sum_{w:\mathsf{fib}_{h_1}(b)} \|\mathsf{fib}_{g_1}(\mathsf{pr}_1 w)\|_n$ вставляет центр стягивания для $\|\mathsf{fib}_{g_1}(\mathsf{pr}_1 w)\|_n$, заданного предположением, что g_1 является n-усеченной; тогда, как в рассматриваемом случае, у этого типа есть очевидный обитатель $\left|(a, \mathsf{refl}_{g_1(a)})\right|_n$, который, по стягиваемости, должен быть равен центру. Второе пропозициональное равенство выражает то, что эквивалентность $\mathsf{fib}_{h_1 \circ g_1}(b) \simeq \mathsf{fib}_{h_2 \circ g_2}(b)$ объединяет второй компонент с $H(a)^{-1}$, и мы имеем $H(a)^{-1} \cdot \mathsf{refl} = H(a)^{-1}$. Читатель может проверить, что другие равенства являются дефинициальными (при условии разумного решения упражнения 4.4). □

Объединяя леммы 7.6.4 и 7.6.5, имеем следующий уникальный результат факторизации:

Теорема 7.6.6. *Для любой $f : A \to B$ пространство $\mathsf{fact}_n(f)$, определенное как*

$$\sum_{(X:\mathcal{U})} \sum_{(g:A\to X)} \sum_{(h:X\to B)} (h \circ g \sim f) \times \mathsf{conn}_n(g) \times \mathsf{trunc}_n(h)\,,$$

является стягиваемым. Его центр стягивания является элемент

$$(\mathsf{im}_n(f), \tilde{f}, \mathsf{pr}_1, \theta, \varphi, \psi) : \mathsf{fact}_n(f)\,,$$

получающийся из леммы 7.6.4, где $\theta : \mathsf{pr}_1 \circ \tilde{f} \sim f$ является канонической гомотопией, φ есть доказательство леммы 7.6.4, а ψ — очевидное доказательство того, что $\mathsf{pr}_1 : \mathsf{im}_n(f) \to B$ имеет n-усеченные слои.

Доказательство. Из леммы 7.6.4 известно, что существует элемент из $\mathsf{fact}_n(f)$, поэтому достаточно показать, что $\mathsf{fact}_n(f)$ является простым высказыванием. Предположим, что имеются две n-факторизации

$$(X_1, g_1, h_1, H_1, \varphi_1, \psi_1) \quad \text{и} \quad (X_2, g_2, h_2, H_2, \varphi_2, \psi_2)$$

для f. Тогда мы имеем поточечно-конкатенированную гомотопию

$$H :\simeq (\lambda a.\, H_1(a) \cdot H_2^{-1}(a)) : (h_1 \circ g_1 \sim h_2 \circ g_2)\,.$$

С помощью унивалентности, характеризации путей и транспортировки в $\sum$-типах, функциональных типах и типах путей, достаточно показать, что

(i) существует эквивалентность $e : X_1 \simeq X_2$,

(ii) существует гомотопия $\varsigma : e \circ g_1 \sim g_2$,

(iii) существует гомотопия $\eta : h_2 \circ e \sim h_1$,

(iv) для всех $a : A$ имеем $\mathsf{ap}_{h_2}(\varsigma(a))^{-1} \cdot \eta(g_1(a)) \cdot H_1(a) = H_2 a)$.

Мы докажем эти четыре утверждения в таком же порядке.

(i) По лемме 7.6.5 имеем расслоенную эквивалентность

$$E(H) : \prod_{b:B} \mathsf{fib}_{h_1}(b) \simeq \mathsf{fib}_{h_2}(b)\,.$$

Это индуцирует эквивалентность пространств расслоений, т.е. имеем

$$\left(\sum_{b:B} \mathsf{fib}_{h_1}(b)\right) \simeq \left(\sum_{b:B} \mathsf{fib}_{h_2}(b)\right).$$

Конечно, также имеются эквивалентности $X_1 \simeq \sum_{(b:B)} \mathsf{fib}_{h_1}(b)$ и $X_2 \simeq \sum_{(b:B)} \mathsf{fib}_{h_2}(b)$, по лемме 4.8.2. Все это приводит к эквивалентности $e : X_1 \simeq X_2$; читатель может проверить, что основная функция e задается формулой

$$e(x) \equiv \mathsf{pr}_1(E(H, h_1(x))(x, \mathsf{refl}_{h_1(x)})).$$

(ii) По лемме 7.6.5 можно выбрать

$$\varsigma(a) :\equiv \mathsf{ap}_{\mathsf{pr}_1}(\overline{E}(H, a)) : e(g_1(a)) = g_2(a).$$

(iii) Для всех $x : X_1$ имеем

$$\mathsf{pr}_2(E(H, h_1(x))(x, \mathsf{refl}_{h_1(x)})) : h_2(e(x)) = h_1(x),$$

приводящую к гомотопии $\eta : h_2 \circ e \sim h_1$.

(iv) По характеризации траекторий в слоях (лемма 4.2.5), путь $\overline{E}(H, a)$ из леммы 7.6.5 дает нам $\eta(g_1(a)) = \mathsf{ap}_{h_2}(\varsigma(a)) \cdot H(a)^{-1}$. Требуемое равенство следует из подстановки определения H и перестановки путей.

$\square$

Стандартные рассуждения приводят к следующему принципу ортогональности.

Теорема 7.6.7. *Пусть $e : A \to B$ является n-связной, а $m : C \to D$ — n-усеченной. Тогда отображение*

$$\varphi : (B \to C) \to \sum_{(h:A\to C)} \sum_{(k:B\to D)} (m \circ h \sim k \circ e)$$

является эквивалентностью.

Набросок доказательства. Для всех (h, k, H) в кообласти, пусть $h = h_2 \circ h_1$ и $k = k_2 \circ k_1$, где h_1 и k_1 являются n-связными, а h_2 и k_2 — n-усеченными. Тогда $f = (m \circ h_2) \circ h_1$ и $f = k_2 \circ (k_1 \circ e)$ являются n-факторизациями для $m \circ h = k \circ e$. Таким образом, между ними существует уникальная эквивалентность. Не трудно (хотя немного утомительно) извлечь из этого то, что $\mathsf{fib}_\varphi((h, k, H))$ является стягиваемым. $\square$

В завершение покажем, что образы сохраняются под обратным образом.

Лемма 7.6.8. *Предположим, что квадрат*

$$\begin{tikzcd} A \arrow[r] \arrow[d, "f"'] & C \arrow[d, "g"] \\ B \arrow[r, "h"'] & D \end{tikzcd}$$

является квадратом обратного образа, и пусть $b : B$. Тогда $\mathsf{fib}_f(b) \simeq \mathsf{fib}_g(h(b))$.

Доказательство. Это следует из склеивания обратных образов (упражнение 2.12), поскольку тип X на диаграмме

$$\begin{tikzcd} X \arrow[r] \arrow[d] & A \arrow[r] \arrow[d, "f"'] & C \arrow[d, "g"] \\ \mathbf{1} \arrow[r, "b"'] & B \arrow[r, "h"'] & D \end{tikzcd}$$

является обратным образом левого квадрата тогда и только тогда, когда он является обратным образом внешнего прямоугольника, в то время как $\mathsf{fib}_f(b)$ является обратным образом квадрата слева и $\mathsf{fib}_g(h(b))$ является обратным образом внешнего прямоугольника. $\square$

Теорема 7.6.9. *Рассмотрим функции $f : A \to B$, $g : C \to D$ и диаграмму*

$$\begin{tikzcd} A \arrow[rr] \arrow[d, "\tilde{f}_n"'] & & C \arrow[d, "\tilde{g}_n"] \\ \mathsf{im}_n(f) \arrow[rr] \arrow[d, "\mathsf{pr}_1"'] & & \mathsf{im}_n(g) \arrow[d, "\mathsf{pr}_1"] \\ B \arrow[rr, "h"'] & & D \end{tikzcd}$$

Если внешний прямоугольник является обратным образом, то тким же является и нижний квадрат (а значит, и верхний квадрат, — упражнение 2.12). Следовательно, образы сохраняются под обратными образами.

Доказательство. Предполагая, что внешний прямоугольник является обратным образом, имеем эквивалентности

$$\begin{aligned} B \times_D \mathsf{im}_n(g) &\equiv \sum_{(b:B)} \sum_{(w:\mathsf{im}_n(g))} h(b) = \mathsf{pr}_1 w \\ &\simeq \sum_{(b:B)} \sum_{(d:D)} \sum_{(w:\|\mathsf{fib}_g(d)\|_n)} h(b) = d \\ &\simeq \sum_{(b:B)} \|\mathsf{fib}_g(h(d))\|_n \\ &\simeq \sum_{(b:B)} \|\mathsf{fib}_f(b)\|_n && \text{(по лемме 7.6.8)} \\ &\equiv \mathsf{im}_n(f)\,. \end{aligned}$$

$\square$

7.7 Модальности

Почти вся теория n-типов и связности может быть построена в гораздо большей общности (этот раздел не будет использоваться в оставшейся части книги).

Наша первая мысль об обобщении теории n-типов может заключаться в том, чтобы использовать лемму 7.3.3 в качестве определения.

Определение 7.7.1. Рефлективный подуниверсум — это предикат $P : \mathcal{U} \to \mathsf{Prop}$ такой, что для любого $A : \mathcal{U}$ имеется тип $\bigcirc A$ такой, что $P(\bigcirc A)$, и отображение $\eta_A : A \to \bigcirc A$, с тем свойством, что, для каждого $B : \mathcal{U}$ с $P(B)$, следующее отображение является эквивалентностью:

$$\begin{cases} (\bigcirc A \to B) \longrightarrow (A \to B) \\ \qquad\qquad f \longmapsto f \circ \eta_A \end{cases}$$

Будем записывать $\mathcal{U}_P :\equiv \{A : U \mid P(A)\}$, так что $A : \mathcal{U}_P$ будет означать, что $A : \mathcal{U}$ и имеет место $P(A)$. Также будем обозначать через $\mathsf{rec}_{\bigcirc}$ квазиобратное к введенному отображению. Обозначение $\bigcirc$ может показаться немного странным, но смысл этого скоро будет понятен.

Для любого рефлективного подуниверсума можно привычным образом доказать все знакомые факты о рефлективных подкатегориях из теории категорий. Например, имеем:

- Тип A находится в $\mathcal{U}_P$ тогда и только тогда, когда $\eta_A : A \to \bigcirc A$ является эквивалентностью.

- $\mathcal{U}_P$ является замкнутым под ретрактами. В частности, A лежит в $\mathcal{U}_P$, если η_A допускает стягивание.

- Операция $\bigcirc$ является функтором в подходящем, с точностью до гомотопической когерентности, смысле, который мы можем сделать точным на высших уровнях, при необходимости.

- Типы в $\mathcal{U}_P$ замкнуты под всеми пределами, такими как произведения и обратные образы. В частности, для всех $A : \mathcal{U}_P$ и $x, y : A$ тип тождественности $(x =_A y)$ также находится в $\mathcal{U}_P$, поскольку он является обратным образом двух функций $\mathbf{1} \to A$.

- Копределы в $\mathcal{U}_P$ могут быть построены путем применения $\bigcirc$ к обычным копределам типов.

Важно отметить, что замкнутость под произведениями распространяется также на «бесконечные произведения», то есть на зависимые функциональные типы.

Теорема 7.7.2. *Если $B : A \to \mathcal{U}_P$ — любое семейство типов в рефлективном подуниверсуме $\mathcal{U}_P$, то $\prod_{(x:A)} B(x)$ также содержится в $\mathcal{U}_P$.*

Доказательство. Для любого $x : A$ рассмотрим функцию $\mathsf{ev}_x : (\prod_{(x:A)} B(x)) \to B(x)$, определенную как $\mathsf{ev}_x(f) :\equiv f(x)$. Поскольку $B(x)$ находится в P, она расширяется до функции

$$\mathsf{rec}_{\bigcirc}(\mathsf{ev}_x) : \bigcirc\left(\prod_{x:A} B(x)\right) \to B(x)\,.$$

Так что, можно определить $h : \bigcirc(\prod_{(x:A)} B(x)) \to \prod_{(x:A)} B(x)$ как $h(z)(x) :\equiv \mathsf{rec}_{\bigcirc}(\mathsf{ev}_x)(z)$. Тогда h является стягиванием для $\eta_{\prod_{(x:A)} B(x)}$, таким образом, $\prod_{(x:A)} B(x)$ содержится в $\mathcal{U}_P$. □

В частности, если $B : \mathcal{U}_P$ и A — произвольный тип, то $(A \to B)$ содержится в $\mathcal{U}_P$. На категорном языке это означает, что любой рефлективный подуниверсум является **экспоненциальным идеалом**. Это, в свою очередь, подразумевает, с помощью стандартной аргументации, что рефлектор сохраняет конечные произведения.

Следствие 7.7.3. *Для любых типов A и B и любого рефлективного подуниверсума, индуцированное отображение $\bigcirc(A \times B) \to \bigcirc(A) \times \bigcirc(B)$ является эквивалентностью.*

Доказательство. Достаточно показать, что $\bigcirc(A)\times\bigcirc(B)$ обладает тем же универсальным свойством, что и $\bigcirc(A\times B)$. Таким образом, пусть $C:\mathcal{U}_P$; имеем

$$\begin{aligned}(\bigcirc(A)\times\bigcirc(B)\to C) &= (\bigcirc(A)\to(\bigcirc(B)\to C))\\ &= (\bigcirc(A)\to(B\to C))\\ &= (A\to(B\to C))\\ &= (A\times B\to C)\end{aligned}$$

используя универсальные свойства для $\bigcirc(A)$ и $\bigcirc(B)$, а также то, что $B\to C$ находится в $\mathcal{U}_P$, так как это верно для C. Нетрудно проверить, что эта эквивалентность задается композицией с $\eta_A\times\eta_B$, что и требовалось. □

Может показаться странным, что каждая рефлективная подкатегория типов автоматически является экспоненциальным идеалом с рефлектором, сохраняющим произведение. Однако это действительно так, классически, в категории *множеств*, по тем же причинам. Просто этот факт обычно не упоминается, поскольку классическая категория множеств — в отличие от категории гомотопических типов — не содержит многих интересных отражающих подкатегорий.

Два основных свойства n-типов *не* разделяются общими рефлективными подуниверсумами: теорема 7.1.8 (замыкание в $\sum$-типах) и теорема 7.3.2 (индукция усечения). Однако аналоги этих двух свойств эквивалентны друг другу.

Теорема 7.7.4. *Для рефлективного подуниверсума $\mathcal{U}_P$ следующие свойства логически эквивалентны.*

(i) Если $A:\mathcal{U}_P$ и $B:A\to\mathcal{U}_P$, то $\sum_{(x:A)}B(x)$ содержится в $\mathcal{U}_P$.

(ii) Для любых $A:\mathcal{U}$, семейства типов $B:\bigcirc A\to\mathcal{U}_P$ и отображения $g:\sum_{(a:A)}B(\eta(a))$ существует $f:\prod_{(z:\bigcirc A)}B(z)$ такая, что $f(\eta(a))=g(a)$ для всех $a:A$.

Доказательство. Предположим, что верно (i). Тогда в ситуации (ii) тип $\prod_{(z:\bigcirc A)}B(z)$ содержится в $\mathcal{U}_P$ и мы имеем $g':A\to\prod_{(z:\bigcirc A)}B(z)$, определенную как $g'(a):\equiv(\eta(a),g(a))$. Таким образом, имеем $\mathsf{rec}_\bigcirc(g'):\bigcirc A\to\prod_{(z:\bigcirc A)}B(z)$, так что $\mathsf{rec}_\bigcirc(g')(\eta(a)):\equiv(\eta(a),g(a))$.

Теперь рассмотрим функции $\mathsf{pr}_1\circ\mathsf{rec}_\bigcirc(g'):\bigcirc A\to\bigcirc A$ и $\mathsf{id}_{\bigcirc A}$. По условию, они становятся равными, когда предварительно компонуются с η. Таким образом, по универсальному свойству $\bigcirc$ они равны уже $p_z:\mathsf{pr}_1(\mathsf{rec}_\bigcirc(g')(z))=z$ для всех z. Теперь можно определить $f(z):\equiv p_{z*}:\mathsf{pr}_2(\mathsf{rec}_\bigcirc(g')(z))$. Используя свойство сопряжения эквивалентности из определения 7.7.1, можно показать, что первая компонента $\mathsf{rec}_\bigcirc(g')(\eta(a))=(\eta(a),g(a))$ равна $p_{\eta(a)}$. Таким образом, его вторая компонента дает $f(\eta(a))=g(a)$, что и требовалось.

Обратно, предположим, что верно (ii), и что $A:\mathcal{U}_P$ и $B:A\to\mathcal{U}_P$. Пусть h будет компоновкой

$$\bigcirc\left(\sum_{x:A}B(x)\right)\xrightarrow{\bigcirc(\mathsf{pr}_1)}\bigcirc A\xrightarrow{(\eta_A)^{-1}}A\,.$$

Тогда для $z:\sum_{(x:A)}B(x)$ имеем

$$\begin{aligned}h(\eta(z)) &= \eta^{-1}(\bigcirc(\mathsf{pr}_1)(\eta(z)))\\ &= \eta^{-1}(\eta(\mathsf{pr}_1(z)))\\ &= \mathsf{pr}_1(z)\,.\end{aligned}$$

Обозначим этот путь через p_z. Теперь, если определить $C : \bigcirc(\sum_{(x:A)} B(x)) \to \mathcal{U}$ как $C(w) :\equiv B(h(w))$, то получим

$$g :\equiv \lambda z.\, p_{z*}(\mathsf{pr}_2(z)) : \prod_{z:\sum_{(x:A)} B(x)} C(\eta(z))\,.$$

Таким образом, предположение дает $f : \prod_{(w:\bigcirc(\sum_{(x:A)} B(x)))} C(w)$ такую, что $f(\eta(z)) = g(z)$. Совместно, h и f дают функцию $k : \bigcirc(\sum_{(x:A)} B(x)) \to \sum_{(x:A)} B(x)$, определенную как $k(w) :\equiv (h(w), f(w))$, тогда как p_z и равенство $f(\eta(z)) = g(z)$ показывают, что k является стягиванием для $\eta_{\sum_{(x:A)} B(x)}$. Следовательно, $\sum_{(x:A)} B(x)$ содержится в $\mathcal{U}_P$. □

Обратите внимание на сходство с обсуждением в §5.5. Универсальное свойство рефлектора рефлексивного подуниверсума подобно принципу рекурсии с его свойством уникальности, а теорема 7.7.4 (ii) подобна соответствующему принципу индукции. В отличие от §5.5, они не эквивалентны здесь, из-за ограничения, которое мы можем устранить только в типах, которые содержатся в $\mathcal{U}_P$. Условие (i) теоремы 7.7.4 фиксирует это разобщение.

Неудивительно, конечно, что если имеется принцип индукции, то можно получить принцип рекурсии. Также можно получить свойство уникальности, если позволить себе исключать типы путей. Это предполагает следующее определение. Заметим, что любой рефлективный подуниверсум может характеризоваться операцией $\bigcirc : \mathcal{U} \to \mathcal{U}$ и функций $\eta_A : A \to \bigcirc A$, так как имеется $P(A) = \mathsf{isequiv}(\eta_A)$.

Определение 7.7.5. Модальность — это операция $\bigcirc : \mathcal{U} \to \mathcal{U}$, для которой существуют

(i) функции $\eta_A^{\bigcirc} : A \to \bigcirc A$ для любого типа A,

(ii) для любого $A : \mathcal{U}$ и любого семейства типов $B : \bigcirc(A) \to \mathcal{U}$, функция

$$\mathsf{ind}_{\bigcirc} : \left(\prod_{a:A} \bigcirc(B(\eta_A^{\bigcirc}(a)))\right) \to \prod_{z:\bigcirc(A)} \bigcirc(B(z))\,,$$

(iii) путь $\mathsf{ind}_{\bigcirc}(f)(\eta_A^{\bigcirc}(a)) = f(a)$ для каждой $f : \prod_{(a:A)} \bigcirc(B(\eta_A^{\bigcirc}(a)))$,

(iv) для любых $z, z' : \bigcirc(A)$ функция $\eta_{z=z'}^{\bigcirc} : (z = z') \to \bigcirc(z = z')$, являющаяся эквивалентностью.

Скажем, что A является **модальным** для $\bigcirc$, если $\eta_A^{\bigcirc} : A \to \bigcirc A$ является эквивалентносью, и для типа модальных типов записываем

$$\mathcal{U}_{\bigcirc} :\equiv \{X : \mathcal{U} \mid X \text{ является } \bigcirc\text{-модальным}\}\,. \tag{7.7.6}$$

Условия (ii) и (iii) очень похожи на условие теоремы 7.7.4(ii), но формулируются с использованием $\bigcirc B(z)$, а не предполагают, что B оценивается в $\mathcal{U}_P$. Это позволяет сформулировать условие только с точки зрения операции $\bigcirc$, вместо того, чтобы требовать, чтобы предикат $P : \mathcal{U} \to \mathsf{Prop}$ был задан заранее (это не совсем удовлетворительно, так как мы все еще должны относиться к P не так тонко в пункте (iv); мы не знаем, следует ли (iv) из (i)-(iii)). Однако, (ii) следует из определения 7.7.5(ii) и (iii), так как для любого $C : \bigcirc A \to \mathcal{U}_{\bigcirc}$ имеем $C(z) \simeq \bigcirc C(z)$, и мы можем выйти за пределы этой эквивалентности.

Как и в случаях с другими принципами индукции, это подразумевает наличие универсального свойства.

Теорема 7.7.7. *Пусть A — тип и $B : \bigcirc(A) \to \mathcal{U}_\bigcirc$. Тогда функция*

$$\left(\_ \circ \eta_A^\bigcirc\right) : \left(\prod_{z:\bigcirc(A)} B(z)\right) \to \left(\prod_{a:A} B(\eta_A^\bigcirc(a))\right)$$

является эквивалентностью.

Доказательство. По определению, операция $\mathsf{ind}_\bigcirc$ является правой обратной к $(\_ \circ \eta_A^\bigcirc)$. Таким образом, нужно всего лишь найти гомотопию

$$\prod_{z:\bigcirc(A)} s(z) = \mathsf{ind}_\bigcirc(s \circ \eta_A^\bigcirc)(z)$$

для каждого $s : \prod_{(z:\bigcirc(A))} B(z)$, а также продемонстрировать ее как левую обратную. По предположению, каждый $B(z)$ является модальным, и поэтому каждый тип $s(z) = R(s \circ \eta_A^\bigcirc)(z)$ также является модальным. Таким образом, достаточно найти функцию типа

$$\prod_{a:A} s(\eta_A^\bigcirc(a)) = \mathsf{ind}_\bigcirc(s \circ \eta_A^\bigcirc)(\eta_A^\bigcirc(a))\,,$$

которая следует из определения 7.7.5(iii). □

В частности, для каждого типа A и любого модального типа B имеем эквивалентность $(\bigcirc A \to B) \simeq (A \to B)$.

Следствие 7.7.8. *Для любой модальности $\bigcirc$, $\bigcirc$-модальные типы образуют рефлективный подуниверсум, удовлетворяющий эквивалентным условиям теоремы* 7.7.4.

Таким образом, модальности можно отождествить с рефлективными подуниверсумами, замкнутыми под $\sum$-типами. Разумеется, название *модальность* пришло из *модальной логики*, в которой можно формировать такие утверждения, как «возможно A» (обычно записываемое как $\Diamond A$) или «обязательно A» (записываемое как $\Box A$). Символ $\bigcirc$ иногда используется для указания произвольного модального оператора. В соответствии с принципом высказывания-как-типы модальность в смысле модальной логики соответствует операции на типах, и определение 7.7.5 представляется разумным кандидатом на то, как должна быть определена такая операция (точнее, мы могли бы назвать эти идемпотенты монадическими модальностями, см. Примечания). Как упоминалось в §3.10, мы можем в основном использовать наречия, чтобы неформально говорить о модальностях, таких как «просто», для пропозиционального усечения, и «чисто», для тождественной модальности (т.е. той, которая определена как $\bigcirc A :\equiv A$).

Для любой модальности $\bigcirc$ определим отображение $f : A \to B$ как **$\bigcirc$-связное**, если $\bigcirc(\mathsf{fib}_f(b))$ является стягиваемым для всех $b : B$, и как **$\bigcirc$-усеченное**, если $\mathsf{fib}_f(b)$ является модальным для всех $b : B$. Вся теория §§ 7.5 и 7.6, которая не затрагивает связи n-типов при разных значениях n, применяется в этой общности дословно. В частности, имеем ортогональную систему факторизации.

Важным классом модальностей, который *не* включает n-усечения, являются *левые точные* модальности: те, для которых функтор $\bigcirc$ сохраняет обратные образы, а также конечные произведения. Это категорификация «топологий Лоувера-Тирней (Lawvere-Tierney)» в теории элементарных топосов, соответствующая, в семантике высших категорий, под-$(\infty, 1)$-топосам. Однако, обсуждение этих вопросов выходит за рамки данной книги.

В упражнениях можно найти некоторые конкретные примеры, отличные от n-усечений.

Примечания

Понятие гомотопического n-типа в классической гомотопической теории известно довольно давно. Но, именно Воеводский понял, что, отталкиваясь от стягиваемости, это понятие может быть определено рекурсивно в гомотопической теории типов.

Свойство «Аксиома K» было так названо Томасом Стрейчером (Thomas Streicher), как свойство типов тождественности, после $\mathcal{J}$ — традиционного обозначения для выделителя типов тождественности. Теорема 7.2.5 принадлежит Хедбергу (Hedberg) [Hed98]; [KECA13] содержит больше информации и обобщений.

Понятия n-связных пространств и функций также являются классическими в гомотопической теории, хотя, как упоминалось ранее, наша индексация для связности функций отличается от классической индексации. Важность результирующей системы факторизации была подчеркнута недавней работой в теории высших топосов Резком (Rezk), Лурье (Lurie) и другими. В частности, результаты этой главы следует сравнить с [Lur09, п.6.5.1]. В §8.6 теория n-связных отображений будет иметь решающее значение для нашего доказательства теоремы о надстройке Фрейденталя (Freudenthal).

Модальные операторы широко изучались в теории *простых* типов; например, [dPGM04]. В рамках теории зависимых типов в [AB04] рассматривается частный случай пропозиционального усечения ((-1)-усечение) в качестве модального оператора. Наш подход значительно расширяет и обобщает эту работу, отображая также идеи теории топосов.

Как правило, модальные операторы возникают (по крайней мере) двумя способами: как $\lozenge$ («возможно»), для которого $A \Rightarrow \lozenge A$, и как $\square$ («обязательно»), для которого $\square A \Rightarrow A$. Когда они также идемпотентны (т.е. $\lozenge A = \lozenge\lozenge A$ или $\square A = \square\square A$), первый может быть отождествлен с рефлективными подкатегориями (или, что то же самое, с идемпотентными монадами), а второй — с корефлективными подкатегориями (или с идемпотентными комонадами). Однако, в теории зависимых типов сложнее иметь дело с комонадическими разновидностями, поскольку они редко стабильны под обратными образами и, следовательно, не могут быть интерпретированы как операции в универсуме $\mathcal{U}$. Иногда есть способы обойти это (см., например, [SS12]), но для простоты, здесь мы придерживаемся монадической разновидности.

Со стороны вычислений, монады (и, следовательно, модальности) используются для моделирования вычислительных эффектов в функциональном программировании [Mog89]. Вычисление считается чистым, если его выполнение не приводит к побочным эффектам (например, отображение сообщения на экране, воспроизведение музыки или передача данных через Интернет). Существуют «чисто функциональные» языки программирования, такие как Haskell, в которых технически возможно записать только чистые функции: побочные эффекты представлены применением «монад» к результирующим типам. Например, функция типа $\mathsf{Int} \to \mathsf{Int}$ является чистой, а функция типа $\mathsf{Int} \to \mathsf{IO}(\mathsf{Int})$ может выполнять ввод и вывод при вычислении результата; операция IO является монадой (это является источником нашего использования наречия «чисто» для тождественной монады, поскольку оно соответствует вычислительно чистым функциям без побочных эффектов). Методы, рассмотренные нами в этой главе, все идемпотентны, тогда как те, которые используются в функциональном программировании, редко являются таковыми, но идеи, по-прежнему, тесно связаны.

Упражнения

Упражнение 7.1.

(i) Используя теорему 7.2.2, покажите, что если $\|A\| \to A$ для любого типа A, то любой тип является множеством.

(ii) Покажите, что если любая сюръективная функция (чисто) расщепляема, т.е. если $\prod_{(b:B)} \|\mathsf{fib}_f(b)\| \to \prod_{(b:B)} \mathsf{fib}_f(b)$ для любой $f : A \to B$, то каждый тип является множеством.

Упражнение 7.2. Для этого упражнения рассмотрим следующее общее понятие копредела. Определим **граф** Γ, состоящий из типа Γ_0 и семейства $\Gamma_1 : \Gamma_0 \to \Gamma_0 \to \mathcal{U}$. **Диаграмма** (типов) над графом Γ состоит из семейства $F : \Gamma_0 \to \mathcal{U}$ вместе с функцией $F_{x,y} : \Gamma_1(x, y) \to F(x) \to F(\mathrm{y})$, для всех $x, y : \Gamma_0$. **Копредел** такой диаграммы представляет собой высший индуктивный тип $\mathsf{colim}(F)$, порожденный

- функцией $\mathsf{inc}_x : F(x) \to \mathsf{colim}(F)$ для всех $x : \Gamma_0$ и
- путем $\mathsf{inc}_y(F_{x,y}(\gamma, a)) = \mathsf{inc}_x(a)$ для всех $x, y : \Gamma_0$, $\gamma : \Gamma_1(x, y)$ и $a : F(x)$.

Существуют также более общие виды копределов (см., например, упражнение 7.16), но и этот достаточно полезен для многих целей.

(i) Постройте граф Γ таким образом, чтобы копределы Γ-диаграмм можно было отождествить с амальгамами, как определено в §6.8. Другими словами, каждый пролет должен индуцировать диаграмму над Γ, чей копредел — это амальгама пролета.

(ii) Постройте граф Γ и диаграмму F над Γ такую, что $F(x) = \mathbf{1}$ для всех x и $\mathsf{colim}(F) = \mathbb{S}^2$. Заметьте, что $\mathbf{1}$ является (-2)-типом, тогда как $\mathbb{S}^2$ не является n-типом для любого конечного n. См. также упражнение 7.16.

Упражнение 7.3. Покажите, что если A является n-типом и $B : A \to n\text{-}\mathsf{Type}$ — семейство n-типов, где $n \geqslant -1$, то W-тип $\mathsf{W}_{(a:A)}B(a)$ (см. §5.3) также является n-типом.

Упражнение 7.4. Используйте лемму 7.5.13, чтобы распространить лемму 7.5.11 на любую пару сечение-ретракция.

Упражнение 7.5. Покажите, что следствие 7.5.9 также работает, как характеризация, в другом направлении: B является n-типом тогда и только тогда, когда каждое отображение к B от n-связного типа является постоянным. В идеале, ваше доказательство должно работать для любой модальности, как в x7.7.

Упражнение 7.6. Докажите, что, для $n \geqslant -1$, тип A является n-связным тогда и только тогда, когда он просто обитаем и для всех $a, b : A$ тип $a =_A b$ является $(n-1)$-связным. Таким образом, поскольку каждый тип является (2)-связным, n-связность типов может быть определена индуктивно, используя только пропозициональные усечения (в частности, A является 0-связным тогда и только тогда, когда $\|A\|$ и $\prod_{(a,b:A)} \|a = b\|$).

Упражнение 7.7. Для $-1 \leqslant n$, $m \leqslant \infty$, пусть $\mathsf{LEM}_{n,m}$ обозначает выражение

$$\prod_{A:n\text{-}\mathsf{Type}} \|A + \neg A\|_m \ ,$$

где $\infty\text{-}\mathsf{Type} :\equiv U$, а $\|X\|_\infty :\equiv X$. Покажите, что:

(i) Если $n = -1$ или $m = -1$, то $\mathsf{LEM}_{n,m}$ является эквивалентом LEM из §3.4.

(ii) Если $n \geqslant 0$ и $m \geqslant 0$, то $\mathsf{LEM}_{n,m}$ несовместимо с унивалентностью.

Упражнение 7.8. Для $-1 \leqslant n,\ m \leqslant \infty$, пусть $\mathsf{AC}_{n,m}$ обозначает выражение

$$\prod_{(X:\mathsf{Set})}\ \prod_{(Y:X\to n\text{-}\mathsf{Type})}\left(\prod_{x:X}\|Y(x)\|_m\right)\to\left\|\prod_{x:X}Y(x)\right\|_m,$$

с условностями, как в упражнении 7.7. Таким образом, $\mathsf{AC}_{0,-1}$ является аксиомой выбора из §3.8, тогда как $\mathsf{AC}_{\infty,\infty}$ есть функция идентификации (если бы мы сформулировали $\mathsf{AC}_{n,m}$ аналогично (3.8.1), а не (3.8.3), то $\mathsf{AC}_{\infty,\infty}$ было бы подобно теореме 2.15.7). Известно, что $\mathsf{AC}_{\infty,-1}$ согласуется с унивалентностью, поскольку она выполняется в симплициальной модели Воеводского.

(i) Без использования унивалентности, покажите, что $\mathsf{LEM}_{n,\infty}$ подразумевает $\mathsf{AC}_{n,m}$ для всех m (с другой стороны, в §10.1.5 мы покажем, что $\mathsf{AC} = \mathsf{AC}_{0,-1}$ подразумевает $\mathsf{LEM} = \mathsf{LEM}_{-1,-1}$).

(ii) Конечно, $\mathsf{AC}_{n,m} \Rightarrow \mathsf{AC}_{k,m}$, если $k \leqslant n$. Существуют ли какие-либо другие утверждения между принципами $\mathsf{AC}_{n,m}$? Является ли $\mathsf{AC}_{n,m}$ согласованным с унивалентностью для любого $m \geqslant 0$ и любого n? (это открытые вопросы)

Упражнение 7.9. Покажите, что $\mathsf{AC}_{n,-1}$ подразумевает, что для любого n-типа , существует просто множество B и сюръекция $B \to A$.

Упражнение 7.10. Определите **n-связную аксиому выбора** как утверждение

> Если X — множество, а $Y : X \to \mathcal{U}$ — семейство типов такое, что каждое $Y(x)$ является n-связным, то $\prod_{(x:X)} Y(x)$ является n-связным.

Отметим, что (-1)-связная аксиома выбора есть $\mathsf{AC}_{\infty,-1}$ из упражнения 7.8.

(i) Докажите, что (-1)-связная аксиома выбора подразумевает n-связную аксиому выбора для всех $n \geqslant 1$.

(ii) Существуют ли какие-либо другие утверждения между n-связными аксиомами выбора и принципами $\mathsf{AC}_{n,m}$? (это открытый вопрос)

Упражнение 7.11. Покажите, что модальность n-усечения не является точной слева для любого $n \geqslant -1$. Это демонстрирует обратный образ, не сохраняющий эту модальность.

Упражнение 7.12. Покажите, что $X \mapsto (\neg\neg X)$ является модальностью.

Упражнение 7.13. Пусть P — простое высказывание.

(i) Покажите, что $X \mapsto (P \to X)$ — левая точная модальность. Она называется **открытой модальностью**, связанной с P.

(ii) Покажите, что $X \mapsto P * X$ — левая точная модальность, где $*$ обозначает присоединение (см. §6.8). Она называется **закрытой модальностью**, связанной с P.

Упражнение 7.14. Пусть $f : A \to B$ — отображение; тип Z называется f**-локальным**, если $(\_ \circ f) : (B \to Z) \to (A \to Z)$ является эквивалентностью.

(i) Докажите, что f-локальные типы образуют рефлективный подуниверсум. Вам необходимо использовать высший индуктивный тип для определения рефлектора (локализации).

(ii) Докажите, что если $B = \mathbf{1}$, то этот подуниверсум является модальностью.

Упражнение 7.15. Покажите, что в отличие от замечания 6.7.1, мы могли бы эквивалентно определить, что $\|A\|_n$ может быть порожден функцией $|\_|_n : A \to \|A\|_n$ вместе с путем $s_r(x) : r(x) = r(\mathsf{base})$, для каждого $r : \mathbb{S}^{n+1} \to \|A\|_n$ и каждого $x : \mathbb{S}^{n+1}$.

Упражнение 7.16. В этом упражнении мы рассмотрим несколько причудливое понятие копредела, отличное от приведенного в упражнении 7.2. Определим **граф с композицией** Γ как граф из упражнения 7.2 с дополнением: для всех $x, y, z : \Gamma_0$ имеется функция $\Gamma_1(y, z) \to \Gamma_1(x, y) \to \Gamma_1(x, z)$, записываемая как $\delta \mapsto \gamma \mapsto \delta \circ \gamma$ (например, любая предкатегория, как в главе 9, представляет собой граф с композицией). **Диаграмма** F над графом с композицией Γ состоит из диаграммы над лежащим в основе графом и гомотопии $\mathsf{cmp}_{x,y,z}(\delta, \gamma) : F_{y,z}(\delta) \circ F_{x,y}(\gamma) \sim F_{x,z}(\delta \circ \gamma)$ для всех $x, y, z : \Gamma_0$, $\gamma : \Gamma_1(x, y)$ и $\delta : \Gamma_1(y, z)$. **Копредел** такой диаграммы представляет собой высший индуктивный тип $\mathsf{colim}(F)$, порожденный

- функцией $\mathsf{inc}_x : F(x) \to \mathsf{colim}(F)$ для любого $x : \Gamma_0$,
- путем $\mathsf{glue}_{x,y}(\gamma, a) : \mathsf{inc}_y(F_{x,y}(\gamma, a)) = \mathsf{inc}_x(a)$ для всех $x, y : \Gamma_0$, $\gamma : \Gamma_1(x, y)$, $a : F(x)$ и
- путем $\mathsf{inc}_z\left(\mathsf{cmp}_{x,y,z}(\delta, \gamma, a)\right) \cdot \mathsf{glue}_{x,z}(\delta \circ \gamma, a) = \mathsf{glue}_{y,z}(\delta, F_{x,y}(\gamma, a)) \cdot \mathsf{glue}_{x,y}(\gamma, a)$, для всех $x, y, z : \Gamma_0$, $\gamma : \Gamma_1(x, y)$ и $\delta : \Gamma_1(y, z)$,

(это «приближение второго порядка» к полностью теоретико-гомотопическим представлениям диаграммы и копредела, которые должны включать в себя «пути когерентности» такого рода на всех высших уровнях. Определение подобных понятий в теории типов является важной открытой проблемой).
Предъявите граф с композицией Γ такой, что Γ_0 является множеством, каждый тип $\Gamma_1(x, y)$ является простым высказыванием, с диаграммой F над Γ, для которой $F(x) = \mathbf{1}$ для всех x, и $\mathsf{colim}(F) = \mathbb{S}^2$.

Упражнение 7.17. При сравнении лемм 7.5.12 и 7.5.13 может возникнуть соблазн предположить, что если $f : A \to B$ является n-связной, а $g : \prod_{(a:A)} P(a) \to Q(f(a))$ индуцирует n-связное отображение $\left(\sum_{(a:A)} P(a)\right) \to \left(\sum_{(b:B)} Q(a)\right)$, то g является послойно n-связным. Приведите контрпример, чтобы показать, что это не так (на самом деле, когда производится обобщенние модальностями, это свойство характеризует левые точные, см. упражнение 7.13).

Упражнение 7.18. Покажите, что если $f : A \to B$ является n-связной, то таковым является и $\|f\|_n : \|A\|_n \to \|B\|_n$.

Упражнение 7.19. Говорят, что тип A является **категорно связным** если, для любых типов B и C, каноническое отображение $e_{A,B,C} : ((A \to B) + (A \to C)) \to (A \to B + C)$, определяемое как

$$e_{A,B,C}(\mathsf{inl}(g)) :\equiv \lambda x.\, \mathsf{inl}(g)$$
$$e_{A,B,C}(\mathsf{inr}(g)) :\equiv \lambda x.\, \mathsf{inr}(g)$$

является эквивалентностью.

i Покажите, что любой связный тип является категорно связным.

ii Покажите, что все категорно связные типы связны тогда и только тогда, когда выполняется LEM (подсказка: рассмотрите $A :\equiv \Sigma P$ такой, что имеет место $\neg\neg P$).

Часть II

Математика

Глава 8

Теория гомотопий

В этой главе мы развиваем некоторую гомотопическую теорию в рамках теории типов. Мы используем синтетический подход к теории гомотопий, представленный в главе 2: пространства, точки, пути и гомотопии являются основными понятиями, которые представлены типами и элементами типов, в частности тождественным типом. Алгебраическая структура путей и гомотопий представлена естественной ∞-группоидной структурой на типах, которая порождается правилами для тождественного типа. Используя высшие индуктивные типы, представленные в главе 6, мы можем описать пространства непосредственно их универсальными свойствами.

Можно отметить несколько интересных аспектов такого синтетического подхода. Во-первых, он объединяет преимущества конкретных моделей (таких как, топологические пространства или симплициальные множества) с преимуществами абстрактных категориальных структур для теории гомотопий (таких как модельные категории Квиллена). С одной стороны, наши доказательства кажутся элементарными и конкретно относятся к точкам, путям и гомотопиям на типах. С другой стороны, наш подход, тем не менее, абстрагируется от какого-либо конкретного представления этих объектов — например, ассоциативность конкатенации путей подтверждается индукцией путей, а не репараметризацией отображений $[0,1] \to X$ или какими-либо условиями технического свойства. Теория типов представляется очень удобным способом изучения абстрактной гомотопической теории ∞-группоидов: используя правила для тождественного типа, мы можем избежать сложной комбинаторики, связанной со многими определениями ∞-группоидов, и привести объяснение, исходя только из структуры, как это необходимо в любом конкретном доказательстве.

Абстрактная природа теории типов означает, что наши доказательства применяются автоматически в различных ситуациях. В частности, как упоминалось ранее, гомотопическая теория типов имеет некую интерпретацию в симплициальных множествах Кана, что является одной из моделей гомотопической теории ∞-группоидов. Таким образом, наши доказательства применимы к этой модели, и перенос их вдоль функтора геометрической реализации от симплициальных множеств к топологическим пространствам дает доказательства соответствующих теорем в классической теории гомотопий. Однако, хотя детали находятся в стадии разработки, мы также можем интерпретировать теорию типов в целом ряде других категорий, которые похожи на категорию ∞-группоидов, таких как $(\infty,1)$-топосы. Таким образом, доказательство результата в теории типов покажет, что оно справедливо и в этих случаях. Этот вид дополнительной общности хорошо известен как свойство обычной категорной логики: унивалентные основания распространяют его и на теорию гомотопий.

Во-вторых, этот синтетический подход предложил новые теоретико-типовые методы и доказательства. Некоторые из наших доказательств являются довольно прямой транскрипцией классических доказательств. Другие имеют более теоретико-типовой привкус и состоят в основном из вычислений с ∞-группоидными операциями в стиле, очень похожим на то, как ученые в области информатики используют теорию типов для рассуждения о компьютерных программах. Одна особенность, которая, по-видимому, позволила использовать эти новые доказательства, состоит в том, что теория типов подчеркивает различные аспекты теории гомотопий, в отличие от других подходов: в то время как инструменты такие, как индукция по пути и универсальные свойства высших индуктивностей доступны в такой среде, как симплициальные множества Кана, теория типов усиливает их важность, потому что они являются единственными примитивными инструментами для работы с этими типами. Сосредоточение внимания на этих инструментах привело к новым описаниям знакомых конструкций, таких как универсальное покрытие круга и расслоение Хопфа, с использованием только принципов рекурсии для высших индуктивных типов. Эти описания являются непосредственными, и многие доказательства в этой главе содержат вычисляемые оценки с такими расслоениями. Еще один новый аспект наших доказательств заключается в том, что они конструктивны (при условии, что унивалентность и высшие индуктивные типы конструктивны); мы опишем применение этого к гомотопическим группам сфер в разделе 8.10.

В-третьих, наш синтетический подход очень подходит для проведенных компьютером доказательств в таких системах, как Coq и Agda. Почти все доказательства, описанные в этой главе, были получены таким образом, и многие из них затем были «деформализованы» для этой книги. Проведенные компьютером доказательства сопоставимы, по размеру и приложенным усилиям, с неформальными доказательствами, представленными здесь, а в некоторых случаях они были даже компактнее и их легче было выполнить.

Прежде чем перейти к изложению результатов, кратко рассмотрим некоторые основные понятия и теоремы из теории гомотопий, для удобства читателя, который не знаком с ними. Также приведем обзор результатов, доказанных в этой главе.

Теория гомотопий является разделом алгебраической топологии и использует инструменты из абстрактной алгебры, такие как теория групп, для исследования свойств пространств. Один из вопросов, которые исследуют теоретики, состоит в том, как определить, являются ли два пространства одинаковыми, где «одинаковость» означает *гомотопическую эквивалентность* (непрерывные отображения от одного к другому и обратно, которые компонуются к тождественности с точностью до гомотопии — это дает возможность «исправлять» отображения, композиция которых не приводит к тождественности). Распространенный способ определить, являются ли два пространства одинаковыми, — вычислить *алгебраические инварианты*, связанные с пространством, которые включают его *гомотопические группы* и *группы гомологий* и *когомологий*. Эквивалентные пространства содержат группы изоморфных гомотопий/(ко)гомологий, поэтому если два пространства имеют разные группы, то они не эквивалентны. Таким образом, эти алгебраические инварианты предоставляют глобальную информацию о пространстве, которую можно использовать для разделения пространств, и дополняют локальную информацию, предоставляемую такими понятиями, как непрерывность. Например, тор локально выглядит как 2-сфера, но у него есть глобальное отличие, потому что он содержит замкнутый туннель, и эта разница видна в гомотопических группах этих двух пространств.

Простейшим примером гомотопической группы является *фундаментальная группа* пространства, которая записывается как $\pi_1(X, x_0)$: для пространства X и точки x_0 в нем, можно построить группу, элементами которой являются петли в точке x_0 (непрерывные пути от x_0 к x_0), рассматриваемые с точностью до гомотопии, с групповыми операциями, заданными тождественным путем (оставляющим на месте), конкатенацией путей и реверсированием путей.

Например, фундаментальная группа 2-сферы тривиальна, а фундаментальная группа тора — нет, и это показывает, что сфера и тор гомотопически не эквивалентны. Интуиция заключается в том, что каждая петля на сфере гомотопна тождественности, потому что ее внутренность может быть заполнена. Напротив, петля на торе, проходящая через отверстие «пончика», не гомотопна тождественности, поэтому существуют нетривиальные элементы в фундаментальной группе.

Высшие гомотопические группы предоставляют дополнительную информацию о пространстве. Зафиксируем точку x_0 в X и рассмотрим постоянный путь refl_{x_0}. Тогда гомотопические классы гомотопий между refl_{x_0} и им же, образуют группу $\pi_2(X, x_0)$, что говорит кое-что о двумерной структуре пространства. Тогда $\pi_3(X, x_0)$ — группа гомотопических классов гомотопий между гомотопиями и т.д. Одной из основных проблем алгебраической топологии является *вычисление гомотопических групп пространства* X, что означает задание изоморфизма групп между $\pi_k(X, x_0)$ и некоторым более прямым описанием группы (например, с помощью таблицы умножения или представления). Несколько удивительно, что это очень сложный вопрос, даже для таких простых пространств, как сферы. Как видно из таблицы 8.1, некоторые закономерности появляются в высших гомотопических группах сфер, но общая формула отсутствует, и многие гомотопические группы сфер в настоящее время все еще неизвестны.

	$\mathbb{S}^0$	$\mathbb{S}^1$	$\mathbb{S}^2$	$\mathbb{S}^3$	$\mathbb{S}^4$	$\mathbb{S}^5$	$\mathbb{S}^6$	$\mathbb{S}^7$	$\mathbb{S}^8$
π_1	0	$\mathbb{Z}$	0	0	0	0	0	0	0
π_2	0	0	$\mathbb{Z}$	0	0	0	0	0	0
π_3	0	0	$\mathbb{Z}$	$\mathbb{Z}$	0	0	0	0	0
π_4	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0	0
π_5	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0	0
π_6	0	0	$\mathbb{Z}_{12}$	$\mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	0
π_7	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z} \times \mathbb{Z}_{12}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$	0
π_8	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}$
π_9	0	0	$\mathbb{Z}_3$	$\mathbb{Z}_3$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$	$\mathbb{Z}_2$
π_{10}	0	0	$\mathbb{Z}_{15}$	$\mathbb{Z}_{15}$	$\mathbb{Z}_{24} \times \mathbb{Z}_3$	$\mathbb{Z}_2$	0	$\mathbb{Z}_{24}$	$\mathbb{Z}_2$
π_{11}	0	0	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$\mathbb{Z}_{15}$	$\mathbb{Z}_2$	$\mathbb{Z}$	0	$\mathbb{Z}_{24}$
π_{12}	0	0	$\mathbb{Z}_2^2$	$\mathbb{Z}_2^2$	$\mathbb{Z}_2$	$\mathbb{Z}_{30}$	$\mathbb{Z}_2$	0	0
π_{13}	0	0	$\mathbb{Z}_{12} \times \mathbb{Z}_2$	$\mathbb{Z}_{12} \times \mathbb{Z}_2$	$\mathbb{Z}_2^3$	$\mathbb{Z}_2$	$\mathbb{Z}_{60}$	$\mathbb{Z}_2$	0

Таблица 8.1. Гомотопические группы сфер [Wik13]. k-я гомотопическая группа π_k n-мерной сферы $\mathbb{S}^n$ изоморфна группе, указанной в каждой ячейке таблицы, где $\mathbb{Z}$ — аддитивная группа целых чисел, а $\mathbb{Z}_m$ — циклическая группа порядка m.

Одним из способов понять эту сложность является соответствие между пространствами и ∞-группоидами, введенное в главе 2. Как обсуждалось в разделе §6.4, 2-сфера представлена высшим индуктивным типом с одной точкой и одной 2-мерной петлей. Таким образом, можно задаться вопросом, почему $\pi_3(\mathbb{S}^2)$ — это $\mathbb{Z}$, когда тип $\mathbb{S}^2$ не имеет образующих, создающих трехмерные клетки. Оказывается, что порождающий элемент $\pi_3(\mathbb{S}^2)$ строится с использованием закона обмена, описанного в доказательстве теоремы 2.1.6: алгебраическая структура ∞-группоида включает в себя нетривиальные взаимодействия между уровнями, и эти взаимодействия создают элементы высших гомотопических групп.

Теория типов обеспечивает естественное окружение для исследования этой структуры, поскольку мы можем легко определить высшие гомотопические группы. Напомним из определения 2.1.8, что для $n : \mathbb{N}$ пространство итерированных петель n-кратного точечного типа (A, a) определяется рекурсивно:

$$\Omega^0(A,a) :\equiv (A,a)$$
$$\Omega^{n+1}(A,a) :\equiv \Omega^n(\Omega(A,a)).$$

Это дает *пространство* (то есть тип) n-мерных петель, которое само имеет высшие гомотопии. Получаем множество n-мерных петель путем усечения (что также было определено в качестве примера в §6.11):

Определение 8.0.1 (гомотопические группы)**.** Для $n \geqslant 1$ и точечного типа (A, a) определим **гомотопические группы** A в точке a как

$$\pi_n(A,a) :\equiv \left\|\Omega^n(A,a)\right\|_0$$

Поскольку $n \geqslant 1$, операции конкатенации пути и инверсии на $\Omega^n(A)$ индуцируют операции на $\pi_n(A)$, превращая его в группу простым способом. Если $n \geqslant 2$, то группа $\pi_n(A)$ является абелевой по аргументу Экмана-Хилтона (теорема 2.1.6). Также удобно записать $\pi_0(A) :\equiv \|A\|_0$, но в этом случае дело обстоит несколько иначе: это не только не группа, но и определено без ссылки на какую-либо отмеченную точку в A.

Это определение подходит для исследования гомотопических групп, потому что индуктивное определение типа X (высшего порядка) представляет X как свободный тип, аналогичный свободному ∞-группоиду, и это представление *определяет*, но не *описывает явно* тождественные типы X высших порядков. Тождественные типы заселяются как образующими (петли, для окружности), так и результатами применения к ним всех операций группоида (тождественность, композиция, инверсия, ассоциативность, взаимообмен, ...). Таким образом, высшее индуктивное представление пространства позволяет задать вопрос: «Каким на самом деле оказывается тождественный тип для X?», хотя для ответа на него может потребоваться серьезная математика. Это многомерное обобщение известного факта теории типов: характеризация тождественного типа X может потребовать некоторой работы, даже если X является обычным индуктивным типом, таким как натуральные числа или логические значения. Например, теорема о том, что $0_\mathbf{2}$ отличается от $1_\mathbf{2}$, не следует сразу из определения; см. §2.12.

Аксиома унивалентности играет существенную роль при вычислении гомотопических групп (без унивалентности теория типов совместима с интерпретацией, в которой все пути, включая, например, петлю на окружности, являются рефлексивными). Мы увидим это ниже при вычислении фундаментальной группы окружности: отображение от $\Omega(\mathbb{S}^1)$ к $\mathbb{Z}$ определяется отображением петли на окружности в автоморфизм множества $\mathbb{Z}$, так что, например, $\mathsf{loop} \cdot \mathsf{loop}^{-1}$ отсылается в $\mathsf{successor} \cdot \mathsf{predecessor}$ (где $\mathsf{successor}$ и $\mathsf{predecessor}$ — автоморфизмы $\mathbb{Z}$, рассматриваемые, в силу унивалентности, как пути в универсуме), а затем, применением автоморфизма к 0. Унивалентность создает нетривиальные пути в универсуме, и это используется для извлечения информации из путей в высших индуктивных типах.

В этой главе сначала вычисляются некоторые гомотопические группы сфер, включая $\pi_k(\mathbb{S}^1)$ (§8.1), $\pi_k(\mathbb{S}^n)$ для $k < n$ (§§ 8.2 и 8.3), $\pi_2(\mathbb{S}^2)$ и $\pi_3(\mathbb{S}^2)$, согласно пути расслоения Хопфа (§8.5) и аргумента длинной точной последовательности (§8.4) и $\pi_n(\mathbb{S}^n)$ с помощью теоремы Фрейденталя о приостановке (§8.6}). Затем обсудим теорему ван Кампена (§8.7), которая характеризует фундаментальную группу амальгамы, и статус принципа Уайтхеда (когда отображение, индуцирующее эквивалентность на всех гомотопических группах, является эквивалентностью?) (§8.8).

Наконец, приведено краткое изложение дополнительных результатов, которые не включены в книгу, таких как $\pi_{n+1}(\mathbb{S}^n)$ для $n \geqslant 3$, теорема Бейкера-Месси и конструкция пространств Эйленберга-Маклейна (§8.10). Предварительные требования к этой главе включают информацию глав 1, 2, 6 и 7, а также некоторых частей главы 3.

8.1 Гомотопическая группа сфер $\pi_1(\mathbb{S}^1)$

Наша цель в этом разделе — показать, что $\pi_1(\mathbb{S}^1) = \mathbb{Z}$. Фактически, мы покажем, что пространство петель $\Omega(\mathbb{S}^1)$ эквивалентно $\mathbb{Z}$. Это более сильное утверждение, потому что $\pi_1(\mathbb{S}^1) = \|\Omega(\mathbb{S}^1)\|_0$ по определению; поэтому, если $\Omega(\mathbb{S}^1) = \mathbb{Z}$, то $\|\Omega(\mathbb{S}^1)\|_0 = \|\mathbb{Z}\|_0$ по конгруэнтности, а $\mathbb{Z}$ — множество по определению (являющееся частным по множеству; см. 6.10.7 и 6.10.11), поэтому $\|\mathbb{Z}\|_0 = \mathbb{Z}$. Более того, знание того, что $\Omega(\mathbb{S}^1)$ является множеством, будет означать, что $\pi_n(\mathbb{S}^1)$ тривиально для $n > 1$, поэтому мы фактически вычислим *все* гомотопические группы $\mathbb{S}^1$.

8.1.1 Введение

Нетрудно определить функции в обоих направлениях между $\Omega(\mathbb{S}^1)$ и $\mathbb{Z}$. Специализируя следствие 6.10.13 к $\mathsf{loop} : \mathsf{base} = \mathsf{base}$, получаем функцию $\mathsf{loop}^- : \mathbb{Z} \to (\mathsf{base} = \mathsf{base})$, определяемую (примерно) как

$$\mathsf{loop}^n = \begin{cases} \underbrace{\mathsf{loop} \cdot \mathsf{loop} \cdot \ldots \cdot \mathsf{loop}}_{n} & \text{если } n > 0\,, \\ \underbrace{\mathsf{loop}^{-1} \cdot \mathsf{loop}^{-1} \cdot \ldots \cdot \mathsf{loop}^{-1}}_{n} & \text{если } n < 0\,, \\ \mathsf{refl}_{\mathsf{base}} & \text{если } n = 0\,. \end{cases}$$

Определение функции $g : \Omega(\mathbb{S}^1) \to \mathbb{Z}$ в другом направлении немного сложнее. Обратите внимание, что функция-преемник $\mathsf{suc} : \mathbb{Z} \to \mathbb{Z}$ является эквивалентностью и, следовательно, индуцирует путь $\mathsf{ua}(\mathsf{suc}) : \mathbb{Z} = \mathbb{Z}$ в универсуме $\mathcal{U}$. Таким образом, принцип рекурсии для $\mathbb{S}^1$ индуцирует отображение $c : \mathbb{S}^1 \to \mathcal{U}$ посредством $c(\mathsf{base}) :\equiv \mathbb{Z}$ и $\mathsf{ap}_c(\mathsf{loop}) := \mathsf{ua}(\mathsf{suc})$. Тогда, имеем $\mathsf{ap}_c : (\mathsf{base} = \mathsf{base}) \to (\mathbb{Z} = \mathbb{Z})$, и можно определить $g(p) :\equiv \mathsf{transport}^{X \mapsto X}(\mathsf{ap}_c(p), 0)$.

С помощью этих определений мы даже можем доказать, что $g(\mathsf{loop}^n) = n$ для любого $n : \mathbb{Z}$, используя принцип индукции леммы 6.10.12 для n (чуть позже мы докажем кое-что более общее). Однако, другое равенство — $\mathsf{loop}^{g(p)} = p$, значительно сложнее. Очевидная вещь, которую стоит попробовать, — это индукция по пути, но она не применяется к петлям, таким как $p : (\mathsf{base} = \mathsf{base})$, у которых фиксированы *обе* конечные точки! Требуется новая идея, которую можно объяснить, как в терминах классической теории гомотопий, так и в терминах теории типов. Начнем с первого.

8.1.2 Классическое доказательство

В классической теории гомотопий существует стандартное доказательство $\pi_1(\mathbb{S}^1) = \mathbb{Z}$ с использованием универсальных покрывающих пространств. Наше доказательство можно рассматривать как теоретико-ти́повую версию этого доказательства, при этом покрывающие пространства появляются здесь как расслоения, слои которых являются множествами. Напомним, что *расслоения* над пространством B в теории гомотопий соответствуют семействам типов $B \to \mathcal{U}$ в теории типов. В частности, для точки $x_0 : B$ семейство типов $(x \mapsto (x_0 = x))$ соответствует *расслоению путей* $P_{x_0}B \to B$, в котором точки $P_{x_0}B$ — это пути в B, начинающиеся с x_0, и отображение в B выбирает другую конечную точку такого пути. Это пространство расслоений $P_{x_0}B$ стягиваемо, поскольку мы можем «отвести» любой путь к его инициальной конечной точке x_0 — мы видели теоретико-типовую версию этого как лемму 3.11.8. Более того, слой над x_0 является пространством петель $\Omega(B, x_0)$ — в теории типов это очевидно из определения пространства петель.

Теперь, в классической теории гомотопий, где $\mathbb{S}^1$ рассматривается как топологическое пространство, можно поступить следующим образом. Рассмотрим «винтовое» отображение $w : \mathbb{R} \to \mathbb{S}^1$, которое выглядит как спираль, спускающаяся на окруность (см. рис.8.1). Это отображение w отсылает каждую точку спирали к точке на окружности, «над которой» она находится. Это расслоение, и слой над каждой точкой изоморфен целым числам. Если поднимается путь, идущий против часовой стрелки вокруг петли внизу, то поднятие производится на один уровень по спирали, увеличивая целое число, связанное с слоем. Точно так же, обход петли вниз по часовой стрелке соответствует спуску на один уровень спирали с уменьшением этого счетчика. Это расслоение называется *универсальным покрытием* окружности.

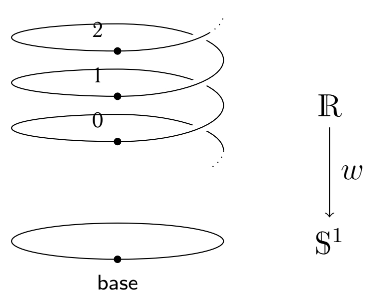


Рис. 8.1. Винтовое отображение в классической топологии

Основным фактом классической теории гомотопий является то, что отображение $E_1 \to E_2$ расслоений над B, которое является гомотопической эквивалентностью между E_1 и E_2, индуцирует гомотопическую эквивалентность на всех слоях (теоретико-типовая версия этого уже встречалась в теореме 4.7.7). Поскольку $\mathbb{R}$ и $P_{\mathsf{base}}\mathbb{S}^1$ являются стягиваемыми топологическими пространствами, они гомотопически эквивалентны, и, следовательно, их слои $\mathbb{Z}$ и $\Omega(\mathbb{S}^1)$ над отмеченной точкой также гомотопически эквивалентны.

8.1.3 Универсальное покрытие в теории типов

Рассмотрим, как можно было бы выразить предыдущее доказательство в теории типов. Уже отмечалось, что расслоение путей в $\mathbb{S}^1$ представлено семейством типов $(x \mapsto (\mathsf{base} = x))$. Также, имеется хороший кандидат для универсального покрытия S^1: это не что иное, как семейство типов $c : \mathbb{S}^1 \to \mathcal{U}$, которое было определено в §8.1.1! По определению, слой этого семейства над base — это $\mathbb{Z}$, в то время как эффект перемещения вокруг loop заключается в добавлении такого же — таким образом, он ведет себя так, как и ожидается, из рис.8.1.

Однако, поскольку мы еще не знаем, что это семейство ведет себя так, как универсальное покрытие (например, что его пространство расслоений просто является связным), то используем для него другое имя. Поэтому, для справки, повторим определение.

Определение 8.1.1 (Универсальное покрытие $\mathbb{S}^1$)**.** Определим $\mathsf{code} : \mathbb{S}^1 \to \mathcal{U}$ круговой рекурсией

$$\begin{aligned} \mathsf{code}(\mathsf{base}) &:\equiv \mathbb{Z} \\ \mathsf{ap}_{\mathsf{code}}(\mathsf{loop}) &:= \mathsf{ua}(\mathsf{succ}) \end{aligned}$$

Мы кратко акцентируемся на определении этого семейства, поскольку оно сильно отличается от того, как обычно определяют покрывающие пространства в классической теории гомотопий. Чтобы определить функцию с помощью круговой рекурсии, нужно найти точку и петлю в кообласти. В данном случае кообласть — это $\mathcal{U}$, а точка, которую мы выбираем, — это $\mathbb{Z}$, что соответствует ожиданию, что слой универсального покрытия должен быть представлен целыми числами. Выбранная петля является изоморфизмом преемника/предшественника на $\mathbb{Z}$, который соответствует тому факту, что обход петли в основании поднимается на один уровень спирали. Для этой части доказательства необходима унивалентность, потому что нам нужно преобразовать *нетривиальную* эквивалентность на $\mathbb{Z}$ в тождественность.

Мы называем это расслоением «кодов», поскольку его элементы являются комбинаторными данными, которые действуют как коды для путей на окружности: целое число n кодирует путь, который проходит по окружности n раз.

Из этого определения легко вычислить, что транспортировка с помощью code использует loop для перехода к функции-преемнику, а loop^{-1} — к функции-предшественнику:

Лемма 8.1.2. $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, x) = x + 1$ *и* $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^{-1}, x) = x - 1$.

Доказательство. Для первого уравнения имеем:

$$\begin{aligned} \mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, x) &= \mathsf{transport}^{A \mapsto A}((\mathsf{code}(\mathsf{loop})), x) && \text{(по лемме 2.3.10)} \\ &= \mathsf{transport}^{A \mapsto A}(\mathsf{ua}(\mathsf{succ}), x) && \text{(вычисление для } \mathsf{rec}_{\mathbb{S}^1}\text{)} \\ &= x + 1 && \text{(вычисление для } \mathsf{ua}\text{)} \end{aligned}$$

Второе уравнение следует из первого, поскольку $\mathsf{transport}^B(p, -)$ и $\mathsf{transport}^B(p^{-1}, -)$ всегда инвертированы, поэтому $\mathsf{transport}^{\mathsf{code}}(p^{-1}, -)$ должен быть обратным к succ. □

Теперь можно понять, что было не так с первым подходом: мы определили f и g только на слоях $\Omega(\mathbb{S}^1)$ и $\mathbb{Z}$, тогда как должны были определить весь морфизм *расслоений* над $\mathbb{S}^1$. В теории типов это означает, что должны быть определены функции, имеющие типы

$$\prod_{x:\mathbb{S}^1} ((\mathsf{base} = x) \to \mathsf{code}(x)) \quad \text{и/или} \tag{8.1.3}$$

$$\prod_{x:\mathbb{S}^1} (\mathsf{code}(x) \to (\mathsf{base} = x)) \tag{8.1.4}$$

а не только их частные случаи, когда x равно base. Это — также пример общего наблюдения в теории типов: при попытке доказать что-то о конкретных обитателях некоторого индуктивного типа часто легче обобщить утверждение так, чтобы оно относилось ко *всем* обитателям этого типа, что тогда можно доказать по индукции. С этой точки зрения, доказательство $\Omega(\mathbb{S}^1) = \mathbb{Z}$ укладывается в тот же образец, что и характеризация тождественных типов копроизведений и натуральных чисел в §§ 2.12 и 2.13.

В этой точке имеются два способа завершить доказательство. Можно продолжить подражание классическому аргументу, построив (8.1.3) или (8.1.4), неважно, какой из них, доказывая, что гомотопическая эквивалентность между пространствами расслоений индуцирует эквивалентность на слоях, а затем то, что пространство расслоений универсального покрытия является стягиваемым. Первое теоретико-типовое доказательство для $\Omega(\mathbb{S}^1) = \mathbb{Z}$ проводилось по этой схеме; мы называем его *теоретико-гомотопическим* доказательством.

Позже, однако, мы обнаружили, что существует альтернативное доказательство, которое имеет более теоретический характер и более точно следует доказательствам в §§ 2.12 и 2.13. В этом доказательстве мы напрямую конструируем как (8.1.3), так и (8.1.4), и доказываем вычислением, что они взаимно обратны. Будем называть его доказательством *кодированием-декодированием*, потому что мы называем функции (8.1.3) и (8.1.4) *кодированием* и *декодированием*, соответственно. Оба доказательства используют одну и ту же конструкцию покрытия, приведенную выше. Если классическое доказательство индуцирует эквивалентность на слоях из эквивалентности пространств расслоений, доказательство кодированием-декодированием строит обратное отображение (*декодирование*) явно, как отображение между слоями. И там, где классическое доказательство использует стягиваемость, доказательство кодированием-декодированием использует индукцию пути, индукцию окружности и целочисленную индукцию. Это те же самые инструменты, которые используются для доказательства стягиваемости — действительно, индукция пути, по сути, *является* стягиваемостью расслоения пути, составленного с помощью $\mathsf{transport}$ — но они применяются по-другому.

Поскольку это книга о теории гомотопических типов, сначала мы представляем доказательство кодированием-декодированием. Случайно попавшему сюда гомотопическому теоретику рекомендуется сразу перейти к теоретико-гомотопическому доказательству (§8.1.5).

8.1.4 Доказательство кодированием-декодированием

Начнем с функции (8.1.3), которая отображает пути в коды:

Определение 8.1.5. Определим $\mathsf{encode} : \prod_{x:\mathbb{S}^1}(\mathsf{base} = x) \to \mathsf{code}(x)$ как

$$\mathsf{encode}\, p :\equiv \mathsf{transport}^{\mathsf{code}}(p, 0)$$

(аргумент x полагаем неявным).

Кодирование определяется поднятием пути в универсальное покрытие, которое определяет эквивалентность, и последующим применением полученной эквивалентности к 0. Интересная особенность этой функции заключается в том, что она вычисляет конкретное число по петле на окружности, когда эта петля представлена с использованием абстрактной группоидной структуры теории гомотопических типов. Чтобы получить представление о том, как это происходит, заметьте, что согласно приведенным выше леммам $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, x)$ — это отображение-преемник, а $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^{-1}, x)$ — отображение-предшественник. Кроме того, $\mathsf{transport}$ является функториальным (глава 2), так что $\mathsf{transport}^{\mathsf{code}}(\mathsf{loop} \cdot \mathsf{loop}, -)$ есть

$$(\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, -)) \circ (\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, -))$$

и т.д. Таким образом, когда p представляет собой композицию вида

$$\mathsf{loop} \cdot \mathsf{loop}^{-1} \cdot \mathsf{loop} \cdot \cdots$$

$\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, -)$ вычислит композицию таких функций, как

$$\mathsf{succ} \circ \mathsf{pred} \circ \mathsf{succ} \circ \cdots$$

Применение этой композиции функций к 0 позволяет вычислить *число витков* пути — сколько раз он проходит по окружности с ориентацией, отмеченной положительным или отрицательным значением, после отмены инверсий. Таким образом, вычислительное поведение encode следует из правил редукции для высоко-индуктивных типов и унивалентности, а также действия $\mathsf{transport}$ на композиции и инверсные объекты.

Заметьте, что экземпляр $\mathsf{encode}' :\equiv \mathsf{encode}_{\mathsf{base}}$ имеет тип $(\mathsf{base} = \mathsf{base}) \to \mathbb{Z}$. Это будет половина желаемой эквивалентности; действительно, это в точности функция g, определенная в §8.1.1.

Аналогично, функция (8.1.4) является обобщением функции loop^- из §8.1.1.

Определение 8.1.6. Определим $\mathsf{decode} : \prod_{x:\mathbb{S}^1} \mathsf{code}(x) \to (\mathsf{base} = x)$ индукцией по окружности, по x. Достаточно указать функцию $\mathsf{code}(\mathsf{base}) \to (\mathsf{base} = \mathsf{base})$, для которой используется loop^-, и показать, что loop^- ожидаемым образом работает с петлей.

Обоснование определения *8.1.6.* Чтобы показать, что loop^{-1} ведет себя должным образом, достаточно указать путь от loop^{-1} к самому себе, который лежит над loop. По определению зависимых путей, это означает путь от

$$\mathsf{transport}^{(x' \mapsto \mathsf{code}(x') \to (\mathsf{base}=x'))}(\mathsf{loop}, \mathsf{loop}^-)$$

к loop^{-1}. Определим такой путь следующим образом:

$$\begin{aligned}
&\mathsf{transport}^{(x' \mapsto \mathsf{code}(x') \to (\mathsf{base}=x'))}(\mathsf{loop}, \mathsf{loop}^-)\\
&\qquad = \mathsf{transport}^{x' \mapsto (\mathsf{base}=x')}(\mathsf{loop}) \circ \mathsf{loop}^- \circ \mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^{-1})\\
&\qquad = (- \centerdot \mathsf{loop}) \circ (\mathsf{loop}^-) \circ \mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^{-1})\\
&\qquad = (- \centerdot \mathsf{loop}) \circ (\mathsf{loop}^-) \circ \mathsf{pred}\\
&\qquad = (n \mapsto \mathsf{loop}^{n-1} \centerdot \mathsf{loop}).
\end{aligned}$$

В первой строке применяется характеризацию $\mathsf{transport}$, когда внешней связкой расслоения является $\to$, что редуцирует $\mathsf{transport}$ до пред- и пост-композиции с $\mathsf{transport}$ в типах области и кообласти. Во второй строке характеризация $\mathsf{transport}$ применяется, когда семейство типов есть $x \mapsto \mathsf{base} = x$, что является пост-композицией путей. В третьей строке используется действие code на loop^{-1} из леммы 8.1.2. А в четвертой строке просто редуцируется функциональная композиция. Таким образом, достаточно показать, что, для всех n, $\mathsf{loop}^{n-1} \centerdot \mathsf{loop} = \mathsf{loop}^n$. А это есть простое применение леммы 6.10.12 с использованием группоидных законов. □

Теперь уже можно показать, что кодирование и декодирование являются квазиобратными. То, что раньше казалось трудным, теперь становится легким!

Лемма 8.1.7. *Для всех $x : \mathbb{S}^1$ и $p : \mathsf{base} = x$ имеет место $\mathsf{decode}_x(\mathsf{encode}_x(p)) = p$.*

Доказательство. Достаточно показать, что $\mathsf{decode}_{\mathsf{base}}(\mathsf{encode}_{\mathsf{base}}(\mathsf{refl}_{\mathsf{base}})) = \mathsf{refl}_{\mathsf{base}}$, используя индукцию по путям. Но $\mathsf{encode}_{\mathsf{base}}(\mathsf{refl}_{\mathsf{base}}) \equiv \{\mathsf{transport}\}^{\mathsf{code}}(\mathsf{refl}_{\mathsf{base}}, 0)$, а $\mathsf{decode}_{\mathsf{base}}(0) \equiv \mathsf{loop}^0 \equiv \mathsf{refl}_{\mathsf{base}}$. □

Другое направление не намного сложнее.

Лемма 8.1.8. *Для всех $x : \mathbb{S}^1$ and $c : \mathsf{code}(x)$ имеем $\mathsf{encode}_x(\mathsf{decode}_x(c)) = c$.*

Доказательство. Доказательство проведем индукцией по окружности. Достаточно сделать это для base, потому что вариант для loop — это путь между путями в $\mathbb{Z}$, который является непосредственным, потому что $\mathbb{Z}$ — множество.

Таким образом, достаточно показать для всех $n : \mathbb{Z}$, что

$$\mathsf{encode}'(\mathsf{loop}^n) = n\,.$$

Доказательство проведем по индукции с использованием леммы 6.10.12.

- В случае 0 результат верен по определению.
- В случае $n+1$,

$$\begin{aligned}
\mathsf{encode}'(\mathsf{loop}^{n+1}) &= \mathsf{encode}'(\mathsf{loop}^n \cdot \mathsf{loop}) && (\text{определение } \mathsf{loop}^-)\\
&= \mathsf{transport}^{\mathsf{code}}((\mathsf{loop}^n \cdot \mathsf{loop}), 0) && (\text{определение } \mathsf{encode})\\
&= \mathsf{transport}^{\mathsf{code}}(\mathsf{loop}, (\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^n, 0))) && (\text{функториальность})\\
&= (\mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^n, 0)) + 1 && (\text{лемма } 8.1.2)\\
&= n + 1\,. && (\text{индуктивное прелположение})
\end{aligned}$$

- Случай с отрицательными значениями аналогичен.

$\square$

Наконец, приходим к выводу теоремы.

Теорема 8.1.9. *Существует семейство эквивалентностей* $\prod_{(x:\mathbb{S}^1)}((\mathsf{base} = x) \simeq \mathsf{code}(x))$.

Доказательство. Отображения encode и decode квазиобратны по леммам 8.1.7 и 8.1.8. $\square$

Создание экземпляра на base дает

Следствие 8.1.10. $\Omega(\mathbb{S}^1, \mathsf{base}) \simeq \mathbb{Z}$.

Простая индукция показывает, что эта эквивалентность требует добавления к композиции, так что $\Omega(\mathbb{S}^1) = \mathbb{Z}$ как группы.

Следствие 8.1.11. $\pi_1(\mathbb{S}^1) = \mathbb{Z}$, *а* $\pi_n(\mathbb{S}^1) = 0$ *для* $n > 1$.

Доказательство. Для $n = 1$ доказательство вытекает из следствия 8.1.10. При $n > 1$ имеем $\|\Omega^n(\mathbb{S}^1)\|_0 = \|\Omega^{n-1}(\Omega\mathbb{S}^1)\|_0 = \|\Omega^{n-1}(\mathbb{Z})\|_0$. А поскольку $\mathbb{Z}$ — множество, то $\Omega^{n-1}(\mathbb{Z})$ является стягиваемым, что и завершает доказательство. $\square$

8.1.5 Теоретико-гомотопическое доказательство

В §8.1.3 было определено предполагаемое универсальное покрытие $\mathsf{code} : \mathbb{S}^1 \to U$ в теории типов, а в §8.1.4 — отображение $\mathsf{encode} : \prod_{(x:\mathbb{S}^1)}(\mathsf{base} = x) \to \mathsf{code}(x)$ от расслоения путей к универсальному покрытию. Для классического доказательства остается показать, что это отображение индуцирует эквивалентность на пространствах расслоений, поскольку оба они являются стягиваемыми, и вывести из этого, что оно должно быть эквивалентностью на каждом слое.

По лемме 3.11.8 известно, что пространство расслоений $\sum_{(x:\mathbb{S}^1)}(\mathsf{base} = x)$ стягиваемо. Для другого пространства имеется:

Лемма 8.1.12. *Тип* $\sum_{(x:\mathbb{S}^1)}\mathsf{code}(x)$ *является стягиваемым.*

Доказательство. Применим лемму сглаживания (лемма 6.12.2) со следующими значениями:

- $A :\equiv \mathbf{1}$ и $B :\equiv \mathbf{1}$, с очевдными функциями f и g. Так что, базовый высший индуктивный тип W в лемме сглаживания эквивалентен $\mathbb{S}^1$.
- $C : A \to \mathcal{U}$ — константа при $\mathbb{Z}$.
- $D : \prod_{(b:B)}(\mathbb{Z} \simeq \mathbb{Z})$ — константа при succ.

Тогда семейство типов $P : \mathbb{S}^1 \to \mathcal{U}$, определенное в лемме сглаживания, эквивалентно $\mathsf{code} : \mathbb{S}^1 \to \mathcal{U}$. Таким образом, из леммы сглаживания следует, что $\sum_{(x:\mathbb{S}^1)}\mathsf{code}(x)$ эквивалентен высшему индуктивному типу, который мы обозначаем как R, со следующими образующими:

- функцией $\mathsf{c} : \mathbb{Z} \to R$,
- для каждого $z : \mathbb{Z}$, путем $\mathsf{p}_z : \mathsf{c}(z) = \mathsf{c}(\mathsf{succ}(z))$.

Можно было бы назвать этот тип **гомотопическими действительными числами**; этот тип играет ту же роль, что и топологическое пространство $\mathbb{R}$ в классическом доказательстве.

Таким образом, осталось показать, что R стягиваемо. В качестве центра стягивания выберем $\mathsf{c}(0)$; тогда надо показать, что $x = \mathsf{c}(0)$ для всех $x : R$. Сделаем это индукцией по R. Во-первых, когда x равно $\mathsf{c}(z)$, необходимо указать путь $q_z : \mathsf{c}(0) = \mathsf{c}(z)$, что можно сделать индукцией по $z : \mathbb{Z}$, используя лемму 6.10.12:

$$\begin{aligned} q_0 &:= \mathsf{refl}_{\mathsf{c}(0)} & \\ q_{n+1} &:= q_n \cdot \mathsf{p}(n) & \text{для } n \geqslant 0 \\ q_{n-1} &:= q_n \cdot \mathsf{p}_{n-1}^{-1} & \text{для } n \leqslant 0 \end{aligned}$$

Во-вторых, мы должны показать, что для любого $z : \mathbb{Z}$ путь q_z переносится вдоль p_z к q_{z+1}. При транспортировке путей это означает, что требуется, чтобы $q_z \cdot \mathsf{p}_z = q_{z+1}$. Это легко сделать индукцией по z, используя определение q_z. Это завершает доказательство того, что R стягиваемо, а значит, таково и $\sum_{(x:\mathbb{S}^1)}\mathsf{code}(x)$. □

Следствие 8.1.13. *Отображение, индуцированное посредством* encode:

$$\sum_{(x:\mathbb{S}^1)}(\mathsf{base} = x) \to \sum_{(x:\mathbb{S}^1)}\mathsf{code}(x)$$

является эквивалентностью.

Доказательство. Оба типа являются стягиваемыми. □

Теорема 8.1.14. $\Omega(\mathbb{S}^1, \mathsf{base}) \simeq \mathbb{Z}$.

Доказательство. Применяем теорему 4.7.7 к encode, используя следствие 8.1.13. □

По сути, эти два доказательства не очень различаются: доказательство кодированием-декодированием можно рассматривать как «редукцию» или «распаковку» теоретико-гомотопического доказательства. У каждого из них имеются свои преимущества; взаимодействие между двумя точками зрения является частью заинтересованности субъекта.

8.1.6 Универсальное покрытие как система тождественностей

Обратите внимание, что расслоение $\mathsf{code} : \mathbb{S}^1 \to \mathcal{U}$ вместе с $0 : \mathsf{code}(\mathsf{base})$ является точечным предикатом в смысле определения 5.8.1. С этой точки зрения можно понять, что доказательство кодированием-декодированием в §8.1.4 содержит доказательство того, что code удовлетворяет теореме 5.8.2(iii), в то время как теоретико-гомотопическое доказательство в §8.1.5 состоит из доказательства того, что code удовлетворяет теореме 5.8.2(iv). Это предполагает третий подход.

Теорема 8.1.15. *Пара* $(\mathsf{code}, 0)$ *является системой тождественностей при* $\mathsf{base} : \mathbb{S}^1$ *в смысле определения* 5.8.1.

Доказательство. Пусть даны $D : \prod_{(x:\mathbb{S}^1)} \mathsf{code}(x) \to \mathcal{U}$ и $d : D(\mathsf{base}, 0)$; мы хотим определить функцию $f : \prod_{(x:\mathbb{S}^1)} \prod_{(c:\mathsf{code}(x))} D(x, c)$. Индукцией по окружности достаточно задать $f(\mathsf{base}) : \prod_{(c:\mathsf{code}(\mathsf{base}))} D(\mathsf{base}, c)$ и проверить, что $\mathsf{loop}_*(f(\mathsf{base})) = f(\mathsf{base})$.

Конечно, $\mathsf{code}(\mathsf{base}) \equiv \mathbb{Z}$. Используя лемму 8.1.2 и индукцию по n, мы можем получить путь $p_n : \mathsf{transport}^{\mathsf{code}}(\mathsf{loop}^n, 0) = n$ для любого целого числа n. Следовательно, по путям в Σ-типах у нас есть путь $\mathsf{pair}^=(\mathsf{loop}^n, p_n) : (\mathsf{base}, 0) = (\mathsf{base}, n)$ в $\Sigma_{(x:\mathbb{S}^1)}\mathsf{code}(x)$. Перенося d по этому пути в расслоении $\widehat{D} : (\Sigma_{(x:\mathbb{S}^1)}\mathsf{code}(x)) \to \mathcal{U}$, ассоциированном с D, получаем элемент $D(\mathsf{base}, n)$ для любого $n : \mathbb{Z}$. Мы определяем этот элемент как $f(\mathsf{base})(n)$:

$$f(\mathsf{base})(n) :\equiv \mathsf{transport}^{\widehat{D}}(\mathsf{pair}^=(\mathsf{loop}^n, p_n), d)\,.$$

Теперь нам нужен $\mathsf{transport}^{\lambda x.\prod_{(c:\mathsf{code}(x))} D(x,c)}(\mathsf{loop}, f(\mathsf{base})) = f(\mathsf{base})$. По лемме 2.9.7 это означает, что нужно показать, что для любого $n : \mathbb{Z}$

$$\mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^=(\mathsf{refl}_{\mathsf{loop}_*(n)}), f(\mathsf{base})(n)\right) =_{D(\mathsf{base},\mathsf{loop}_*(n))} f(\mathsf{base})(\mathsf{loop}_*(n))\,.$$

Теперь имеется путь $q : \mathsf{loop}_*(n) = n + 1$, поэтому, перемещаясь по нему, достаточно показать, что

$$\mathsf{transport}^{D(\mathsf{base})}\left(q, \mathsf{transport}^{\widehat{D}}(\mathsf{pair}^=(\mathsf{refl}_{\mathsf{loop}_*(n)}), f(\mathsf{base})(n))\right) \\ =_{D(\mathsf{base},n+1)} \mathsf{transport}^{D(\mathsf{base})}\left(q, f(base)(\mathsf{loop}_*(n))\right)\,.$$

По леммам о транспортировке и зависимом применении это эквивалентно

$$\mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^=(\mathsf{loop}, q), f(\mathsf{base})(n)\right) =_{D(\mathsf{base},n+1)} f(\mathsf{base})(n+1)\,.$$

Однако, расширяя определение $f(\mathsf{base})$, имеем

$$\begin{aligned}
\mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^{=}(\mathsf{loop},q), f(\mathsf{base})(n)\right) &= \mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^{=}(\mathsf{loop},q), \mathsf{transport}^{\widehat{D}}(\mathsf{pair}^{=}(\mathsf{loop}^{n},p_n),d)\right)\\
&= \mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^{=}(\mathsf{loop}^{n},p_n) \cdot \mathsf{pair}^{=}(\mathsf{loop},q),d\right)\\
&= \mathsf{transport}^{\widehat{D}}\left(\mathsf{pair}^{=}(\mathsf{loop}^{n+1},p_{n+1}),d\right)\\
&= f(\mathsf{base})(n+1)\,.
\end{aligned}$$

Мы использовали функториальность транспортировки, характеризацию композиции в Σ-типах (что было упражнением для читателя) и лемму, связывающую p_n и q с p_{n+1}, которую мы оставляем читателю для формулировки и доказательства.

Это завершает построение $f : \prod_{(x:\mathbb{S}^1)} \prod_{(c:\mathsf{code}(x))} D(x,c)$. Поскольку

$$f(\mathsf{base},0) \equiv \mathsf{pair}^{=}(\mathsf{loop}^{0},p_0)_*(d) = \mathsf{refl}_{\mathsf{base}*}(d) = d,$$

мы показали, что (code, 0) является системой тождественностей. □

Следствие 8.1.16. *Для любого* $x : \mathbb{S}^1$ *имеем* $(\mathsf{base} = x) \simeq \mathsf{code}(x)$.

Доказательство. Следует из теоремы 5.8.2. □

Конечно, это доказательство также содержит по существу те же элементы, что и предыдущие два. Грубо говоря, можно сказать, что оно объединяет доказательства определения 8.1.6 и леммы 8.1.8, проводя необходимое индуктивное рассуждение только один раз в общем случае.

Замечание 8.1.17. Заметьте, что все приведенные выше доказательства того, что $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$ существенно используют аксиому унивалентности. Это неизбежно: унивалентность или что-то подобное необходимо для доказательства $\pi_1(\mathbb{S}^1) \simeq \mathbb{Z}$. При отсутствии унивалентности логично предположить, что утверждение «все типы являются множествами» (также известное как «уникальность доказательств тождественности» или «Аксиома K», как обсуждается в §7.2) означает, что $\pi_1(\mathbb{S}^1) \simeq \mathbf{1}$. Фактически (не)тривиальность $\pi_1(\mathbb{S}^1)$ точно определяет, все ли типы являются множествами: доказательство леммы 6.4.1, наоборот, показало, что если $\mathsf{loop} = \mathsf{refl}_{\mathsf{base}}$, то все типы являются множествами.

8.2 Связность надстроек

Напомним из §7.5, что тип A называется n-**связным**, если $\|A\|_n$ является стягиваемым. Цель этого раздела — доказать, что операция надстройки из §6.5 увеличивает связность.

Теорема 8.2.1. *Если A является n-связным, то надстройка A является (n+1)-связной.*

Доказательство. В §6.8 было отмечено, что надстройка A — это амальгама $\mathbf{1} \sqcup^{A} \mathbf{1}$, поэтому нужно доказать, что следующий тип является стягиваемым:

$$\left\|\mathbf{1} \sqcup^{A} \mathbf{1}\right\|_{n+1}.$$

Из теоремы 7.4.12 известно, что $\left\|\mathbf{1} \sqcup^{A} \mathbf{1}\right\|_{n+1}$ является амальгамой в $(n+1)$-Type диаграммы

$$\begin{tikzcd}
\|A\|_{n+1} \arrow[r] \arrow[d] & \|\mathbf{1}\|_{n+1} \\
\|\mathbf{1}\|_{n+1} &
\end{tikzcd}$$

Учитывая, что $\|\mathbf{1}\|_{n+1} = \mathbf{1}$, тип $\left\|\mathbf{1} \sqcup^A \mathbf{1}\right\|_{n+1}$ также является амальгамой следующей диаграммы в $(n+1)$-Type (поскольку обе диаграммы эквивалентны)

$$\mathscr{D} = \begin{tikzcd} \|A\|_{n+1} \arrow[r] \arrow[d] & \mathbf{1} \\ \mathbf{1} \end{tikzcd}$$

Теперь докажем, что $\mathbf{1}$ также является амальгамой $\mathscr{D}$ в $(n+1)$-Type. Пусть E будет $(n+1)$-усеченным типом; нужно доказать, что следующее отображение является эквивалентностью

$$\begin{cases} (1 \to E) & \longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ \qquad y & \longmapsto (y, y, \lambda u.\, \mathsf{refl}_{y(\star)}) \end{cases}$$

где, напомним, $\mathsf{cocone}_{\mathscr{D}}(E)$ — это тип

$$\sum_{(f:\mathbf{1}\to E)} \sum_{(g:\mathbf{1}\to E)} \left(\|A\|_{n+1} \to (f(\star) =_E g(\star))\right) .$$

Отображение $\begin{cases} (1 \to E) & \longrightarrow \mathsf{cocone}_{\mathscr{D}}(E) \\ \qquad y & \longmapsto (y, y, \lambda u.\, \mathsf{refl}_{y(\star)}) \end{cases}$ является эквивалентностью, поэтому также имеем

$$\mathsf{cocone}_{\mathscr{D}}(E) = \sum_{(x:E)} \sum_{(y:E)} \left(\|A\|_{n+1} \to (x =_E y)\right) .$$

Теперь A является n-связным, а следовательно и $\|A\|_{n+1}$, поскольку $\left\|\|A\|_{n+1}\right\|_n = \|A\|_n = \mathbf{1}$, а $(x =_E y)$ является n-усеченным, потому что E является $(n+1)$-усеченным. Следовательно, по следствию 7.5.9, следующее отображение является эквивалентностью

$$\begin{cases} (x =_E y) & \longrightarrow \left(\|A\|_{n+1} \to (x =_E y)\right) \\ \qquad p & \longmapsto \lambda z.p \end{cases}$$

Следовательно, имеем

$$\mathsf{cocone}_{\mathscr{D}}(E) = \sum_{(x:E)} \sum_{(y:E)} (x =_E y) .$$

Но следующее отображение является эквивалентностью

$$\begin{cases} E & \longrightarrow \Sigma_{(x:E)}\Sigma_{(y:E)}(x =_E y) \\ x & \longmapsto (x, x, \mathsf{refl}_x) \end{cases}$$

Следовательно,

$$\mathsf{cocone}_{\mathscr{D}}(E) = E .$$

В итоге получаем эквивалентность

$$(\mathbf{1} \to E) \simeq \mathsf{cocone}_{\mathscr{D}}(E) .$$

Теперь можно развернуть определения, чтобы получить явное выражение этого отображения, и, как легко видеть, это именно то отображение, которое фигурировало в начале.

Таким образом, доказано, что $\mathbf{1}$ является амальгамой для $\mathscr{D}$ в $(n+1)$-Type. Используя уникальность амальгам, получаем, что $\left\|\mathbf{1} \sqcup^A \mathbf{1}\right\|_{n+1} = \mathbf{1}$, а из этого следует, что надстройка для A является $(n+1)$-связной. $\square$

Следствие 8.2.2. *Для всех $n:\mathbb{N}$ сфера $\mathbb{S}^n$ является $(n-1)$-связной.*

Доказательство. Докажем это индукцией по n. Для $n=0$ надо показать, что $\mathbb{S}^0$ просто обитаем, что верно. Пусть $n:\mathbb{N}$ таково, что $\mathbb{S}^n$ является $(n-1)$-связным. По определению, $\mathbb{S}^{n+1}$ является надстройкой для $\mathbb{S}^n$, следовательно, по предыдущей лемме, $\mathbb{S}^{n+1}$ является n-связным. □

8.3 $\pi_{k\leqslant n}$ *n*-связного пространства и $\pi_{k<n}(\mathbb{S}^n)$

Пусть (A,a) — точечный тип и $n:\mathbb{N}$. Напомним из примера 6.11.4, что если $n>0$, то множество $\pi_n(A,a)$ обладает групповой структурой, а если $n>1$, то эта группа является абелевой.

Теперь можно обсудить гомотопические группы n-усеченных и n-связных типов.

Лемма 8.3.1. *Если A является n-усеченным и $a:A$, то $\pi_k(A,a)=\mathbf{1}$, для всех $k>n$.*

Доказательство. Пространство петель n-типа является $(n-1)$-типом, следовательно, $\Omega^k(A,a)$ — $(n-k)$-тип, и имеем, что $(n-k)\leqslant -1$, так что, $\Omega^k(A,a)$ — это просто предложение. Но $\Omega^k(A,a)$ обитаем, поэтому, на самом деле, он является стягиваемым и $\pi_k(A,a)=\left\|\Omega^k(A,a)\right\|_0=\|\mathbf{1}\|_0=\mathbf{1}$. □

Лемма 8.3.2. *Если A является n-связным и $a:A$, то $\pi_k(A,a)=\mathbf{1}$, для всех $k\leqslant n$.*

Доказательство. Имеется следующая последовательность равенств:

$$\pi_k(A,a)=\left\|\Omega^k(A,a)\right\|_0=\Omega^k\left(\|(A,a)\|_k\right)=\Omega^k\left(\left\|\|(A,a)\|_n\right\|_k\right)=\Omega^k\left(\|\mathbf{1}\|_k\right)=\Omega^k(\mathbf{1})=\mathbf{1}\,.$$

Третье равенство использует то, что $\|-\|_k\circ\|-\|_n=\|-\|_k$ при $k\leqslant n$, а в четвертом равенстве используется n-связность A. □

Следствие 8.3.3. $\pi_k(\mathbb{S}^n)=\mathbf{1}$, *для $k<n$.*

Доказательство. Сфера $\mathbb{S}^n$ является $(n-1)$-связной, согласно следствию 8.2.2, поэтому можно применить лемму 8.3.2. □

8.4 Последовательности слоев и длинная точная последовательность

Если кообласть функции $f:X\to Y$ снабжена отмеченной точкой $y_0:Y$, то мы ссылаемся на слой $F:\equiv \mathsf{fib}_f(y_0)$ для f над y_0 как на **слой для** f (если Y связно, то F определяется с точностью до эквивалентности; см. упражнение 8.5). Теперь покажем, что если X также является точечным и f сохраняет отмеченные точки, то существует связь между гомотопическими группами F, X, и Y в виде *длинной точной последовательности*. Мы выводим это с помощью *последовательности слоев*, ассоциированной с таким f.

Определение 8.4.1. **Точечное отображение** между точечными типами (X,x_0) и (Y,y_0) — это отображение $f:X\to Y$ вместе с путем $f_0:f(x_0)=y_0$.

Для любых точечных типов (X, x_0) и (Y, y_0) существует точечное отображение $(\lambda x. y_0) : X \to Y$, которое постоянно в отмеченной точке. Мы называем это **нулевым отображением** и иногда записываем как $0 : X \to Y$.

Напомним, что каждый точечный тип (X, x_0) имеет пространство петель $\Omega(X, x_0)$. Заметим теперь, что эта операция функториальна на точечных отображениях.

Определение 8.4.2. Для точечного отображения между точечными типами$f : X \to Y$ определим точечное отображение $\Omega f : \Omega X \to \Omega Y$ как

$$(\Omega f)(p) :\equiv f_0^{-1} \cdot f(p) \cdot f_0 .$$

Путь $(\Omega f)_0 : (\Omega f)(\mathsf{refl}_{x_0}) = \mathsf{refl}_{y_0}$, который демонстрирует Ωf как точечное отображение, является очевидным путем типа

$$f_0^{-1} \cdot f(\mathsf{refl}_{x_0}) \cdot f_0 = \mathsf{refl}_{y_0} .$$

На точечных отображениях есть еще один функтор, который переводит $f : X \to Y$ к $\mathsf{pr}_1 : \mathsf{fib}_f(y_0) \to X$. Когда f является точечной, мы считаем, что $\mathsf{fib}_f(y_0)$ будет точечным с отмеченной точкой (x_0, f_0), и в этом случае pr_1 — также точечное отображение, со свидетельством $(\mathsf{pr}_1)_0 :\equiv \mathsf{refl}_{x_0}$. Таким образом, эта операцию может быть итерирована.

Определение 8.4.3. Последовательность слоев точечного отображения $f : X \to Y$ — это бесконечная последовательность точечных типов и точечных отображений

$$\ldots \xrightarrow{f^{(n+1)}} X^{(n+1)} \xrightarrow{f^{(n)}} X^{(n)} \xrightarrow{f^{(n-1)}} \ldots \xrightarrow{f^{(2)}} X^{(2)} \xrightarrow{f^{(1)}} X^{(1)} \xrightarrow{f^{(0)}} X^{(0)}$$

рекурсивно определяемых как

$$X^{(0)} :\equiv Y \qquad X^{(1)} :\equiv X \qquad f^{(0)} :\equiv f$$

и

$$\begin{aligned} X^{(n+1)} &:\equiv \mathsf{fib}_{f^{(n-1)}}(x_0^{(n-1)}) \\ f^{(n)} &:\equiv \mathsf{pr}_1 \qquad\qquad : X^{(n+1)} \to X^{(n)} \end{aligned}$$

где $x_0^{(n)}$ обозначает отмеченную точку $X^{(n)}$, выбранную рекурсивно, как указано выше.

Таким образом, любая смежная пара отображений в этой последовательности слоев имеет вид

$$X^{(n+1)} \equiv \mathsf{fib}_{f^{(n-1)}}(x_0^{(n-1)}) \xrightarrow{f^{(n)} \equiv \mathsf{pr}_1} X^{(n)} \xrightarrow{f^{(n-1)}} X^{(n-1)} .$$

В частности, имеем $f^{(n-1)} \circ f^{(n)} = 0$. Теперь заметим, что типы, встречающиеся в этой последовательности, представляют собой повторяющиеся пространства петель базового пространства Y, пространства расслоений X, и слоя $F :\equiv \mathsf{fib}_f(y_0)$; аналогично для отображений.

Лемма 8.4.4. *Пусть $f : X \to Y$ — точечное отображение точечного пространства. Тогда:*

(i) Слой $f^{(1)} :\equiv \mathsf{pr}_1 : \mathsf{fib}_f(y_0) \to X$ эквивалентен ΩY.

(ii) Аналогично, слой $f^{(2)} : \Omega Y \to \mathsf{fib}_f(y_0)$ эквивалентен ΩX.

(iii) *При этих эквивалентностях точечное отображение $f^{(3)} : \Omega X \to \Omega Y$ отождествляется с точечным отображением $\Omega f \circ (-)^{-1}$.*

Доказательство. Для (i) имеем

$$\begin{aligned}
\mathsf{fib}_f(x_0) &:\equiv \sum_{z:\mathsf{fib}_f(y_0)} (\mathsf{pr}_1(z) = x_0) \\
&\simeq \sum_{(x:X)} \sum_{(p:f(x)=y_0)} (x = x_0) && \text{(упражнение 2.10)} \\
&\simeq (f(x_0) = f(y_0)) && \text{(поскольку } \Sigma_{(x:X)}(x = x_0) \text{ стягиваемо)} \\
&\simeq (y_0 = y_0) && \text{(по } (f_0 \cdot -)) \\
&\equiv \Omega Y\,.
\end{aligned}$$

Прослеживая до конца, мы видим, что эта эквивалентность отсылает $((x, p), q)$ к $f_0^{-1} \cdot f(q^{-1}) \cdot p$, в то время как ее обратная отсылает $r : y_0 = y_0$ к $((x_0, f_0 \cdot r), \mathsf{refl}_{x_0})$. В частности, отмеченная точка $((x_0, f_0), \mathsf{refl}_{x_0})$ из $\mathsf{fib}_{f^{(1)}}(x_0)$ отсылается к $f_0^{-1} \cdot f(\mathsf{refl}_{x_0}^{-1}) \cdot f_0$, которое равно refl_{y_0}. Следовательно, эта эквивалентность — точечное отображение (см. упражнение 8.7). Более того, при этой эквивалентности, $f^{(2)}$ отождествляется с $\lambda r.\, (x_0, f_0 \cdot r) : \Omega Y \to \mathsf{fib}_f(y_0)$.

Пункт (ii) следует немедленно, применением (i) к $f^{(1)}$, вместо f. Поскольку $(f^{(1)})_0 :\equiv \mathsf{refl}_{x_0}$, при этой эквивалентности $f^{(3)}$ отождествляется с отображением $\Omega X \to \mathsf{fib}_{f^{(1)}}(x_0)$, определенным как $s \mapsto ((x_0, f_0), s)$. Таким образом, когда мы компонуем $\mathsf{fib}_{f^{(1)}}(x_0) \simeq \Omega Y$ с предыдущей эквивалентностью, то видим, что s отображается к $f_0^{-1} \cdot f(s^{-1}) \cdot f_0$, по определению $(\Omega f)(s^{-1})$. Мы опускаем доказательство того, что это равенство точечных отображений, а не только функций. □

Таким образом, последовательность слоев от $f : X \to Y$ может быть представлена как

$$\ldots \longrightarrow \Omega^2 X \xrightarrow{\Omega^2 f} \Omega^2 Y \xrightarrow{-\Omega\partial} \Omega F \xrightarrow{-\Omega i} \Omega X \xrightarrow{-\Omega f} \Omega Y \xrightarrow{\partial} F \xrightarrow{i} X \xrightarrow{f} Y\,.$$

где минус обозначает композицию с инверсией пути $(-)^{-1}$. Обратите внимание, что по упражнению 8.6, имеет место

$$\Omega\left(\Omega f \circ (-)^{-1}\right) \circ (-)^{-1} = \Omega^2 f \circ (-)^{-1} \circ (-)^{-1} = \Omega^2 f\,.$$

Таким образом, на k-кратных отображениях петель имеются знаки «$-$», когда k нечетно.

Из этой последовательности слоев выведем *точную последовательность точечных множеств*. Пусть A и B — множества, а $f : A \to B$ - функция, и напомним из определения 7.6.3 определение *образа* $\mathsf{im}(f)$, который можно рассматривать как подмножество в B:

$$\mathsf{im}(f) :\equiv \{b : B \mid \exists (a : A).\, f(a) = b\}\,.$$

Если A и B, кроме того, указаны с помощью отмеченных точек a_0 и b_0, а f — это точечное отображение, мы определяем **ядро** f как следующее подмножество в A:

$$\mathsf{ker}(f) :\equiv \{x : A \mid f(x) = b_0\}\,.$$

Конечно, это просто слой f над отмеченной точкой b_0; это подмножество в A, потому что B — множество.

Заметьте, что любая группа является точечным множеством с единичным элементом в качестве отмеченной точки, а любой гомоморфизм группы является точечным отображением. В этом случае ядро и образ согласуются с общепринятой нотацией теории групп.

Определение 8.4.5. Точная последовательность точечных множеств — это (возможно, ограниченная) последовательность точечных множеств и точечных отображений:

$$\ldots \xrightarrow{f^{(n+1)}} A^{(n+1)} \xrightarrow{f^{(n)}} A^{(n)} \xrightarrow{f^{(n-1)}} A^{(n-1)} \xrightarrow{f^{(n-2)}} \ldots$$

такая, что, для каждого n, образ $f^{(n)}$, как подмножество в $A^{(n)}$, равен ядру $f^{(n-1)}$. Другими словами, для всех $a : A^{(n)}$, имеем

$$(f^{(n-1)}(a) = a_0^{(n-1)}) \iff \exists(b : A^{(n+1)}).\,(f^{(n)}(b) = a)\,.$$

где $a_0^{(n)}$ обозначает отмеченную точку в $A^{(n)}$.

Обычно, большинство или все точечные множества в точной последовательности являются группами, и часто — абелевыми группами. Когда говорят о **точной последовательности групп**, предполагается, что, кроме того, отображения являются гомоморфизмами групп, а не просто точечными отображениями.

Теорема 8.4.6. *Пусть $f : X \to Y$ — точечное отображение между точечными пространствами со слоем $F :\equiv \mathsf{fib}_f(y_0)$. Тогда имеется следующая длинная точная последовательность, состоящая из групп, за исключением последних трех членов, и абелевых групп, за исключением последних шести.*

$$\begin{tikzcd}
\vdots & \vdots & \vdots \arrow[dll] \\
\pi_k(F) \arrow[r] & \pi_k(X) \arrow[r] & \pi_k(Y) \arrow[dll] \\
\vdots & \vdots & \vdots \arrow[dll] \\
\pi_2(F) \arrow[r] & \pi_2(X) \arrow[r] & \pi_2(Y) \arrow[dll] \\
\pi_1(F) \arrow[r] & \pi_1(X) \arrow[r] & \pi_1(Y) \arrow[dll] \\
\pi_0(F) \arrow[r] & \pi_0(X) \arrow[r] & \pi_0(Y)
\end{tikzcd}$$

Доказательство. Начнем с того, что покажем, что 0-усечение последовательности слоев является точной последовательностью точечных множеств. Так что, нужно показать, что для любой смежной пары отображений в последовательности слоев:

$$\mathsf{fib}_f(z_0) \xrightarrow{g} W \xrightarrow{f} Z$$

с $g :\equiv \mathsf{pr}_1$, последовательность

$$\|\mathsf{fib}_f(z_0)_0\| \xrightarrow{\|g\|_0} \|W\|_0 \xrightarrow{\|f\|_0} \|Z\|_0$$

существует, т.е., что $\mathsf{im}\,(\|g\|_0) \subseteq \mathsf{ker}\,(\|f\|_0)$ и $\mathsf{ker}\,(\|f\|_0) \subseteq \mathsf{im}\,(\|g\|_0)$.

Первое включение эквивалентно $\|g\|_0 \circ \|f\|_0 = 0$, которое выполняется в силу функториальности $\|-\|_0$ и того факта, что $g \circ f = 0$. Для второго, предположим, что $w' : \|W\|_0$ и

$p' : \|f\|_0(w') = |z_0|_0$ и покажем, что существует $t : \mathsf{fib}_f(z_0)$ такое, что $g(t) = w'$. Поскольку наша цель — простое высказывание, можно предположить, что w' имеет вид $|w|_0$ для некоторого $w : W$. Теперь, по теореме 7.3.12, $p' : |f(w)|_0 = |z_0|_0$ влечет $p'' : \|f(w) = z_0\|_{-1}$, поэтому с помощью дальнейшей индукции усечения можно предположить, что $p : f(w) = z_0$. Но теперь имеется $|(w,p)|_0 : |\mathsf{fib}_f(z_0)|_0$, образ которого под $\|g\|_0$ равен $|w|_0 \equiv w'$, что и требовалось.

Таким образом, применяя $\|-\|_0$ к последовательности слоев f, получаем длинную точную последовательность, включающую точечные множества $\pi_k(F)$, $\pi_k(X)$ и $\pi_k(Y)$ в требуемом порядке. И, конечно же, π_k — это группа для $k \geqslant 1$, являющаяся 0-усечением пространства петель, и абелева группа для $k \geqslant 2$ по аргументу Экмана-Хилтона (теорема 2.1.6). Более того, лемма 8.4.4 позволяет идентифицировать отображения $\pi_k(F) \to \pi_k(X)$ и $\pi_k(X) \to \pi_k(Y)$ в этой точной последовательности, как $(-1)^k\pi_k(i)$ и $(-1)^k\pi_k(f)$, соответственно.

В более широком смысле, каждое отображение в этой длинной точной последовательности, кроме последних трех, имеет вид $\|\Omega h\|_0$ или $\|-\Omega h\|_0$ для некоторых h. В первом случае это гомоморфизм групп, а во втором — гомоморфизм, если группы абелевы; в других отношениях это «антигомоморфизм». Однако, ядро и образ гомоморфизма групп не изменяются, когда происходит замена его отрицательным, и, следовательно, это точность любой последовательности, включающей такое. Так что, можно изменить нашу длинную точную последовательность, чтобы получить последовательность, включающую непосредственно $\pi_k(i)$ и $\pi_k(f)$, в которой все отображения являются гомоморфизмами групп (кроме последних трех). $\square$

Обычные свойства точных последовательностей абелевых групп доказываются как обычно. В частности, имеем:

Лемма 8.4.7. *Предположим, что дана точная последовательность абелевых групп:*

$$K \longrightarrow G \xrightarrow{f} H \longrightarrow Q\,.$$

(i) *Если $K = 0$, то f — инъекция.*

(ii) *Если $Q = 0$, то f — сюръекция.*

(iii) *Если $K = Q = 0$, то f — изоморфизм.*

Доказательство. Поскольку ядро f является образом $K \to G$, если $K = 0$, то ядро f есть $\{0\}$; следовательно, f инъективна, потому что это морфизм групп. Точно так же, поскольку образ f является ядром $H \to Q$, если $Q = 0$, то образ f — это H целиком, поэтому f сюръективна. Наконец, (iii) следует из (i) и (ii) по теореме 4.6.3. $\square$

В качестве непосредственного приложения теперь можно количественно определить, каким образом n-связность отображения сильнее, чем индуцирование эквивалентности на n-усечениях.

Следствие 8.4.8. *Пусть $f : A \to B$ — n-связна, $a : A$ и $b :\equiv f(a)$. Тогда*:

(i) *Если $k \leqslant n$, то $\pi_k(f) : \pi_k(A,a) \to \pi_k(B,b)$ — изоморфизм.*

(ii) *Если $k = n+1$, то $\pi_k(f) : \pi_k(A,a) \to \pi_k(B,b)$ — сюръекция.*

Доказательство. При $k = 0$, пункт (i) следует из леммы 7.5.14, замечая при этом, что $\pi_0(f) \equiv \|f\|_0$. Для $k = 0$ пункт (ii) следует из упражнения 7.18, замечая при этом, что функция сюръективна тогда и только тогда, когда она (-1)-связна по лемме 7.5.2. При $k > 0$ в составе длинной точной последовательности имеется точная последовательность

$$\pi_k\left(\mathsf{fib}_f(b)\right) \longrightarrow \pi_k\left(A, a\right) \xrightarrow{f} \pi_k\left(B, b\right) \longrightarrow \pi_{k-1}\left(\mathsf{fib}_f(b)\right) .$$

Теперь, поскольку f является n-связным, $\left\|\mathsf{fib}_f(b)\right\|_n$ стягиваемо. Следовательно, если $k \leqslant n$, то $\pi_k(\mathsf{fib}_f(b)) = \left\|\Omega^k(\mathsf{fib}_f(b))\right\|_0 = \Omega^k(\left\|\mathsf{fib}_f(b)\right\|_k)$ также стягиваемо. Таким образом, $\pi_k(f)$ является изоморфизмом для $k \leqslant n$ по лемме 8.4.7(iii), а для $k = n+1$ это сюръекция по лемме 8.4.7(ii). □

В §8.8 мы увидим, что верно и обратное к следствию 8.4.8.

8.5 Расслоение Хопфа

В этом разделе мы определим **расслоение Хопфа**.

Теорема 8.5.1 (Расслоение Хопфа)**.** *Существует расслоение H над $\mathbb{S}^2$, слой над отмеченной точкой которого есть $\mathbb{S}^1$, а пространство расслоений которого — $\mathbb{S}^3$.*

Расслоение Хопфа позволит вычислить несколько гомотопических групп сфер. Действительно, оно дает следующую длинную точную последовательность гомотопических групп (см. §8.4):

$$\begin{tikzcd}
\pi_k(\mathbb{S}^1) \arrow[r] & \pi_k(\mathbb{S}^3) \arrow[r] & \pi_k(\mathbb{S}^2) \arrow[dll] \\
\vdots & \vdots & \vdots \arrow[dll] \\
\pi_2(\mathbb{S}^1) \arrow[r] & \pi_2(\mathbb{S}^3) \arrow[r] & \pi_2(\mathbb{S}^2) \arrow[dll] \\
\pi_1(\mathbb{S}^1) \arrow[r] & \pi_1(\mathbb{S}^3) \arrow[r] & \pi_1(\mathbb{S}^2)
\end{tikzcd}$$

Мы уже вычислили все $\pi_n(\mathbb{S}^1)$, и $\pi_k(S^n)$ для $k < n$, поэтому получается следующее:

$$\begin{tikzcd}
0 \arrow[r] & \pi_k(\mathbb{S}^3) \arrow[r] & \pi_k(\mathbb{S}^2) \arrow[dll] \\
\vdots & \vdots & \vdots \arrow[dll] \\
0 \arrow[r] & \pi_3(\mathbb{S}^3) \arrow[r] & \pi_3(\mathbb{S}^2) \arrow[dll] \\
0 \arrow[r] & 0 \arrow[r] & \pi_2(\mathbb{S}^2) \arrow[dll] \\
\mathbb{Z} \arrow[r] & 0 \arrow[r] & 0
\end{tikzcd}$$

В частности, получаем следующий результат:

Следствие 8.5.2. *Имеем $\pi_2(\mathbb{S}^2) \simeq \mathbb{Z}$ и $\pi_k(\mathbb{S}^3) \simeq \pi_k(\mathbb{S}^2)$ для каждого $k \geqslant 3$ (где отображение индуцировано расслоением Хопфа, рассматриваемым как отображение пространства расслоений $\mathbb{S}^3$ в базовое пространство $\mathbb{S}^2$).*

Фактически, можно сказать больше: последовательность слоев расслоения Хопфа покажет, что $\Omega^3(\mathbb{S}^3)$ является слоем отображения от $\Omega^3(\mathbb{S}^2)$ к $\Omega^2(\mathbb{S}^1)$. Поскольку $\Omega^2(\mathbb{S}^1)$ является стягиваемым, то имеем $\Omega^3(\mathbb{S}^3) \simeq \Omega^3(\mathbb{S}^2)$. В классической теории гомотопий этот факт был бы заключением следствия 8.5.2 и теоремы Уайтхеда, но теорема Уайтхеда не обязательно верна в теории гомотопических типов (see §8.8). Однако мы не будем использовать здесь более точную версию.

8.5.1 Расслоения над амальгамами

Начнем с леммы, поясняющей, как строить расслоения над амальгамами.

Лемма 8.5.3. *Пусть $\mathscr{D} = (Y \xleftarrow{j} X \xrightarrow{k} Z)$ — пролет, и предположим, что имеются*

- *Расслоения $E_Y : Y \to \mathcal{U}$ и $E_Z : Z \to \mathcal{U}$.*
- *Эквивалентность e_X между $E_Y \circ j : X \to \mathcal{U}$ и $E_Z \circ k : X \to \mathcal{U}$, т.е.*

$$e_X : \prod_{x:X} E_Y(j(x)) \simeq E_Z(k(x)) \,.$$

Тогда можно построить расслоение $E : Y \sqcup^X Z \to U$ такое, что

- *Для всех $y : Y$, $E(\mathsf{inl}(y)) \equiv E_Y(y)$.*
- *Для всех $z : Z$, $E(\mathsf{inr}(z)) \equiv E_Z(z)$.*
- *Для всех $x : X$, $E(\mathsf{glue}(x)) = \mathsf{ua}(e_X(x))$ (заметьте, что обе стороны уравнения содержат пути из $\mathcal{U}$ от $E_Y(j(x))$ к $E_Z(k(x))$).*

Более того, пространство этого расслоения умещается в следующем квадрате амальгамы:

$$\begin{tikzcd}
\sum_{(x:X)} E_Y(j(x)) \arrow[r, "\mathsf{id}\times e_X", "\sim"'] \arrow[d, "j\times\mathsf{id}"'] & \sum_{(x:X)} E_Z(k(x)) \arrow[r, "k\times\mathsf{id}"] & \sum_{(z:Z)} E_Z(z) \arrow[d, "\mathsf{inr}"] \\
\sum_{(y:Y)} E_Y(y) \arrow[rr, "\mathsf{inl}"'] & & \sum_{(t:Y\sqcup^X Z)} E(t)
\end{tikzcd}$$

Доказательство. Определим E по принципу рекурсии амальгамы $Y \sqcup^X Z$. Для этого нужно указать значение E на элементах $\mathsf{inl}(y)$, $\mathsf{inr}(z)$ этой формы и действие E на путях $\mathsf{glue}(x)$, поэтому можно просто выбрать следующие значения:

$$\begin{aligned}
E(\mathsf{inl}(y)) &:\equiv E_Y(y) \,, \\
E(\mathsf{inr}(z)) &:\equiv E_Z(z) \,, \\
E(\mathsf{glue}(x)) &:\equiv \mathsf{ua}(e_X(x)) \,.
\end{aligned}$$

Чтобы понять, что пространство этого расслоения является амальгамой, применим лемму сглаживания (лемма 6.12.2) со следующими значениями:

- $A :\equiv Y + Z$, $B :\equiv X$ и $f, g : B \to A$, определенные как $f(x) :\equiv \mathsf{inl}(j(x))$, $g(x) :\equiv \mathsf{inr}(k(x))$,
- семейство типов $C : A \to \mathcal{U}$, определенное посредством

$$C(\mathsf{inl}(y)) :\equiv E_Y(y) \qquad \text{и} \qquad C(\mathsf{inr}(z)) :\equiv E_Z(z)\,,$$

- семейство эквивалентностей $D : \prod_{(b:B)} C(f(b)) \simeq C(g(b))$, определенное как e_X.

Базовый высший индуктивный тип W в лемме сглаживания эквивалентен амальгаме $Y \sqcup^X Z$, а семейство типов $P : Y \sqcup^X Z \to \mathcal{U}$ эквивалентно E, определенному выше.

Таким образом, лемма сглаживания утверждает, что $\sum_{(t:Y\sqcup^X Z)} E(t)$ эквивалентен высшему индуктивному типу $E^{\mathsf{tot}\prime}$ со следующими образующими:

- функцией $\mathsf{z} : \sum_{(a:Y+Z)} C(a) \to E^{\mathsf{tot}\prime}$,
- для каждого $x : X$ и $t : E_Y(j(x))$, путем $\mathsf{z}(\mathsf{inl}(j(x)), t) = \mathsf{z}(\mathsf{inr}(k(x)), e_X(t))$.

Снова используя лемму сглаживания или прямое вычисление, легко увидеть, что $\sum_{(a:Y+Z)} C(a) \simeq \sum_{(y:Y)} E_Y(y) + \sum_{(z:Z)} E_Z(z)$, следовательно, $E^{\mathsf{tot}\prime}$ эквивалентен высшему индуктивному типу E^{tot} с образующими:

- функцией $\mathsf{inl} : \sum_{(y:Y)} E_Y(y) \to E^{\mathsf{tot}}$,
- функцией $\mathsf{inr} : \sum_{(z:Z)} E_Z(z) \to E^{\mathsf{tot}}$,
- для каждой пары $(x, t) : \sum_{(x:X)} E_Y(j(x))$, путем $\mathsf{glue}(x, t) : \mathsf{inl}(j(x), t) = \mathsf{inr}(k(x), e_X(t))$.

Таким образом, пространство расслоений E является амальгамой пространств расслоений E_Y и E_Z, что и требовалось. □

8.5.2 Конструкция Хопфа

Определение 8.5.4. H-пространство состоит из

- типа A,
- отмеченной точки $e : A$,
- бинарной операции $\mu : A \times A \to A$ и
- для каждого $a : A$, равенства $\mu(e, a) = a$ и $\mu(a, e) = a$.

Лемма 8.5.5. *Пусть A — связное* H-*пространство. Тогда для каждого $a : A$, отображения $\mu(a, -) : A \to A$ и $\mu(-, a) : A \to A$ являются эквивалентностями.*

Доказательство. Докажем, что для любого $a : A$ отображение $\mu(a, -)$ является эквивалентностью. Другое утверждение симметрично. Утверждение, что $\mu(a, -)$ является эквивалентностью, соответствует семейству типов $P : A \to \mathsf{Prop}$, и его доказательство соответствует поиску сечения этого семейства типов.

Тип Prop — это множество (теорема 7.1.11), поэтому можно определить новое семейство типов $P' : \|A\|_0 \to \mathsf{Prop}$ согласно $P'(|a|_0) :\equiv P(a)$. Но A связно по предположению, поэтому $\|A\|_0$ стягиваемо. Это означает, что для того, чтобы найти сечение P', достаточно найти точку в слое P' над $|e|_0$. Но у нас есть $P'(|e|_0) = P(e)$, который населен, потому что $\mu(e, -)$ равен тождественному отображению по определению H-пространства, следовательно, это эквивалентность.

Мы доказали, что для любого $x : \|A\|_0$ предложение $P'(x)$ истинно, поэтому, в частности, для каждого $a : A$ предложение $P(a)$ истинно, потому что $P(a)$ есть $P'(|a|_0)$. □

Определение 8.5.6. Пусть A — связное H-пространство. Определим расслоение над ΣA, используя лемму 8.5.3.

Учитывая, что ΣA является амальгамой $\mathbf{1} \sqcup^A \mathbf{1}$, можно определить расслоение над ΣA, указав

- два расслоения над $\mathbf{1}$ (т.е. два типа F_1 и F_2), и
- семейство $e : A \to (F_1 \simeq F_2)$ эквивалентностей между F_1 и F_2, по одному на каждый элемент из A.

Будем использовать A вместо F_1 и F_2, а в качестве $a : A$ — эквивалентность $\mu(a, -)$ для $e(a)$.

Согласно лемме 8.5.3 имеем следующую диаграмму:

$$\begin{tikzcd}
A & A \times A \arrow[l, "\mathsf{pr}_2"'] \arrow[r, "\mu"] \arrow[d, "\mathsf{pr}_1"'] & A \arrow[d] \\
\mathbf{1} \arrow[from=u] & A \arrow[l] \arrow[r] & \mathbf{1}
\end{tikzcd}$$

а только что построенное расслоение является расслоением над ΣA, пространство расслоений которого является амальгамой верхней строки этой диаграммы.

Более того, с $f(x, y) :\equiv (\mu(x, y), y)$ имеем следующую диаграмму:

$$\begin{tikzcd}
A \arrow[d, "\mathsf{id}"'] & A \times A \arrow[l, "\mathsf{pr}_2"'] \arrow[r, "\mu"] \arrow[d, "f"] & A \arrow[d, "\mathsf{id}"] \\
A & A \times A \arrow[l, "\mathsf{pr}_2"] \arrow[r, "\mathsf{pr}_1"'] & A
\end{tikzcd}$$

Эта диаграмма коммутативна, а ее три вертикальных отображения являются эквивалентностями; обратной к f является функция g , определенная формулой

$$g(u, v) :\equiv (\mu(-, v)^{-1}(u), v)\,.$$

Это показывает, что две строки на последней диаграмме эквивалентны (следовательно, равны) пролетам, поэтому пространство расслоения, которое было построено, эквивалентно амальгаме нижней строки. И, по определению, эта амальгама является *соединением* A с самим собой (см. §6.8). Таким образом, доказана:

Лемма 8.5.7. *Для связного* H*-пространства* A *существует расслоение, называемое* ***конструкцией Хопфа,*** *над* ΣA *со слоем* A *и пространством расслоений* $A * A$.

8.5.3 Расслоение Хопфа

Сначала построим структуру H-пространства на окружности $\mathbb{S}^1$, отсюда, по лемме 8.5.7, получим расслоение над $\mathbb{S}^2$ со слоем $\mathbb{S}^1$ и пространство расслоений $\mathbb{S}^1 * \mathbb{S}^1$. Затем докажем, что это соединение эквивалентно $\mathbb{S}^3$.

Лемма 8.5.8. *На окружности $\mathbb{S}^1$ существует структура* H-*пространства.*

Доказательство. В качестве отмеченной точки структуры H-пространства мы выберем base. Теперь нужно определить операцию умножения $\mu : \mathbb{S}^1 \times \mathbb{S}^1 \to \mathbb{S}^1$. Определим каррированную форму $\widetilde{\mu} : \mathsf{S}^1 \to (\mathbb{S}^1 \to \mathbb{S}^1)$ для μ рекурсией на $\mathbb{S}^1$:

$$\widetilde{\mu}(\mathsf{base}) :\equiv \mathsf{id}_{\mathbb{S}^1} \qquad \text{и} \qquad \widetilde{\mu}(\mathsf{loop}) :\equiv \mathsf{funext}(h)\,.$$

где $h : \prod_{(x:\mathbb{S}^1)}(x = x)$ — функция, определенная в лемме 6.4.2, со свойством $h(\mathsf{base}) :\equiv \mathsf{loop}$.

Теперь просто нужно доказать, что $\mu(x, \mathsf{base}) = \mu(\mathsf{base}, x) = x$ для каждого $x : \mathbb{S}^1$. По определению, если $x : \mathbb{S}^1$, имеем $\mu(\mathsf{base}, x) = \widetilde{\mu}(\mathsf{base})(x) = \mathsf{id}_{\mathbb{S}^1}(x) = x$. Для равенства $\mu(x, \mathsf{base}) = x$ проведем доказательство индукцией по $x : \mathbb{S}^1$:

- Если x есть base, то $\mu(\mathsf{base}, \mathsf{base}) = \mathsf{base}$ по определению, так что имеем

$$\mathsf{refl}_{\mathsf{base}} : \mu(\mathsf{base}, \mathsf{base}) = \mathsf{base}\,.$$

- Когда x изменяется вдоль loop, докажем, что

$$\mathsf{refl}_{\mathsf{base}} \cdot \mathsf{ap}_{\lambda x.\,x}(\mathsf{loop}) = \mathsf{ap}_{\lambda x.\,\mu(x,\mathsf{base})}(\mathsf{loop}) \cdot \mathsf{refl}_{\mathsf{base}}\,.$$

  Левая часть равна loop, и для правой части имеем:

$$\begin{aligned}
\mathsf{ap}_{\lambda x.\,\mu(x,\mathsf{base})}(\mathsf{loop}) \cdot \mathsf{refl}_{\mathsf{base}} &= \mathsf{ap}_{\lambda x.\,(\widetilde{\mu}(x))(\mathsf{base})}(\mathsf{loop}) \\
&= \mathsf{happly}(\mathsf{ap}_{\lambda x.\,(\widetilde{\mu}(x))}(\mathsf{loop}), \mathsf{base}) \\
&= \mathsf{happly}(\mathsf{funext}(h), \mathsf{base}) \\
&= h(\mathsf{base}) \\
&= \mathsf{loop}\,.
\end{aligned}$$

□

Теперь напомним из §6.8, что *соединение* $A*B$ типов A и B является амальгамой диаграммы

$$A \xleftarrow{\mathsf{pr}_1} A \times B \xrightarrow{\mathsf{pr}_2} B\,.$$

Лемма 8.5.9. *Операция соединения ассоциативна: если* A, B *и* C *—- типы, то имеется эквивалентность* $(A * B) * C \simeq A * (B * C)$.

Доказательство. Определим отображение $f : (A * B) * C \to A * (B * C)$ по индукции. Сначала определяем $f \circ \mathsf{inl} : A*B \to A*(B*C)$, что выполняется по индукции, тогда $f \circ \mathsf{inr} : C \to A*(B*C)$, затем $\mathsf{ap}_f \circ \mathsf{glue} : \prod_{(t:(A*B)\times C)} f(\mathsf{inl}(\mathsf{pr}_1(t))) = f(\mathsf{inr}(\mathsf{pr}_2(t)))$, что устанавливается индукцией по первой первой компоненте t:

$$\begin{aligned}
(f \circ \mathsf{inl})(\mathsf{inl}(a)) &:\equiv \mathsf{inl}(a)\,, \\
(f \circ \mathsf{inl})(\mathsf{inr}(b)) &:\equiv \mathsf{inr}(\mathsf{inl}(b))\,, \\
\mathsf{ap}_{f\circ\mathsf{inl}}(\mathsf{glue}(a, b)) &:= \mathsf{glue}(a, \mathsf{inl}(b))\,, \\
f(\mathsf{inr}(c)) &:\equiv \mathsf{inr}(\mathsf{inr}(c))\,, \\
\mathsf{ap}_f(\mathsf{glue}(\mathsf{inl}(a), c)) &:= \mathsf{glue}(a, \mathsf{inr}(c))\,, \\
\mathsf{ap}_f(\mathsf{glue}(\mathsf{inr}(b), c)) &:= \mathsf{ap}_{\mathsf{inr}}(\mathsf{glue}(b, c))\,, \\
\mathsf{apd}_{\lambda x.\,\mathsf{ap}_f(\mathsf{glue}(x,c))}(\mathsf{glue}(a, b)) &:= \text{"}\mathsf{apd}_{\lambda x.\,\mathsf{glue}(a,x)}(\mathsf{glue}(b, c))\text{"}\,.
\end{aligned}$$

Обратите внимание, что правая часть последнего уравнения имеет тип

$$\mathsf{transport}^{\lambda x.\,\mathsf{inl}(a)=\mathsf{inr}(x)}(\mathsf{glue}(b,c),\mathsf{glue}(a,\mathsf{inl}(b)))=\mathsf{glue}(a,\mathsf{inr}(c))$$

тогда как она должна быть типа

$$\mathsf{transport}^{\lambda x.\,f(\mathsf{inl}(a))=f(\mathsf{inr}(x))}(\mathsf{glue}(a,b),\mathsf{ap}_f(\mathsf{glue}(\mathsf{inl}(a),c)))=\mathsf{ap}_f(\mathsf{glue}(\mathsf{inr}(b),c))\,.$$

Но согласно предыдущим пунктам, определения обоих этих типов эквивалентны следующему типу:

$$\mathsf{glue}(a,\mathsf{inr}(c))=\mathsf{glue}(a,\mathsf{inl}(b))\cdot\mathsf{ap}_{\mathsf{inr}}(\mathsf{glue}(b,c))\,,$$

и поэтому можно принудительно получить необходимый элемент с помощью эквивалентности. Аналогично можно определить и отображение $g : A*(B*C)\to(A*B)*C$; проверка того, что f и g обратны друг другу, является рутинным и утомительным, но, по существу, простым действием. □

Более концептуальный набросок доказательства выглядит следующим образом.

Доказательство. Рассмотрим следующую диаграмму, где отображения являются очевидными проекциями:

$$\begin{tikzcd}
A & A\times C \arrow[l] \arrow[r] & A\times C \\
A\times B \arrow[u] \arrow[d] & A\times B\times C \arrow[l] \arrow[r] \arrow[u] \arrow[d] & A\times C \arrow[u] \arrow[d] \\
B & B\times C \arrow[l] \arrow[r] & C
\end{tikzcd}$$

Взятие копредела столбцов дает следующую диаграмму, копредел которой есть $(A*B)*C$:

$$A*B \longleftarrow (A*B)\times C \longrightarrow C$$

С другой стороны, взятие копредела строк приводит к диаграмме, копредел которой равен $A*(B*C)$.

Следовательно, используя теорему типа Фубини для копределов (которую мы пока не доказали, поскольку ее доказательство требует долгих и утомительных вычислений), получаем эквивалентность $(A*B)*C\simeq A*(B*C)$. □

Лемма 8.5.10. *Для любого типа A, существует эквивалентность $\Sigma A\simeq\mathbf{2}*A$.*

Доказательство. Просто определить два отображения back и forth (назад и вперед) и доказать, что они обратны друг другу. Проработка деталей оставлена читателю в качестве упражнения. □

Теперь можно построить расслоение Хопфа.

Теорема 8.5.11. *Существует расслоение над $\mathbb{S}^2$ со слоем $\mathbb{S}^1$ и пространством расслоений $\mathbb{S}^3$.*

Доказательство. Доказано, что $\mathbb{S}^1$ имеет структуру H-пространства (см. лемму 8.5.8), поэтому по лемме 8.5.7 существует расслоение над $\mathbb{S}^2$ со слоем $\mathbb{S}^1$ и пространством расслоений $\mathbb{S}^1*\mathbb{S}^1$. Но по двум предыдущим результатам и лемме 6.5.1 имеем:

$$\mathbb{S}^1*\mathbb{S}^1=(\Sigma\mathbf{2})*\mathbb{S}^1=(\mathbf{2}*\mathbf{2})*\mathbb{S}^1=\mathbf{2}*(\mathbf{2}*\mathbb{S}^1)=\Sigma(\Sigma\mathbb{S}^1)=\mathbb{S}^3\,.$$

□

8.6 Теорема Фрейденталя о надстройке

Прежде чем доказывать теорему Фрейденталя о надстройке, потребуются вспомогательные леммы о связности. В главе 7 доказаны утверждения о n-связных отображениях и n-типах для фиксированного n; теперь нас интересует, что происходит, когда мы меняем n. Например, в лемме 7.5.7 было показано, что n-связные отображения характеризуются «принципом индукции» относительно семейств n-типов. Если мы хотим «индуцировать вдоль» n-связное отображение в семейство k-типов для $k > n$, то не сразу узнаем, что существует функция по такому принципу индукции, но следующая лемма говорит, что по крайней мере уровень нашего незнания можно измерить.

Лемма 8.6.1. *Если $f : A \to B$ является n-связной, а $P : B \to k\text{-}\mathsf{Type}$ — это семейство k-типов для $k \geqslant n$, то индуцированная функция*

$$(-\circ f) : \left(\prod_{(b:B)} P(b)\right) \to \left(\prod_{(a:A)} P(f(a))\right)$$

является $(k-n-2)$-усеченной.

Доказательство. Проведем индукцию по натуральному числу $k-n$. При $k=n$ — это лемма 7.5.7. Для индуктивного шага предположим, что f является n-связной, а P — семейство $(k+1)$-типов. Для демонстрации того, что $(-\circ f)$ является $(k-n-1)$-усеченной, положим $\ell : \prod_{(a:A)} P(f(a))$; тогда имеем

$$\mathsf{fib}_{(-\circ f)}(\ell) \simeq \sum_{(g:\prod_{(b:B)} P(b))} \; \prod_{(a:A)} g(f(a)) = \ell(a)\,.$$

Пусть (g,p) и (h,q) принадлежат этому типу, поэтому $p : g\circ f \sim \ell$ и $q : h \circ f \sim \ell$; тогда также имеем

$$\big((g,p) = (h,q)\big) \simeq \left(\sum_{(r:g\sim h)} r\circ f = p \centerdot q^{-1}\right)\,.$$

Однако, здесь правая часть — это слой отображения

$$(-\circ f) : \left(\prod_{(b:B)} Q(b)\right) \to \left(\prod_{(a:A)} P(f(a))\right)$$

где $Q(b) :\equiv (g(b) = h(b))$. Поскольку P — семейство $(k+1)$-типов, Q — семейство of k-типов, то индуктивная гипотеза подразумевает, что этот слой является $(k-n-2)$-типом. Таким образом, все пространства путей $\mathsf{fib}_{(\ \circ f)}(\ell)$ являются $(k-n-2)$-типами, поэтому — это $(k-n-1)$-тип. □

Напомним, что если (A,a_0) и (B,b_0) являются точечными типами, то их **клин** $A \vee B$ определяется как амальгама для $A \xleftarrow{a_0} \mathbf{1} \xrightarrow{b_0} B$. Существует каноническое отображение $i : A \vee B \to A \times B$, определенное двумя отображениями $\lambda a.\,(a, b_0)$ и $\lambda b.\,(a_0, b)$; следующая лемма по существу говорит, что это отображение сильно связно, если таковыми являются A и B. Однако, это будет удобнее, и доказывать, и использовать, если мы воспользуемся характеризацией связности из леммы 7.5.7 и универсальным свойством клина (обобщенным на семейства типов).

Лемма 8.6.2 (Лемма о связности клина)**.** *Предположим, что* (A, a_0) *и* (B, b_0) *являются* n*- и* m*-связными точечными типами, соответственно, с* $n, m \geqslant 0$*, и пусть* $P : A \to B \to (n+m)\text{-}\mathsf{Type}$*. Тогда, для любых* $f : \prod_{(a:A)} P(a, b_0)$ *и* $g : \prod_{(b:B)} P(a_0, b)$ *с* $p : f(a_0) = g(b_0)$*, существует* $h : \prod_{(a:A)} \prod_{(b:B)} P(a, b)$ *с гомотопиями*

$$q : \prod_{(a:A)} h(a, b_0) = f(a) \qquad \text{и} \qquad r : \prod_{(b:B)} h(a_0, b) = g(b)$$

такими, что $p = q(a_0)^{-1} \cdot r(b_0)$.

Доказательство. Определим $Q : A \to \mathcal{U}$ как

$$Q(a) :\equiv \sum_{k:\prod_{(b:B)} P(a,b)} (f(a) = k(b_0)) \,.$$

Тогда имеем $(g, p) : Q(a_0)$. Поскольку $a_0 : \mathbf{1} \to A$ — $(n-1)$-связно, если Q — семейство $(n-1)$-типов, то будем иметь $\ell : \prod_{(a:A)} Q(a)$, так что $\ell(a_0) = (g, p)$, и в этом случае можно определить $h(a, b) :\equiv \mathsf{pr}_1(\ell(a))(b)$. Однако, при фиксированном a, тип $Q(a)$ является слоем над $f(a)$ отображения

$$\left(\prod_{(b:B)} P(a, b)\right) \to P(a, b_0)$$

заданного пред-композицией с $b_0 : \mathbf{1} \to B$. Поскольку $b_0 : \mathbf{1} \to B$ является $(m-1)$-связным, чтобы этот слой был $(n-1)$-усеченным, по лемме 8.6.1 достаточно, чтобы каждый тип $P(a, b)$ был $(n+m)$-типом, который и был принят. □

Пусть (X, x_0) — точечный тип, и напомним определение надстройки ΣX из §6.5, с конструкторами $\mathsf{N}, \mathsf{S} : \Sigma X$ и $\mathsf{merid} : X \to (\mathsf{N} = \mathsf{S})$. Рассмотрим ΣX как точечное пространство с отмеченной точкой N, так что, имеем $\Omega\Sigma X :\equiv (\mathsf{N} =_{\Sigma X} \mathsf{N})$. Тогда существует каноническое отображение

$$\begin{aligned} &\sigma X \to \Omega\Sigma X \\ &\sigma(x) :\equiv \mathsf{merid}(x) \cdot \mathsf{merid}(x_0)^{-1} \,. \end{aligned}$$

Замечание 8.6.3. В классической алгебраической топологии рассматривается *редуцированная надстройка*, в которой путь $\mathsf{merid}(x_0)$ сворачивается до точки, идентифицирующей N и S. Редуцированные и нередуцированные надстройки гомотопически эквивалентны, поэтому различие невидимо для нашего чисто теоретико-гомотопического взгляда, а высшие индуктивные типы в любом случае позволяют «идентифицировать» точки до высшего пути, и нет смысла рассматривать редуцированные надстройки в гомотопической теории типов. Однако «нередуцируемость» нашей надстройки является причиной (возможно, неожиданного) появления $\mathsf{merid}(x_0)^{-1}$ в определении σ.

Теперь нашей целью является доказательство следующей теоремы.

Теорема 8.6.4 (Теорема Фрейденталя о надстройке)**.** *Предположим, что* X n*-связно и точечно, с* $n \geqslant 0$*. Тогда отображение* $\sigma : X \to \Omega\Sigma(X)$ *является* $2n$*-связным.*

Будем использовать метод кодирования-декодирования, но применяемый в несколько ином виде. До сих пор, в большинстве случаев, мы использовали его, чтобы охарактеризовать пространство петель $\Omega(A, a_0)$ некоторого типа как эквивалент некоторого другого типа B, создавая семейство $\mathsf{code} : A \to \mathcal{U}$ с $\mathsf{code}(a_0) :\equiv B$ и семейство эквивалентностей $\mathsf{decode} : \prod_{(x:A)} \mathsf{code}(x) \simeq (a_0 = x)$.

Однако в этом случае мы хотим показать, что $\sigma : X \to \Omega\Sigma X$ является $2n$-связным. Мы могли бы использовать усеченную версию предыдущего метода, такую, как мы увидим в §8.7, чтобы доказать, что $\|X\|_{2n} \to \|\Omega\Sigma X\|_{2n}$ является эквивалентностью, но это утверждение немного слабее, чем $2n$-связность отображения (см. следствия 8.4.8 и 8.8.5). Однако, обратите внимание, что в общем случае, чтобы доказать, что $\mathsf{decode}(x)$ является эквивалентностью, мы могли бы тем же самым образом доказать, что его слои стягиваемы, и мы все равно сможем использовать индукцию по базовому типу. Можно обобщить это, чтобы доказать связность отображения в пространстве петель, т.е. что *усечения* его слоев стягиваемы. Более того, вместо того, чтобы строить code и decode по отдельности, можно напрямую построить семейство *кодов для усечений слоев*.

Определение 8.6.5. Если X является n-связным и точечным с $n \geqslant 0$, то существует семейство

$$\mathsf{code} : \prod_{(y:\Sigma X)} (\mathsf{N} = y) \to \mathcal{U} \tag{8.6.6}$$

такое, что

$$\mathsf{code}(\mathsf{N}, p) :\equiv \|\mathsf{fib}_\sigma(p)\|_{2n} \equiv \left\|\sum\nolimits_{(x:X)} (\mathsf{merid}(x) \cdot \mathsf{merid}(x_0))^{-1} = p\right\|_{2n} \tag{8.6.7}$$

$$\mathsf{code}(\mathsf{S}, q) :\equiv \|\mathsf{fib}_{\mathsf{merid}}(q)\|_{2n} \equiv \left\|\sum\nolimits_{(x:X)} (\mathsf{merid}(x)) = q\right\|_{2n} \tag{8.6.8}$$

Наша конечная цель — доказать, что $\mathsf{code}(y, p)$ является стягиваемым для всех $y : \Sigma X$ и $p : \mathsf{N} = y$. Использование этого с $y :\equiv \mathsf{N}$ покажет, что все слои σ являются $2n$-связными, следовательно, $2n$-связно и σ.

Обоснование определения 8.6.5. Определим $\mathsf{code}(y, p)$ индукцией по $y : \Sigma X$, где первыми двумя случаями являются (8.6.7) и (8.6.8). Остается построить, для каждого $x_1 : X$, зависимый путь

$$\mathsf{code}(\mathsf{N}) =^{\lambda y.\,(\mathsf{N}=y)\to\mathcal{U}}_{\mathsf{merid}(x_1)} \mathsf{code}(\mathsf{S})\,.$$

По лемме 2.9.6 это равносильно заданию семейства путей

$$\prod_{(q:\mathsf{N}=\mathsf{S})} \mathsf{code}(\mathsf{N})\left(\mathsf{transport}^{\lambda y.\,(\mathsf{N}=y)}(\mathsf{merid}(x_1)^{-1}, q)\right) = \mathsf{code}(\mathsf{S})(q)\,.$$

А по унивалентности и транспортировке в типах путей это равносильно семейству эквивалентностей

$$\prod_{(q:\mathsf{N}=\mathsf{S})} \mathsf{code}(\mathsf{N})\left(N, q \cdot \mathsf{merid}(x_1)^{-1}\right) \simeq \mathsf{code}(\mathsf{S})(q)\,.$$

Определим семейство отображений

$$\prod_{(q:\mathsf{N}=\mathsf{S})} \mathsf{code}(\mathsf{N})\left(N, q \cdot \mathsf{merid}(x_1)^{-1}\right) \to \mathsf{code}(\mathsf{S})(q)\,. \tag{8.6.9}$$

и далее покажем, что все они эквивалентны. Итак, пусть $q : \mathsf{N} = \mathsf{S}$; по универсальному свойству усечения и определениям $\mathsf{code}(\mathsf{N}, -)$ и $\mathsf{code}(\mathsf{S}, -)$, достаточно определить, для каждого $x_2 : X$, отображение

$$\big(\mathsf{merid}(x_2) \cdot \mathsf{merid}(x_0)^{-1} = q \cdot \mathsf{merid}(x_1)^{-1}\big) \to \left\| \textstyle\sum_{(x:X)} (\mathsf{merid}(x) = q) \right\|_{2n}.$$

Теперь, для любых $x_1, x_2 : X$, этот тип является $2n$-усеченным, а X — n-связный. Таким образом, по лемме 8.6.2, достаточно определить это отображение, когда x_1 равно x_0, когда x_2 равно x_0, и проверить, что они согласованы, когда оба являются x_0.

Когда x_1 есть x_0, предположением является $r : \big(\mathsf{merid}(x_2) \cdot \mathsf{merid}(x_0)^{-1} = q \cdot \mathsf{merid}(x_0)^{-1}\big)$. Таким образом, убрав $\mathsf{merid}(x_0)^{-1}$ из r, получая $r' : \mathsf{merid}(x_2) = q$, можно определить этот образ как $|(x_2, r')|_{2n}$.

Когда x_2 равно x_0, предположением является $r : \big(\mathsf{merid}(x_0) \cdot \mathsf{merid}(x_0)^{-1} = q \cdot \mathsf{merid}(x_0)^{-1}\big)$. Преобразуя это, получим $r'' : \mathsf{merid}(x_1) = q$, и можно определить этот образ как $|(x_1, r'')|_{2n}$.

Наконец, когда и x_1, и x_2 равны x_0, достаточно показать, что результирующие r' и r'' согласованы; это простая лемма о композиции путей, что завершает определение (8.6.9). Чтобы показать, что это семейство эквивалентностей, поскольку быть эквивалентностью — это просто утверждение, а $x_0 : \mathbf{1} \to X$ (по крайней мере) (-1)-связно, достаточно предположить, что x_1 есть x_0. В этом случае, исследуя приведенную выше конструкцию, видим, что это, по сути, $2n$-усечение функции, убирающее $\mathsf{merid}(x_0)^{-1}$, что является эквивалентностью. □

В дополнение к (8.6.7) и (8.6.8) нужно будет извлечь из конструкции code некоторую информацию о том, как она действует на путях. Для этого воспользуемся следующей леммой.

Лемма 8.6.10. *Пусть $A : \mathcal{U}$, $B : A \to \mathcal{U}$, $C : \prod_{(a:A)} B(a) \to \mathcal{U}$, а также $a_1, a_2 : A$ с $m : a_1 = a_2$ и $b : B(a_2)$. Тогда функция*

$$\mathsf{transport}^{\widehat{C}}(\mathsf{pair}^{=}(m, t), -) : C(a_1, \mathsf{transport}^B(m^{-1}, b)) \to C(a_2, b),$$

где $t : \mathsf{transport}^B(\mathsf{transport}^B(m^{-1}, b)) = b$ — очевидный связный путь, а $\widehat{C} : (\sum_{(a:A)} B(a)) \to \mathcal{U}$ — это некаррированная форма C, равносильна эквивалентности, полученной унивалентностью из

$$\begin{aligned} C(a_1, \mathsf{transport}^B(m^{-1}, b)) &= \mathsf{transport}^{\lambda a.\, B(a) \to \mathcal{U}}(m, C(a_1))(b) && \text{(используя (2.9.4))} \\ &= C(a_2, b) && \text{(согласно } \mathsf{happly}(\mathsf{apd}_C(m), b)) \end{aligned}$$

Доказательство. Используя индукцию пути, можно предположить, что a_2 есть a_1, а m равно refl_{a_1}, и в этом случае обе функции тождественны. □

Применим эту лемму с $A :\equiv \Sigma X$, $B :\equiv \lambda y.\, (\mathsf{N} = y)$ и $C :\equiv \mathsf{code}$, в то время как $a_1 :\equiv \mathsf{N}$, $a_2 :\equiv \mathsf{S}$ and $m :\equiv \mathsf{merid}(x_1)$ для некоторого $x_1 : X$, и, наконец, $b :\equiv q$ — некоторый путь $\mathsf{N} = \mathsf{S}$. Правило вычисления для индукции над ΣX отождествляет $\mathsf{apd}_C(m)$ с путем, построенным определенным образом из унивалентности и функциональной экстенсиональности. Вторая функция, описанная в лемме 8.6.10, по существу состоит в отмене этих применений унивалентности и экстенсиональности, возвращении к конкретным функциям (8.6.9), которые были определены с помощью леммы 8.6.2. Следовательно, лемма 8.6.10 формулирует то, что транспортирование по $\mathsf{pair}^{=}(q, t)$ по существу восстанавливает эти функции.

Наконец, по построению, когда x_1 или x_2 совпадает с x_0, а исходные данные находится в образе $|-|_{2n}$, мы более точно знаем, что это за функции. Таким образом, для любого $x_2 : X$, имеем

$$\mathsf{transport}^{\hat{\mathsf{code}}}(\mathsf{pair}^{=}(\mathsf{merid}(x_0), t), |(x_2, r)|_{2n}) = |(x_1, r')|_{2n} \tag{8.6.11}$$

где $r : \mathsf{merid}(x_2) \cdot \mathsf{merid}(x_0)^{-1} = \mathsf{transport}^{B}(\mathsf{merid}(x_0)^{-1}, q)$ произвольно, как и раньше, а $r' : \mathsf{merid}(x_2) = q$ получается из r, путем отождествления его конечной точки с $q \cdot \mathsf{merid}(x_0)^{-1}$ и изъятием $\mathsf{merid}(x_0)^{-1}$. Аналогично, для любого $x_1 : X$, имеем

$$\mathsf{transport}^{\hat{\mathsf{code}}}(\mathsf{pair}^{=}(\mathsf{merid}(x_1), t), |(x_0, r)|_{2n}) = |(x_1, r'')|_{2n} \tag{8.6.12}$$

где $r : \mathsf{merid}(x_0) \cdot \mathsf{merid}(x_0)^{-1} = \mathsf{transport}^{B}(\mathsf{merid}(x_1)^{-1}, q)$, а $r'' : \mathsf{merid}(x_1) = q$ получается путем определения его конечной точки и перестановкой путей.

Доказательство теоремы 8.6.4. Покажем, что функция $\mathsf{code}(y, p)$ стягиваема для каждого $y : \Sigma X$ и $p : \mathsf{N} = y$. Сначала выберем центр стягивания, скажем $c(y, p) : \mathsf{code}(y, p)$. Это соответствует определению функции encode в предыдущих доказательствах, поэтому определим ее с использованием $\mathsf{transport}$. Заметьте, что в особом случае, когда y есть N, а p — $\mathsf{refl}_{\mathsf{N}}$, имеем

$$\mathsf{code}(\mathsf{N}, \mathsf{refl}_{\mathsf{N}}) \equiv \left\| \sum_{(x:X)} (\mathsf{merid}(x) \cdot \mathsf{merid}(x_0)^{-1} = \mathsf{refl}_{\mathsf{N}} \right\|_{2n} .$$

Таким образом, можно выбрать $c(\mathsf{N}, \mathsf{refl}_{\mathsf{N}}) :\equiv \left|(x_0, \mathsf{rinv}_{\mathsf{merid}(x_0)})\right|_{2n}$, где rinv_q — очевидный путь $q \cdot q^{-1} = \mathsf{refl}$ для любого q. Теперь можно получить $c : \prod_{(y:\Sigma X)} \prod_{(p:\mathsf{N}=y)} \mathsf{code}(y, p)$ индукцией пути по p, но ниже будет важно, что также можно дать конкретное определение в терминах транспортирования:

$$c(y, p) :\equiv \mathsf{transport}^{\hat{\mathsf{code}}}(\mathsf{pair}^{=}(p, \mathsf{tid}_p), c(\mathsf{N}, \mathsf{refl}_{\mathsf{N}})) ,$$

где $\hat{\mathsf{code}} : \left(\sum_{(y:\Sigma X)} (\mathsf{N} = y)\right) \to \mathcal{U}$ — некаррированная версия code, а $\mathsf{tid}_p : p_*(\mathsf{refl}) = p$ — стандартная лемма.

Далее, надо показать, что каждый элемент из $\mathsf{code}(y, p)$ равен $c(y, p)$. Опять же, с помощью индукции пути достаточно предположить, что y есть N, а p — $\mathsf{refl}_{\mathsf{N}}$. Фактически, докажем это в более общем виде, когда y есть N, а p произвольно. То есть, покажем, что для любых $p : \mathsf{N} = \mathsf{N}$ и $d : \mathsf{code}(\mathsf{N}, p)$ имеем $d = c(\mathsf{N}, p)$. Поскольку это равенство является $(2n-1)$-типом, можно предположить, что d имеет вид $|(x_1, r)|_{2n}$ для некоторых $x_1 : X$ и $r : \mathsf{merid}(x_1) \cdot \mathsf{merid}(x_0)^{-1} = p$.

Теперь, применяя далее индукцию пути, можно предположить, что r — рефлексивность, а p — $\mathsf{merid}(x_1) \cdot \mathsf{merid}(x_0)^{-1}$ (поэтому, выше, мы обобщили на произвольный p). Таким образом, надо доказать, что

$$\left|(x_1, \mathsf{refl}_{\mathsf{merid}(x_1) \cdot \mathsf{merid}(x_0)^{-1}})\right|_{2n} = c\left(\mathsf{N}, \mathsf{refl}_{\mathsf{merid}(x_1) \cdot \mathsf{merid}(x_0)^{-1}}\right) . \tag{8.6.13}$$

По определению, правая часть этого равенства имеет вид

$$\begin{aligned}
&\mathsf{transport}^{\hat{\mathsf{code}}}\left(\mathsf{pair}^{=}(\mathsf{merid}(x_1) \cdot \mathsf{merid}(x_0)^{-1}, |(x_0, _)|_{2n}\right) \\
&\quad = \mathsf{transport}^{\hat{\mathsf{code}}}\left(\mathsf{pair}^{=}(\mathsf{merid}(x_0)^{-1}, _), \mathsf{transport}^{\hat{\mathsf{code}}}\left(\mathsf{pair}^{=}(\mathsf{merid}(x_1), _), |(x_0, _)|_{2n}\right)\right) \\
&\qquad = \mathsf{transport}^{\hat{\mathsf{code}}}\left(\mathsf{pair}^{=}(\mathsf{merid}(x_0)^{-1}, _), |(x_1, _)|_{2n}\right) = |(x_1, _)|_{2n} ,
\end{aligned}$$

где подчеркивание "_" должно быть заполнено подходящими когерентными путями. Здесь первый шаг — это функториальность транспортирования, второй вызывает (8.6.12), а третий вызывает (8.6.11) (с транспортированием, направленным в обратную сторону). Таким образом, имеется та же первая компонента, что и в левой части (8.6.13). Мы предоставляем читателю возможность убедиться, что все когерентные пути сокращаются, давая рефлексивность во втором компоненте. $\square$

Следствие 8.6.14 (Эквивалентность Фрейденталя)**.** *Предположим, что X является n-связным и точечным, с $n \geqslant 0$. Тогда $\|X\|_{2n} \simeq \|\Omega\Sigma X\|_{2n}$.*

Доказательство. По теореме 8.6.4, σ является $2n$-связным. По лемме 7.5.14, следовательно, это эквивалентность на $2n$-усечениях. $\square$

Одним из важных следствий теоремы Фрейденталя о надстройке является то, что гомотопические группы сфер стабильны в определенном пролете (это диагонали с северо-востока на юго-запад в таблице 8.1).

Следствие 8.6.15 (Устойчивость для сфер)**.** *Если $k \leqslant 2n-2$, то $\pi_{k+1}(\mathbb{S}^{n+1}) = \pi_k(\mathbb{S}^n)$.*

Доказательство. Пусть $k \leqslant 2n-2$. По следствию 8.2.2, $\mathbb{S}^n$ является $(n-1)$-связным. Следовательно, по следствию 8.6.14,

$$\|\Omega(\Sigma(\mathbb{S}^n))\|_{2(n-1)} = \|\mathbb{S}^n\|_{2(n-1)} .$$

По лемме 7.3.15, поскольку $k \leqslant 2(n-1)$, применение $\|-\|_k$ к обеим сторонам показывает, что это равенство выполняется для k:

$$\|\Omega(\Sigma(\mathbb{S}^n))\|_k = \|\mathbb{S}^n\|_k . \tag{8.6.16}$$

Тогда основная идея доказательства состоит в том, что опускается проверка того, что эти эквивалентности надлежащим образом действуют на отмеченные точки этих пространств, и что при $k > 0$ эти эквивалентности поддерживают умножение:

$$\begin{aligned}
\pi_{k+1}(\mathbb{S}^1) &\equiv \left\|\Omega^{k+1}(\mathbb{S}^{n+1})\right\|_0 \\
&\equiv \left\|\Omega^{k}(\Omega(\mathbb{S}^{n+1}))\right\|_0 \\
&\equiv \left\|\Omega^{k}(\Omega(\Sigma(\mathbb{S}^{n})))\right\|_0 \\
&= \Omega^k\left(\|(\Omega(\Sigma(\mathbb{S}^n)))\|_k\right) && \text{(по теореме 7.3.12)} \\
&= \Omega^k\left(\|\mathbb{S}^n\|_k\right) && \text{(по (8.6.16))} \\
&= \left\|\Omega^k(\mathbb{S}^n)\right\|_0 && \text{(по теореме 7.3.12)} \\
&\equiv \pi_k(\mathbb{S}^n) .
\end{aligned}$$

$\square$

Это означает, что после того, как вычислена одна запись в одной из этих стабильных диагоналей, можно узнать и все остальные. Например:

Теорема 8.6.17. *$\pi_n(\mathbb{S}^n) = \mathbb{Z}$ для любого $n \geqslant 1$.*

Доказательство. Доказательство проводится индукцией по n. Уже имеем $\pi_1(\mathbb{S}^1) = \mathbb{Z}$ (следствие 8.1.11) и $\pi_2(\mathbb{S}^2) = \mathbb{Z}$ (следствие 8.5.2}). Когда $n \geqslant 2$, то $n \leqslant (2n-2)$. Следовательно, согласно следствию 8.6.15, $\pi_{n+1}(\mathbb{S}^{n+1}) = \pi_n(\mathbb{S}^n)$, и эта эквивалентность, в сочетании с индуктивной гипотезой, дает требуемый результат. $\square$

Следствие 8.6.18. $\mathbb{S}^{n+1}$ *не является n-типом, для любого $n \geqslant -1$.*

Следствие 8.6.19. $\pi_3(\mathbb{S}^2) = \mathbb{Z}$.

Доказательство. По следствию 8.5.2, $\pi_3(\mathbb{S}^2) = \pi_3(\mathbb{S}^3)$. Но, по теореме 8.6.17, $\pi_3(\mathbb{S}^3) = \mathbb{Z}$. □

8.7 Теорема Ван Кампена

Теорема ван Кампена позволяет вычислить фундаментальную группу π_1 (гомотопической) амальгамы пространств. Традиционно она формулируется для топологического пространства X, которое является объединением двух открытых подпространств $\mathcal{U}$ и V, но с точки зрения теории гомотопий это просто удобный способ убедиться, что X является амальгамой $\mathcal{U}$ и V над их пересечением. Мы докажем версию теоремы ван Кампена для произвольных амальгам.

В этом разделе будет представлено доказательство теоремы ван Кампена, в котором используется тот же метод кодирования-декодирования, что был применен для $\pi_1(\mathbb{S}^1)$ в §8.1. Существует также более теоретико-гомотопический подход; см. упражнение 9.11.

Нам потребуется более совершенная версия метода кодирования-декодирования. В §8.1 (а также в §§ 2.12 и 2.13) мы использовали его, чтобы охарактеризовать пространство путей (высшего) индуктивного типа W, получив, как следствие, характеристику пространства петель $\Omega(W)$, а значит, и его 0-усечения $\pi_1(W)$. В теореме ван Кампена наша цель состоит только в том, чтобы охарактеризовать фундаментальную группу $\pi_1(W)$, и у нас нет явного описания пространств петель или пространств путей для использования.

Оказывается, можно использовать ту же технику непосредственно для усеченной версии расслоения путей, тем самым охарактеризовав не только фундаментальную *группу* $\pi_1(W)$, но и весь фундаментальный *группоид*. В частности, для типа X, запишем $\Pi_1 X : X \to X \to \mathcal{U}$ для 0-усечения его тождественного типа, то есть $\Pi_1 X(x,y) :\equiv \|x = y\|_0$. Обратите внимание, что мы индуцировали группоидные операции

$$\begin{aligned}
(- \cdot -) &: \Pi_1 X(x,y) \to \Pi_1 X(y,z) \to \Pi_1 X(x,z) \\
(-)^{-1} &: \Pi_1 X(x,y) \to \Pi_1 X(y,x) \\
\mathsf{refl}_x &: \Pi_1 X(x,x) \\
\mathsf{ap}_f &: \Pi_1 X(x,y) \to \Pi_1 Y(fx,fy)
\end{aligned}$$

для которых мы используем те же обозначения, что и в соответствующих операциях на путях.

8.7.1 «Наивная» теорема ван Кампена

Начнем с «наивной» версии теоремы ван Кампена, которая полезна, хотя и не так, как классическая версия. В §8.7.2 мы улучшим ее до более содержательной, классической версии.

Для типов A, B, C и функций $f : A \to B$, $g : A \to C$, пусть P будет их амальгамой $B \sqcup^A C$. Как мы видели в §6.8, P — это высший индуктивный тип, порожденный

- $i : B \to P$,
- $j : C \to P$, и

- путем $kx : ifx = jgx$, для каждого $x : A$.

Определим $\mathsf{code} : P \to P \to \mathcal{U}$ двойной индукцией по P, следующим образом.

- $\mathsf{code}(ib, ib')$ является множеством-частным (см. §6.10) типа последовательностей

$$(b, p_0, x_1, q_1, y_1, p_1, x_2, q_2, y_2, p_2, \ldots, y_n, p_n, b')$$

  где

  - $n : \mathbb{N}$;
  - $x_k : A$ и $y_k : A$ для $0 < k \leqslant n$;
  - $p_0 : \Pi_1 B(b, fx_1)$ и $p_n : \Pi_1 B(fy_n, b')$ для $n > 0$, и $p_0 : \Pi_1 B(b, b')$ для $n = 0$;
  - $p_k : \Pi_1 B(fy_k, fx_{k+1})$ для $1 \leqslant k < n$;
  - $q_k : \Pi_1 C(gx_k, gy_k)$ для $1 \leqslant k \leqslant n$.

  Частное порождается следующими равенствами:

$$(\ldots, q_k, y_k, \mathsf{refl}_{fy_k}, y_k, q_{k+1}, \ldots) = (\ldots, q_k \cdot q_{k+1}, \ldots)$$
$$(\ldots, p_k, x_k, \mathsf{refl}_{gx_k}, x_k, p_{k+1}, \ldots) = (\ldots, p_k \cdot p_{k+1}, \ldots)$$

  (см. замечание 8.7.3 ниже). Мы предоставляем читателю определить этот тип последовательностей именно как индуктивный.

- $\mathsf{code}(jc, jc')$ идентичен, но роли B и C поменялись местами. Аналогичным образом поменяем в обозначениях роли x и y, а также p и q.

- $\mathsf{code}(ib, jc)$ и $\mathsf{code}(jc, ib)$ похожи, но с изменением четности, так что они начинаются одним типом, а заканчиваются другим.

- Для $a : A$ и $b : B$ мы требуем эквивалентности

$$\mathsf{code}(ib, ifa) \simeq \mathsf{code}(ib, jga)\,. \tag{8.7.1}$$

  Мы определяем ее, как состоящей из двух функций, определенных на последовательностях, формулой

$$(\ldots, y_n, p_n, fa) \mapsto (\ldots, y_n, p_n, a, \mathsf{refl}_{ga}, ga)\,,$$
$$(\ldots, x_n, p_n, a, \mathsf{refl}_{fa}, fa) \leftarrow\!\shortmid (\ldots, x_n, p_n, ga)\,.$$

  Обе эти функции, как легко видеть, поддерживают отношения эквивалентности и, следовательно, определяют функции на типах кодов. Композиция слева-направо-налево — это

$$(\ldots, y_n, p_n, fa) \mapsto (\ldots, y_n, p_n, a, \mathsf{refl}_{ga}, a, \mathsf{refl}_{fa}, fa)$$

  которая равна единице порождающим равенством частного. Другая составная часть устроена аналогично. Таким образом, мы определили эквивалентность (8.7.1).

- Аналогично нам потребуются эквивалентности

$$\begin{aligned}\mathsf{code}(jc, ifa) &\simeq \mathsf{code}(jc, jga)\\ \mathsf{code}(ifa, ib) &\simeq (jga, ib)\\ \mathsf{code}(ifa, jc) &\simeq (jga, jc)\end{aligned}$$

  Аналогично нам потребуются эквивалентности которые определены точно так же (последние два — добавлением термов рефлексивности в начале, а не в конце).

- Наконец, для $a, a' : A$, следующая диаграмма является коммутативной:

$$\begin{tikzcd} \mathsf{code}(ifa, ifa') \arrow[r] \arrow[d] & \mathsf{code}(ifa, jga') \arrow[d] \\ \mathsf{code}(jga, ifa') \arrow[r] & \mathsf{code}(jga, jga') \end{tikzcd} \tag{8.7.2}$$

  Это равносильно утверждению, что если мы добавляем что-то в начало, а затем — что-то в конец последовательности, то могли бы сделать это в другом порядке.

Замечание 8.7.3. Можно было бы ожидать увидеть в определении code некоторые дополнительные порождающие уравнения для множества-частного, такие как

$$\begin{aligned}(\ldots, p_{k-1} \cdot fw, x'_k, q_k, \ldots) &= (\ldots, p_{k-1}, x_k, gw \cdot q_k, \ldots) &\quad& (\text{for } w : \Pi_1 A(x_k, x'_k))\\ (\ldots, q_k \cdot gw, y'_k, p_k, \ldots) &= (\ldots, q_k, y_k, fw \cdot p_k, \ldots). && (\text{for } w : \Pi_1 A(y_k, y'_k))\end{aligned}$$

Однако в этом нет необходимости! Фактически, они следуют автоматически использованием индукции пути по w. В этом основное отличие «наивной» теоремы Ван Кампена от более изощренной, которая будет рассмотрена в следующем подразделе.

Продолжая, можно охарактеризовать транспортирование в расслоении code:

- Для $p : b =_B b'$ и $u : P$, имеем

$$\mathsf{transport}^{b \mapsto \mathsf{code}(u, ib)}(p, (\ldots, y_n, p_n, b)) = (\ldots, y_n, p_n \cdot p, b').$$

- Для $q : c =_C c'$ и $u : P$, имеем

$$\mathsf{transport}^{c \mapsto \mathsf{code}(u, jc)}(q, (\ldots, x_n, q_n, c)) = (\ldots, x_n, q_n \cdot q, c').$$

Здесь мы злоупотребляем нотацией, используя одно и то же имя для пути в X и его образа в $\Pi_1 X$. Заметьте, что транспортирование в $\Pi_1 X$ также задается конкатенацией с (образом) путем. Отсюда можно доказать приведенные выше утверждения индукцией по u. Также имеем:

- Для $a : A$ и $u : P$,

$$\mathsf{transport}^{v \mapsto \mathsf{code}(u, v)}(ha, (\ldots, y_n, p_n, fa)) = (\ldots, y_n, p_n, a, \mathsf{refl}_{ga}, ga).$$

  По существу, это следует из определения code.

Также построим функцию
$$r : \prod_{u:P} \mathsf{code}(u,u)$$
индукцией по u следующим образом:
$$rib :\equiv (b, \mathsf{refl}_b, b)$$
$$rjc :\equiv (c, \mathsf{refl}_c, c)$$
и для rka используем составное равенство
$$\begin{aligned}(ka,ka)_*(fa, \mathsf{refl}_{fa}, fa) &= (ga, \mathsf{refl}_{ga}, a, \mathsf{refl}_{fa}, a, \mathsf{refl}_{ga}, ga)\\ &= (ga, \mathsf{refl}_{ga}, ga)\end{aligned}$$
где первое равенство следует из выше приведенного наблюдения о транспортировке в code, а второе является экземпляром отношения множества-частного, используемого для определения code.

Теперь докажем следующую теорему.

Теорема 8.7.4 (Наивная теорема ван Кампена). *Для всех $u, v : P$ существует эквивалентность*
$$\Pi_1 P(u,v) \simeq \mathsf{code}(u,v)\,.$$

Доказательство. Для определения функции
$$\mathsf{encode} : \Pi_1 P(u,v) \to \mathsf{code}(u,v)$$
достаточно определить функцию $(u =_P v) \to \mathsf{code}(u,v)$, поскольку $\mathsf{code}(u,v)$ — это множество. Сделаем это транспортированием:
$$\mathsf{encode}(p) :\equiv \mathsf{transport}^{v \mapsto \mathsf{code}(u,v)}(p, r(u))\,.$$
Теперь для определения
$$\mathsf{decode} : \mathsf{code}(u,v) \to \Pi_1 P(u,v)$$
действуем, как обычно, индукцией по $u, v : P$. В каждом случае для u и v, применяем i или j ко всем равенствам p_k и q_k в зависимости от ситуации и объединяем результаты в P, используя h для идентификации конечных точек. Например, когда $u \equiv ib$ и $v \equiv ib'$, то определяем
$$\begin{aligned}&\mathsf{decode}(b, p_0, x_1, q_1, y_1, p_1, \ldots, y_n, p_n, b') :\equiv\\ &\qquad (p_0) \cdot h(x_1) \cdot j(q_1) \cdot h(y_1)^{-1} \cdot i(p_1) \cdot \cdots \cdot h(y_n)^{-1} \cdot i(p_n)\,. \qquad (8.7.5)\end{aligned}$$
При этом соблюдаются, отношение эквивалентности множество-частного, и эквивалентности, такие как (8.7.1), поскольку $h : fi \sim gj$ естественно, а f и g — функториальны.

Как обычно, чтобы показать, что композиция
$$\Pi_1 P(u,v) \xrightarrow{\mathsf{encode}} \mathsf{code}(u,v) \xrightarrow{\mathsf{decode}} \Pi_1 P(u,v)$$
является тождественной, сначала уберем 0-усечение (поскольку ко-область — это множество), а затем применим индукцию пути. Вход refl_u направляется к ru, затем возвращается к refl_u (применяя дальнейшую индукцию по u для декомпозиции $\mathsf{decode}(ru)$).

Наконец, рассмотрим композицию

$$\mathsf{code}(u,v) \xrightarrow{\mathsf{decode}} \Pi_1 P(u,v) \xrightarrow{\mathsf{encode}} \mathsf{code}(u,v)\,.$$

Продолжим индукцию по $u, v : P$. Когда $u \equiv ib$ и $v \equiv ib'$, эта композиция есть

$$\begin{aligned}(b, p_0, x_1, q_1, y_1, p_1, \ldots, y_n, p_n, b') \mapsto & \left(ip_0 \cdot hx_1 \cdot jq_1 \cdot hy_1{}^{-1} \cdot ip_1 \cdot \cdots \cdot hy_n{}^{-1} \cdot ip_n\right)_*(rib) \\ &= (ip_n)_* \cdots (jq_1)_*(hx_1)_*(ip_0)_*(b, \mathsf{refl}_b, b) \\ &= (ip_n)_* \cdots (jq_1)_*(hx_1)_*(b, p_0, ifx_1) \\ &= (ip_n)_* \cdots (jq_1)_*(b, p_0, x_1, \mathsf{refl}_{gx_1}, jgx_1) \\ &= (ip_n)_* \cdots (b, p_0, x_1, q_1, jgy_1) \\ &= \quad \vdots \\ &= (b, p_0, x_1, q_1, y_1, p_1, \ldots, y_n, p_n, b')\,.\end{aligned}$$

то есть тождественная функция (точнее, здесь необходим неявный индуктивный аргумент). Остальные три точечных случая аналогичны, а случаи путей тривиальны, поскольку все типы являются множествами. □

Теорема 8.7.4 позволяет вычислять фундаментальные группы многих типов при условии, что A является множеством, поскольку в этом случае каждая функция $\mathsf{code}(u, v)$, по определению, является множеством-частным *множества* по отношению.

Пример 8.7.6. Пусть $A :\equiv \mathbf{2}$, $B :\equiv \mathbf{1}$ и $C :\equiv \mathbf{1}$. Тогда $P \simeq \mathbb{S}^1$. Анализируя определение, скажем, $\mathrm{code}(i(\star), i(*))$, видим, что все пути также могут быть тривиальными, поэтому единственная информация находится в последовательности элементов $x_1, y_1, \ldots, x_n, y_n : \mathbf{2}$. Более того, если имеет место $x_k = y_k$ или $y_k = x_{k+1}$ для любого k, то отношения множества-частного позволяют исключить оба этих элемента. Таким образом, каждая такая последовательность эквивалентна канонической *приведенной*, в которой нет двух одинаковых соседних элементов. Ясно, что такая сокращенная последовательность однозначно определяется ее длиной (натуральным числом n) вместе с, если $n > 1$, информацией о том, совпадает ли x_1 с $0_{\mathbf{2}}$ или с $1_{\mathbf{2}}$, поскольку это однозначно определяет остальную часть последовательности. И эти данные, конечно, можно идентифицировать с помощью целого числа, когда n — абсолютное значение, а x_1 кодирует знак. Таким образом мы получаем обратно $\pi_1(\mathbb{S}^1) \cong \mathbb{Z}$.

Поскольку теорема 8.7.4 утверждает только биекцию семейств множеств, этот изоморфизм, $\pi_1(\mathbb{S}^1) \cong \mathbb{Z}$, также является только биекцией множеств. Однако мы могли бы определить операцию конкатенации для code (путем конкатенации последовательностей) и показать, что encode и decode образуют изоморфизм относительно этой структуры (на языке главы 9 это были бы «предгруппоиды»). Разбор деталей оставляем читателю.

Пример 8.7.7. В более общем смысле, пусть $B :\equiv \mathbf{1}$ и $C :\equiv \mathbf{1}$, но A произвольно, так что P является надстройкой для A. Тогда снова, пути p_k и q_k тривиальны, поэтому единственная информация в коде пути — это последовательность элементов $x_1, y_1, \ldots, x_n, y_n : A$. Первые два образующих равенства говорят о том, что смежные равные элементы могут быть изъяты, поэтому имеет смысл думать об этой последовательности как о слове формы

$$x_1 y_1^{-1} x_2 y_2^{-1} \cdots x_n y_n^{-1}$$

в группе. Действительно, она похожа на свободную группу на A (или, что то же самое, на $\|A\|_0$; см. замечание 6.11.8), но мы рассматриваем только слова, начинающиеся с неинвертированного элемента , чередуются перевернутые и неинвертированные элементы и заканчиваются перевернутым. Это эффективно уменьшает размер образующей установки на единицу. Например, если A имеет точку $a : A$, то можно отождествить $\pi_1(\Sigma A)$ с группой, представленной $\|A\|_0$ как образующими с отношением $|a|_0 = e$; подробности см. в упражнениях 8.10 и 8.11.

Пример 8.7.8. Пусть $A :\equiv \mathbf{1}$, B и C произвольны, так что f и g просто снабжают B и C отмеченными точками, скажем, b и c. Тогда P — это *клин* $B \vee C$ пространств B и C (копроизведение в категории, базирующейся на пространствах). В данном случае тривиальны элементы x_k и y_k, поэтому единственной информацией является последовательность петель $(p_0, q_1, p_1, \ldots, p_n)$ с $p_k : \pi_1(B, b)$ и $q_k : \pi_1(C, c)$. Такие последовательности по модулю наложенного нами отношения эквивалентности легко отождествляются с явным описанием *свободного произведения* групп $\pi_1(B, b)$ и $\pi_1(C, c)$, построенного в §6.11. Таким образом, имеем $\pi_1(B \vee C) \cong \pi_1(B) * \pi_1(C)$.

Однако, теорема 8.7.4 перестает быть полной классической теоремой Ван Кампена, которая рассматривает случай, когда A не обязательно является множеством, и утверждает, что $\pi_1(B \sqcup^A C) \cong \pi_1(B) *_{\pi_1(A)} \pi_1(C)$ (с отмеченной точкой, исходящей из A). Действительно, заключение теоремы 8.7.4 вообще ничего не говорит о $\pi_1(A)$; пути в A «встроены в построение частного» теоретико-типовым способом, который затрудняет извлечение явной информации, в том смысле, что $\mathsf{code}(u, v)$ является множеством-частным не-множества по отношению. По этой причине в следующем пункте мы рассмотрим более подходящую версию теоремы ван Кампена.

8.7.2 Теорема ван Кампена с множеством отмеченных точек

Усовершенствование от ван Кампена, которое мы здесь представляем, близко аналогичному усовершенствованию в классической алгебраической топологии, где тип A снабжен *множеством S отмеченных точек*. Фактически, для нашего доказательства оказывается ненужным предполагать, что «множество отмеченных точек» является *множеством* — с таким же успехом это может быть произвольный тип; полезность предположения, что S является множеством, проявит себя позже, когда теорема будет применяться для получения вычислений. Важно то, что S содержит по крайней мере одну точку в каждой связной компоненте A. Мы заявляем это в теории типов, говоря, что у имеется тип S и функция $k : S \to A$ которая является сюръективной, т.е. (-1)-связной. Если $S \equiv A$ и k - тождественная функция, то получаем наивную теорему ван Кампена. Другой пример, который следует иметь в виду, — это когда A является точечным и (0-)связным, с $k : \mathbf{1} \to A$ точкой: согласно леммам 7.5.2 и 7.5.11 это отображение сюръективно только тогда, когда A является 0-связным.

Пусть $A, B, C, f, g, P, i, j, h$ — такие же, как в предыдущем разделе. Определим теперь, учитывая сюръективное отображение $k : S \to A$, вспомогательный тип, улучшающий связность k. Пусть T — высший индуктивный тип, порожденный

- функцией $\ell : S \to T$, и
- функцией $m : (ks =_A ks') \to (\ell s =_T \ell s')$, для любых $s, s' : S$.

Существует очевидная индуцированная функция $\bar{k} : T \to A$ такая, что $\bar{k}\ell = k$, и произвольное $p : ks = ks'$, эквивалентное составному $ks = \bar{k}\ell s \overset{\bar{k}mp}{=} \bar{k}\ell s' = ks'$.

Лемма 8.7.9. *$\bar{k}$ является 0-связной.*

Доказательство. Мы должны показать, что для всех $a : A$, 0-усечение типа $\sum_{(t:T)}(\bar{k}t = a)$ стягиваемо. Поскольку стягиваемость — это просто утверждение, а k является (-1)-связным, то можно предположить, что $a = ks$ для некоторого $s : S$. Теперь можно рассматривать центр стягивания как $|(\ell s, q)|_0$, где q — это равенство $\bar{k}\ell s = ks$.

Осталось показать, что для любого $\phi : \left\|\sum_{(t:T)}(\bar{k}t = ks)\right\|_0$ имеем $\phi = |(\ell s, q)|_0$. Поскольку последнее является простым высказыванием, и, в частности, множеством, можно предположить, что $\phi = |(t, p)|_0$ для $t : T$ и $p : \bar{k}t = ks$.

Теперь можно провести индукцию по $t : T$. Если $t \equiv \ell s'$, то $ks' = \bar{k}\ell s' \overset{p}{=} ks$ дает посредством m равенство $\ell s = \ell s'$. Следовательно, по определению $\bar{k}$ и равенства в гомотопических слоях, получаем равенство $(ks', p) = (ks, q)$, а значит, $|(ks', p)|_0 = |(ks, q)|_0$. Далее надо показать, что при изменении t вдоль m эти равенства согласуются. Но они являются равенствами в множестве (а именно, $\left\|\sum_{(t:T)}(\bar{k}t = ks)\right\|_0$), и, следовательно, это согласование происходит автоматически. □

Замечание 8.7.10. T можно рассматривать как (гомотопический) ко-уравнитель «ядерной пары» k. Если бы S и A были множествами, то (-1)-связность k означала бы, что A является 0-усечением этого ко-уравнителя (см. главу 10). Для общих типов теория высших топосов предполагает, что (-1)-связность k, вместо этого, будет означать, что A является копределом (он же «геометрическая реализация») «симплициального ядра» k. Тип T является копределом «1-скелета» этого симплициального ядра, поэтому имеет смысл повысить связность k на 1. В более общем смысле, можно ожидать, что копредел n-скелета увеличит связность на n.

Теперь определим $\mathsf{code} : P \to P \to \mathcal{U}$ двойной индукцией следующим образом.

- $\mathsf{code}(ib, ib')$ теперь является множеством-частным типа последовательностей

$$(b, p_0, x_1, q_1, y_1, p_1, x_2, q_2, y_2, p_2, \dots, y_n, p_n, b')$$

где

  - $n : \mathbb{N}$,
  - $x_k : S$ и $y_k : S$ для $0 < k \leqslant n$,
  - $p_0 : \Pi_1 B(b, fkx_1)$ и $p_n : \Pi_1 B(fky_n, b')$ для $n > 0$, и $p_0 : \Pi_1 B(b, b')$ для $n = 0$,
  - $p_k : \Pi_1 B(fky_k, fkx_{k+1})$ для $1 \leqslant k < n$,
  - $q_k : \Pi_1 C(gkx_k, gky_k)$ для $1 \leqslant k \leqslant n$.

Это частное порождает следующие равенства (см. замечание 8.7.3):

$$\begin{aligned}
(\dots, q_k, y_k, \mathsf{refl}_{fy_k}, y_k, q_{k+1}, \dots) &= (\dots, q_k \cdot q_{k+1}, \dots) \\
(\dots, p_k, x_k, \mathsf{refl}_{gx_k}, x_k, p_{k+1}, \dots) &= (\dots, p_k \cdot p_{k+1}, \dots) \\
(\dots, p_{k-1} \cdot fw, x_k', q_k, \dots) &= (\dots, p_{k-1}, x_k, gw \cdot q_k, \dots) && (\text{для } w : \Pi_1 A(kx_k, kx_k')) \\
(\dots, q_k \cdot gw, y_k', p_k, \dots) &= (\dots, q_k, y_k, fw \cdot p_k, \dots). && (\text{для } w : \Pi_1 A(ky_k, ky_k'))
\end{aligned}$$

Ниже нам понадобится определение случая для decode для такой последовательности, которая, как и раньше, объединяет все пути p_k и q_k вместе с экземплярами h чтобы получить элемент $\Pi_1 P(ifb, ifb')$, ср. (8.7.5). Как и ранее, остальные три случая практически идентичны.

- Для $a : A$ и $b : B$, мы требуем эквивалентности

$$\mathsf{code}(ib, ifa) \simeq \mathsf{code}(ib, jga)\,. \tag{8.7.11}$$

Поскольку code является множественно-значным, по лемме 8.7.9 можно считать, что $a = \bar{k}t$ для некоторого $t : T$. Затем можно провести индукцию по t. Если $t \equiv \ell s$ для $s : S$, то определим (8.7.11) как в §8.7.1:

$$(\dots, y_n, p_n, fks) \mapsto (\dots, y_n, p_n, s, \mathsf{refl}_{gks}, gks)\,,$$
$$(\dots, x_n, p_n, s, \mathsf{refl}_{fks}, fks) \leftarrow\!\shortmid (\dots, x_n, p_n, gks)\,.$$

Они удовлетворяют отношениям эквивалентности и, как и раньше, определяют квазиобратные. Теперь предположим, что t изменяется вдоль $m_{s,s'}(w)$ для некоторого $w : ks = ks'$; мы должны показать, что (8.7.11) не нарушает транспортирование вдоль $\bar{k}mw$. По определению $\bar{k}$, это, по сути, сводится к транспортированию по самому w. Характеризуя транспортирование в типах путей, нужно показать, что

$$w_*(\dots, y_n, p_n, fks) = (\dots, y_n, p_n \cdot fw, fks')$$

отображается согласно (8.7.11) к

$$w_*(\dots, y_n, p_n, s, \mathsf{refl}_{gks}, gks) = (\dots, y_n, p_n, s, \mathsf{refl}_{gks} \cdot gw, gks')\,.$$

Но это прямо следует из новых образующих, которые мы задали на отношении множества-частного, определяющего code.

- Остальные три необходимые эквивалентности определяются аналогично.

- Наконец, поскольку коммутативность (8.7.2) является простым высказыванием, в силу (-1)-связности k можно предположить, что $a = ks$ и $a' = ks'$, и в этом случае требуемое следует точно так же, как и раньше.

Теорема 8.7.12 (ван Кампен с множеством отмеченных точек)**.** *Для всех $u, v : P$ существует эквивалентность*

$$\Pi_1 P(u, v) \simeq \mathsf{code}(u, v)$$

с code*, определенным как в данном разделе.*

Доказательство. В основном так же, как и ранее. Чтобы показать, что decode удовлетворяет новым образующим отношения частных, мы используем естественность h. И чтобы показать, что decode удовлетворяет эквивалентностям, таким как (8.7.11), нужно провести индукцию по $\bar{k}$ и по T , чтобы декомпозировать эти эквивалентности на их определения, но тогда это снова становится просто функториальностью f и g. Остальное просто. В частности, для $\mathsf{encode} \circ \mathsf{decode}$ не требуется дополнительных аргументов, поскольку целью является доказательство равенства в множестве, и поэтому случай для h тривиален. □

Теорема 8.7.12 позволяет вычислить фундаментальную группу пространства A, даже если A не является множеством, при условии, что S — множество, поскольку в этом случае каждый $\mathsf{code}(u, v)$ есть, по определению, множество-частное множества по отношению. В таком аспекте, это улучшение по сравнению с теоремой 8.7.4.

Пример 8.7.13. Предположим, что $S :\equiv \mathbf{1}$, так что A имеет отмеченную точку $a :\equiv k(\star)$ и является связным. Тогда code для петель в амальгаме можно отождествить с чередующимися последовательностями петель в $\pi_1(B, f(a))$ и $\pi_1(C, g(a))$ по модулю отношения эквивалентности, которое позволяет перемещать элементы $\pi_1(A, a)$ между ними (после применения f и g, соответственно). Таким образом, $\pi_1(P)$ можно отождествить с *амальгамированным свободным произведением* $\pi_1(B) *_{\pi_1(A)} \pi_1(C)$ (амальгама в категория групп), построенным в §6.11. Это (в случае, когда B и C — открытые подпространства в P, а A — их пересечение), вероятно, является наиболее классической версией теоремы ван Кампена.

Пример 8.7.14. В качестве частного случая примера 8.7.13 предположим дополнительно, что $C :\equiv \mathbf{1}$, так что P является ко-расслоением B/A. Тогда каждая петля в C является рефлексивностью, поэтому отношения в code для путей позволяют свернуть все последовательности в одну петлю в B. Дополнительные отношения требуют умножения слева, справа или внутри на элемент в образе $\pi_1(A)$ — это тождественность. Таким образом, можно отождествить $\pi_1(B/A)$ с фактор-группой $\pi_1(B)$ по нормальной подгруппе, порожденной образом $\pi_1(A)$.

Пример 8.7.15. В качестве еще одного частного случая примера 8.7.14, пусть $B :\equiv \mathbb{S}^1 \vee \mathbb{S}^1$, $A :\equiv \mathbb{S}^1$, а $f : A \to B$ выбирает составную петлю $p \cdot q \cdot p^{-1} \cdot q^{-1}$, где p и q — порождающие петли в двух копиях $\mathbb{S}^1$ содержащие B. Тогда P — представление тора T^2. В самом деле, нетрудно отождествить P с представлением T^2 как описано в §6.7, используя конус на конкретной петле. Таким образом, $\pi_1(T^2)$ является фактором свободной группы на двух образующих (т.е., $\pi_1(B)$) по отношению $p \cdot q \cdot p^{-1} \cdot q^{-1} = 1$. Это явно производит свободную *абелеву* группу на двух образующих, которая есть $\mathbb{Z} \times \mathbb{Z}$.

Пример 8.7.16. Более общо, любой комплекс CW может быть получен путем многократного «срезания» сфер, как описано в §6.7. То есть, мы начинаем с множества X_0 точек («0-клеток»), которое является «0-скелетом» комплекса CW. Берем амальгаму

$$
\begin{tikzcd}
S_1 \times \mathbb{S}^0 \arrow[r, "f_1"] \arrow[d] & X_0 \arrow[d] \\
\mathbf{1} \arrow[r] & X_1
\end{tikzcd}
$$

для некоторого множества S_1 клеток и некоторого семейства f_1 «прикрепляющих отображений», получая «1-скелет» X_1. Затем берем амальгаму

$$
\begin{tikzcd}
S_2 \times \mathbb{S}^1 \arrow[r, "f_2"] \arrow[d] & X_1 \arrow[d] \\
\mathbf{1} \arrow[r] & X_2
\end{tikzcd}
$$

для некоторого множества S_2 2-клеток и некоторого семейства f_2 прикрепляющих отображений, получая 2-скелет X_2, и т.д. Фундаментальная группа каждой амальгамы может быть вычислена с помощью теоремы ван Кампена: мы получаем группу, представленную образующими, полученными из 1-скелета, и отношениями, полученными из S_2 и f_2. Амальгамы после этого этапа не изменяют фундаментальную группу, поскольку $\pi_1(\mathbb{S}^n)$ тривиально для $n > 1$ (см. §8.3).

Пример 8.7.17. В частности, предположим, что дано любое представление (множества-)группы $G = \langle X \mid R \rangle$, где X — множество образующих, а R — множество слов в этих образующих. Пусть $B :\equiv \bigvee_X \mathbb{S}^1$ и $A :\equiv \bigvee_R \mathbb{S}^1$, причем $f : A \to B$ отправляет каждую копию $\mathbb{S}^1$ к соответствующему слову в образующих петлях B. Отсюда следует, что $\pi_1(P) \cong G$; таким образом, мы построили связный тип, фундаментальной группой которого является G. Поскольку любая группа имеет представление, любая группа является фундаментальной группой некоторого типа. Если мы 1-усечем такой тип, то получим тип, единственной нетривиальной гомотопической группой которого является G; это называется **пространством Эйленберга-МакЛейна** $K(G, 1)$.

8.8 Теорема и принцип Уайтхеда

В классической теории гомотопии отображение $f : A \to B$, которое индуцирует изоморфизм $\pi_n(A, a) \cong \pi_n(B, f(a))$ для всех точек a в A (а также изоморфизм $\pi_0(A) \cong \pi_0(B)$) обязательно является гомотопической эквивалентностью, если пространства A и B хорошо себя ведут (например, имеют гомотопические типы CW-комплексов). Этот факт известен как *теорема Уайтхеда*. Фактически, «плохие» пространства, с которыми теорема Уайтхеда не справляется, невидимы для теории типов. Грубо говоря, хороших топологических пространств достаточно для представления ∞-группоидов, а теория гомотопических типов имеет дело непосредственно с ∞-, а не с фактическими топологическими пространствами. Таким образом, можно было бы ожидать, что теорема Уайтхеда верна для унивалентных оснований.

Однако это *не* тот случай: теорема Уайтхеда недоказуема. Фактически, существуют известные модели теории типов, в которых она не может быть истинной, хотя и по совершенно другим причинам, чем ее несостоятельность для некорректных топологических пространств. Эти модели являются «не-гиперполными ∞-топосами» (см. [Lur09]); грубо говоря, они состоят из пучков ∞-группоидов над ∞-мерными базовыми пространствами.

Поэтому с фундаментальной точки зрения мы можем говорить о *принципе Уайтхеда* как об «аксиоме классичности», сродни LEM и AC. Это можно настойчиво предполагать, но это не часть теории типов, основанная на вычислениях, и не выполняется во всех естественных моделях. При работе с теоретико-множественными основами этот принцип скрыт: он не может не выполняться в мире, где ∞-группоиды построены из множеств (с использованием топологических пространств, симплициальных множеств или любой другой подобной модели).

Это может показаться странным, но на самом деле не должно вызывать удивления. Гомотопическая теория типов — это *абстрактная* теория гомотопических типов, тогда как теория гомотопий топологических пространств или симплициальных множеств в теории множеств является *конкретной* моделью этой теории, точно так же, как целые числа являются конкретной моделью абстрактной теории колец. Следует ожидать, что любая конкретная модель будет обладать особыми свойствами, которые не присущи соответствующей абстрактной теории, но которые мы можем иногда принимать в качестве дополнительных аксиом (например, целые числа являются областью главных идеалов, но не все кольца таковы).

Описание каких-либо моделей теории типов выходит за рамки этой книги, поэтому мы не будем объяснять, как принцип Уайтхеда может не работать в некоторых из них. Тем не менее, мы можем доказать, что это верно всякий раз, когда задействованные типы n-усечены для некоторого конечного n, с помощью «нисходящей» индукции по n. Помимо того, что он представляет самостоятельный интерес (например, подразумевает существенную уникальность пространств Эйленберга-МакЛейна), мы надеемся, что доказательство этого результата даст некоторое интуитивное объяснение того, почему мы не можем надеяться доказать аналогичную теорему без гипотезы усечения.

Начнем со следующей модификации теоремы 4.6.3, которая в конечном итоге обеспечит индукционный шаг в доказательстве усеченного принципа Уайтхеда. Его можно рассматривать как теоретико-типовую, ∞-группоидальную версию классического утверждения, что вполне точный и эссенциально сюръективный функтор является эквивалентностью категорий.

Теорема 8.8.1. *Предположим, что функция $f : A \to B$ такова, что*

(i) $\|f\|_0 : \|A\|_0 \to \|B\|_0$ *сюръективно и*

(ii) для любых $x, y : A$, функция $\mathsf{ap}_f : (x =_A y) \to (f(x) =_B f(y))$ является эквивалентностью.

Тогда f — эквивалентность.

Доказательство. Обратите внимание, что (ii) — это в точности утверждение, что f является вложением, ср. §4.6. Таким образом, по теореме 4.6.3 достаточно показать, что f сюръективна, т.е. что для любого $b : B$ имеем $\|\mathsf{fib}_f(b)\|_{-1}$. Предположим, что b задано; тогда, поскольку $\|f\|_0$ сюръективно, просто существует $a : A$ такое, что $\|f\|_0(|a|_0) = |b|_0$. А поскольку наша цель — простое высказывание, то можно предположить его при таком a. Тогда имеем $|f(a)|_0 = \|f\|_0(|a|_0) = |b|_0$, следовательно, $\|f(a) = b\|_{-1}$. Опять же, поскольку наша цель по-прежнему — простое высказывание, можно предположить, что $f(a) = b$. Следовательно, $\mathsf{fib}_f(b)$ обитаем, а значит, просто обитаем. □

Поскольку гомотопические группы представляют собой усечения пространств петель, а не пространств путей, нам нужно изменить эту теорему, чтобы говорить об этом. Напомним про отображение Ωf из определения 8.4.2.

Следствие 8.8.2. *Предположим, что функция $f : A \to B$ такова, что*

(i) $\|f\|_0 : \|A\|_0 \to \|B\|_0$ *биективно и*

(ii) *для любых $x, y : A$, функция $\Omega f : \Omega(A, x) \to \Omega(B, f(x))$ является эквивалентностью.*

Тогда f — эквивалентность.

Доказательство. По теореме 8.8.1 достаточно показать, что $\mathsf{ap}_f : (x =_A y) \to (f(x) =_B f(y))$ является эквивалентностью для любых $x, y : A$. По следствию 4.4.6 можно считать, что $f(x) =_B f(y)$. В частности, $|f(x)|_0 = |f(y)|_0$, поэтому, поскольку $\|f\|_0$ является эквивалентностью, имеем $|x|_0 = |y|_0$, следовательно, $|x = y|_{-1}$. Поскольку мы пытаемся доказать простое высказывание (ap_f является эквивалентностью), можно считать, что задано $p : x = y$. Но теперь следующий квадрат коммутативен вплоть до гомотопии:

$$\begin{tikzcd}
\Omega(A, x) \arrow[r, "- \cdot p"] \arrow[d, "\Omega f"'] & (x =_A) \arrow[d, "\mathsf{ap}_f"] \\
\Omega(B, f(x)) \arrow[r, "- \cdot f(p)"'] & (f(x) =_B f(y))
\end{tikzcd}$$

Верхнее и нижнее отображения эквивалентны, а левое отображение — по предположению. Следовательно, по свойству 2 из 3, то же самое верно и для правого отображения. □

Теперь можно доказать усеченный принцип Уайтхеда.

Теорема 8.8.3. *Предположим, что A и B являются n-типами, а $f : A \to B$ таково, что*

(i) $\|f\|_0 : \|A\|_0 \to \|B\|_0$ *биективно, и*

(ii) $\pi_k(f) : \pi_k(A, x) \to \pi_k(B, f(x))$ — *биекция, для всех $k \geqslant 1$ и всех $x : A$.*

Тогда f — эквивалентность.

Условие (i) почти аналогично (ii), когда $k = 0$, за исключением того, что оно не ссылается ни на какую отмеченную точку $x : A$.

Доказательство. Проведем индукцию по n. Когда $n = -2$, утверждение тривиально. Так что, предположим, что это верно для всех функций между n-типами, и пусть A и B будут $(n+1)$-типами, а $f : A \to B$, как указано выше. Первое условие следствия 8.8.2 выполняется по предположению, поэтому достаточно показать, что, для любого $x : A$, функция $\Omega f : \Omega(A, x) \to \Omega(B, f(x))$ является эквивалентностью.

Поскольку $\Omega(A, x)$ и $\Omega(B, f(x))$ являются n-типами, можно применить предположение индукции. Нужно проверить, что $\|\Omega f\|_0$ является биекцией, и что для всех $k \geqslant 1$ и $p : x = x$ отображение $\pi_k(\Omega f) : \pi_k(x = x, p) \to \pi_k(f(x) = f(x), \Omega f(p))$ является биекцией. Первое утверждение выполняется по предположению, поскольку $\|\Omega f\|_0 \equiv \pi_1(f)$. Чтобы доказать второе утверждение, сначала обобщим его: для всех $y : A$ и $q : x = y$ имеем $\pi_k(\mathsf{ap}_f) : \pi_k(x = y, q) \to \pi_k(f(x) = f(y), \mathsf{ap}_f(q))$. Отсюда следует желаемое утверждение, поскольку, когда $y :\equiv x$, имеем $\pi_k(\Omega f) = \pi_k(\mathsf{ap}_f)$ по модулю определения их отмеченных точек $\Omega f(p) = \mathsf{ap}_f(p)$. Чтобы доказать обобщение, достаточно индукцией по пути доказать его, когда q есть refl_a. В этом случае имеем $\pi_k(\mathsf{ap}_f) = \pi_k(\Omega f) = \pi_{k+1}(f)$, и $\pi_{k+1}(f)$ — биекция, согласно исходным предположениям. □

Заметьте, что если A и B не являются n-типами для любого конечного n, то индукция не может начаться.

Следствие 8.8.4. *Если A — 0-связный n-тип и $\pi_k(A, a) = 0$ для всех k и $a : A$, то A является стягиваемым.*

Доказательство. Примените теорему 8.8.3 к отображению $A \to \mathbf{1}$. □

Следствие 8.8.5. *Для $n \geqslant 0$, отображение $f : A \to B$ является n-связным тогда и только тогда, когда верно следующее*:

(i) $\|f\|_0 : \|A\|_0 \to \|B\|_0$ *является изоморфизмом*;

(ii) *для любых $a : A$ и $k \leqslant n$, отображение $\pi_k(f) : \pi_k(A, a) \to \pi_k(B, f(a))$ является изоморфизмом*;

(iii) *для любого $a : A$, отображение $\pi_{n+1}(f) : \pi_{n+1}(A, a) \to \pi_{n+1}(B, f(a))$ сюръективно.*

Доказательство. Направление «только тогда» — это следствие 8.4.8. И наоборот, по длинной точной последовательности расслоения (теорема 8.4.6), из гипотез следует, что $\pi_k(\mathsf{fib}_f(f(a)) = 0$ для всех $k \leqslant n$ и $a : A$, и что $\|\mathsf{fib}_f(f(a))\|_0$ стягиваемо. Поскольку $\pi_k(\mathsf{fib}_f(f(a)) = \pi_k(\|\mathsf{fib}_f(f(a)\|_n)$ для $k \leqslant n$, и $\|\mathsf{fib}_f(f(a))\|_n$ — n-стягиваемо, по следствию 8.8.4 это стягиваемо для любого a.

Осталось показать, что $\|\mathsf{fib}_f(b)\|_n$ стягиваемо для $b : B$, не обязательно имеющего форму $f(a)$. Однако, по предположению существует $x : \|A\|_0$ с $|b|_0 = \|f\|_0\,(x)$. Поскольку стягиваемость — это простое высказывание, можно предположить, что x имеет форму $|a|_0$ для $a : A$, и в этом случае $|b|_0 = \|f\|_0\,(|a|_0) = |f(a)|_0$, и, следовательно, $\|b = f(a)\|_{-1}$. Опять же, поскольку стягиваемость — это просто утверждение, можно предположить, что $b = f(a)$; отсюда и следует требуемый результат. □

Отображение f такое, что $\|f\|_0$ является биекцией, а $\pi_k(f)$ является биекцией для всех k, называется **∞-связным** или **слабой эквивалентностью**. Это эквивалентно вопросу о том, что f будет n-связным для всех n. Тип Z называется **∞-усеченным** или **гиперполным**, если индуцированное отображение

$$(- \circ f) : (B \to Z) \to (A \to Z)$$

является эквивалентностью всякий раз, когда f — ∞-связно, то есть если для Z каждое ∞-связное отображение является эквивалентностью. Затем, если мы хотим принять принцип Уайтхеда в качестве аксиомы, то можем использовать любую из следующих эквивалентных форм.

- Каждая ∞-связная функция является эквивалентностью.
- Каждый тип является ∞-усеченным.

В моделях высших топосов, ∞-усеченные типы образуют рефлективный подуниверсум в смысле §7.7 («гиперпополнение» $(\infty, 1)$-топоса), но неизвестно, верно ли это в целом.

Может быть неочевидно, что *существуют* какие-то типы, которые не являются n-типами для любого n, но на самом деле они есть. Действительно, в классической теории гомотопии $\mathbb{S}^n$ обладает этим свойством для любого $n \geqslant 2$. Мы еще не доказали этот факт в гомотопической теории типов, но существуют и другие типы, которые, как мы можем доказать, имеют «бесконечный уровень усечения».

Пример 8.8.6. Предположим, имеется $B : \mathbb{N} \to \mathcal{U}$ такой, что для каждого n, тип $B(n)$ содержит n-петлю, которая не адекватна n-кратной рефлексивности, скажем, $p_n : \Omega^n(B(n), b_n)$ с $p_n \neq \mathsf{refl}^n_{b_n}$ (например, можно было бы определить $B(n) :\equiv \mathbb{S}^n$, с p_n образом $1 : \mathbb{Z}$ при изоморфизме $\pi_n(\mathbb{S}^n) \cong \mathbb{Z}$). Рассмотрим $C :\equiv \prod_{(n:\mathbb{N})} B(n)$, с точкой $c : C$, определенной как $c(n) :\equiv b_n$. Поскольку пространства петель коммутируют с произведениями, для любого m имеем

$$\Omega^m(C, c) \simeq \prod_{(n:\mathbb{N})} \Omega^m(B(n), b_n)\,.$$

Согласно этой эквивалентности, refl^m_c соответствует функции $(n \mapsto \mathsf{refl}^m_{b_n})$. Теперь определим q_m в правом типе как

$$q_m(n) :\equiv \begin{cases} p_n & m = n\,, \\ \mathsf{refl}^m_{b_n} & m \neq n\,. \end{cases}$$

Если бы было $q_m = (n \mapsto \mathsf{refl}^m_{b_n})$, то имело бы место $p_n = \mathsf{refl}^n_{b_n}$, но это не так. Таким образом, $q_m \neq (n \mapsto \mathsf{refl}^m_{b_n})$, и поэтому существует точка $\Omega^m(C, c)$, которая не адекватна refl^m_c. Следовательно, C не является m-типом ни для какого $m : \mathbb{N}$.

Мы ожидаем, что также будет возможно показать, что универсум $\mathcal{U}$, сам по себе, не является n-типом для любого n, используя тот факт, что он содержит высшие индуктивные типы, такие, как $\mathbb{S}^n$, для всех n. Однако, этого пока еще не сделано.

8.9 Общая формулировка метода кодирования-декодирования

Мы использовали метод кодирования-декодирования для характеристики пространств путей различных типов, включая копроизведения (теорема 2.12.5), натуральные числа (теорема 2.13.1), усечения (теорема 7.3.12), окружность (следствие 8.1.10), надстройки (теорема 8.6.4) и амальгамы (теорема 8.7.12). Варианты этой техники используются в доказательствах многих других теорем, упомянутых во введении к этой главе, таких как прямое доказательство $\pi_n(\mathbb{S}^n)$, теорема Бейкерса-Месси и построение пространств Эйленберга-Маклейна. Хотя заманчиво попытаться абстрагировать метод в виде леммы, осуществить это сложно, потому что для разных задач требуются несколько различающиеся варианты. Например, различные варианты одного и того же метода могут быть использованы для характеристики пространства петель (как в теореме 2.12.5 и следствии 8.1.10) или всего пространства путей (как в теореме 2.13.1), чтобы дать полную характеризацию пространства петель (например, $\Omega^1(\mathbb{S}^1)$) или только для характеристики некоторого его усечения (например, ван Кампена), и для вычисления гомотопических групп или для доказательства того, что отображение n-связно (например, Фрейденталь и Бейкерс-Месси).

Однако, для конкретных вариантов метода можно сформулировать леммы. Доказательства этих лемм почти тривиальны, главное — уточнить метод, изложив сами доказательства лишь в общих чертах. В простейшем случае используется метод кодирования-декодирования для характеристики пространства петель, как в теореме 2.12.5 и следствии 8.1.10.

Лемма 8.9.1 (Кодирование-декодирование для пространств петель)**.** *Для точечного типа* (A, a_0) *и расслоения* $\mathsf{code} : A \to \mathcal{U}$*, если*

(i) $c_0 : \mathsf{code}(a_0)$*,*

(ii) $\mathsf{decode} : \prod_{(x:A)} \mathsf{code}(x) \to (a_0 = x)$*,*

(iii) для всех $c : \mathsf{code}(a_0)$*,* $\mathsf{transport}^{\mathsf{code}}(\mathsf{decode}(c), c_0) = c$*, и*

(iv) $\mathsf{decode}(c_0) = \mathsf{refl}$*,*

то $(a_0 = a_0)$ *эквивалентно* $\mathsf{code}(a_0)$.

Доказательство. Определим $\mathsf{encode} : \prod_{(x:A)}(a_0 = x) \to \mathsf{code}(x)$ как

$$\mathsf{encode}_x(\alpha) = \mathsf{transport}^{\mathsf{code}}(\alpha, c_0)\,.$$

Покажем, что encode_{a_0} и decode_{a_0} являются квазиобратными. Композиция $\mathsf{encode}_{a_0} \circ \mathsf{decode}_{a_0}$ является непосредственной согласно предположению (iii). Для другой композиции показываем

$$\prod_{(x:A)} \prod_{(p:a_0=x)} \mathsf{decode}_x(\mathsf{encode}_x p) = p\,.$$

Индукцией по пути достаточно показать, что $\mathsf{decode}_{a_0}(\mathsf{encode}_{a_0}\mathsf{refl}) = \mathsf{refl}$. После сокращения $\mathsf{transport}$, достаточно показать, что $\mathsf{decode}_{a_0}(c_0) = \mathsf{refl}$, что является предположением (iv). $\square$

Если требуется послойная эквивалентность между $(a_0 = -)$ и code, достаточно усилить условие (iii) до

$$\prod_{(x:A)} \prod_{(c:\mathsf{code}(x))} \mathsf{encode}_x(\mathsf{decode}_x c) = c\,.$$

Однако, для вычисления пространства циклов (например, $\Omega(\mathbb{S}^1)$), это более сильное предположение не требуется.

Другой вариант, который часто встречается при вычислении гомотопических групп, характеризует усечение пространства петель:

Лемма 8.9.2 (Кодирование-декодирование для усечения пространств петель)**.** *Пусть имеется точечный тип* (A, a_0) *и расслоение* $\mathsf{code} : A \to \mathcal{U}$*, где, для каждого* x*,* $\mathsf{code}(x)$ *является* k*-типом. Определим*

$$\mathsf{encode} : \prod_{(x:A)} \left\| a_0 = x \right\|_k \to \mathsf{code}(x)$$

путем рекурсии усечения (используя тот факт, что $\mathsf{code}(x)$ *является* k*-типом), отображая* $\alpha : a_0 = x$ *к* $\mathsf{transport}^{\mathsf{code}}(\alpha, c_0)$*. Предположим:*

(i) $c_0 : \mathsf{code}(a_0)$*,*

(ii) $\mathsf{decode} : \prod_{(x:A)} \mathsf{code}(x) \to \left\| a_0 = x \right\|_k$*,*

(iii) $\mathsf{encode}_{a_0}(\mathsf{decode}_{a_0} c) = c$ *для всех* $c : \mathsf{code}(a_0)$*, и*

(iv) $\mathsf{decode}(c_0) = |refl|$*.*

Тогда $\left\| a_0 = a_0 \right\|_k$ *эквивалентно* $\mathsf{code}(a_0)$*.*

Доказательство. То, что $\mathsf{decode} \circ \mathsf{encode}$ является тождеством, сразу следует из пункта (iii). Чтобы доказать $\mathsf{encode} \circ \mathsf{decode}$, мы сначала проводим индукцию усечения, с помощью которой достаточно показать

$$\prod_{(x:A)} \prod_{(p:a_0=x)} \mathsf{decode}_x(\mathsf{encode}_x(|p|_k)) = |p|_k\ .$$

Индукция усечения разрешена, потому что пути k-типа относятся к k-типу. Чтобы продемонстрировать этот тип, проводим индукцию по пути и, после сокращения encode, используем предположение (iv). □

8.10 Дополнительные результаты

В этой главе доказательства не приводятся, а результаты были получены в рамках гомотопической теории типов.

Теорема 8.10.1. *Существует* k *такое, что для всех* $n \geqslant 3$*,* $\pi_{n+1}(\mathbb{S}^n) = \mathbb{Z}_k$*.*

Примечания к доказательству. Доказательство состоит из вычисления $\pi_4(\mathbb{S}^3)$ с обращением к стабильности (следствие 8.6.15). В классической формулировке этого результата k равен 2. Хотя мы еще не проверили, что k на самом деле равен 2, наше вычисление $\pi_4(\mathbb{S}^3)$ является конструктивным, как и все остальные доказательства теорем этой главы (точнее, в нем не используются какие-либо дополнительные аксиомы, такие как LEM или AC, что делает его столь же конструктивным, как и унивалентность и высшие индуктивные типы). Таким образом, принимая во внимание вычислительную интерпретацию гомотопической теории типов, мы могли бы запустить доказательство на компьютере, чтобы убедиться, что k равен 2. Этот пример довольно интригующий, потому что это первое вычисление гомотопической группы, для которого нам не нужно было знать ответ заранее. □

Теорема 8.10.2 (Теорема Блейкерса-Месси)**.** *Пусть имеются отображения $f : C \to X$ и $g : C \to Y$. Рассматривая сначала амальгаму $X \sqcup^C Y$ для f и g, а затем обратный образ ее включений $\mathsf{inl} : X \to X \sqcup^C Y \leftarrow Y : \mathsf{inr}$, получаем индуцированное отображение $C \to X \times_{(X \sqcup^C Y)} Y$.*

Если f является i-связным, а g — j-связным, то это индуцированное отображение — $(i+j)$-связно. Другими словами, для любых точек $x : X$, $y : Y$, соответствующий слой $C_{x,y}$ для $(f,g) : C \to X \times Y$ дает приближение к пространству путей $\mathsf{inl}(x) =_{X \sqcup^C Y} \mathsf{inr}(y)$ в указанной амальгаме.

Следует отметить, что в классической алгебраической топологии теорема Блекерса-Месси часто формулируется в несколько иной форме, где отображения f и g заменяются включениями подкомплексов CW-комплексов, а гомотопические амальгамы и гомотопические обратные образы — объединением и пересечением, соответственно. Чтобы выразить теорему в гомотопической теории типов, мы должны заменить такие понятия гомотопически-инвариантными. Мы видели другой пример этого в теореме Ван Кампена (§8.7), где нам пришлось заменить объединение открытых подмножеств гомотопической амальгамой.

Теорема 8.10.3 (Пространства Эйленберга-МакЛейна)**.** *Для любой абелевой группы G и положительного целого числа n, существует n-тип $K(G,n)$ такой, что $\pi_n(K(G,n)) = G$ и $\pi_k(K(G,n)) = 0$ для $k \neq n$.*

Теорема 8.10.4 (Покрытие пространств)**.** *Для связного пространства A существует эквивалентность между покрывающими пространствами над A и множествами с действием $\pi_1(A)$.*

Примечания

Теоремы в этой главе являются стандартными результатами классической теории гомотопий; многие описаны в [Hat02]. В этих заметках мы рассматриваем развитие их новых синтетических доказательств в гомотопической теории типов. В таблице 8.2 перечислены такие теоретико-гомотопические теоремы и указано, проверено ли их доказательство компьютерными программами. Почти все эти результаты были получены в течение весеннего семестра IAS в 2013 году в рамках значительных совместных усилий. Многие люди внесли свой вклад в эти результаты, например, будучи главным автором доказательства, предлагая проблемы для работы, участвуя во многих дискуссиях и семинарах по этим проблемам или давая отзывы о результатах. Следующие люди были главными авторами первых доказательств вышеуказанных теорем в гомотопической теории типов. Если не указано иное, для теорем, которые были проверены компьютером, основные авторы также были первыми, кто формализовал доказательство с помощью программного помощника компьютерных доказательств.

Теорема	Статус
$\pi_1(\mathbb{S}^1)$	✓✓
$\pi_{k<n}(\mathbb{S}^n)$	✓✓
Длинная точная последовательность гомотопических групп	✓✓
Пространство расслоения Хопфа есть $\mathbb{S}^3$	✓
$\pi_2(\mathbb{S}^2)$	✓✓
$\pi_3(\mathbb{S}^2)$	✓
$\pi_n(\mathbb{S}^n)$	✓✓
$\pi_4(\mathbb{S}^3)$	✓
Теорема Фрейденталя о надстройке	✓✓
Теорема Бейкерса-Месси	✓✓
Пространства Эйленберга-Маклейна $K(G,n)$	✓✓
Теорема ван Кампена	✓✓
Покрывающие пространства	✓✓
Принцип Уайтхеда для n-типов	✓✓

Таблица 8.2. Теоремы из теории гомотопий, доказанные вручную (✓) и с помощью компьютера (✓✓)

- Шульман (Shulman) дал теоретико-гомотопическое вычисление $\pi_1(\mathbb{S}^1)$. Позже Ликата (Licata) обнаружил доказательство кодирования-декодирования и метод кодирования-декодирования.
- Брунери (Brunerie) вычислил $\pi_{k<n}(\mathbb{S}^n)$. Позже Ликата (Licata) предоставил версию для кодирования-декодирования.
- Воеводский (Voevodsky) построил длинную точную последовательность гомотопических групп.
- Ламсдейн (Lumsdaine) построил расслоение Хопфа. Брюнери (Brunerie) доказал, что его пространство расслоений есть $\mathbb{S}^3$, тем самым вычислив $\pi_2(\mathbb{S}^2)$ и $\pi_3(\mathbb{S}^3)$.
- Ликата (Licata) и Брунери (Brunerie) дали прямое вычисление $\pi_n(\mathbb{S}^n)$.
- Ламсдейн (Lumsdaine) доказал теорему Фрейденталя (Freudenthal) о надстройке; Ликата (Licata) и Ламсдэйн (Lumsdaine) формализовали это доказательство.
- Ламсдейн (Lumsdaine), Финстер (Finster) и Ликата (Licata) доказали теорему Бейкера-Месси (Blakers-Massey); Ламсдейн (Lumsdaine), Брунери (Brunerie), Ликата (Licata) и Хоу (Hou) формализовали его.
- Ликата (Licata) дал вычисление кодирования-декодирования $\pi_2(\mathbb{S}^2)$ и вычисление $\pi_n(\mathbb{S}^n)$ с использованием теоремы о надстройке Фрейденталя (Freudenthal); используя аналогичные методы, он построил $K(G,n)$.
- Шульман (Shulman) доказал теорему ван Кампена (van Kampen); Хоу (Hou) формализовал это доказательство.
- Ликата (Licata) доказал теорему Уайтхеда (Whitehead) для n-типов.
- Брунери (Brunerie) вычислил $\pi_4(\mathbb{S}^3)$.

- Хоу (Hou) основал и формализовал теорию покрывающих пространств.

Взаимодействие между теорией гомотопии и теорией типов имело решающее значение для развития этих результатов. Например, первое доказательство того, что $\pi_1(\mathbb{S}^1) = \mathbb{Z}$, было доказано в §8.1.5, которое следует классическому теоретическому доказательству гомотопии. Теоретико-типовой анализ этого доказательства привел к развитию метода кодирования-декодирования. Первое вычисление $\pi_2(\mathbb{S}^2)$ также следовало классическим методам, но быстро привело к доказательству кодирования-декодирования этого результата. Вычисление кодирования-декодирования было обобщенно на $\pi_n(\mathbb{S}^n)$, что, в свою очередь, привело к доказательству теоремы о надстройке Фрейденталя (Freudenthal) путем объединения аргумента кодирования-декодирования с классическими теоретико-гомотопическими рассуждениями о связности, что, в свою очередь, привело к теореме Блекерса-Месси (Blakers-Massey) и пространствам Эйленберга-Маклейна (Eilenberg-MacLane). Стремительная разработка этой серии результатов иллюстрирует перспективы нашего нового понимания связей между этими двумя темами.

Упражнения

Упражнение 8.1. Докажите, что гомотопические группы сохраняют произведения: $\pi_n(A \times B) \simeq \pi_n(A) \times \pi_n(B)$.

Упражнение 8.2. Докажите, что если A — множество с разрешимым равенством (см. определение 3.4.3), то его надстройка ΣA является 1-типом (открытый вопрос — доказуемо ли это без предположения разрешимого равенства).

Упражнение 8.3. Определите $\mathbb{S}^\infty$ как копредел последовательности $\mathbb{S}^0 \to \mathbb{S}^1 \to \mathbb{S}^2 \to \cdots$. Докажите, что $\mathbb{S}^\infty$ является стягиваемым.

Упражнение 8.4. Определите $\mathbb{S}^\infty$ как высший индуктивный тип, пораждаемый

- двумя точками $N : \mathbb{S}^\infty$, $\mathsf{S} : \mathbb{S}^\infty$, и
- путем $\mathsf{merid}(x) : \mathsf{N} = \mathsf{S}$, для каждого $x : \mathbb{S}^\infty$.

Другими словами, $\mathbb{S}^\infty$ — это собственная подвеска. Докажите, что $\mathbb{S}^\infty$ является стягиваемым.

Упражнение 8.5. Пусть $f : X \to Y$ — функция и Y — связно. Покажите, что для любых $y_1, y_2 : Y$ имеет место $\|\mathsf{fib}_f(y_1) \simeq \mathsf{fib}_f(y_2)\|$.

Упражнение 8.6. Для любого точечного типа A, пусть $i_A : \Omega A \to \Omega A$ обозначает инверсию петель, $i_A :\equiv \lambda p.\, p^{-1}$. Покажите, что $i_{\Omega A} : \Omega^2 A \to \Omega^2 A$ эквивалентно $\Omega(i_A)$.

Упражнение 8.7. Определите **точечную эквивалентность** как точечное отображение, основной функцией которого является эквивалентность.

(i) Покажите, что тип точечных эквивалентностей между точечными типами (X, x_0) и (Y, y_0) эквивалентен $(X, x_0) =_{\mathcal{U}_\bullet} (Y, y_0)$.

(ii) Переформулируйте понятие точечной эквивалентности в терминах точечных квазиобратных и точечных гомотопий в одном из когерентных стилей из главы 4.

Упражнение 8.8. Следуя примеру расслоения Хопфа в §8.5, определите **младшее расслоение Хопфа** как расслоение (то есть семейство типов) над $\mathbb{S}^1$, слой над отмеченной точкой которого есть $\mathbb{S}^0$, а пространство расслоений — $\mathbb{S}^1$. Это также называется «скрученным двойным накрытием» окружности $\mathbb{S}^1$.

Упражнение 8.9. Снова следуя примеру расслоения Хопфа в §8.5, определите аналогичное расслоение над $\mathbb{S}^4$, слой над отмеченной точкой которого есть $\mathbb{S}^3$, а пространство расслоений — $\mathbb{S}^7$. Это открытая проблема в гомотопической теории типов (такое расслоение, как известно, существует в классической теории гомотопий).

Упражнение 8.10. Продолжая пример 8.7.7, докажите, что если A имеет точку $a : A$, то можно отождествить $\pi_1(\Sigma A)$ с группой, представленной $\|A\|_0$, как образующими с отношением $|a|_0 = e$. Затем покажите, что если предполагается исключить середину, это также свободная группа на $\|A\|_0 \setminus \{|a|_0\}$.

Упражнение 8.11. Снова продолжая пример 8.7.7, но на этот раз не предполагая, что A — точечный, покажите, что можно отождествить $\pi_1(\Sigma A)$ с группой, представленной образующими $\|A\|_0 \times \|A\|_0$ и отношениями

$$(a,b) = (b,a)^{-1}, \qquad (a,c) = (a,b)\cdot(b,c), \qquad \text{и} \qquad (a,a) = e.$$

Глава 9

Теория категорий

Из разделов математики теория категорий, возможно, наименее удобна для теоретических основ. Одна из проблем состоит в том, что большая часть теории категорий инвариантна относительно более слабых понятий «тождественности», чем равенство, таких как изоморфизм в категории или эквивалентность категорий, в некотором смысле, который теория множеств не может уловить. Но это та же проблема, которую решает аксиома унивалентности для типов, отождествляя равенство с эквивалентностью. Таким образом, в унивалентных основаниях имеет смысл рассмотреть понятие «категория», в котором равенство объектов аналогичным образом отождествляется с изоморфизмом.

Если игнорировать проблемы с размером в математике, основанной на множествах, то категория состоит просто из *множества* A_0 объектов и, для любых $x, y \in A_0$, *множества* $\mathsf{hom}_A(x, y)$ морфизмов. При унивалентных основаниях «наивное» определение категории просто имитировало бы это с *типом* объектов и *типами* морфизмов. Если мы позволим этим типам содержать произвольную высшую гомотопию, то должны будем наложить высшие условия когерентности, что приведет к некоторому понятию $(\infty, 1)$-категории, но в настоящее время наша цель более скромная. Мы рассматриваем только 1-категории, и поэтому ограничиваем типы $\mathsf{hom}_A(x, y)$ множествами, т.е. 0-типами. Если мы не налагаем никаких дополнительных условий, то будем называть такую конструкцию *предкатегорией*.

Если добавить требование, чтобы тип A_0 объектов был множеством, то в итоге получим определение, которое ведет себя так же, как традиционное теоретико-множественное. Следуя Toby Bartels, мы называем это понятие *строгой категорией*. В качестве альтернативы, можно затребовать обобщенную версию аксиомы унивалентности, отождествляя $(x =_{A_0} y)$ с типом $\mathsf{iso}(x, y)$ изоморфизмов от x к y. Поскольку мы рассматриваем последний вариант как обычно «правильное» определение, будем называть его просто *категорией*.

Хорошим примером различия между тремя понятиями категории является утверждение «каждый вполне точный и эссенциально сюръективный функтор является эквивалентностью категорий», которое в классической теории категорий, основанной на множествах, эквивалентно аксиоме выбора.

(i) Для строгих категорий это по-прежнему эквивалентно аксиоме выбора.

(ii) Для предкатегорий не существует согласующейся аксиомы выбора, которая могла бы сделать ее истинной.

(iii) Для категорий это доказуемо *без* какой-либо аксиомы выбора.

Мы докажем последнее утверждение в этой главе, а также другие приятные свойства категорий, например что эквивалентные категории равны (как элементы типа категорий). Мы также опишем универсальный способ «насыщения» предкатегории A в категорию $\widehat{A}$, который мы называем ее *пополнением Резка* (Rezk completion), хотя также разумно его назвать *стековым пополнением* (stack completion; см. Примечания).

Пополнение Резка также проливает свет на понятие эквивалентности категорий. Например, функтор $A \to \widehat{A}$ всегда вполне точен и эссенциально сюръективен, отсюда «слабая эквивалентность». А это влечет то, что предкатегория является категорией именно тогда, когда она «рассматривает» все вполне точные и эссенциально сюръективные функторы как эквивалентности; таким образом, наше понятие «категория» уже заложено в понятие «вполне точный и эссенциально сюръективный функтор».

Мы предполагаем, что читатель знаком с классической теорией категорий. Напомним, что всякий раз, когда мы используем $\mathcal{U}$, это обозначает некий универсум типов, но, возможно, различный в разные моменты времени; все, что мы говорим, остается верным для любого согласующегося выбора уровней универсума. Мы будем использовать основные понятия гомотопической теории типов из глав 1 и 2 и пропозициональное усечение из главы 3, но не многое другое из части I, за исключением того, что наша вторая конструкция пополнения Резка будет использовать высший индуктивный тип.

9.1 Категории и предкатегории

В классической математике имеется ряд эквивалентных определений категории. В нашем случае, поскольку имеются зависимые типы, естественно выбрать стрелки в качестве семейства типов, индексируемых объектами. Это соответствует тому, как hom-типы всегда используются в теории категорий: мы даже не рассматриваем сравнение двух стрелок, если не знаем, что их домены и кодомены совпадают. Более того, кажется очевидным, что для теории 1-категорий все hom-типы должны быть множествами. Это приводит к следующему определению.

Определение 9.1.1. ПредкатегорияA содержит следующее.

(i) Тип A_0, элементы которого называются **объектами**. Будем писать $a : A$ для $a : A_0$.

(ii) Для любых $a, b : A$, множество $\mathrm{hom}_A(a, b)$, элементы которого называются **стрелками** или **морфизмами**.

(iii) Для каждого $a : A$, морфизм $1_a : \mathrm{hom}_A(a, a)$, называемый **тождественным морфизмом**.

(iv) Для любых $a, b, c : A$, функцию

$$\mathrm{hom}_A(b, c) \to \mathrm{hom}_A(a, b) \to \mathrm{hom}_A(a, c)\,,$$

называемую **композицией**, и означающей $g \mapsto f \mapsto g \circ f$, иногда обозначаемой как gf.

(v) Для любых $a, b : A$ и $f : \mathrm{hom}_A(a, b)$, имеем $f = 1_b \circ f$ и $f = f \circ 1_a$.

(vi) Для любых $a, b, c, d : A$ и

$$f : \mathrm{hom}_A(a, b), \qquad g : \mathrm{hom}_A(b, c), \qquad h : \mathrm{hom}_A(c, d),$$

имеет место $h \circ (g \circ f) = (h \circ g) \circ f$.

Проблема с понятием предкатегории состоит в том, что для объектов $a, b : A$, имеются два, возможно разных, понятия «одинаковости». С одной стороны, имеется тип $a =_{A_0} b$. Но с другой стороны, есть стандартное категорное понятие *изоморфизма*.

Определение 9.1.2. Морфизм $f : \hom_A(a, b)$ является **изоморфизмом**, если существует морфизм $g : \hom_A(b, a)$ такой, что $g \circ f = 1_a$ и $f \circ g = 1_b$. Будем обозначать через $a \cong b$ тип таких изоморфизмов.

Лемма 9.1.3. *Для любой $f : \hom_A(a, b)$, тип «f является изоморфизмом» является простым высказыванием. Следовательно, для любых $a, b : A$ тип $a \cong b$ является множеством.*

Доказательство. Пусть даны $g : \hom_A(b, a)$, $\eta : (1_a = g \circ f)$ и $\epsilon : (f \circ g = 1_b)$, и, аналогично, g', η' и ϵ'. Надо показать, что $(g, \eta, \epsilon) = (g', \eta', \epsilon')$. Но поскольку все $\hom$-совокупности являются множествами, их типы тождественности являются простыми высказываниями, поэтому достаточно показать, что $g = g'$. А для этого имеем

$$g' = 1_a \circ g' = (g \circ f) \circ g' = g \circ (f \circ g') = g \circ 1_b = g\,,$$

используя η и ϵ'. □

Если $f : a \cong b$, то будем писать f^{-1} для ее обратной, которая по лемме 9.1.3 однозначно определена.

Единственным отношением между этими двумя понятиями тождества, которые имеются в предкатегории, является следующее.

Лемма 9.1.4 (idtoiso)**.** *Если A — предкатегория и $a, b : A$, то*

$$(a = b) \to (a \cong b)\,.$$

Доказательство. Индукцией по тождественности можно считать, что a и b совпадают. Но тогда имеет место $1_a : \hom_A(a, a)$, что, очевидно, является изоморфизмом. □

Очевидно, эта ситуация аналогична той, которая побудила нас ввести аксиому унивалентности. Фактически мы имеем следующее:

Пример 9.1.5. Существует предкатегория $\mathcal{S}et$ с типами объектов Set и с $\hom_{\mathcal{S}et}(A, B) :\equiv (A \to B)$. Тождественными морфизмами являются тождественные функции, а композиция — это композиция функций. Для этой предкатегории лемма 9.1.4 эквивалентна отображению (сужение на множествах) idtoeqv из §2.10.

Конечно, если быть более пунктуальными, мы должны были обозначить эту категорию как $\mathcal{S}et_{\mathcal{U}}$, поскольку ее объекты — это только *малые множества* относительно универсума $\mathcal{U}$.

Так что, естественно дать следующее определение.

Определение 9.1.6. Категория — это предкатегория, такая что для всех $a, b : A$, функция $\mathsf{idtoiso}_{a,b}$ из леммы 9.1.4 является эквивалентностью.

В частности, в категории, если $a \cong b$, то $a = b$.

Пример 9.1.7. Из аксиомы унивалентности немедленно следует, что $\mathcal{S}et$ является категорией. Можно также показать, используя унивалентность, что любая предкатегория структур уровня множества, таких как группы, кольца, топологические пространства и т.д., является категорией; см. §9.8.

Отметим также следующее.

Лемма 9.1.8. *В категории, тип объектов является* 1*-типом.*

Доказательство. Достаточно показать, что для любых $a, b : A$, тип $a = b$ является множеством. Но $a = b$ эквивалентно $a \cong b$, которое является множеством. □

Будем записывать $\mathsf{isotoid}$ для обратного $(a \cong b) \to (a = b)$ отображения $\mathsf{idtoiso}$ из леммы 9.1.4. Следующее отношение между ними весьма важно.

Лемма 9.1.9. *Для* $p : a = a'$, $q : b = b'$ *и* $f : \hom_A(a, b)$, *имеем*

$$(p, q)_*(f) = \mathsf{idtoiso}(q) \circ f \circ \mathsf{idtoiso}(p)^{-1}. \tag{9.1.10}$$

Доказательство. По индукции можно считать, что p и q есть refl_a и refl_b, соответственно. Тогда левая часть (9.1.10) — это просто f. Но по определению, $\mathsf{idtoiso}(\mathsf{refl}_a)$ есть 1_a, а $\mathsf{idtoiso}(\mathsf{refl}_b)$ есть 1_b, поэтому правая часть (9.1.10) равна $1_b \circ f \circ 1_a$, что равно f. □

Аналогично можно показать

$$\mathsf{idtoiso}(p^{-1}) = (\mathsf{idtoiso}(p))^{-1} \tag{9.1.11}$$

$$\mathsf{idtoiso}(p \cdot q) = \mathsf{idtoiso}(q) \circ \mathsf{idtoiso}(p) \tag{9.1.12}$$

$$\mathsf{isotoid}(f \circ e) = \mathsf{isotoid}(e) \cdot \mathsf{isotoid}(f) \tag{9.1.13}$$

и т.д.

Пример 9.1.14. Предкатегория, в которой каждое множество $\hom_A(a, b)$ является простым высказыванием, эквивалентно типу A_0, снабженному простым отношением «$\leqslant$», которое является рефлексивным ($a \leqslant a$) и транзитивным (если $a \leqslant b$ и $b \leqslant c$, то $a \leqslant c$), называется **предпорядком**.

В предпорядке, $f : a \leqslant b$ является изоморфизмом только тогда, когда существует некоторое $g : b \leqslant a$. Таким образом, $a \cong b$ — это простое высказывание, выражающее то, что $a \leqslant b$ и $b \leqslant a$. Следовательно, предпорядок A является категорией только тогда, когда (1) каждый тип $a = b$ является простым высказыванием и (2) для любых $a, b : A_0$ существует функция $(a \cong b) \to (a = b)$. Другими словами, A_0 должен быть множеством, а $\leqslant$ должно быть антисимметричным (если $a \leqslant b$ и $b \leqslant a$, то $a = b$). Тогда это называется **(частичным) порядком** или **частично упорядоченным множеством**.

Пример 9.1.15. Если A — категория, то A_0 является множеством тогда и только тогда, когда для любых $a, b : A_0$, тип $a \cong b$ является простым высказыванием. Это эквивалентно тому, что любой изоморфизм в A является тождественностью; таким образом, это более сильное, чем классическое понятие «скелетальной» категории. Такие категории иногда называют **тощими** (gaunt) [BSP11]. На самом деле, не существует понятия «скелетальность» для наших категорий, если не считать таковым само определение 9.1.6.

Пример 9.1.16. Для любого 1-типа Xсуществует категория, с X в качестве типа объектов и с $\hom(x, y) :\equiv (x = y)$. Если X — множество, она называется **дискретной** категорией на X. Вообще, она называется **группоидом** (см. упражнение 9.6).

Пример 9.1.17. Для *любого* типа X, существует предкатегория, с X в качестве типа объектов и с $\hom(x, y) :\equiv \|x = y\|_0$. Операция композиции

$$\|y = z\|_0 \to \|x = y\|_0 \to \|x = z\|_0$$

определяется индукцией по усечению из конкатенации $(y = z) \to (x = y) \to (x = z)$. Мы называем такую предкатегорию **фундаментальным предгруппоидом** X (фактически, мы уже встречались с ним в §8.7; см. также упражнение 9.11).

Пример 9.1.18. Существует предкатегория с типом объектов $\mathcal{U}$, с $\hom(X, Y) :\equiv \|X = Y\|_0$ и композицией, определяемой индукцией по усечению из обычной композиции $(Y \to Z) \to (X \to Y) \to (X \to Z)$. Такую предкатегорию назовем **гомотопической прекатегорией типов**.

Пример 9.1.19. Пусть $\mathcal{R}el$ — следующая предкатегория.

- Ее объектами являются множества.
- $\hom_{\mathcal{R}el}(X, Y) = X \to Y \to \mathsf{Prop}$.
- Для множества X, имеет место $1_X(x, x') :\equiv (x = x')$.
- Для $R : \hom_{\mathcal{R}el}(X, Y)$ и $S : \hom_{\mathcal{R}el}(Y, Z)$, их композиция определяется как

$$(S \circ R)(x, z) :\equiv \left\| \sum_{(y:Y)} R(x, y) \times S(y, z) \right\| .$$

Предположим, что $R : \hom_{\mathcal{R}el}(X, Y)$ — изоморфизм, с обратным S. Отметим следующее.

(i) Если $R(x, y)$ и $S(y', x)$, то $(R \circ S)(y', y)$, и, следовательно, $y' = y$. Аналогично, если $R(x, y)$ и $S(y, x')$, то $x = x'$.

(ii) Для любого x, имеем $x = x$, так что, $(S \circ R)(x, x)$. Таким образом, просто существует $y : Y$ такой, что $R(x, y)$ и $S(y, x)$.

(iii) Предположим, что $R(x, y)$. По (ii) существует просто y' с $R(x, y')$ и $S(y', x)$. Но тогда, по (i), будет $y' = y$, и, следовательно, $y' = y$, поскольку Y — множество. Следовательно, транспортируя $S(y', x)$ по этому равенству, получим $S(y, x)$. В заключение, $R(x, y) \to S(y, x)$. Аналогично, $S(y, x) \to R(x, y)$.

(iv) Если $R(x, y)$ и $R(x, y')$, то, по (iii), $S(y', x)$, так что, по (i), $y = y'$. Таким образом, для любого x существует не более одного y такого, что $R(x, y)$. И, согласно (ii), такой y существует.

В заключение, если $R : \hom_{\mathcal{R}el}(X, Y)$ является изоморфизмом, то для каждого $x : X$ существует ровно один $y : Y$ такой, что $R(x, y)$, и верно двойственное утверждение. Таким образом, существует функция $f : X \to Y$ отсылающая каждый x к этому y, что является эквивалентностью; следовательно, $X = Y$. Приложив немного усилий, заключаем, что $\mathcal{R}el$ — это категория.

Теперь можно ограничиться рассмотрением категорий, а не предкатегорий. Вместо этого мы разработаем ряд концепций для предкатегорий и категорий, чтобы подчеркнуть, насколько лучше ведут себя категории по сравнению как с предкатегориями, так и с обычными категориями в классической математике.

Мы также увидим в §§ 9.6–9.7, что в несколько более экзотических контекстах существуют варианты использования определенных видов предкатегорий, отличных от категорий, каждая из которых «фиксирует» равенство объектов «по-своему». Это подчеркивает «предварительность» предкатегорий: они являются сырьем, из которого могут быть определены несколько важных категорных структур.

9.2 Функторы и преобразования

Следующие определения довольно очевидны и не нуждаются в изменении нотации.

Определение 9.2.1. Пусть A и B — предкатегории. **Функтор** $F : A \to B$ характеризуется следующим:

(i) функцией $F_0 : A_0 \to B_0$, обычно также обозначаемой F;

(ii) для любых $a, b : A$, функцией $F_{a,b} : \hom_A(a, b) \to \hom_B(Fa, Fb)$, обычно также обозначаемой F;

(iii) для каждого $a : A$, имеет место $F(1_a) = 1_{Fa}$;

(iv) для любых $a, b, c : A$ и $f : \hom_A(a, b)$ and $g : \hom_A(b, c)$, имеем

$$F(g \circ f) = Fg \circ Ff\,.$$

Заметьте, что индукцией по тождественности функтор также сохраняет idtoiso.

Определение 9.2.2. Для функторов $F, G : A \to B$, **естественное преобразование** $\gamma : F \to G$ характеризуется

(i) для каждого a:A, морфизмом $\gamma_a : \hom_B(Fa, Ga)$ («компоненты»);

(ii) для любых $a, b : A$ и $f : \hom_A(a, b)$, «аксиомой естественности» $Gf \circ \gamma_a = \gamma_b \circ Ff$.

Поскольку каждый тип $\hom_B(Fa, Gb)$ является множеством, его тип тождественности — это просто утверждение. Таким образом, аксиома естественности — это просто утверждение, поэтому тождественность естественных преобразований определяется тождественностью их компонент. В частности, для любых F и G, тип естественных преобразований от F к G снова является множеством.

Точно так же, тождественность функторов определяется тождественностью функций $A_0 \to B_0$ и (транспортированных по ним) соответствующих функций на $\hom$-множествах.

Определение 9.2.3. Для предкатегорий A, B, существует предкатегория B^A, называемая **предкатегорией функторов**, и определяемая как

- $(B^A)_0$ — тип функторов от A к B;

- $\hom_{B^A}(F,G)$ — тип естественных преобразований от F к G.

Обоснование определения. Определим $(1_F)_a :\equiv 1_{Fa}$. Естественность следует благодаря аксиомам единиц предкатегории. Для $\gamma : F \to G$ и $\delta : G \to H$ определим $(\delta \circ \gamma)_a :\equiv \delta_a \circ \gamma_a$. Естественность влечет ассоциативность. Точно так же, законы единицы и ассоциативности для B^A следуют из законов для B. □

Лемма 9.2.4. *Естественное преобразование $\gamma : F \to G$ является изоморфизмом в B^A тогда и только тогда, когда каждый компонент γ_a является изоморфизмом в B.*

Доказательство. Если γ — изоморфизм, то имеется $\delta : G \to F$, которое является его обратным. По определению композиции в B^A, $(\delta\gamma)_a \equiv \delta_a\gamma_a$ и аналогично для $\gamma\delta$. Таким образом, $\delta\gamma = 1_F$ и $\gamma\delta = 1_G$ подразумевают $\delta_a\gamma_a = 1_{Fa}$ и $\gamma_a\delta_a = 1_{Ga}$, поэтому γ_a — изоморфизм.

Наоборот, предположим, что каждый γ_a является изоморфизмом, с обратным, скажем, δ_a. Определим естественное преобразование $\delta : G \to F$ с компонентами δ_a; для аксиомы естественности имеем

$$Ff \circ \delta_a = \delta_b \circ \gamma_b \circ Ff \circ \delta_a = \delta_b \circ Gf \circ \gamma_a \circ \delta_a = \delta_b \circ Gf\,.$$

Теперь, поскольку композиция и тождественность естественных преобразований определены на их компонентах, то имеем $\gamma\delta = 1_G$ и $\delta\gamma = 1_F$. □

Следующий результат является основным.

Теорема 9.2.5. *Если A — предкатегория, а B — категория, то B^A — категория.*

Доказательство. Пусть $F, G : A \to B$; надо показать, что $\mathsf{idtoiso} : (F = G) \to (F \cong G)$ есть эквивалентность.

Чтобы предъявить обратную эквивалентность, предположим, что $\gamma : F \cong G$ — естественный изоморфизм. Тогда для любого $a : A$, имеется изоморфизм $\gamma_a : Fa \cong Ga$, следовательно, тождественность $\mathsf{isotoid}(\gamma_a) : Fa = Ga$. Согласно функциональной экстенсиональности, имеем тождественность $\bar{\gamma} : F_0 =_{(A_0 \to B_0)} G_0$.

Теперь, поскольку две последние аксиомы функтора являются простыми высказываниями, чтобы показать, что $F = G$, достаточно показать, что для любых $a, b : A$, функции

$$F_{a,b} : \hom_A(a,b) \to \hom_B(Fa, Fb) \qquad \text{и}$$
$$G_{a,b} : \hom_A(a,b) \to \hom_B(Ga, Gb)$$

становятся равными при транспортировании вдоль $\bar{\gamma}$. При вычислении функциональной экстенсиональности, когда она применяется к a, $\bar{\gamma}$ становится равным $\mathsf{isotoid}(\gamma_a)$. Но, по лемме 9.1.9, транспортирование $Ff : \hom_B(Fa, Fb)$ вдоль $\mathsf{isotoid}(\gamma_a)$ и $\mathsf{isotoid}(\gamma_b)$ эквивалентно компоновке $\gamma_b \circ Ff \circ (\gamma_a)^{-1}$, которая, по естественности γ, равна Gf.

Это завершает определение функции $(F \cong G) \to (F = G)$. Теперь рассмотрим композицию

$$(F = G) \to (F \cong G) \to (F = G)\,.$$

Поскольку $\hom$-наборы являются множествами, их типы тождественности являются простыми высказываниями, поэтому, чтобы показать, что две тождественности $p, q : F = G$ равны, достаточно показать, что $p =_{F_0 = G_0} q$. Но в определении $\bar{\gamma}$, если γ имеет форму $\mathsf{idtoiso}(p)$, то γ_a будет равно $\mathsf{idtoiso}(p_a)$ (это легко доказать индукцией по p). Таким образом, $\mathsf{isotoid}(\gamma_a)$ будет равно p_a, и поэтому, в силу функциональной экстенсиональности, будем иметь $\bar{\gamma} = p$, то и было нужно.

Наконец, рассмотрим композицию

$$(F \cong G) \to (F = G) \to (F \cong G)\,.$$

Поскольку тождественность естественных преобразований может быть проверена покомпонентно, достаточно показать, что для каждого a имеет место $\mathsf{idtoiso}(\bar{\gamma})_a = \gamma_a$. Но, как отмечалось выше, имеется $\mathsf{idtoiso}(\bar{\gamma})_a = \mathsf{idtoiso}((\bar{\gamma})_a)$, а $(\bar{\gamma})_a = \mathsf{isotoid}(\gamma_a)$ путем вычисления функциональной экстенсиональности. Поскольку $\mathsf{isotoid}$ и $\mathsf{idtoiso}$ — взаимообратные, имеем $\mathsf{idtoiso}(\bar{\gamma})_a = \gamma_a$, что и требовалось. □

В частности, естественно изоморфные функторы между категориями (в отличие от предкатегорий) равны.

Теперь определим все обычные способы композиции функторов и естественных преобразований.

Определение 9.2.6. Для функторов $F : A \to B$ и $G : B \to C$, их композиция $G \circ F : A \to C$ задается

- композицией $(G_0 \circ F_0) : A_0 \to C_0$;
- для любых $a, b : A$, композицией

$$(G_{Fa,Fb} \circ F_{a,b}) : \hom_A(a,b) \to \hom_C(GFa, GFb)\,.$$

Проверить выполнение соответствующих аксиом несложно.

Определение 9.2.7. Для функторов $F : A \to B$, $G, H : B \to C$ и естественного преобразования $\gamma : G \to H$, композиция $(\gamma F) : GF \to HF$ задается

- для каждого $a : A$, компонентой γ_{Fa}.

Естественность проверить несложно. Аналогично, для γ, как указано выше, и для $K : C \to D$, композиция $(K\gamma) : KG \to KH$ задается

- для каждого $b : B$, компонентой $K(\gamma_b)$.

Лемма 9.2.8. *Для функторов $F, G : A \to B$ и $H, K : B \to C$, и естественных преобразований $\gamma : F \to G$ и $\delta : H \to K$, имеет место*

$$(\delta G)(H\gamma) = (K\gamma)(\delta F)\,.$$

Доказательство. Достаточно проверить покомпонентно: при $a : A$ имеем

$$\begin{aligned}
((\delta G)(H\gamma))_a &\equiv (\delta G)_a (H\gamma)_a \\
&\equiv \delta_{Ga} \circ H(\gamma_a) \\
&= K(\gamma_a) \circ \delta_{Fa} && \text{(по естественности } \delta\text{)} \\
&\equiv (K\gamma)_a \circ (\delta F)_a \\
&\equiv ((K\gamma)(\delta F))_a\,.
\end{aligned}$$

□

Классически, это определяет «горизонтальную композицию» $\gamma : F \to G$ и $\delta : H \to K$ как общее значение $(\delta G)(H\gamma)$ и $(K\gamma)(\delta F)$. Мы воздержимся от этого, потому что, хотя эти две трансформации равны, они не равны по определению (*дефинициально*). Также, отсюда следует, что можно однозначно использовать символ $\circ$ (или сопоставление) для всех видов композиции: существует только один способ компоновки двух естественных преобразований (в отличие от компоновки естественного преобразования с функтором, с обеих сторон).

Лемма 9.2.9. *Композиция функторов ассоциативна: $H(GF) = (HG)F$.*

Доказательство. Поскольку композиция функций ассоциативна, это немедленно следует для действий над объектами и на всех $\hom$. А поскольку $\hom$-совокупности являются множествами, к остальным данным это относится автоматически. $\square$

Равенство в лемме 9.2.9 также не является дефинициальным (композиция функций дефинициально ассоциативна, но аксиомы, поддерживающие функтор, также должны быть скомпонованы, а это нарушает дефинициальную ассоциативность). По этой причине необходимо быть убежденным в *согласованности* ассоциативности.

Лемма 9.2.10. *Лемма 9.2.9 согласована, т.е. следующий пятиугольник равенств является коммутативным:*

$$\begin{tikzcd}[column sep=tiny]
 & K(H(GF)) \arrow[dl, equal] \arrow[dr, equal] & \\
(KH)(GF) \arrow[d, equal] & & K((HG)F) \arrow[d, equal] \\
((KH)G)F \arrow[rr, equal] & & (K(HG))F
\end{tikzcd}$$

Доказательство. Как и в лемме 9.2.9, это очевидно для действий над объектами, а все остальное следует автоматически. $\square$

В дальнейшем мы будем злоупотреблять обозначениями, записывая $H \circ G \circ F$ или HGF вместо $H(GF)$ либо $(HG)F$, при необходимости транспортируя их по лемме 9.2.9. Также имеется аналогичный результат согласованности и для единиц.

Лемма 9.2.11. *Для функтора $F : A \to B$ имеются равенства $(1_B \circ F) = F$ и $(F \circ 1_A) = F$такие, что, также и для $G : B \to C$, следующий треугольник равенств является коммутативным.*

$$\begin{tikzcd}[column sep=tiny]
G \circ (1_B \circ F) \arrow[rr, equal] \arrow[dr, equal] & & (G \circ 1_B) \circ F \arrow[dl, equal] \\
 & G \circ F &
\end{tikzcd}$$

См. упражнения 9.4 и 9.5 для ознакомления с дальнейшим развитием этих идей.

9.3 Сопряжения

Определение сопряженных функторов несложно; главный интересный аспект возникает из релевантности доказательства.

Определение 9.3.1. Функтор $F : A \to B$ является **левым сопряженным**, если существуют

- функтор $G : B \to A$;
- естественное преобразование $\eta : 1_A \to GF$ (**единица**);
- естественное преобразование $\epsilon : FG \to 1_B$ (**коединица**);
- $(\epsilon F)(F\eta) = 1_F$;
- $(G\epsilon)(\eta G) = 1_G$.

Последние два уравнения называются **треугольными тождествами** или **зигзагообразными тождествами**. Мы предоставляем читателю формулировку аналогичного определения правых сопряженных.

Лемма 9.3.2. *Если A — категория (но B может быть только предкатегорией), то тип «F — левый сопряженный» является простым высказыванием.*

Доказательство. Предположим, что даны: (G, η, ϵ) с треугольными тождествами, а также (G', η', ϵ'). Определим $\gamma : G \to G'$ как $(G'\epsilon)(\eta' G)$, а $\delta : G' \to G$ как $(G\epsilon')(\eta G')$. Тогда

$$\begin{aligned}
\delta\gamma &= (G\epsilon')(\eta G')(G'\epsilon)(\eta' G) \\
&= (G\epsilon')(GFG'\epsilon)(\eta G'FG)(\eta' G) \\
&= (G\epsilon)(G\epsilon' FG)(GF\eta' G)(\eta G) \\
&= (G\epsilon)(\eta G) \\
&= 1_G
\end{aligned}$$

используя лемму 9.2.8 и треугольные тождества. Точно так же получается $\gamma\delta = 1_{G'}$, так что, γ — естественный изоморфизм $G \cong G'$. По теореме 9.2.5, имеем тождественность $G = G'$.

Теперь нужно знать, что когда η и ϵ транспортируются по этому тождеству, они становятся равными η' и ϵ'. По лемме 9.1.9 эта транспортировка задается композицией с γ или δ в зависимости от ситуации. Для η это дает

$$(G'\epsilon F)(\eta' GF)\eta = (G'\epsilon F)(G'F\eta)\eta' = \eta'$$

используя лемму 9.2.8 и треугольное тождество. Случай с ϵ аналогичен. Наконец, треугольные тождества автоматически транспортируются корректно, поскольку $\hom$-наборы являются множествами. $\square$

В §9.5 мы приведем еще одно доказательство леммы 9.3.2.

9.4 Эквивалентности

В теории категорий принято определять *эквивалентность категорий* как функтор $F : A \to B$ такой, что существует функтор $G : B \to A$ и естественные изоморфизмы $FG \cong 1_B$ и $GF \cong 1_A$. Однако, в отличие от свойства быть сопряжением, это не было бы простым высказыванием без его усечения, по тем же причинам, по которым тип квази-обратных обладает некорректным поведением (см. §4.1). Но, как и в §4.2, этого можно избежать, используя обычное понятие *сопряженной* эквивалентности.

Определение 9.4.1. Функтор $F : A \to B$ является **эквивалентностью (пред) категорий**, если он является левым сопряженным, для которого η и ϵ — изоморфизмы. Через $A \simeq B$ обозначим тип эквивалентности категорий от A к B.

По леммам 9.1.3 и 9.3.2, если A — категория, то тип «F — эквивалентность предкатегорий» является простым высказыванием.

Лемма 9.4.2. *Если для $F : A \to B$ существуют $G : B \to A$ и изоморфизмы $GF \cong 1_A$, $FG \cong 1_B$, то F — эквивалентность предкатегорий.*

Доказательство. Такое же, как доказательство теоремы 4.2.3 для эквивалентностей типов. □

Определение 9.4.3. Говорят, что функтор $F : A \to B$ является **точным**, если, для всех $a, b : A$, функция

$$F_{a,b} : \hom_A(a, b) \to \hom_B(Fa, Fb)$$

является инъективной, и **полным**, если, для всех $a, b : A$, эта функция сюръективна. Если она, и инъективна, и сюръективна, (следовательно, каждая $F_{a,b}$ является эквивалентностью), то говорят, что функтор F **вполне точен**.

Определение 9.4.4. Говорят, что функтор $F : A \to B$ является **расщепляемым эссенциально сюръективным**, если для всех $b : B$ существует $a : A$ такой, что $Fa \cong b$.

Лемма 9.4.5. *Для любых предкатегорий A, B и функтора $F : A \to B$, следующие типы эквивалентны.*

(i) F является эквивалентностью предкатегорий.

(ii) F вполне точен и является расщепляемым эссенциально сюръективным.

Доказательство. Предположим, что F — эквивалентность предкатегорий, с заданными G, η, ϵ. Тогда имеется функция

$$\begin{aligned} \hom_B(Fa, Fb) &\to \hom_A(a, b)\,, \\ g &\mapsto {\eta_b}^{-1} \circ G(g) \circ \eta_a\,. \end{aligned}$$

Для $f : \hom_A(a, b)$, имеем

$${\eta_b}^{-1} \circ G(F(f)) \circ \eta_a = {\eta_b}^{-1} \circ \eta_b \circ f = f$$

а для $g : \hom_B(Fa, Fb)$ —

$$\begin{aligned} F({\eta_b}^{-1} \circ G(g) \circ \eta_a) &= F({\eta_b}^{-1}) \circ F(G(g)) \circ F(\eta_a) \\ &= \epsilon_{Fb} \circ F(G(g)) \circ F(\eta_a) \\ &= g \circ \epsilon_{Fa} \circ F(\eta_a) \\ &= g \end{aligned}$$

используя естественность ϵ, и (дважды) треугольные тождества. Таким образом, $F_{a,b}$ — эквивалентность, поэтому F вполне точен. Наконец, для любого $b : B$, имеем $Gb : A$ и $\epsilon_b : FGb \cong b$.

С другой стороны, предположим, что F вполне точен и является расщепляемым эссенциально сюръективным. Определим $G_0 : B_0 \to A_0$, отсылая $b : B$ к $a : A$, задаваемое указанным эссенциальным расщеплением, и записывая ϵ_b для аналогично указанного изоморфизма $FGb \cong b$.

Теперь для любого $g : \hom_B(b, b')$, определим $G(g) : \hom_A(Gb, Gb')$ как единственный морфизм, такой что $F(G(g)) = (\epsilon_{b'})^{-1} \circ g \circ \epsilon_b$ (который существует, поскольку F вполне точен). Наконец, для $a : A$ определим $\eta_a : \hom_A(a, GFa)$ как единственный морфизм такой, что $F\eta_a = {\epsilon_{Fa}}^{-1}$. Просто проверить, что G — функтор и что (G, η, ϵ) характеризует F как эквивалентность предкатегорий.

Теперь рассмотрим компоновку (i)→(ii)→(i). Мы явно восстанавливаем ту же функцию $G_0 : B_0 \to A_0$. Для действия G на hom-множествах надо показать, что для $g : \hom_B(b, b')$, $G(g)$ является (обязательно уникальным) морфизмом, таким что $F(G(g)) = (\epsilon_{b'})^{-1} \circ g \circ \epsilon_b$. Но это выражение выполняется ввиду предполагаемой естественности ϵ. Мы также явно восстанавливаем ϵ, в то время как η однозначно характеризуется $F\eta_a = {\epsilon_{Fa}}^{-1}$ (что является одним из треугольных тождеств, которые, как предполагается, выполняются в структуре эквивалентности предкатегорий). Таким образом, рассмотренная компоновка эквивалентна тождественности.

Наконец, рассмотрим другую компоновку (ii)→(i)→(ii). Поскольку, чтобы быть вполне точным — это простое высказывание, достаточно заметить, что мы восстанавливаем, для каждого $b : B$, то же самое $a : A$ и изоморфизм $Fa \cong b$. Но это ясно, поскольку мы использовали эту функцию и изоморфизм для определения G_0 и ϵ в (i), которые, в свою очередь, являются именно тем, что мы снова использовали для восстановления (ii). Таким образом, эти компоновки в обоих направлениях эквивалентны тождественностям, следовательно, имеем эквивалентность (i)≃(ii). □

Однако, если A не является категорией, то ни один из типов в лемме 9.4.5 не обязательно может быть простым высказыванием. Это заставляет задуматься и над следующими понятиями.

Определение 9.4.6. Функтор $F : A \to B$ **эссенциально сюръективен**, если, для всех $b : B$, *просто* существует $a : A$, такое, что $Fa \cong b$. Говорят, что F является **слабой эквивалентностью**, если он вполне точен и эссенциально сюръективен.

Быть слабой эквивалентностью — это *всегда* просто утверждение. Однако, для категорий нет разницы между эквивалентностями и слабыми категориями.

Лемма 9.4.7. *Если $F : A \to B$ вполне точен и A является категорией, то для любого $b : B$ тип $\sum_{a:A}(Fa \cong b)$ является простым высказыванием. Следовательно, функтор между категориями является эквивалентностью тогда и только тогда, когда он является слабой эквивалентностью.*

Доказательство. Предположим, что заданы (a, f) и (a', f') в $\sum_{a:A}(Fa \cong b)$. Тогда ${f'}^{-1} \circ f$ является изоморфизмом $Fa \cong Fa'$. Поскольку F вполне точен, имеем $g : a \cong a'$ с $Fg = {f'}^{-1} \circ f$. А поскольку A — категория, то $p : a = a'$ с $\mathsf{idtoiso}(p) = g$. Теперь, $Fg = {f'}^{-1} \circ f$ влечет $((F_0)(p))_*(f) = f'$, следовательно (характеризуя равенства в зависимых парных типах), $(a, f) = (a', f')$.

Таким образом, для вполне точных функторов, область определения которых является категорией, эссенциальная сюръективность эквивалентна расщепленной эссенциальной сюръективности, и поэтому, быть слабой эквивалентностью равносильно тому, чтобы быть эквивалентностью. □

Имеется важное преимущество нашей теории категорий перед подходами, основанными на множествах. При чисто основанном на множестве определении категории, утверждение «каждый вполне точный и эссенциально сюръективный функтор является эквивалентностью категорий» эквивалентно аксиоме выбора AC. У нас же, она открыто предоставляется как теоретико-категорная версия принципа единственного выбора (§3.9; фактически, это свойство характеризует категории среди предкатегорий; см. §9.9).

С другой стороны, следующая характеристика эквивалентности категорий, возможно, даже более полезна.

Определение 9.4.8. Функтор $F : A \to B$ является **изоморфизмом (пред)категорий**, если F вполне точен и $F_0 : A_0 \to B_0$ является эквивалентностью типов.

Это определение является исключением из общего правила (см. §2.4) использования слова «изоморфизм» только для множеств и подобных множествам объектов. Однако здесь он имеет соответствующий оттенок, потому что для общих предкатегорий изоморфизм сильнее эквивалентности.

Заметьте, что быть изоморфизмом предкатегорий — это всегда просто свойство. Обозначим через $A \cong B$ тип изоморфизмов (пред)категорий от A к B.

Лемма 9.4.9. *Для предкатегорий A, B и функтора $F : A \to B$, следующие утверждения эквивалентны.*

(i) F — это изоморфизм предкатегорий.

(ii) Существуют $G : B \to A$, $\eta : 1_A = GF$ и $\epsilon : FG = 1_B$ такие, что

$$\mathsf{ap}_{(\lambda H.\, FH)}(\eta) = \mathsf{ap}_{(\lambda K.\, KF)}(\epsilon^{-1}) \,. \tag{9.4.10}$$

(iii) Просто существуют $G : B \to A$, $\eta : 1_A = GF$ и $\epsilon : FG = 1_B$.

Отметим, что если B_0 не является 1-типом, то (9.4.10) не может быть простым высказыванием.

Доказательство. Прежде всего отметим, что поскольку $\hom$-наборы являются множествами, равенства между равенствами функторов однозначно определяются их частями-объектами. Таким образом, в силу функциональной экстенсиональности, (9.4.10) эквивалентно

$$(F_0)(\eta_0)_a = (\epsilon_0)^{-1}{}_{F_0 a} \,. \tag{9.4.11}$$

для всех $a : A_0$. Заметьте, что это в точности треугольное тождество для G_0, η_0 и ϵ_0, которое служит доказательством того, что F_0 является половиной сопряженной эквивалентности типов.

Теперь предположим, что верно (i). Пусть $G_0 : B_0 \to A_0$ будет обратным к F_0, с $\eta_0 : \mathsf{id}_{A_0} = G_0 F_0$ и $\epsilon_0 : F_0 G_0 = \mathsf{id}_{B_0}$, удовлетворяющие тождеству треугольника, что в точности соответствует (9.4.11). Определим $G_{b,b'} : \hom_B(b,b') \to \hom_A(G_0 b, G_0 b')$ как

$$G_{b,b'}(g) :\equiv (F_{G_0 b, G_0 b'})^{-1}\Big(\mathsf{idtoiso}((\epsilon_0)^{-1}{}_{b'}) \circ g \circ \mathsf{idtoiso}((\epsilon_0)_b)\Big)$$

(в предположении, что F вполне точен). Поскольку $\mathsf{idtoiso}$ переводит обратное в обратное, а конкатенацию в композицию, и F — функтор, то отсюда следует, что G — функтор.

По определению, имеем $(GF)_0 \equiv G_0 F_0$, что эквивалентно id_{A_0} посредством η_0. Чтобы получить $1_A = GF$, нужно показать, что при транспортировании по η_0, тождественная функция $\hom_A(a,a')$ становится равной составной $G_{Fa,Fa'} \circ F_{a,a'}$. Другими словами, для любой $f : \hom_A(a,a')$, мы должны иметь

$$\begin{aligned}
&\mathsf{idtoiso}((\eta_0)_{a'}) \circ f \circ \mathsf{idtoiso}((\eta_0)^{-1}{}_a) \\
&\qquad = (F_{GFa, GFa'})^{-1}\Big(\mathsf{idtoiso}((\epsilon_0)^{-1}{}_{Fa'}) \circ F_{a,a'}(f) \circ \mathsf{idtoiso}((\epsilon_0)_{Fa})\Big) \,.
\end{aligned}$$

Но это эквивалентно

$$(F_{GFa,GFa'})\Big(\mathsf{idtoiso}((\eta_0)_{a'}) \circ f \circ \mathsf{idtoiso}({(\eta_0)^{-1}}_a)\Big) \\ = \mathsf{idtoiso}({(\epsilon_0)^{-1}}_{Fa'}) \circ F_{a,a'}(f) \circ \mathsf{idtoiso}((\epsilon_0)_{Fa})\,.$$

что следует из функториальности F, того факта, что F сохраняет idtoiso, и (9.4.11). Таким образом, имеем $\eta : 1_A = GF$.

С другой стороны, имеем $(FG)_0 \equiv F_0 G_0$, что равно id_{B_0} посредством ϵ_0. Чтобы получить $FG = 1_B$, нужно показать, что при транспортировании вдоль ϵ_0, тождественная функция $\hom_B(b, b')$ становится равна составной $F_{Gb,Gb'} \circ G_{b,b'}$. То есть, для любой $g : \hom_B(b, b')$, мы должны иметь

$$F_{Gb,Gb'}\Big((F_{Gb,Gb'})^{-1}\Big(\mathsf{idtoiso}({(\epsilon_0)^{-1}}_{b'}) \circ g \circ \mathsf{idtoiso}((\epsilon_0)_b)\Big)\Big) \\ = \mathsf{idtoiso}(({\epsilon_0}^{-1})_{b'}) \circ g \circ \mathsf{idtoiso}((\epsilon_0)_b)\,.$$

Фактически, $(F_{Gb,Gb'})^{-1}$ является обратным к $F_{Gb,Gb'}$. И мы замечаем, что (9.4.10) эквивалентно (9.4.11), поэтому выполняется (ii).

Обратно, предположим, что верно (ii); тогда части-объекты G, η и ϵ вместе с (9.4.11) показывают, что F_0 является эквивалентностью типов. А для $a, a' : A_0$ определим $\overline{G}_{a,a'} : \hom_B(Fa, Fa') \to \hom_A(a, a')$ как

$$\overline{G}_{a,a'}(g) :\equiv \mathsf{idtoiso}(\eta^{-1})_{a'} \circ G(g) \circ \mathsf{idtoiso}(\eta)_a\,. \tag{9.4.12}$$

По естественности $\mathsf{idtoiso}(\eta)$, для любой $f : \hom_A(a, a')$, имеем

$$\begin{aligned}\overline{G}_{a,a'}(F_{a,a'}(f)) &= \mathsf{idtoiso}(\eta^{-1})_{a'} \circ G(F(f)) \circ \mathsf{idtoiso}(\eta)_a \\ &= \mathsf{idtoiso}(\eta^{-1})_{a'} \circ \mathsf{idtoiso}(\eta)_{a'} \circ f \\ &= f\,.\end{aligned}$$

С другой стороны, для $g : \hom_B(Fa, Fa')$ имеем

$$\begin{aligned}F_{a,a'}(\overline{G}_{a,a'}(g)) &= F(\mathsf{idtoiso}(\eta^{-1})_{a'}) \circ F(G(g)) \circ F(\mathsf{idtoiso}(\eta)_a) \\ &= \mathsf{idtoiso}(\epsilon)_{Fa'} \circ F(G(g)) \circ \mathsf{idtoiso}(\epsilon^{-1})_{Fa} \\ &= \mathsf{idtoiso}(\epsilon)_{Fa'} \circ \mathsf{idtoiso}(\epsilon^{-1})_{Fa'} \circ g \\ &= g\end{aligned}$$

(здесь необходимы леммы относительно совместимости idtoiso и вискеринга, которые оставлены читателю для их формулировки и доказательства). Таким образом, $F_{a,a'}$ — эквивалентность, поэтому F вполне точен; то есть (i) выполняется.

Теперь, компоновка (i)→(ii)→(i) эквивалентна тождеству, поскольку (i) является простым высказыванием. С другой стороны, прослеживая вышеупомянутые конструкции, ясно, что компоновка (ii)→(i)→(ii), по существу, сохраняет части-объекты G_0, η_0, ϵ_0 и часть-объект (9.4.10). И в последних трех случаях часть-объект — это все, что имеется, поскольку hom-наборы — это множества.

Таким образом, достаточно показать, что мы восстанавливаем действие G на hom-множествах. Другими словами, надо показать, что если $g : \hom_B(b, b')$, то

$$G_{b,b'}(g) = \overline{G}_{G_0 b, G_0 b'}\Big(\mathsf{idtoiso}({(\epsilon_0)^{-1}}_{b'}) \circ g \circ \mathsf{idtoiso}((\epsilon_0)_b)\Big)$$

где $\overline{G}$ определено в (9.4.12). Однако, что следует из функториальности G и другого треугольного тождества, которое мы видели в главе 4, это эквивалентно (9.4.11).

Поскольку (i) — простое высказывание, как и (ii), то достаточно показать, что они логически эквивалентны (iii). Конечно, (ii)$\to$(iii), поэтому допустим, что верно (iii). Поскольку (i) — простое высказывание, можно считать заданными G, η и ϵ. Тогда G_0 вместе с η и ϵ подразумевают, что F_0 является эквивалентностью. Более того, также имеются естественные изоморфизмы $\mathsf{idtoiso}(\eta) : 1_A \cong GF$ и $\mathsf{idtoiso}(\epsilon) : FG \cong 1_B$, поэтому по лемме 9.4.2, функтор F является эквивалентностью предкатегорий и, в частности, вполне точен. □

Из леммы 9.4.9 (ii) и idtoiso в категориях функторов, немедленно заключаем, что любой изоморфизм предкатегорий является эквивалентностью. Для предкатегорий обратное может не выполняться.

Пример 9.4.13. Пусть X — тип, а $x_0 : X$ — элемент, и пусть X_{ch} обозначает *хаотическую* или *недискретную* предкатегорию на X. По определению, имеем $(X_{\mathrm{ch}})_0 :\equiv X$ и $\hom_{X_{\mathrm{ch}}}(x, x') :\equiv \mathbf{1}$, для всех x, x'. Тогда единственный функтор $X_{\mathrm{ch}} \to \mathbf{1}$ является эквивалентностью предкатегорий, но не изоморфизмом, если X не стягиваемо.

Этот пример также показывает, что предкатегория может быть эквивалентна категории, но сама по себе не является категорией. Конечно, если предкатегория *изоморфна* категории, тогда она должна быть категорией.

Однако, для категорий эти два понятия совпадают.

Лемма 9.4.14. *Для категорий A и B, функтор $F : A \to B$ является эквивалентностью категорий тогда и только тогда, когда он является изоморфизмом категорий.*

Доказательство. Поскольку оба понятия являются простыми свойствами, достаточно показать, что они логически эквивалентны. Сначала предположим, что F — эквивалентность категорий с заданными (G, η, ϵ). Мы уже знаем, что F вполне точен. По теореме 9.2.5, естественные изоморфизмы η и ϵ дают тождества $1_A = GF$ и $FG = 1_B$, следовательно, в частности, и тождества $\mathsf{id}_A = G_0 \circ F_0$, $F_0 \circ G_0 = id_B$. Таким образом, F_0 — эквивалентность типов.

Наоборот, предположим, что F вполне точен, а F_0 эквивалентен типам, скажем, с обратным G_0. Тогда для каждого $b : B$ имеем $G_0 b : A$ и тождество $FGb = b$, а следовательно, изоморфизм $FGb \cong b$. Таким образом, по лемме 9.4.5, F — эквивалентность категорий. □

Конечно, есть еще третье понятие одинаковости для (пред)категорий: равенство. Однако, из аксиомы унивалентности следует, что оно совпадает с изоморфизмом.

Лемма 9.4.15. *Если A и B - предкатегории, то функция*

$$(A = B) \to (A \cong B)$$

(определенная индукцией по тождественному функтору) является эквивалентностью типов.

Доказательство. Как обычно, для зависимых типов сумм, задание элемента для $A = B$ эквивалентно заданию

- тождественности $P_0 : A_0 = B_0$,
- для любых $a, b : A_0$, тождественности

$$P_{a,b} : \hom_A(a, b) = \hom_B(P_{0*}(a), P_{0*}(b)),$$

- тождественностям $(P_{a,a})_*(1_a) = 1_{P_{0*}(a)}$ и $(P_{a,c})_*(gf) = (P_{b,c})_*(g) \circ (P_{a,b})_*(f)$

(снова используется тот факт, что типы тождественности $\hom$-множеств являются простыми высказываниями). Однако, по унивалентности, это равносильно

- эквивалентности типов $F_0 : A_0 \simeq B_0$,
- для любых $a, b : A_0$, эквивалентности типов

$$F_{a,b} : \hom_A(a,b) \simeq \hom_B(F_0(a), F_0(b))\,,$$

- тождественностям $F_{a,a}(1_a) = 1_{F_0(a)}$ и $F_{a,c}(gf) = F_{b,c}(g) \circ F_{a,b}(f)$.

Но эта функция — в точности функтор $F : A \to B$, который является изоморфизмом категорий. И, индукцией по тождеству, эквивалентность $(A = B) \simeq (A \cong B)$ есть эквивалентность, полученная по индукции. □

Таким образом, для категорий равенство также совпадает с эквивалентностью. Можно интерпретировать это как утверждение, что категории, функторы и естественные преобразования образуют не только пред-2-категорию, но и 2-категорию (см. упражнение 9.4).

Теорема 9.4.16. *Если A и B — категории, то функция*

$$(A = B) \to (A \simeq B)$$

(определенная индукцией по тождественному функтору) является эквивалентностью типов.

Доказательство. По леммам 9.4.14 и 9.4.15. □

Как следствие, тип категорий является 2-типом. Поскольку $A \simeq B$ является подтипом типа функторов от A к B, которые являются объектами категории, это 1-тип; следовательно, тождественные типы $A = B$ также являются 1-типами.

9.5 Лемма Йонеды

Напомним, что имеется категория $\mathcal{S}et$, объектами которой являются множества, а морфизмами — функции. Теперь покажем, что каждая предкатегория имеет $\mathcal{S}et$-значный $\hom$-функтор. Сначала нужно определить обратные и произведения (пред)категорий.

Определение 9.5.1. Для предкатегории A, **обратная** к ней, A^{op}, является предкатегорией с объектами того же типа, с $\hom_{A^{\mathrm{op}}}(a,b) :\equiv \hom_A(b,a)$ и с тождественностями и композицией, унаследованными от A.

Определение 9.5.2. Для предкатегорий A и B, их **произведение** $A \times B$ является предкатегорией с $(A \times B)_0 :\equiv A_0 \times B_0$ и

$$\hom_{A\times B}((a,b),(a',b')) :\equiv \hom_A(a,a') \times \hom_B(b,b').$$

Тождественности определяются как $1_{(a,b)} :\equiv (1_a, 1_b)$, а композиция — как $(g,g')(f,f') :\equiv ((gf),(g'f'))$.

Лемма 9.5.3. *Для предкатегорий A, B, C, следующие типы эквивалентны:*

(i) *функторы* $A \times B \to C$,

(ii) *функторы* $A \to C^B$.

Доказательство. Для $F : A \times B \to C$и любого $a : A$, очевидно, имеется функтор $F_a : B \to C$. Это дает функцию $A_0 \to (C^B)_0$. Далее, для любого $f : \hom_A(a, a')$, имеем, для любого $b : B$ морфизм $F_{(a,b),(a',b)}(f, 1_b) : F_a(b) \to F_{a'}(b)$. Это компоненты естественного преобразования $F_a \to F_{a'}$. Функториальность легко проверить, поэтому имеется функтор $\hat{F} : A \to C^B$.

Наоборот, предположим, что дано $G : A \to C^B$. Тогда, для любых $a : A$ и $b : B$, имеется объект $G(a)(b) : C$, дающий функцию $A_0 \times B_0 \to C_0$. А для $f : \hom_A(a, a')$ и $g : \hom_B(b, b')$, имеется морфизм

$$G(a')_{b,b'}(g) \circ G_{a,a'}(f)_b = G_{a,a'}(f)_{b'} \circ G(a)_{b,b'}(g)$$

в $\hom_C(G(a)(b), G(a')(b'))$. Функториальность снова легко проверить, так что имеется функтор $\check{G} : A \times B \to C$.

Наконец, также ясно, что эти операции обратные. □

Теперь, для любой предкатегории A, имеется hom-функтор

$$\hom_A : A^{\mathsf{op}} \times A \to \mathcal{S}et\,.$$

Он применяет пару $(a, b) : (A^{op})_0 \times A_0 :\equiv A_0 \times A_0$ к множеству $\hom_A(a, b)$. Для морфизма $(f, f') : \hom_{A^{op} \times A}((a, b), (a', b'))$, по определению, имеем $f : \hom_A(a', a)$ и $f' : \hom_A(b, b')$, поэтому можно определить

$$\begin{aligned}(\hom_A)_{(a,b),(a',b')}(f, f') &:\equiv (g \mapsto (f' g f)) \\ &: \hom_A(a, b) \to \hom_A(a', b')\,.\end{aligned}$$

Функториальность проверить несложно.

Следовательно, по лемме 9.5.3, имеется индуцированный функтор $\mathbf{y} : A \to \mathcal{S}et^{A^{\mathsf{op}}}$, который называется **вложением Йонеды**.

Теорема 9.5.4 (Лемма Йонеды)**.** *Для любой предкатегории A, любого $a : A$ и любого функтора $F : \mathcal{S}et^{A^{\mathsf{op}}}$, имеется изоморфизм*

$$\hom_{\mathcal{S}et^{A^{\mathsf{op}}}}(\mathbf{y}a, F) \cong Fa\,. \tag{9.5.5}$$

Причем, он является естественным, и для a, и для F.

Доказательство. В естественном преобразовании $\alpha : \mathbf{y}a \to F$ рассмотрим компонент $\alpha_a : \mathbf{y}a(a) \to Fa$. Поскольку $\mathbf{y}a(a) :\equiv \hom_A(a, a)$, имеем $1_a : \mathbf{y}a(a)$, так что $\alpha_a(1_a) : Fa$. Это дает функцию $(\alpha \mapsto \alpha_a(1_a))$, действующую в (9.5.5) слева направо.

В обратном направлении, учитывая $x : Fa$, определим $\alpha : \mathbf{y}a \to F$ следующим образом:

$$\alpha_{a'}(f) :\equiv F_{a,a'}(f)(x)\,.$$

Естественность легко проверить, так что это дает функцию, действующую в (9.5.5) справа налево.

Чтобы показать, что они обратны друг другу, сначала предположим, что $x : Fa$. Тогда с α, определенным, как указано выше, имеем $\alpha_a(1_a) = F_{a,a}(1_a)(x) = 1_{Fa}(x) = x$. С другой стороны, если предположить, что задано $\alpha : \mathbf{y}a \to F$, и определить x, как указано выше, то, для любого $f : \hom_A(a', a)$, будем иметь

$$\begin{aligned}\alpha_{a'}(f) &= \alpha_{a'}(\mathbf{y}a_{a,a'}(f)(1_a)) \\ &= (\alpha_{a'} \circ \mathbf{y}a_{a,a'}(f))(1_a) \\ &= (F_{a,a'}(f) \circ \alpha_a)(1_a) \\ &= F_{a,a'}(f)(\alpha_a(1_a)) \\ &= F_{a,a'}(f)(x)\,.\end{aligned}$$

Таким образом, обе компоновки являются тождественностями. Доказательство естественности оставляется читателю. □

Следствие 9.5.6. *Вложение Йонеды* $\mathbf{y} : A \to \mathcal{S}et^{A^{\mathrm{op}}}$ *является вполне точным.*

Доказательство. По теореме 9.5.4, имеем

$$\hom_{\mathcal{S}et^{A^{\mathrm{op}}}}(\mathbf{y}a, \mathbf{y}b) \cong \mathbf{y}b(a) \equiv \hom_A(a, b)\,.$$

Легко проверить, что этот изоморфизм на самом деле является действием $\mathbf{y}$ на $\hom$-множествах. □

Следствие 9.5.7. *Если A — категория, то $\mathbf{y}_0 : A_0 \to (\mathcal{S}et^{A^{\mathrm{op}}})_0$ — это вложение. В частности, если $\mathbf{y}a = \mathbf{y}b$, то $a = b$.*

Доказательство. По следствию 9.5.6 9.5.6, $\mathbf{y}$ индуцирует изоморфизм на множествах изоморфизмов. Но, поскольку A и $\mathcal{S}et^{A^{\mathrm{op}}}$ — категории, а $\mathbf{y}$ — функтор, это эквивалентно изоморфизму тождественных типов, который является определением вложения. □

Определение 9.5.8. Функтор $F : \mathcal{S}et^{A^{\mathrm{op}}}$ называется **представимым**, если существует $a : A$ и изоморфизм $\mathbf{y}a \cong F$.

Теорема 9.5.9. *Если A — категория, то тип «F представим» — простое высказывание.*

Доказательство. По определению «F представимо» — это просто слой $\mathbf{y}_0$ над F. Поскольку $\mathbf{y}_0$ является вложением по следствию 9.5.7, этот слой — простое высказывание. □

В частности, в категории, любые два представления одного и того же функтора эквивалентны. Мы можем использовать это, чтобы дать другое доказательство леммы 9.3.2. Сначала приведем характеризацию сопряжений с точки зрения представимости.

Лемма 9.5.10. *Для любых предкатегорий A, B и функтора $F : A \to B$, следующие типы эквивалентны.*

(i) *F — левый сопряженный.*

(ii) *Для каждого $b : B$, функтор $(a \mapsto \hom_B(Fa, b))$ от A^{op} к $\mathcal{S}et$ является представимым.*

Доказательство. Элемент типа (ii) состоит из функции $G_0 : B_0 \to A_0$ вместе с, для каждого $a : A$ и $b : B$, изоморфизмом

$$\gamma_{a,b} : \hom_B(Fa, b) \cong \hom_A(a, G_0 b)$$

таким, что $\gamma_{a,b}(g \circ Ff) = \gamma_{a',b}(g) \circ f$ для $f : \hom_A(a, a')$.

С учетом этого, для $a : A$, определим $\eta_a :\equiv \gamma_{a,Fa}(1_{Fa})$, а для $b : B$, определим $\epsilon_b :\equiv (\gamma_{Gb,b})^{-1}(1_{Gb})$. Теперь, для $g : \hom_B(b, b')$, определим

$$G_{b,b'}(g) :\equiv \gamma_{Gb,b'}(g \circ \epsilon_b)\,.$$

Проверки того, что G является функтором, а η и ϵ — естественными преобразованиями, удовлетворяющими тождествам треугольника, точно такие же, как и в классическом случае, и, поскольку все они являются простыми высказываниями, нет необходимости обращать внимание на их значения. Так что, имеется функция (ii)$\to$(i).

В другом случае, если F является левым сопряженным, имеем, конечно же, точно определенный G_0, и можно взять $\gamma_{a,b}$ в качестве компоновки

$$\hom_B(Fa, b) \xrightarrow{G_{Fa,b}} \hom_A(GFa, Gb) \xrightarrow{(-\circ\eta_a)} \hom_A(a, Gb)\,.$$

Это, очевидно, естественно, так как таковым является η, и имеет обратное, задаваемое как

$$\hom_A(a, Gb) \xrightarrow{F_{a,Gb}} \hom_B(Fa, FGb) \xrightarrow{(\epsilon_b\circ-)} \hom_A(Fa, b)$$

(по треугольным тождествам). Таким образом, имеем (i)$\to$(ii).

Очевидно, что для компоновки (ii)$\to$(i)$\to$(ii), функция G_0 сохраняется, поэтому достаточно проверить, что мы получили обратно γ. Но новое γ определяется так, чтобы перевести $f : \hom_B(Fa, b)$ в

$$\begin{aligned} G(f) \circ \eta_a &\equiv \gamma_{GFa,b}(f \circ \epsilon_{Fa}) \circ \eta_a \\ &= \gamma_{GFa,b}(f \circ \epsilon_{Fa} \circ F\eta_a) \\ &= \gamma_{GFa,b}(f) \end{aligned}$$

что согласуется со старым γ.

Наконец, для (i)$\to$(ii)$\to$(i), мы возвращаемся к функтору G на объектах. Новый $G_{b,b'} : \hom_B(b, b') \to \hom_A(Gb, Gb')$ определяется так, чтобы перевести g в

$$\begin{aligned} \gamma_{Gb,b'}(g \circ \epsilon_b) &\equiv G(g \circ \epsilon_b) \circ \eta_{Gb} \\ &= G(g) \circ G\epsilon_b \circ \eta_{Gb} \\ &= G(g) \end{aligned}$$

так что он совпадает со старым. Новое η_a определяется как $\gamma_{a,Fa}(1_{Fa}) \equiv G(1_{Fa}) \circ \eta_a$, поэтому оно равно старому η_a. И, наконец, новое ϵ_b определяется как $(\gamma_{Gb,b})^{-1}(1_{Gb}) \equiv \epsilon_b \circ F(1_{Gb})$, что также совпадает со старым ϵ_b. □

Следствие 9.5.11. *Если A — категория и $F : A \to B$, то тип «F — левый сопряженный» — это простое высказывание.*

Доказательство. По теореме 9.5.9, если A — категория, то тип из леммы 9.5.10(ii) является простым высказыванием. □

9.6 Строгие категории

Определение 9.6.1. Строгая категория — это предкатегория, типом объектов которой является множество.

В соответствии с математическим принципом отвлекающего маневра, строгая категория не обязательно является категорией. Фактически, категория является строгой категорией именно тогда, когда она тощая (пример 9.1.15). В большинстве случаев теория категорий касается категорий, но не строгих, а иногда хочется рассмотреть строгие категории. Основная мотивация этого состоит в том, что строгие категории имеют более строгое понятие «одинаковости», чем эквивалентность, а именно изоморфизм (или, что эквивалентно, по лемме 9.4.15, равенство).

Рассмотрим один источник строгих категорий.

Пример 9.6.2. Пусть A — предкатегория, а $x : A$ — некоторый ее объект. Тогда существует предкатегория $\mathsf{mono}(A, x)$ такая, что:

- ее объекты состоят из $y : A$ и мономорфизма $m : \hom_A(y, x)$ (как принято, $m : \hom_A(y, x)$ является **мономорфизмом** (или **моно**), если $(m \circ f = m \circ g) \Rightarrow (f = g)$);
- ее морфизмы от (y, m) к (z, n) являются произвольными морфизмами от y к z в A (не обязательно с учетом m и n).

Равенство $(y, m) = (z, n)$ объектов в $\mathsf{mono}(A, x)$ состоит из равенства $p : y = z$ и равенства $p_*(m) = n$, что по лемме 9.1.9 равносильно равенству $m = n \circ \mathsf{idtoiso}(p)$. Поскольку $\hom$-наборы являются множествами, тип таких равенств является простым высказыванием. Но поскольку m и n являются мономорфизмами, тип морфизмов f таких, что $m = n \circ f$, также является простым высказыванием. Таким образом, если A — категория, то $(y, m) = (z, n)$ — простое высказыванием, и, следовательно, $\mathsf{mono}(A, x)$ — строгая категория.

Этот пример можно дуализировать и обобщить по-разному. Вот интересное применение строгих категорий.

Пример 9.6.3. Пусть E/F — конечное расширение полей Галуа, а G — его группа Галуа. Тогда существует строгая категория, объектами которой являются промежуточные поля $F \subseteq K \subseteq E$, а морфизмами — гомоморфизмы полей, которые фиксируют F поточечно (но не обязательно коммутируют с включениями в E). Существует еще одна строгая категория, объектами которой являются подгруппы $H \subseteq G$, а морфизмами — морфизмы G-sets $G/H \to G/K$. Основная теорема теории Галуа гласит, что эти две предкатегории изоморфны (а не просто эквивалентны).

9.7 †-категории

Также стоит упомянуть полезный вид предкатегории, тип объектов которой не является множеством, и которая, к тому же, не является категорией.

Определение 9.7.1. †-предкатегория — это предкатегория A, обладающая следующими свойствами.

(i) Для любых $x, y : A$, имеется функция $(-)^\dagger : \hom_A(x, y) \to \hom_A(y, x)$.

(ii) Для каждого $x : A$, имеем $(1_x)^\dagger = 1_x$.

(iii) Для всех f, g, имеет место $(g \circ f)^\dagger = f^\dagger \circ g^\dagger$.

(iv) Для любой f, верно $(f^\dagger)^\dagger = f^\dagger$.

Определение 9.7.2. Морфизм $f : \hom_A(x, y)$ в †-предкатегории является **унитарным**, если $f^\dagger \circ f = 1_x$ и $f \circ f^\dagger = 1_y$.

Конечно, каждый унитарный морфизм является изоморфизмом, а унитарность — это простое высказывание. Таким образом, для любых $x, y : A$, имеется множество унитарных изоморфизмов от x к y, которое мы обозначаем как $(x \cong^\dagger y)$.

Лемма 9.7.3. *Если $p : (x = y)$, то* $\mathsf{idtoiso}(p)$ *является унитарным.*

Доказательство. По индукции, можно считать, что p есть refl_x. Но тогда $(1_x)^\dagger \circ 1_x = 1_x \circ 1_x = 1_x$ и, аналогично, для $1_x \circ (1_x)^\dagger$. □

Определение 9.7.4. **†-категория** — это †-предкатегория такая, что для всех $x, y : A$, функция

$$(x = y) \to (x \cong^\dagger y)$$

из леммы 9.7.3 является эквивалентностью.

Пример 9.7.5. Категория $\mathcal{R}el$ из примера 9.1.19 превращается ва †-предкатегорию, если определить $(R^\dagger)(y, x) :\equiv R(x, y)$. Доказательство того, что $\mathcal{R}el$ является категорией, на самом деле показывает, что каждый изоморфизм унитарен; следовательно, $\mathcal{R}el$ также является †-категорией.

Пример 9.7.6. Любой группоид становится †-категорией, если определить $f^\dagger :\equiv f^{-1}$.

Пример 9.7.7. Пусть $\mathcal{H}ilb$ — предкатегория:

- с объектами — конечномерными векторными пространствами, снабженными внутренним произведением $\langle -, - \rangle$;

- с морфизмами — произвольными линейными отображениями.

Согласно стандартной линейной алгебре, любое линейное отображение $f : V \to W$ между конечномерными внутренними пространствами произведений имеет однозначно определенное сопряжение $f^\dagger : W \to V$, характеризуемое посредством $\langle fv, w \rangle = \langle v, f^\dagger w \rangle$. Таким образом, $\mathcal{H}ilb$ становится †-предкатегорией. Более того, линейный изоморфизм унитарен именно тогда, когда он является **изометрией**, т.е. $\langle fv, fw \rangle = \langle v, w \rangle$. Отсюда следует, что $\mathcal{H}ilb$ является †-категорией, но не категорией (не всякий линейный изоморфизм унитарен).

Для †-категорий с классической основой было разработано много общей теории. Ранее было замечено, что унитарные, а не произвольные, изоморфизмы, являются правильным понятием «одинаковости» для объектов †-категории, что вызвало некоторую панику среди категорных теоретиков. Теория гомотопических типов решает эту проблему, идентифицируя †-категории, такие как строгие категории, просто как предкатегории другого типа.

9.8 Принцип структурной тождественности

Принцип структурной тождественности — это неформальный принцип, который выражает идентичность изоморфных структур. Мы стремимся доказать общий абстрактный результат, который может быть применен к широкому семейству понятий структуры, когда структуры могут быть много-сортными или даже зависимо-сортными, бесконечными или даже высших порядков.

Самый простой вид одно-сортной структуры может быть представлен типом без дополнительной структуры. Аксиома унивалентности выражает принцип структурной тождественности этого понятия структуры в строгой форме: для типов A, B, каноническая функция $(A = B) \to (A \simeq B)$ является эквивалентностью.

Начнем с предкатегории X. В приложении к одно-сортным структурам первого порядка, X будет категорией $\mathcal{U}$-малых множеств, где $\mathcal{U}$ — универсум унивалентного типа.

Определение 9.8.1. Нотация структуры (P, H) над X включает следующее.

(i) Семейство типов $P : X_0 \to \mathcal{U}$. Для каждого $x : X_0$ элементы из Px называются (P, H)**-структурами** на x.

(ii) Для $x, y : X_0$, $f : \hom_X(x, y)$ и $\alpha : Px$, $\beta : Py$, простое высказывание

$$H_{\alpha\beta}(f)\,.$$

Если $H_{\alpha\beta}(f)$ истинно, будем говорить, что f является (P, H)**-гомоморфизмом** от α к β.

(iii) Для $x : X_0$ и $\alpha : Px$, имеет место $H_{\alpha\alpha}(1_x)$.

(iv) Для $x, y, z : X_0$ и $\alpha : Px$, $\beta : Py$, $\gamma : Pz$, если $f : \hom_X(x, y)$ и $g : \hom_X(y, z)$, то верно

$$H_{\alpha\beta}(f) \to H_{\beta\gamma}(g) \to H_{\alpha\gamma}(g \circ f)\,.$$

Когда (P, H) — нотация структуры, для $\alpha, \beta : Px$ определим

$$(\alpha \leqslant_x \beta) :\equiv H_{\alpha\beta}(1_x)\,.$$

Согласно (iii) и (iv), это есть предпорядок (пример 9.1.14) с его типом объектов Px. Скажем, что (P, H) — это **стандартная нотация структуры**, если этот предпорядок фактически является частичным порядком, для всех $x : X$.

Заметим, что для стандартной нотации о структуре, каждый тип Px должен быть множеством. Теперь определим, для любой нотации структуры (P, H), **предкатегорию** (P, H)**-структур**, $A = \mathsf{Str}_{(P,H)}(X)$.

- Тип объектов A — это тип $A_0 :\equiv \sum_{(x:X_0)} Px$. Если $a \equiv (x, \alpha) : A_0$, можно записать $|a| :\equiv x$.
- Для $(x, \alpha) : A_0$ и $(y, \beta) : A_0$, определим

$$\hom_A((x, \alpha), (y, \beta)) :\equiv \{f : x \to y \mid H_{\alpha\beta}(f)\}\,.$$

Композиция и тождественности наследуются от X; условия (iii) и (iv) гарантируют, что они поднимутся до A.

Теорема 9.8.2 (Принцип структурной тождественности). *Если X — категория, а (P, H) — стандартная нотация структуры над X, то предкатегория $\mathsf{Str}_{(P,H)}(X)$ является категорией.*

Доказательство. По определению равенства в зависимых парных типах, выполнение равенства $(x, \alpha) = (y, \beta)$ включает выполнение

- равенства $p : x = y$ и
- равенства $p_*(\alpha) = \beta$.

Поскольку P является множественно-значным, последнее является простым высказыванием. С другой стороны, ясно, что изоморфизм $(x, \alpha) \cong (y, \beta)$ в $\mathsf{Str}_{(P,H)}(X)$ состоит из

- изоморфизма $f : x \cong y$ в X, такого что
- $H_{\alpha\beta}(f)$ и $H_{\beta\alpha}(f^{-1})$.

Конечно, второе из них также является простым высказыванием. А поскольку X — категория, функция $(x = y) \to (x \cong y)$ является эквивалентностью. Таким образом, достаточно показать, что для любого $p : x = y$ и для любых $(\alpha : Px)$, $(\beta : Py)$, имеем $p_*(\alpha) = \beta$ тогда и только тогда, когда верно, и $H_{\alpha\beta}(\mathsf{idtoiso}(p))$, и $H_{\beta\alpha}(\mathsf{idtoiso}(p)^{-1})$.

Направление «только тогда» — это просто наличие функции $\mathsf{idtoiso}$ для категории $\mathsf{Str}_{(P,H)}(X)$. Для направления «если» индукцией по p можно считать, что $y \equiv x$ и $p \equiv \mathsf{refl}_x$. Однако, в этом случае, $\mathsf{idtoiso}(p) \equiv 1_x$ и, следовательно, $\mathsf{idtoiso}(p)^{-1} = 1_x$. Таким образом, $\alpha \leqslant_x \beta$ и $\beta \leqslant_x \alpha$, что подразумевает $\alpha = \beta$, поскольку (P, H) — стандартная нотация структуры. □

В качестве примера, эта методология дает альтернативный способ выразить доказательство теоремы 9.2.5.

Пример 9.8.3. Пусть A — предкатегория, а B — категория. Существует предкатегория B^{A_0}, объектами которой являются функции $A_0 \to B_0$, и чье множество морфизмов от $F_0 : A_0 \to B_0$ к $G_0 : A_0 \to B_0$ есть $\prod_{(a:A_0)} \hom_B(F_0 a, G_0 a)$. Композиция и тождественности наследуются непосредственно от B. Легко показать, что $\gamma : \hom_{B^{A_0}}(F_0, G_0)$ является изоморфизмом именно тогда, когда каждый компонент γ_a является изоморфизмом, так что имеем $(F_0 \cong G_0) \simeq \prod_{(a:A_0)}(F_0 a \cong G_0 a)$. Более того, отображение $\mathsf{idtoiso} : (F_0 = G_0) \to (F_0 \cong G_0)$ of B^{A_0} есть композиция

$$(F_0 = G_0) \longrightarrow \prod_{(a:A_0)} (F_0 a = G_0 a) \longrightarrow \prod_{(a:A_0)} (F_0 a \cong G_0 a) \longrightarrow (F_0 \cong G_0)\,,$$

в которой первое отображение является эквивалентностью по функциональной экстенсиональности, второе — потому что оно является зависимым произведением эквивалентностей (поскольку B — категория), а третье — как отмечалось выше. Таким образом, B^{A_0} — это категория.

Теперь определим нотацию структуры на B^{A_0}, для которой $P(F_0)$ является типом операций $F : \prod_{(a,a':A_0)} \hom_A(a, a') \to \hom_B(F_0 a, F_0 a')$, которые расширяют F_0 до функтора (т.е. сохраняются композиция и тождественности). Это множество, поскольку каждое $\hom_B(-,-)$ является таковым. Для таких F и G, определим $\gamma : \hom_{B^{A_0}}(F_0, G_0)$ как гомоморфизм, если он образует естественное преобразование. В определении 9.2.3, по существу, было проверено, что это нотация структуры. Более того, если F и F' — структуры на F_0 и тождество является естественным преобразованием от F к F', то, для любого $f : \hom_A(a, a')$, имеем $F'f = F'f \circ 1_{F_0 a} = 1_{F_0 a} \circ Ff = Ff$.

Применяя функциональную экстенсиональность, заключаем, что $F = F'$. Таким образом, имеется *стандартная* нотация структуры, а значит, по теореме 9.8.2, предкатегория B^A является категорией.

В качестве другого примера рассмотрим категории структур для сигнатуры первого порядка. Определим **сигнатуру первого порядка**, Ω, состоящую из множеств Ω_0 и Ω_1 функциональных символов, $\omega : \Omega_0$, и символов отношений, $\omega : \Omega_1$, каждое из которых является множеством и имеет арность $|\omega|$. **Ω-структура** a состоит из множества $|a|$ вместе с назначением $|\omega|$-арной функции $\omega^a : |a|^{|\omega|} \to |a|$ на $|a|$ каждому функциональному символу, ω, и назначению $|\omega|$-арного отношения ω^a на $|a|$, присваивая простое высказывание $\omega^a x$ каждому $x : |a|^{|\omega|}$, каждому символу отношения. И учитывая Ω-структуры a, b, функция $f : |a| \to |b|$ является **гомоморфизмом** $a \to b$**,** если она сохраняет структуру; т.е. если для каждого символа ω сигнатуры и любого $x : |a|^{|\omega|}$,

(i) $f(\omega^a x) = \omega^b(f \circ x)$ if $\omega : \Omega_0$, и

(ii) $\omega^a x \to \omega^b(f \circ x)$ if $\omega : \Omega_1$.

Заметим, что каждый $x : |a|^{|\omega|}$ является функцией $x : |\omega| \to |a|$, так что $f \circ x : b^\omega$.

Теперь предположим, что даны, (унивалентный) универсум $\mathcal{U}$ и $\mathcal{U}$-малая сигнатура Ω; т.е. $|\Omega|$ является $\mathcal{U}$-малым множеством, и для каждого $\omega : |\Omega|$, множество $|\omega|$ является $\mathcal{U}$-малым. Тогда имеется категория $\mathcal{S}et_{\mathcal{U}}$ $\mathcal{U}$-малых множеств. Мы хотим определить предкатегорию $\mathcal{U}$-малых Ω-структур над $\mathcal{S}et_{\mathcal{U}}$ и использовать теорему 9.8.2, чтобы показать, что она является категорией.

Используется сигнатура первого порядка Ω, чтобы дать стандартную нотацию структуры (P, H) над $\mathcal{S}et_{\mathcal{U}}$.

Определение 9.8.4.

(i) Для каждого $\mathcal{U}$-малого множества x определим

$$Px :\equiv P_0 x \times P_1 x\,,$$

где

$$P_0 x :\equiv \prod_{(\omega:\Omega_0)} x^{|\omega|} \to x, \text{ и}$$

$$P_1 x :\equiv \prod_{(\omega:\Omega_1)} x^{|\omega|} \to \mathsf{Prop}_{\mathcal{U}}\,.$$

(ii) Для $\mathcal{U}$-малых множеств x, y и $\alpha : P^\omega x,\ \beta : P^\omega y,\ f : x \to y$, определим

$$H_{\alpha\beta}(f) :\equiv H_{0,\alpha\beta}(f) \wedge H_{1,\alpha\beta}(f)\,,$$

где

$$H_{0,\alpha\beta}(f) :\equiv \forall(\omega : \Omega_0).\, \forall(u : x^{|\omega|}).\, f(\alpha u) = \beta(f \circ u)\,, \text{ и}$$

$$H_{1,\alpha\beta}(f) :\equiv \forall(\omega : \Omega_0).\, \forall(u : x^{|\omega|}).\, \alpha u \to \beta(f \circ u)\,.$$

Довольно трудоемко проверить, что (P, H) является стандартной нотацией структуры над $\mathcal{S}et_{\mathcal{U}}$ и, следовательно, можно использовать теорему 9.8.2, по которой предкатегория $Str_{(P,H)}(\mathcal{S}et_{\mathcal{U}})$ является категорией. Остается только заметить, что это, по существу, то же самое, что предкатегория $\mathcal{U}$-малых Ω-структур над $\mathcal{S}et_{\mathcal{U}}$.

9.9 Пополнение Резка

В этом разделе мы представим универсальный способ замены предкатегории категорией. Фактически, будут описаны два способа. Оба основываются на том факте, что «категории рассматривают слабые эквивалентности как эквивалентности».

Чтобы доказать это, начнем с пары лемм, которые являются полностью стандартной теорией категорий. Они тщательно сформулированы, чтобы убедиться, что элиминатор для $\|-\|_{-1}$ используется корректно. В классической теории категорий следовало бы быть столь же осторожным, если бы кто-то захотел избежать аксиомы выбора: каждый раз, когда требуется определить функцию, нужно как-то однозначно охарактеризовать ее значения.

Лемма 9.9.1. *Если A, B, C — предкатегории, а $H : A \to B$ — эссенциально сюръективный функтор, то $(- \circ H) : C^B \to C^A$ является точным.*

Доказательство. Пусть $F, G : B \to C$ и $\gamma, \delta : F \to G$ таковы, что $\gamma H = \delta H$; надо показать, что $\gamma = \delta$. Итак, пусть $b : B$; покажем, что $\gamma_b = \delta_b$. Это простое высказывание, поэтому, поскольку H эссенциально сюръективен, можно считать, что даны $a : A$ и изоморфизм $f : Ha \cong b$. Но теперь имеем

$$\gamma_b = G(f) \circ \gamma_{Ha} \circ F(f^{-1}) = G(f) \circ \delta_{Ha} \circ F(f^{-1}) = \delta_b\,.$$

□

Лемма 9.9.2. *Если A, B, C — предкатегории, а $H : A \to B$ — эссенциально сюръективный и полный функтор, то $(- \circ H) : C^B \to C^A$ является вполне точным.*

Доказательство. Остается показать полноту. Пусть $F, G : B \to C$ и $\gamma : FH \to GH$. Мы утверждаем, что для любого $b : B$, тип

$$\sum_{(g:\hom_C(Fb,Gb))} \prod_{(a:A)} \prod_{(f:Ha\cong b)} (\gamma_a = Gf^{-1} \circ g \circ Ff) \tag{9.9.3}$$

стягиваем. Поскольку стягиваемость — это простое высказывание, а H эссенциально сюръективен, то можно считать, что заданы $a_0 : A$ и $h : Ha_0 \cong b$.

Теперь рассмотрим $g :\equiv Gh \circ \gamma_{a_0} \circ Fh^{-1}$. Тогда для любых других $a : A$ и $f : Ha \cong b$, надо показать, что $\gamma_a = Gf^{-1} \circ g \circ Ff$. Поскольку H является полным, просто существует морфизм $k : \hom_A(a, a_0)$ такой, чтот $Hk = h^{-1} \circ f$. И поскольку нашей целью является простое высказывание, можно предположить, что задано некоторое такое k. Тогда имеем

$$\begin{aligned}
\gamma_a &= GHk^{-1} \circ \gamma_{a_0} \circ FHk \\
&= Gf^{-1} \circ Gh \circ \gamma_{a_0} \circ Fh^{-1} \circ Ff \\
&= Gf^{-1} \circ g \circ Ff\,.
\end{aligned}$$

Таким образом, (9.9.3) обитаем. Осталось показать, что это простое высказывание. Пусть $g, g' : \hom_C(Fb, Gb)$ таковы, что для всех $a : A$ и $f : Ha \cong b$, имеет место, и $(\gamma_a = Gf^{-1} \circ g \circ Ff)$, и $(\gamma_a = Gf^{-1} \circ g' \circ Ff)$. Зависимые типы произведений — это простые высказывания, поэтому все, что нам остается доказать, — это $g = g'$. Но это простое высказывание, поэтому можно предположить, что $a_0 : A$ и $h : Ha_0 \cong b$, и в этом случае

$$g = Gh \circ \gamma_{a_0} \circ Fh^{-1} = g'\,.$$

Это доказывает, что (9.9.3) стягиваемо для всех $b : B$. Теперь определим $\delta : F \to G$, рассматривая δ_b как единственный g в (9.9.3) для этого b. Чтобы убедиться, что это естественно, предположим, что задано $f : \hom_B(b, b')$; надо показать, что $Gf \circ \delta_b = \delta_{b'} \circ Ff$. Как и ранее, можно принять $a : A$ и $h : Ha \cong b$, а также $a' : A$ и $h' : Ha' \cong b'$. Поскольку H является полным, а также эссенциально сюръективным, можно также считать, что $k : \hom_A(a, a')$ с $Hk = h'^{-1} \circ f \circ h$.

Поскольку γ естественно, то $GHk \circ \gamma_a = \gamma_{a'} \circ FHk$. Используя определение δ, имеем

$$\begin{aligned}
Gf \circ \delta_b &= Gf \circ Gh \circ \gamma_a \circ Fh^{-1} \\
&= Gh' \circ GHk \circ \gamma_a \circ Fh^{-1} \\
&= Gh' \circ \gamma_{a'} \circ FHk \circ Fh^{-1} \\
&= Gh' \circ \gamma_{a'} \circ Fh'^{-1} \circ Ff \\
&= \delta_{b'} \circ Ff\,.
\end{aligned}$$

Таким образом, δ естественно. Наконец, для любого $a : A$, применяя определение δ_{Ha} к a и 1_a, получаем $\gamma_a = \delta_{Ha}$. Следовательно, $\delta \circ H = \gamma$. □

Остальная часть этой леммы выстаивается почти точно так же, с категоризацией C, вставленной на одном решающем шаге, который выделен здесь курсивом для акцентирования: это момент, когда мы пытаемся определить функцию в *объекты* без использования выбора, и поэтому надо соблюдать осторожность с тем, что означает «однозначно заданный» объект. В классической теории категорий все, что можно сказать, это то, что данный объект задан с точностью до единственного изоморфизма, но в теоретико-множественных основах этого недостаточно, чтобы предоставить функцию без обращения к AC. Однако, в унивалентных основаниях, если C — категория, то изоморфизм — это равенство, и имеется соответствующий вид уникальности (а именно, обитание в стягиваемом пространстве).

Теорема 9.9.4. *Если A, B — предкатегории, C — категория, а $H : A \to B$ — слабая эквивалентность, то $(- \circ H) : C^B \to C^A$ — изоморфизм.*

Доказательство. По теореме 9.2.5, C^B и C^A являются категориями. Таким образом, по лемме 9.4.14 достаточно показать, что $(- \circ H)$ — эквивалентность. Но поскольку из предыдущих двух лемм известно, что он вполне точен, по лемме 9.4.7 достаточно показать, что он эссенциально сюръективен. Итак, предположим, что $F : A \to C$; требуется, чтобы просто существовал $G : B \to C$ такой, что $GH \cong F$.

Для каждого $b : B$, пусть X_b будет типом, элементы которого содержат:

(i) $c : C$;

(ii) для каждого $a : A$ и $h : Ha \cong b$, изоморфизм $k_{a,h} : Fa \cong c$;

(iii) для любых (a, h) и (a', h'), как в (ii), и любой $f : \hom_A(a, a')$ такой, что $h' \circ Hf = h$, имеем $k_{a',h'} \circ Ff = k_{a,h}$.

Можно утверждать, что для любого $b : B$, тип X_b является стягиваемым. Действительно, поскольку это простое высказывание, можно считать, что даны $a_0 : A$ и $h_0 : Ha_0 \cong b$. Пусть $c^0 :\equiv Fa_0$. Затем, для $a : A$ и $h : Ha \cong b$, поскольку H вполне точен, существует единственный изоморфизм $g_{a,h} : a \to a_0$ с $Hg_{a,h} = h_0{}^{-1} \circ h$. Определим $k^0_{a,h} :\equiv Fg_{a,h}$. Наконец, если $h' \circ Hf = h$, то ${h_0}^{-1} \circ h' \circ Hf = {h_0}^{-1} \circ h$, следовательно, $g_{a',h'} \circ f = g_{a,h}$, и тогда $k^0_{a',h'} \circ Ff = k^0_{a,h}$. Таким образом, X_b обитаем.

Теперь предположим, что даны другие $(c^1, k^1) : X_b$. Тогда $k^1_{a_0,h_0} : c^0 \equiv Fa_0 \cong c^1$. *Поскольку C — категория, имеем $p : c^0 = c^1$ с* $\mathsf{idtoiso}(p) = k^1_{a_0,h_0}$. И, для любого $a : A$ и $h : Ha \cong b$, используя (iii) для (c^1, k^1) с $f :\equiv g_{a,h}$, имеет место

$$k^1_{a,h} = k^1_{a_0,h_0} \circ k^0_{a,h} = p_*\left(k^0_{a,h}\right) .$$

Это дает необходимые основания для равенства $(c^0, k^0) = (c^1, k^1)$, завершая доказательство того, что X_b является стягиваемым.

Теперь, поскольку X_b стягиваем для каждого b, тип $\prod_{(b:B)} X_b$ также является стягиваемым. В частности, он обитаем, поэтому имеется функция, назначающая каждому $b : B$, c и k. Определим $G_0(b)$ этим c, что дает функцию $G_0 : B_0 \to C_0$.

Далее, нужно определить действие G на морфизмы. Для любых $b, b' : B$ и $f : \hom_B(b, b')$, пусть Y_f будет типом, элементы которого состоят из:

(iv) морфизма $g : \hom_C(Gb, Gb')$ такого, что

(v) для любого $a : A$ и $h : Ha \cong b$, и любого $a' : A$ и $h' : Ha' \cong b'$, и любого $\ell : \hom_A(a, a')$, имеем
$$(h' \circ H\ell = f \circ h) \to (k_{a',h'} \circ F\ell = g \circ k_{a,h}) .$$

Можно утверждать, что для любых b, b' и f, тип Y_f является стягиваемым. Действительно, поскольку это простое высказывание, можно предположить это для $a_0 : A$ и $h_0 : Ha_0 \cong b$, и для любого $a'_0 : A$ и $h'_0 : Ha'_0 \cong b'$. Тогда, поскольку H вполне точен, существует единственный $\ell_0 : \hom_A(a_0, a'_0)$ такой, что $h'_0 \circ H\ell_0 = f \circ h_0$. Определим $g_0 :\equiv k_{a'_0,h'_0} \circ F\ell_0 \circ (k_{a_0,h_0})^{-1}$.

Теперь, для любых a, h, a', h' и ℓ таких, чтоt $(h' \circ H\ell = f \circ h)$, имеем $h^{-1} \circ h_0 : Ha_0 \cong Ha$, следовательно, существует единственный $m : a_0 \cong a$ с $Hm = h^{-1} \circ h_0$и, следовательно, $h \circ Hm = h_0$. Точно так же, имеется единственный $m' : a'_0 \cong a'$ с $h' \circ Hm' = h'_0$. Теперь, согласно (iii), имеем $k_{a,h} \circ Fm = k_{a_0,h_0}$ и $k_{a',h'} \circ Fm' = k_{a'_0,h'_0}$. Также,

$$\begin{aligned} Hm' \circ H\ell_0 &= (h')^{-1} \circ h'_0 \circ H\ell_0 \\ &= (h')^{-1} \circ f \circ h_0 \\ &= (h')^{-1} \circ f \circ h \circ h^{-1} \circ h_0 \\ &= H\ell \circ Hm \end{aligned}$$

и, следовательно, $m' \circ \ell_0 = \ell \circ m$ поскольку H вполне точен. Наконец, можно вычислить

$$\begin{aligned} g_0 \circ k_{a,h} &= k_{a'_0,h'_0} \circ F\ell_0 \circ (k_{a_0,h_0})^{-1} \circ k_{a,h} \\ &= k_{a'_0,h'_0} \circ F\ell_0 \circ Fm^{-1} \\ &= k_{a'_0,h'_0} \circ (Fm')^{-1} \circ F\ell \\ &= k_{a',h'} \circ F\ell\,. \end{aligned}$$

Это завершает доказательство того, что Y_f обитаем. Чтобы показать его стягиваемость, поскольку hom-наборы являются множествами, достаточно взять еще один $g_1 : \hom_C(Gb, Gb')$ удовлетворяющий (v), и показать, что $g_0 = g_1$. Однако, имеются определенные $a_0, h_0, a'_0, h'_0, \ell_0$,а (v) подразумевает, что, как g_0, так и g_1, должны быть равны $k_{a'_0,h'_0} \circ F\ell_0 \circ (k_{a_0,h_0})^{-1}$.

Это завершает доказательство того, что Y_f стягиваемо, для любых $b, b' : B$ и $f : \hom_B(b, b')$. Следовательно, имеется функция, назначающая каждому такому f его единственного обитателя; обозначим эту функцию как $G_{b,b'} : \hom_B(b, b') \to \hom_C(Gb, Gb')$. Доказательство того, что G — функтор, несложно; в любом случае можно выбрать a, h и применить (v).

Наконец, для любого $a_0 : A$, определяя $c :\equiv Fa_0$ и $k_{a,h} :\equiv Fg$, где $g : \hom_A(a, a_0)$ является единственным изоморфизмом с $Hg = h$, имеется элемент X_{Ha_0}. Таким образом, он равен указанному; следовательно, $GHa = Fa$. Аналогично, для $f : \hom_A(a_0, a_0')$, можно определить элемент Y_{Hf} транспортированием по этим равенствам, который, следовательно, должен быть равен указанному. Следовательно, $GH = F$ и, значит, $GH \cong F$, что и требовалось. □

Следовательно, если предкатегория A допускает функтор слабой эквивалентности $A \to \widehat{A}$ в категорию, то это его «отражение» в категории: любой функтор от A в категорию будет эссенциально однозначно факторизован через $\widehat{A}$. Приведем две конструкции такой слабой эквивалентности.

Теорема 9.9.5. *Для любой предкатегории A существует категория $\widehat{A}$ и слабая эквивалентность $A \to \widehat{A}$.*

Первое доказательство. Пусть $\widehat{A}_0 :\equiv \{F : \mathcal{S}et^{A^{op}} \mid \exists (a : A).\, (\mathbf{y}a \cong F)\}$, с hom-множествами, унаследованными от $\mathcal{S}et^{A^{op}}$. Тогда включение $\widehat{A} \to \mathcal{S}et^{A^{op}}$ является вполне точным и вложением в объекты. Поскольку $\mathcal{S}et^{A^{op}}$ — категория (по теореме 9.2.5, так как $\mathcal{S}et$ такова же, в силу унивалентности), $\widehat{A}$ также является категорией.

Пусть $A \to \widehat{A}$ — вложение Йонеды. Оно является вполне точным, по следствию 9.5.6, и эссенциально сюръективным, по определению $\widehat{A}_0$. Таким образом, это слабая эквивалентность. □

Это блестящее и краткое доказательство, но у него есть недостаток, заключающийся в том, что оно увеличивает уровень универсума. Если A — категория в универсуме $\mathcal{U}$, то в этом доказательстве $\mathcal{S}et$ должно быть не меньше, чем $\mathcal{S}et_{\mathcal{U}}$. Тогда $\mathcal{S}et_{\mathcal{U}}$ и $(\mathcal{S}et_{\mathcal{U}})^{A^{op}}$ сами по себе не являются категориями в $\mathcal{U}$, а только в высшем универсуме, и *априори* то же самое верно и для $\widehat{A}$. Можно представить себе аксиому изменения размера, которая могла бы справиться с этим, но также возможно дать прямую конструкцию, используя высшие индуктивные типы.

Второе доказательство. Определим высший индуктивный тип $\widehat{A}_0$ со следующими конструкторами:

- функция $i : A_0 \to \widehat{A}_0$;
- для любых $a, b : A$ и $e : a \cong b$, равенство $je : ia = ib$;
- для каждого $a : A$, равенство $j(1_a) = \mathsf{refl}_{ia}$;
- для любых $(a, b, c : A)$, $(f : a \cong b)$ и $(g : b \cong c)$, равенство $j(g \circ f) = j(f) \cdot j(g)$;
- 1-усечение: для всех $x, y : \widehat{A}_0$, $p, q : x = y$ and $r, s : p = q$, равенство $r = s$.

Отметим, что для любых $a, b : A$ и $p : a = b$, имеет место $j(\mathsf{idtoiso}(p)) = i(p)$. Это следует с помощью индукции пути по p и третьему конструктору.

Тип $\widehat{A}_0$ будет типом объектов $\widehat{A}$; теперь построим всю остальную структуру (следующее доказательство может принести большую пользу с помощью компьютерной проверки: оно широкое и неглубокое, с множеством коротких случаев, которые необходимо рассмотреть, и большая часть работы состоит из записи того, что необходимо проверить).

Шаг 1. Определим семейство $\hom_{\widehat{A}} : \widehat{A}_0 \to \widehat{A}_0 \to \mathsf{Set}$ двойной индукцией по $\widehat{A}_0$. Поскольку Set относится к 1-типу, то можно игнорировать конструктор 1-усечения. Когда x и y имеют форму ia и ib, мы используем $\hom_{\widehat{A}}(ia, ib) :\equiv \hom_A(a, b)$. Осталось рассмотреть все остальные возможные пары конструкторов.

Сначала сохраним $x = ia$ фиксированным. Если y изменяется в соответствии с тождеством $je : ib = ib'$, для некоторого $e : b \cong b'$, то требуется тождество $\hom_A(a, b) = \hom_A(a, b')$. По унивалентности, достаточно указать эквивалентность $\hom_A(a, b) \simeq \hom_A(a, b')$. Мы считаем, что это функция $(e \circ -) : \hom_A(a, b) \to \hom_A(a, b')$. Чтобы убедиться в том, что это эквивалентность, возьмем обратное ей выражение как $(e^{-1} \circ -)$, причем свидетельства инверсии вытекают из того факта, что e^{-1} является обратным к e в A.

Если y изменяется в соответствии с тождеством $j(1_b) = \mathsf{refl}_{ib}$, потребуется тождество $(1_b \circ -) = \mathsf{refl}_{\hom_A(a,b)}$; это следует из аксиомы тождества $1_b \circ g = g$ предкатегории. Аналогично, если y изменяется в соответствии с тождеством $j(g \circ f) = j(f) \cdot j(g)$, то потребуется тождество $((g \circ f) \circ -) = (g \circ (f \circ -))$, что следует из ассоциативности.

Теперь рассмотрим другие конструкторы для x. Пусть x меняется в соответствии с тождеством $j(e) : ia = ia'$, для некоторого $e : a \cong a'$; мы снова должны иметь дело со всеми конструкторами для y. Если y есть ib, то потребуется тождество $\hom_A(a, b)\hom_A(a', b)$. По унивалентности, это может происходить из эквивалентности, и для этого можно использовать $(- \circ e^{-1})$, с обратным $(- \circ e)$.

По-прежнему, с изменением x вдоль $j(e)$, предположим, что y также изменяется вдоль $j(f)$ для некоторого $f : b \cong b'$. Затем нужно знать, что две конкатенированные тождественности

$$\hom_A(a, b) = \hom_A(a', b) = \hom_A(a', b') \qquad \text{и}$$
$$\hom_A(a, b) = \hom_A(a, b') = \hom_A(a', b')$$

идентичны. Это следует из ассоциативности: $(f \circ -) \circ e^{-1} = f \circ (- \circ e^{-1})$. Два других конструктора для y тривиальны, поскольку они являются двукратными равенствами по множествам.

Для следующих двух конструкторов x, все конструкторы для y, кроме первого, также тривиальны. Когда x изменяется вдоль $j(1_a) = \mathsf{refl}_{ia}$ и y есть ib, мы снова используем аксиому тождества. Точно так же, когда x изменяется вдоль $j(g \circ f)j(f) \cdot j(g)$, снова используется ассоциативность. Это завершает построение $\hom_{\widehat{A}} : \widehat{A}_0 \to \widehat{A}_0 \to \mathsf{Set}$.

Шаг 2. Мы придаем предкатегорную структуру на $\widehat{A}$, всегда индукцией по $\widehat{A}_0$. Теперь удаление производится в множествах ($\hom$-множества $\widehat{A}$), поэтому работать со всеми конструкторами, кроме первых двух, нетривиально.

Для тождеств, если x есть ia, то имеем $\hom_{\widehat{A}}(x, x) \equiv \hom_A(a, a)$, и определяем $1_x :\equiv 1_{ia}$. Если x изменяется вдоль je для $e : a \cong a'$, надо показать, что $\mathsf{transport}^{x \mapsto \hom_{\widehat{A}}(x,x)}(je, 1_{ia}) = 1_{ia'}$. Но, по определению, $\hom_{\widehat{A}}$, транспортирование по je задается компоновкой с e и e^{-1}, и имеет место $e \circ 1_{ia} \circ e^{-1} = 1_{ia'}$.

Для композиции, если x, y, z равны ia, ib, ic, соответственно, то $\hom_{\widehat{A}}$ сокращается до $\hom_A$, и можно определить композицию в $\widehat{A}$, которая будет композицией в A. И когда x, y или z

меняются вдоль je, мы проверяем следующие равенства:

$$\begin{aligned} e \circ (g \circ f) &= (e \circ g) \circ f, \\ g \circ f &= (g \circ e^{-1}) \circ (e \circ f), \\ (g \circ f) \circ e^{-1} &= g \circ (f \circ e^{-1}). \end{aligned}$$

Наконец, аксиомы ассоциативности и унитальности — это простые высказывания, поэтому все конструкторы, кроме первого, тривиальны. Но в этом случае имеем соответствующие аксиомы в A.

Шаг 3. Покажем, что $\widehat{A}$ — категория. То есть надо показать, что для всех $x, y : \widehat{A}$, функция $\mathsf{idtoiso} : (x = y) \to (x \cong y)$ является эквивалентностью. Сначала определим, для всех $x, y : \widehat{A}$, функцию $k_{x,y} : (x \cong y) \to (x = y)$ по индукции. Как и раньше, поскольку наша цель - множество, достаточно разобраться с первыми двумя конструкторами.

Когда x и y равны ia и ib, соответственно, имеем $\hom_{\widehat{A}}(ia, ib) \equiv \hom_A(a, b)$, с унаследованными композицией и тождественностями, так что $(ia \cong ib)$ эквивалентно $(a \cong b)$. Но теперь имеется конструктор $j : (a \cong b) \to (ia = ib)$.

Далее, если y изменяется вдоль $j(e)$ для некоторого $e : b \cong b'$, надо показать, что для $f : a \cong b$ имеем $j(j(e)_*(f)) = j(f) \cdot j(e)$. Но, по определению $\hom_{\widehat{A}}$, на равенствах, транспортирование по $j(e)$ эквивалентно пост-компоновке с e, поэтому это равенство следует из последнего конструктора $\widehat{A}_0$. Оставшийся случай, когда x изменяется вдоль $j(e)$, для $e : a \cong a'$, аналогичен. Это завершает определение $k : \prod_{(x,y:\widehat{A}_0)}(x \cong y) \to (x = y)$.

Теперь мы должны показать, что если $p : x = y$, то $k(\mathsf{idtoiso}(p)) = p$. Индукцией по p можно предположить, что это refl_x, и, следовательно, $\mathsf{idtoiso}(p) \equiv 1_x$. Теперь рассуждаем индукцией по $x : \widehat{A}_0$, и поскольку наша цель — простое высказывание (т.к. $\widehat{A}_0$ относится к 1-типу), все конструкторы, кроме первого, тривиальны. Но если x равен ia, то $k(1_{ia}) \equiv j(1_a)$, который равен refl_{ia}, посредством третьего конструктора $\widehat{A}_0$.

Чтобы завершить доказательство того, что $\widehat{A}$ является категорией, надо показать, что если $f : x \cong y$, то $\mathsf{idtoiso}(k(f)) = f$. По индукции можно предположить, что x и y суть ia и ib, соответственно, и в этом случае f должно возникать из изоморфизма $g : a \cong b$, и мы имеем $k(f) \equiv j(g)$. Однако, для любого p, имеется $\mathsf{idtoiso}(p) = p_*(1)$, поэтому, в частности, $\mathsf{idtoiso}(j(g)) = j(g)_*(1_{ia})$. И, по определению $\hom_{\widehat{A}}$ на равенствах, это получается композицией 1_{ia} с эквивалентностью g, следовательно, равно g.

Обратите внимание на сходство этого шага с методом кодирования-декодирования, используемым в §§ 2.12 и 2.13 и в главе 8. Мы снова характеризуем типы идентичности более высшего индуктивного типа (здесь, $\widehat{A}_0$) путем рекурсивного определения семейства кодов (здесь, $(x, y) \mapsto (x \cong y)$) и функций кодирования и декодирования индукцией по $\widehat{A}_0$ и по путям.

Шаг 4. Определим слабую эквивалентность $I : A \to \widehat{A}$. Возьмем $I_0 :\equiv i : A_0 \to \widehat{A}_0$, и, построив $\hom_{\widehat{A}}$, получим функции $I_{a,b} : \hom_A(a, b) \to \hom_{\widehat{A}}(Ia, Ib)$, образующие функтор $I : A \to \widehat{A}$. Этот функтор вполне точен по построению, поэтому остается показать, что он эссенциально сюръективен. То есть, для всех $x : \widehat{A}$, необходимо, чтобы просто существовал $a : A$ такой, что $Ia \cong x$. Как всегда, будем рассуждать индукцией по x, и поскольку целью является простое высказывание, все конструкторы, кроме первого, тривиальны. Но если x равен ia, то, конечно, имеется $a : A$ и $Ia \equiv ia$, следовательно, $Ia \cong ia$ (заметим, что если бы мы пытались доказать, что I *разделен* эссенциально сюръективно, мы бы запутались, потому что ничего не знаем о равенствах в A_0, и, следовательно, не имеем возможности иметь дело с другими конструкторами). $\square$

Мы называем конструкцию $A \mapsto \widehat{A}$ **пополнением Резка**, хотя также имеется аргумент (исходящий из семантики высших топосов) для названия ее **пополнением стека**.

Мы видели, что большинство прекатегорий, возникающих на практике, являются категориями, поскольку они построены из $\mathcal{S}et$, являющейся категорией согласно аксиоме унивалентности. Однако есть несколько ситуаций, когда для получения категории необходимо пополнение Резка.

Пример 9.9.6. Напомним из примера 9.1.17, что, для любого типа X, существует предгруппоид с X, в качестве типа объектов, и $\hom(x, y) :\equiv \|x = y\|_0$. Его пополнение Резка является *фундаментальным группоидом* X. Принимая во внимание, что группоиды эквивалентны 1-типам, нетрудно отождествить этот группоид с $\|X\|_1$.

Пример 9.9.7. Напомним из примера 9.1.18, что существует предкатегория с типом объектов $\mathcal{U}$ и с $\hom(X, Y) :\equiv \|X \to Y\|_0$. Ее пополнение Резка можно назвать **гомотопической категорией типов**. Ее тип объектов можно определить с помощью $\|\mathcal{U}\|_1$ (см. упражнение 9.9).

Пополнение Резка также позволяет показать, что понятие «категория» определяется понятием «слабая эквивалентность предкатегорий». Таким образом, в той мере, в какой неизбежна последняя, она является формирователем.

Теорема 9.9.8. *Предкатегория C является категорией тогда и только тогда, когда для любой слабой эквивалентности предкатегорий $H : A \to B$, индуцированный функтор $(- \circ H) : C^B \to C^A$ является изоморфизмом предкатегорий.*

Доказательство. «Только тогда» — это теорема 9.9.4. В ином случае, пусть H будет $I : A \to \widehat{A}$. Тогда, поскольку $(-\circ I)_0$ является эквивалентностью, существует $R : \widehat{A} \to A$ такой, что $RI = 1_A$. Следовательно, $IRI = I$, но опять же, поскольку $(- \circ I)_0$ является эквивалентностью, отсюда следует, что $IR = 1_{\widehat{A}}$. По лемме 9.4.9(iii), I — изоморфизм предкатегорий. Но поскольку $\widehat{A}$ — категория, то таковой является и A. $\square$

Примечания

Первоначальное определение категорий, конечно же, опиралось на теоретико-множественные основы, так что совокупность объектов категории образовывала множество (или, для больших категорий, класс). Со временем стало ясно, что все «теоретико-категорные» свойства объектов инвариантны относительно изоморфизма, и что равенство объектов в категории обычно не является очень полезным понятием. Многочисленные авторы [Bla79, Fre76, Mak95, Mak01] обнаружили, что логика с зависимой типизацией позволяет сформулировать определение категории без привлечения какого-либо понятия равенства для объектов, и что утверждения, доказываемые в этой логике, являются именно «теоретико-категорными» утверждениями, инвариантными относительно изоморфизма.

Хотя большая часть теории категорий кажется инвариантной при изоморфизме объектов и при эквивалентности категорий, есть несколько интересных исключений, которые привели к философским дискуссиям о том, что значит быть «теоретико-категорным». Так, пример 9.6.3 был предложен Питером Мэем (Peter May) в списке рассылки по категориям в мае 2010 года как ситуация, когда имеет значение, что две категории (определенные, как обычно, в теории множеств) изоморфны, а не только эквивалентны. Случай с †-категориями также несколько сбивал с толку сторонников изоморфно-инвариантной версии теории категорий, поскольку «правильное» понятие тождественности между объектами †-категории является не обычным, а *унитарным* изоморфизмом.

Категории, удовлетворяющие принципу «насыщенности» или «унивалентности», как в определении 9.1.6, впервые были рассмотрены Хофманном (Hofmann) и Штрайхером (Streicher) [HS98]. Затем эта заинтересованность возникла независимо у Воеводского (Voevodsky), Шульмана (Shulman) и, возможно, других примерно в то же время, несколько лет спустя, а ситуация была формализована Аренсом (Ahrens) и Капулькиным (Kapulkin) [AKS13]. Предложенная конструкция помещает все вышеперечисленные примеры в единый контекст: некоторые предкатегории являются категориями, другие — строгими категориями и так далее. Общая теорема о том, что «изоморфизм влечет равенство» для большого класса алгебраических структур (в предположении аксиомы унивалентности), была доказана Коквандом (Coquand) и Даниэльссоном (Danielsson); формулировка принципа структурной идентичности в §9.8 принадлежит Акселю (Aczel).

Независимо от философских соображений относительно теории категорий, Резк (Rezk) [Rez01] обнаружил, что при определении понятия $(\infty, 1)$-категории очень удобно использовать не просто *множество* объектов с пространствами морфизмов между ними, но *пространство* объектов, включающих в себя все эквивалентности и гомотопии между ними. Это дает очень хорошо работающую модель для $(\infty, 1)$-категорий как конкретных симплициальных пространств, которые Резк назвал *пространствами расслоений Сигала* (Segal). Одним из особенно хороших аспектов этой модели является аналог леммы 9.4.14: отображение пространств расслоений Сигала является эквивалентностью только тогда, когда оно является послойной эквивалентностью симплициальных пространств.

При интерпретации в модели симплициального множества унивалентных оснований Воеводского наши предкатегории подобны усеченному аналогу «пространств Сигала» Резка, в то время как наши категории соответствуют его «пространствам расслоений Сигала». Вместо этого строгие категории соответствуют (ослабленной и усеченной версии) так называемым «категориям Сегала». Известно, что категории Сигала и пространства расслоений Сигала являются эквивалентными моделями для $(\infty, 1)$-категорий (см., например, [Ber09]), так что в модели симплициального множества категории и строгие категории порождают «эквивалентные» теории категорий — хотя как мы видели, первые по-прежнему имеют много преимуществ. Однако в более общей категорной семантике высших топосов строгая категория соответствует внутренней категории (в традиционном смысле) в соответствующих 1-топосах пучков, а категория соответствует *стеку*. Последние, как правило, являются более подходящей разновидностью «категории», по сравнению с топосом.

В контексте Резка то, что мы назвали «пополнением Резка», соответствует замене слоев в модельной категории для пространств расслоений Сигала. Поскольку оно построено с использованием аргумента трансфинитной индукции, то наиболее точно соответствует нашей второй конструкции как высший индуктивный тип. Однако, в моделях высших топосов гомотопической теории типов пополнение по Резку соответствует *пополнению стека*, которое может быть построено, либо с помощью трансфинитной индукции [JT91], либо с использованием вложения Йонеды [Bun79].

Упражнения

Упражнение 9.1. Для предкатегории A и $a : A$, определите **предкатегорию срезов** A/a. Покажите, что если A — категория, то A/a также является категорией.

Упражнение 9.2. Для любого множества X, докажите, что категория срезов $\mathcal{S}et/X$ эквивалентна категории функторов $\mathcal{S}et^X$, причем в последнем случае X рассматривается как дискретная категория.

Упражнение 9.3. Докажите, что функтор является эквивалентностью категорий тогда и только тогда, когда он является *правым* сопряженным, единица и коединица которого являются изоморфизмами.

Упражнение 9.4. Дайте определение понятию **пред-2-категории**. Покажите, что предкатегории, функторы и естественные преобразования, определенные в §9.2, образуют пред-2-категорию. Аналогично, определите **пред-бикатегорию**, заменив равенства (например, такие, как в леммах 9.2.9 и 9.2.11) естественными изоморфизмами, удовлетворяющими аналогичным условиям когерентности. Определите функцию от пред-2-категорий к пред-бикатегории и покажите, что она становится эквивалентностью, когда ограничивается и ко-ограничивается теми, чьи hom-предкатегории являются категориями.

Упражнение 9.5. Определите **2-категорию** как пред-2-категорию, удовлетворяющую условию, аналогичному условию определения 9.1.6. Убедитесь, что пред-2-категория категорий $\mathcal{C}at$ является 2-категорией. Какая часть этой главы может быть посвящена произвольной 2-категории?

Упражнение 9.6. Определите 2-категорию, объекты которой являются 1-типами, морфизмы — функциями, а 2-морфизмы — гомотопиями. Докажите, что она эквивалентна, в надлежащем смысле, полной под-2-категории of $\mathcal{C}at$, натянутой на *группоиды* (категории, в которых каждая стрелка является изоморфизмом).

Упражнение 9.7. Напомним, что *строгая категория* — это предкатегория, тип объектов которой — множество. Докажите, что пред-2-категория строгих категорий эквивалентна следующей пред-2-категории.

- Ее объектами являются категории A, оснащенные сюръекцией $p_A : A'_0 \to A_0$, где A'_0 — множество.
- Ее морфизмы — функторы $F : A \to B$, оснащенные функцией $F'_0 : A'_0 \to B'_0$ такой, что $p_B \circ F'_0 = F_0 \circ p_A$.
- Ее 2-морфизмы — это просто естественные преобразования.

Упражнение 9.8. Определите пред-2-категорию †-категорий, которая имеет †-структуры в своих hom-предкатегориях. Покажите, что две †-категории равны именно тогда, когда они «унитарно эквивалентны» в подходящем смысле.

Упражнение 9.9. Докажите, что функция $X \to Y$ является эквивалентностью тогда и только тогда, когда ее образ в гомотопической категории примера 9.9.7 является изоморфизмом. Покажите, что тип объектов этой категории — $\|\mathcal{U}\|_1$.

Упражнение 9.10. Постройте †-пополнение Резка †-предкатегории в †-категории и придайте соответствующее универсальное свойство.

Упражнение 9.11. Используя фундаментальные (пред)группоиды из примеров 9.1.17 и 9.9.6 и пополнение Резка из §9.9, дайте другое доказательство теоремы ван Кампена (§8.7).

Упражнение 9.12. Пусть X и Y — множества, а $p : Y \to X$ — сюръекция.

(i) Определите, для любой предкатегории A, категорию $\mathrm{Desc}(A, p)$ **нисходящих данных** в A относительно p.

(ii) Покажите, что любая предкатегория A является **предстеком** для p, т.е. канонический функтор $A^X \to \mathrm{Desc}(A, p)$ является вполне точным.

(iii) Покажите, что если A — категория, то это **стек** для p, т.е. $A^X \to \mathrm{Desc}(A, p)$ — эквивалентность.

(iv) Покажите, что утверждение «каждая строгая категория является стеком для любой сюръекции множеств» эквивалентно аксиоме выбора.

Глава 10

Теория множеств

Наша концепция множеств как типов с сугубо простой гомотопической характеристикой, см. §3.1, сильно отличается от множеств теории Цермело-Френкеля, которые образуют совокупную иерархию со сложной вложенной структурой членства. Для многих математических целей теоретико-гомотопические множества так же хороши, как и множества Цермело-Френкеля, но имеются и важные отличия.

Мы начинаем эту главу в §10.1 с демонстрации того, что категория $\mathcal{S}et$ имеет (большей частью) обычные свойства категории множеств. В конструктивных, предикативных, унивалентных основаниях — это «ΠW-предтопос»; тогда как, если мы предполагаем пропозициональное изменение размера (§3.5), — это элементарный топос, а если предполагаем LEM и AC, то это — модель *элементарной теории категории множеств* Ловера (Lawvere).Lawvere Этого достаточно, чтобы гарантировать, что множества в гомотопической теории типов ведут себя как множества, используемые большинством математиков вне теории множеств.

Далее мы исследуем некоторые объекты, традиционно относящиеся к «теории множеств». В §§ 10.2–10.4 изучаются кардинальные и ординальные числа. В теории множеств они определяются с использованием глобального отношения принадлежности, но мы увидим, что аксиома унивалентности допускает не менее удобный, но более «структурный», подход.

Наконец, в §10.5 мы рассматриваем возможность построения *внутри* гомотопической теории типов кумулятивной иерархии множеств, снабженной бинарным отношением принадлежности, аналогичным таковому в теории множеств Цермело-Френкеля. Это объединяет высшие индуктивные типы с идеями из области алгебраической теории множеств.

В этой главе часто будут использоваться традиционные логические обозначения, описанные в §3.7. В дополнение к основной теории глав 2 и 3, используются высшие индуктивные типы для копределов и частных, как в §§ 6.8 и 6.10, а также кое-что из теории усечения из главы 7, в частности, система факторизации из §7.6 в случае $n = -1$. В §10.3 мы используем индуктивное семейство (§5.7) для описания обоснованности, а в §10.5 — более сложный высший индуктивный тип для демонстрации кумулятивной иерархии.

10.1 Категория множеств

Напомним, что в главе 9 мы определили категорию $\mathcal{S}et$, состоящую из всех 0-типов (в некотором универсуме $\mathcal{U}$) и отображений между ними, и отметили, что это именно категория (а не только предкатегория). Рассмотрим последовательно уровни структуры, которыми обладает $\mathcal{S}et$.

10.1.1 Пределы и копределы

Поскольку множества замкнуты относительно произведений, универсальное свойство произведений в теореме 2.15.2 сразу показывает, что $\mathcal{S}et$ имеет конечные произведения. Фактически, бесконечные произведения так же легко следуют из эквивалентности

$$\left(X \to \prod_{a:A} B(a)\right) \simeq \left(\prod_{a:A}(X \to B(a))\right).$$

В упражнении 2.11 мы видели, что обратный образ $f : A \to C$ и $g : B \to C$ можно определить как $\sum_{(a:A)} \sum_{(b:B)} f(a) = g(b)$; это множество, если A, B, C являются множествами и наследуют корректное универсальное свойство. Таким образом, $\mathcal{S}et$ — *полная* категория в очевидном смысле.

Поскольку множества замкнуты относительно $+$ и содержат $\mathbf{0}$, $\mathcal{S}et$ имеет конечные копроизведения. Точно так же, поскольку $\sum_{(a:A)} B(a)$ является множеством всякий раз, когда A и каждое $B(a)$ — множества, это обеспечивает копроизведение семейства B в $\mathcal{S}et$. Наконец, как показано в §7.4, обратные образы существуют в n-типах, в частности, в $\mathcal{S}et$. Таким образом, $\mathcal{S}et$ также является *кополным*.

10.1.2 Образы

Далее, покажем, что $\mathcal{S}et$ является **регулярной категорией**, т.е.:

(i) $\mathcal{S}et$ конечно полна;

(ii) ядерная пара $\mathsf{pr}_1, \mathsf{pr}_2 : (\sum_{(x,y:A)} f(x) = f(y)) \to A$ любой функции $f : A \to B$ имеет коуравнитель;

(iii) обратные образы регулярных эпиморфизмов являются регулярными эпиморфизмами.

Напомним, что **регулярный эпиморфизм** — это морфизм, являющийся уравнителем некоторой пары отображений. Таким образом, в (iii) требуется, чтобы обратный образ коуравнителя снова был коуравнивателем, но не обязательно для пары обратного образа.

Очевидным кандидатом на роль коуравнителя ядерной пары для $f : A \to B$ является *образ* f, как определено в §7.6. Напомним, что мы определили $\mathsf{im}(f) :\equiv \sum_{(b:B)} \|\mathsf{fib}_f(b)\|$ с функциями $\tilde{f} : A \to \mathsf{im}(f)$ и $i_f : \mathsf{im}(f) \to B$:

$$\tilde{f} :\equiv \lambda a.\ \left(f(a), \left|(a, \mathsf{refl}_{f(a)})\right|\right)$$
$$i_f :\equiv \mathsf{pr}_1$$

встроенными в диаграмму

$$\begin{tikzcd}
\sum_{(x,y:A)} f(x) = f(y) \arrow[r, shift left, "\mathsf{pr}_1"] \arrow[r, shift right, "\mathsf{pr}_2"'] & A \arrow[r, "\tilde{f}"] \arrow[dr, "f"'] & \mathsf{im}(f) \arrow[d, dotted, "i_f"] \\
& & B
\end{tikzcd}$$

Напомним также, что функция $f : A \to B$ называется *сюръективной*, если $\forall (b : B).\ \|\mathsf{fib}_f(B)\|$, или, эквивалентно, $\forall (b : B).\, \exists (a : A).\, f(a) = b$, и — *инъективной*, если $\forall (a, a' : A).\, (f(a) =$

$f(a')) \Rightarrow (a = a')$, или, что то же самое, если каждый из ее слоев является простым высказыванием. Поскольку это (-1)-связные и (-1)-усеченные отображения в смысле главы 7, из общей теории следует, что $\tilde{f}$ сюръективна, а i_f инъективна, и что эта факторизация сохраняется при переходе к обратному образу.

Теперь отождествим сюръективность и инъективность с соответствующими теоретико-категорными понятиями. Сначала заметим, что категорные мономорфизмы и эпиморфизмы имеют несколько более сильную эквивалентную формулировку.

Лемма 10.1.1. *Для морфизма $f : \hom_A(a, b)$ из категории A, следующие утверждения эквивалентны.*

(i) *f является* ***мономорфизмом****: для всех $x : A$ и $g, h : \hom_A(x, a)$, если $f \circ g = f \circ h$, то $g = h$.*

(ii) *(Если A является обратным образом) диагональное отображение $a \to a \times_b a$ является изоморфизмом.*

(iii) *Для всех $x : A$ и $k : \hom_A(x, b)$, тип $\sum_{(h:homA(x,a))}(k = f \circ h)$ — простое высказывание.*

(iv) *Для всех $x : A$ и $g : \hom_A(x, a)$, тип $\sum_{(h:homA(x,a))}(f \circ g = f \circ h)$ является стягиваемым.*

Доказательство. Эквивалентность условий (i) и (ii) следует из стандартной теории категорий. Теперь рассмотрим функцию $(f \circ -) : \mathsf{hom}_A(x, a) \to \mathsf{hom}_A(x, b)$ между множествами. По условию (i), она инъективна, а по (iii), ее слои являются простыми высказываниями; следовательно, они эквивалентны. А (iii) влечет (iv), если взять $k :\equiv f \circ g$ и учесть то, что простое высказывание обитаемости является стягиваемым. Наконец, (iv) влечет (i), поскольку, если $p : f \circ g = f \circ h$, то (g, refl) и (h, p) — оба населяют тип в (iv), следовательно, являются равными, и поэтому $g = h$. □

Лемма 10.1.2. *Функция $f : A \to B$ между множествами является инъективной тогда и только тогда, когда она является мономорфизмом в $\mathcal{S}et$.*

Доказательство. Оставлено читателю. □

Конечно же, **эпиморфизм** — это мономорфизм в противоположной категории. Теперь покажем, что в $\mathcal{S}et$, эпиморфизмы — это в точности сюръекции, а также, в точности коуравнители (регулярные эпиморфизмы).

Коуравнитель пары отображений $f, g : A \to B$ в $\mathcal{S}et$ определяется как 0-усечение общего (гомотопического) коуравнителя. Для ясности, можно назвать его **множество-коуравнителем**. Его универсальное свойство удобно выразить следующим образом.

Лемма 10.1.3. *Пусть $f, g : A \to B$ — функции между множествами A и B. (Множество-ко)уравнитель $c_{f,g} : B \to Q$ обладает тем свойством, что для любого множества C и любой функции $h : B \to C$ с $h \circ f = h \circ g$, тип*

$$\sum_{k:Q \to C} (k \circ c_{f,g} = h)$$

является стягиваемым.

Лемма 10.1.4. *Для любой функйии $f : A \to B$ между множествами, следующие свойства эквивалентны.*

(i) f — *эпиморфизм.*

(ii) *Рассмотрим диаграмму амальгамы*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d] & B \arrow[d, "\iota"] \\ \mathbf{1} \arrow[r, "t"'] & C_f \end{tikzcd}$$

в $\mathcal{S}et$, определяющую отображающий конус. Тогда тип C_f является стягиваемым.

(iii) f *является сюръективной.*

Доказательство. Пусть $f : A \to B$ — функция между множествами, и предположим, что это эпиморфизм; покажем, что C_f является стягиваемым. Конструктор $\mathbf{1} \to C_f$ из C_f дает элемент $t : C_f$. Надо показать, что

$$\prod_{x:C_f} x = t\,.$$

Заметим, что $x = t$ — простое высказывание, поэтому можно использовать индукцию по C_f. Конечно, когда x равно t, имеем $\mathsf{refl}_t : t = t$, поэтому достаточно найти

$$I_0 : \prod_{b:B} \iota(b) = t$$
$$I_1 : \prod_{a:A} \alpha_1(a)^{-1} \cdot I_0(f(a)) = \mathsf{refl}_t$$

где $\iota : B \to C_f$ и $\alpha_1 : \prod_{(a:A)} \iota(f(a)) = t$ — другие конструкторы из C_f. Отметим, что α_1 — это гомотопия от $\iota \circ f$ к $\mathsf{const}_t \circ f$, поэтому находим элементы

$$(\iota, \mathsf{refl}_{\iota\circ f})\,, (\mathsf{const}_t, \alpha_1) : \sum_{h:B\to C_f} \iota \circ f \sim h \circ f\,.$$

По двойственной к лемме 10.1.1 (iv) (и экстенсиональности функции) существует путь

$$\gamma : (\iota, \mathsf{refl}_{\iota\circ f}) = (\mathsf{const}_t, \alpha_1)\ .$$

Следовательно, можно определить $I_0(b) :\equiv \mathsf{happly}(\mathsf{ap}_{\mathsf{pr}_1}(\gamma), b) : \iota(b) = t$. Также, имеем

$$\mathsf{ap}_{\mathsf{pr}_2}(\gamma) : \mathsf{ap}_{\mathsf{pr}_1}(\gamma)_*\,(\mathsf{refl}_{\iota\circ f}) = \alpha_1\,.$$

Это транспортирование включает пред-композицию с f, которая коммутативна с happly. Таким образом, из транспортирования в типах путей получаем $I_0(f(a)) = \alpha_1(a)$, для любого $a : A$, что дает I_1.

Теперь предположим, что C_f стягиваемо; покажем, что f сюръективна. Сначала конструируем семейство типов $P : C_f \to \mathsf{Prop}$ рекурсией на C_f, что допустимо, поскольку Prop является множеством. В точечных конструкторах определяем

$$P(t) :\equiv \mathbf{1}\,,$$
$$P(\iota(b)) :\equiv \|\mathsf{fib}_f(b)\|\ .$$

Чтобы завершить построение P, осталось указать путь $\|\mathsf{fib}_f(f(a))\| =_{\mathsf{Prop}} \mathbf{1}$ для всех $a : A$. Однако, $\|\mathsf{fib}_f(f(a))\|$ населено посредством $(f(a), \mathsf{refl}_{f(a)})$. Поскольку это простое высказывание, это означает, что оно стягиваемо и, следовательно, эквивалентно, а значит, равно $\mathbf{1}$. Это завершает определение P. Теперь, поскольку C_f предполагается стягиваемым, то отсюда следует, что $P(x)$ эквивалентно $P(t)$ для любого $x : C_f$. В частности, $P(\iota(b)) \equiv \|\mathsf{fib}_f(b)\|$ эквивалентно $P(t) \equiv \mathbf{1}$ для каждого $b : B$, а значит, стягиваемо. Таким образом, f сюръективна.

Наконец, предположим, что $f : A \to B$ сюръективна, и рассмотрим множество C и две функции $g, h : B \to C$ со свойством $g \circ f = h \circ f$. Поскольку f предполагается сюръективной, для всех $b : B$, тип $\|\mathsf{fib}_f(b)\|$ является стягиваемым. Таким образом, имеем следующие эквивалентности:

$$\begin{aligned}
\prod_{b:B}(g(b) = h(b)) &\simeq \prod_{b:B} (\|\mathsf{fib}_f(b)\| \to (g(b) = h(b))) \\
&\simeq \prod_{b:B} (\mathsf{fib}_f(b) \to (g(b) = h(b))) \\
&\simeq \prod_{(b:B)} \prod_{(a:A)} \prod_{(p:f(a)=b)} g(b) = h(b) \\
&\simeq \prod_{a:A} g(f(a)) = h(f(a))
\end{aligned}$$

используя во второй строке то, что $g(b) = h(b)$ является простым высказыванием, поскольку C - это множество. Но по предположению, существует элемент последнего типа. □

Теорема 10.1.5. *Категория $\mathcal{S}et$ регулярна. Более того, сюръективные функции между множествами являются регулярными эпиморфизмами.*

Доказательство. Стандартной леммой в теории категорий является то, что категория является регулярной, если она допускает конечные пределы и сохраняет обратные образы ортогональной системы факторизации $(\mathcal{E}, \mathcal{M})$ с мономорфизмами $\mathcal{M}$в этом случае $\mathcal{E}$ автоматически состоит из регулярных эпиморфизмов (см., например, [Joh02, A1.3.4]). Существование системы факторизации доказано в теореме 7.6.6. □

Лемма 10.1.6. *Обратные образы регулярных эпиморфизмов из $\mathcal{S}et$ являются регулярными эпиморфизмами.*

Доказательство. В теореме 7.6.9 было показано, что обратные образы n-связных функций являются n-связными. По теореме 10.1.5, это свойство достаточно применить при $n = -1$. □

Одним из следствий того, что $\mathcal{S}et$ является регулярной категорией, является то, что у нас имеется операция «image» над подмножествами. То есть, при заданной $f : A \to B$, любое подмножество $P : \mathcal{P}(A)$ (то есть предикат $P : A \to \mathsf{Prop}$) имеет **образ**, который является подмножеством в B. Это может быть определено непосредственно, как $\{y : B \mid \exists (x : A).\, f(x) = y \wedge P(x)\}$, или косвенно, как образ (в предшествующем смысле) составной функции

$$\{x : A \mid P(x)\} \to A \xrightarrow{f} B\,.$$

Мы также иногда будем использовать общее обозначение $\{f(x) \mid P(x)\}$ для образа P.

10.1.3 Частные

Теперь, когда мы знаем, что $\mathcal{S}et$ регулярно, чтобы показать, что $\mathcal{S}et$ является точным, нужно убедиться, что каждое отношение эквивалентности эффективно. Другими словами, для отношения эквивалентности $R : A \to A \to \mathsf{Prop}$, существует коуравнитель c_R пары $\mathsf{pr}_1, \mathsf{pr}_2 : \sum_{(x,y:A)} R(x,y) \to A$ и, кроме того, pr_1 и pr_2 образуют ядро пары из c_R.

Мы уже видели в §6.10 два общих способа построения фактор-множества по отношению эквивалентности $R : A \to A \to \mathsf{Prop}$. Первый можно охарактеризовать как множество-коуравнитель двух проекций.

$$\mathsf{pr}_1, \mathsf{pr}_2 : \left(\sum_{x,y:A} R(x,y)\right) \to A\,.$$

Важным свойством такого частного является следующее.

Определение 10.1.7. Отношение $R : A \to A \to \mathsf{Prop}$ называется **эффективным**, если квадрат

$$\begin{tikzcd}
\sum_{(x,y:A)} R(x,y) \arrow[r, "\mathsf{pr}_1"] \arrow[d, "\mathsf{pr}_2"'] & A \arrow[d, "c_R"] \\
A \arrow[r, "c_R"'] & A/R
\end{tikzcd}$$

является обратным образом.

Поскольку стандартный обратный образ c_R и сам есть $\sum_{(x,y:A)}(c_R(x) = c_R(y))$, то по теореме 4.7.7 это эквивалентно тому, что каноническое преобразование $\prod_{(x,y:A)} R(x,y) \to (c_R(x) = c_R(y))$ является послойной эквивалентностью.

Лемма 10.1.8. *Предположим, что (A, R) является отношением эквивалентности. Тогда существует эквивалентность*

$$(c_R(x) = c_R(y)) \simeq R(x,y)$$

для любых $x, y : A$. Другими словами, отношения эквивалентности эффективны.

Доказательство. Начнем с расширения R до отношения $\widetilde{R} : A/R \to A/R \to \mathsf{Prop}$, которое, как затем покажем, эквивалентно тождественному типу на A/R. Определим $\widetilde{R}$ двойной индукцией по A/R (заметьте, что Prop является множеством по унивалентности для простых высказываний). Определим $\widetilde{R}(c_R(x), c_R(y)) :\equiv R(x,y)$. Для $r : R(x,x')$ и $s : R(y,y')$, транзитивность и симметричность R дают эквивалентность от $R(x,y)$ к $R(x',y')$. Это завершает определение $\widetilde{R}$.

Осталось показать, что $\widetilde{R}(w,w') \simeq (w = w')$ для любых $w, w' : A/R$. Направление $(w = w') \to \widetilde{R}(w,w')$ следует из транспортировки, как только мы покажем, что $\widetilde{R}$ рефлексивно, что является простой индукцией. Другое направление $\widetilde{R}(w,w') \to (w = w')$ является простым высказыванием, поэтому, поскольку $c_R : A \to A/R$ сюръективно, достаточно предположить, что w и w' имеют вид $c_R(x)$ и $c_R(y)$. Но в этом случае имеем каноническое отображение $\widetilde{R}(c_R(x), c_R(y)) :\equiv R(x,y) \to (c_R(x) = c_R(y))$ (еще раз обратите внимание на проявление метода кодирования-декодирования). □

Вторая конструкция частных представляет собой множество классов эквивалентности R (подмножество его степенного множества):

$$A \mathbin{/\!\!/} R :\equiv \{P : A \to \mathsf{Prop} \mid P \text{ — класс эквивалентности } R\} .$$

Это требует пропозиционального изменения размера, чтобы оставаться в том же универсуме, что и A с R.

Обратите внимание, что если рассматривать R как функцию от A к $A \to \mathsf{Prop}$, то $A \mathbin{/\!\!/} R$ эквивалентно $\mathsf{im}(R)$, как это построено в §10.1.2. А в теореме 10.1.5 было показано, что образы являются коуравнителями. Тогда, в частности, сразу получаем диаграмму коуравнителя

$$\textstyle\sum_{x,y:A} R(x) = R(y) \overset{\mathsf{pr}_1}{\underset{\mathsf{pr}_2}{\rightrightarrows}} A \longrightarrow A \mathbin{/\!\!/} R \ .$$

Мы можем использовать это, чтобы дать альтернативное доказательство того, что любое отношение эквивалентности эффективно, и что два определения частных согласованы.

Теорема 10.1.9. *Для любой функции $f : A \to B$ между любыми двумя множествами, отношение $\mathsf{ker}(f) : A \to A \to \mathsf{Prop}$, заданное как $\mathsf{ker}(f, x, y) :\equiv (f(x) = f(y))$, эффективно.*

Доказательство. Будем использовать то, что, для $\mathsf{pr}_1, \mathsf{pr}_2 : (\sum_{(x,y:A)} f(x) = f(y)) \to A$, коуравнителем является $\mathsf{im}(f)$. Заметим, что ядерная пара функции

$$c_f :\equiv \lambda a.\, \big(f(a), \big\|a, \mathsf{refl}_{f(a)}\big\|\big) : A \to \mathsf{im}(f)$$

состоит из двух проекций

$$\mathsf{pr}_1, \mathsf{pr}_2 : \left(\sum_{x,y:A} c_f(x) = c_f(y)\right) \to A\,.$$

Для любых $x, y : A$, имеем эквивалентности

$$\begin{aligned} c_f(x) = c_f(y) &\simeq \left(\sum_{p:f(x)=f(y)} p_*\left(\big\|(x, \mathsf{refl}_{f(x)})\big\|\right) = \big\|(y, \mathsf{refl}_{f(x)})\big\|\right) \\ &\simeq (f(x) = f(y))\,, \end{aligned}$$

где последняя эквивалентность верна, поскольку $\|\mathsf{fib}_f(b)\|$ — простое высказывание для любого $b : B$. Таким образом,

$$\left(\sum_{x,y:A} c_f(x) = c_f(y)\right) \simeq \left(\sum_{x,y:A} f(x) = f(y)\right)$$

и, следовательно, можно заключить, что $\mathsf{ker} f$ является эффективным отношением для любой функции f. □

Теорема 10.1.10. *Отношения эквивалентности эффективны, и существует эквивалентность $A/R \simeq A \mathbin{/\!\!/} R$.*

Доказательство. Необходимо проанализировать диаграмму коуравнителя

$$\textstyle\sum_{(x,y:A)} R(x) = R(y) \overset{\mathsf{pr}_1}{\underset{\mathsf{pr}_2}{\rightrightarrows}} A \longrightarrow A \mathbin{/\!\!/} R \ .$$

По аксиоме унивалентности, тип $R(x) = R(y)$ эквивалентен типу гомотопий от $R(x)$ к $R(y)$, что эквивалентно $\prod_{(z:A)} R(x,z) \simeq R(y,z)$. Поскольку R — отношение эквивалентности, последнее пространство эквивалентно $R(x,y)$. В итоге получаем $(R(x) = R(y)) \simeq R(x,y)$, поэтому R эффективно, поскольку оно эквивалентно эффективному отношению. К тому же, диаграмма

$$\textstyle\sum_{(x,y:A)} R(x,y) \underset{\mathsf{pr}_2}{\overset{\mathsf{pr}_1}{\rightrightarrows}} A \longrightarrow A \mathbin{/\!\!/} R$$

— диаграмма коуравнителя. Поскольку коуравнители единственны с точностью до эквивалентности, отсюда следует, что $A/R \simeq A \mathbin{/\!\!/} R$. □

Завершим этот раздел упоминанием о возможной третьей конструкции фактор-множества A по отношению эквивалентности R. Рассмотрим предкатегорию с объектами A и $\hom$-множествами R; тип объектов из пополнения Резка (см. §9.9) этой предкатегории будет тогда частным. Читателю предлагается уточнить детали.

10.1.4 $\mathcal{S}$et — это ΠW-предтопос

Понятие ΠW-*предтопоса* — то есть, локально декартово замкнутой категории с дизъюнктивными конечными копроизведениями, эффективными отношениями эквивалентности и инициальными алгебрами для полиномиальных эндофункторов — задумано как «предикативное» понятие топоса, т.е. как категория «предикативных множеств», которая может служить целям конструктивной математики, что обычная категория множеств выполняет в классической математике.

Обычно, в конструктивной теории типов прибегают к внешнему построению «сетоидов» — точному пополнению — для получения категории с определенными замкнутыми свойствами. Именно, частные с хорошим поведением требуются для многих математических построений, которые обычно включают (неконструктивные) степенные множества. Примечательно, что унивалентные основания обеспечивают эти конструкции *изнутри* (через высшие индуктивные типы), не требуя подобных внешних построений. Это представляет собой мощное преимущество нашего подхода, в чем мы убедимся в последующих примерах.

Теорема 10.1.11. *Категория $\mathcal{S}et$ является* ΠW-*предтопосом.*

Доказательство. У нас имеется инициальный объект $\mathbf{0}$ и конечные дизъюнктивные суммы $A + B$. Они устойчивы под обратным образом просто потому, что последний имеет правый сопряженный. Действительно, $\mathcal{S}et$ локально декартово замкнуто, поскольку для любого отображения $f : A \to B$ между множествами «фибрантная замена» $\sum_{(a:A)} f(a) = b$ эквивалентна на A (над B), и у нас есть зависимые типы функций для замены. Мы только что показали, что $\mathcal{S}et$ регулярна (теорема 10.1.5) и что частные эффективны (лемма 10.1.8). Таким образом, имеем локально декартово замкнутый предтопос. Наконец, поскольку n-типы замкнуты относительно образования W-типов (упражнение 7.3), и, согласно теореме 5.4.7, W-типы являются инициальными алгебрами для полиномиальных эндофункторов, то ясно, что $\mathcal{S}et$ является ΠW-предтопосом. □

Возникает естественный вопрос: что вообще мешает $\mathcal{S}et$ быть (элементарным) топосом? В дополнение к уже упомянутой структуре, топос имеет *классификатор подобъектов*: точечный объект, классифицирующий (по классам эквивалентности) мономорфизмы (на самом деле, при наличии классификатора подобъектов все становится несколько проще: необходимо надлежащее декартово замыкание, чтобы получить копределы). В гомотопической теории типов унивалентность означает, что тип $\mathsf{Prop} :\equiv \sum_{(X:\mathcal{U})} \mathsf{isProp}(X)$ классифицирует мономорфизмы (аргументом, аналогичным §4.8), но в целом он такой же большой, как окружающий универсум $\mathcal{U}$. Таким образом, это «множество» в смысле 0-типа, но он не «малое» в смысле того, что является объектом в $\mathcal{U}$, следовательно, оно не является объектом категории $\mathcal{S}et$. Однако, если мы примем подходящую форму пропозиционального изменения размера (см. §3.5), то сможем найти малую версию Prop, так что $\mathcal{S}et$ станет элементарным топосом.

Теорема 10.1.12. *Если существует тип $\Omega : \mathcal{U}$ всех простых высказываний, то категория $\mathcal{S}et_{\mathcal{U}}$ является элементарным топосом.*

Достаточным условием для этого является закон исключения третьего в «простой пропозициональной» форме, которую мы обозначили как LEM; так как тогда мы имеем $\mathsf{Prop} = \mathbf{2}$, который *является* малым, и который тогда также классифицирует все простые высказывания. Более того, в теории топосов хорошо известным достаточным условием для LEM является аксиома выбора, которая, конечно, часто считается аксиомой в классической теории множеств. В следующем разделе мы кратко исследуем связь между этими условиями в нашей установке.

10.1.5 Аксиома выбора подразумевает исключение третьего

Начнем с леммы.

Лемма 10.1.13. *Если A — простое высказывание, то его надстройка $\Sigma(A)$ является множеством, а A эквивалентно $\mathsf{N} =_{\Sigma(A)} \mathsf{S}$.*

Доказательство. Чтобы показать, что $\Sigma(A)$ является множеством, определим семейство $P : \Sigma(A) \to \Sigma(A) \to \mathcal{U}$ со свойством, что $P(x,y)$ является простым высказыванием для любых $x,y : \Sigma(A)$, которое эквивалентно его тождественному типу $\mathsf{id}_{\Sigma(A)}$. Определим:

$$P(\mathsf{N},\mathsf{N}) :\equiv \mathbf{1} \qquad\qquad P(\mathsf{S},\mathsf{N}) :\equiv A$$
$$P(\mathsf{N},\mathsf{S}) :\equiv A \qquad\qquad P(\mathsf{S},\mathsf{S}) :\equiv \mathbf{1}$$

Мы должны проверить, что это определение сохраняет пути. Для любого $a : A$, существует меридиан $\mathsf{merid}(a) : \mathsf{N} = \mathsf{S}$, поэтому мы также должны иметь

$$P(\mathsf{N},\mathsf{N}) = P(\mathsf{N},\mathsf{S}) = P(\mathsf{S},\mathsf{N}) = P(\mathsf{S},\mathsf{S})\,.$$

Но, поскольку A населено элементами a, оно эквивалентно $\mathbf{1}$, поэтому имеем

$$P(\mathsf{N},\mathsf{N}) \simeq P(\mathsf{N},\mathsf{S}) \simeq P(\mathsf{S},\mathsf{N}) \simeq P(\mathsf{S},\mathsf{S})\,.$$

Аксиома унивалентности превращает их в искомые равенства. Кроме того, $P(x,y)$ является простым высказыванием для всех $x,y : \Sigma(A)$, что доказывается индукцией по x и y, и с использованием того факта, что быть простым высказыванием — это простое высказывание.

Заметим, что P — рефлексивное отношение. Следовательно, можно применить теорему 7.2.2, поэтому достаточно построить $\tau : \prod_{(x,y:\Sigma(A))} P(x,y) \to (x = y)$. Сделаем это двойной индукцией. Когда x есть N, определяем $\tau(\mathsf{N})$ как

$$\tau(\mathsf{N},\mathsf{N},u) :\equiv \mathsf{refl}_{\mathsf{N}} \qquad \text{и} \qquad \tau(\mathsf{N},\mathsf{S},a) :\equiv \mathsf{merid}(a)\,.$$

Если A населен элементами a, то $\mathsf{merid}(a) : \mathsf{N} = \mathsf{S}$, и также понадобится $\mathsf{merid}(a)_*(\tau(\mathsf{N},\mathsf{N})) = \tau(\mathsf{N},\mathsf{S})$. Это получается за счет функциональной экстенсиональности, используя то, что для всех $x : A$,

$$\mathsf{merid}(a)_*(\tau(\mathsf{N},\mathsf{N},x)) = \tau(\mathsf{N},\mathsf{N},x) \cdot \mathsf{merid}(a)^{-1} \equiv \mathsf{refl}_{\mathsf{N}} \cdot \mathsf{merid}(a) = \mathsf{merid}(a) = \mathsf{merid}(x) \equiv \tau(\mathsf{N},\mathsf{S},x)\,.$$

Симметричным образом можно определить $\tau(\mathsf{S})$ как

$$\tau(\mathsf{S},\mathsf{N},a) :\equiv \mathsf{merid}(a)^{-1} \qquad \text{и} \qquad \tau(\mathsf{S},\mathsf{S},u) :\equiv \mathsf{refl}_{\mathsf{S}}\,.$$

Чтобы завершить построение τ, нужно проверить $\mathsf{merid}(a)_*(\tau(\mathsf{N})) = \tau(\mathsf{S})$, для любого $a : A$. Проверка проводится аналогично индукцией по второму аргументу τ.

Таким образом, по теореме 7.2.2 имеем, что $\Sigma(A)$ — это множество и что $P(x,y) \simeq (x = y)$ для всех $x, y : \Sigma(A)$. Если положить $x :\equiv \mathsf{N}$ и $y :\equiv \mathsf{S}$, получим $A \simeq (\mathsf{N} =_{\Sigma(A)} \mathsf{S})$, что и требовалось. □

Теорема 10.1.14 (Diaconescu)**.** *Аксиома выбора подразумевает закон исключения третьего.*

Доказательство. Будем использовать эквивалентную форму выбора, данную в лемме 3.8.2. Рассмотрим простое высказывание A. Функция $f : \mathbf{2} \to \Sigma(A)$, определенная посредством $f(0_{\mathbf{2}}) :\equiv \mathsf{N}$ и $f(1_{\mathbf{2}}) :\equiv \mathsf{S}$, является сюръективной. Действительно, имеется $(0_{\mathbf{2}}, \mathsf{refl}_{\mathsf{N}}) : \mathsf{fib}_f(\mathsf{N})$ и $(1_{\mathbf{2}}, \mathsf{refl}_{\mathsf{S}}) : \mathsf{fib}_f(\mathsf{S})$. Поскольку $\|\mathsf{fib}_f(x)\|$ — простое высказывание, по индукции следует заявленная сюръективность.

По лемме 10.1.13 надстройка $\Sigma(A)$ является множеством, поэтому по аксиоме выбора просто существует сечение $g : \Sigma(A) \to \mathbf{2}$ функции f. Поскольку равенство на $\mathbf{2}$ разрешимо, получаем

$$(g(f(0_{\mathbf{2}})) = g(f(1_{\mathbf{2}}))) + \neg(g(f(0_{\mathbf{2}})) = g(f(1_{\mathbf{2}})))\,,$$

и, поскольку g является сечением f, то инъективно,

$$(f(0_{\mathbf{2}}) = f(1_{\mathbf{2}})) + \neg(f(0_{\mathbf{2}}) = f(1_{\mathbf{2}}))\,.$$

Наконец, поскольку $(f(0_{\mathbf{2}}) = f(1_{\mathbf{2}})) = (N = S) = A$ по лемме 10.1.13, имеем $A + \neg A$. □

Теорема 10.1.15. *Если аксиома выбора верна, то категория $\mathcal{S}et$ является четко обозначенным логическим элементарным топосом с выбором.*

Доказательство. Поскольку AC влечет LEM, имеются логические элементарные топосы с выбором по теореме 10.1.12 и следующему за ней замечанию. Мы оставляем доказательство четкой обозначенности в качестве упражнения для читателя (упражнение 10.3). □

Замечание 10.1.16. Условия для категории, упомянутые в этой теореме, известны как аксиомы Ловера для *элементарной теории категории множеств*. [Law05].

10.2 Кардинальные числа

Определение 10.2.1. Тип кардинальных чисел — это 0-усечение типа Set множеств:

$$\mathsf{Card} :\equiv \|\mathsf{Set}\|_0\,.$$

Таким образом, **кардинальное число**, или **кардинал**, является обитателем в $\mathsf{Card} \equiv \|\mathsf{Set}\|_0$. Технически, конечно, имеется отдельный тип $\mathsf{Card}_{\mathcal{U}}$, ассоциированный с каждым $\mathcal{U}$.

Как обычно для усечений, если A — множество, то $|A|_0$ обозначает его образ в канонической проекции $\mathsf{Set} \to \|\mathsf{Set}\|_0 \equiv \mathsf{Card}$; мы обозначаем через $|A|_0$ **кардинальность** (кардинальное число) A. По определению, Card — это множество. Оно также наследует структуру полукольца из Set.

Определение 10.2.2. Операция **кардинального сложения**

$$(- + -) : \mathsf{Card} \to \mathsf{Card} \to \mathsf{Card}$$

определяется индукцией по усечению:

$$|A|_0 + |B|_0 :\equiv |A + B|_0 \ .$$

Обоснование определения. Поскольку $\mathsf{Card} \to \mathsf{Card}$ — это множество, для определения $(\alpha + -) : \mathsf{Card} \to \mathsf{Card}$, для всех $\alpha : \mathsf{Card}$, по индукции достаточно предположить, что α есть $|A|_0$, для некоторого $A : \mathsf{Set}$. Теперь мы хотим определить $(|A|_0 + -) : \mathsf{Card} \to \mathsf{Card}$, т.е. определить $|A|_0 + \beta : \mathsf{Card}$, для всех $\beta : \mathsf{Card}$. Однако, поскольку Card — это множество, по индукции достаточно предположить, что β есть $|B|_0$, для некоторого $B : \mathsf{Set}$. Но тогда можно определить $|A|_0 + |B|_0$ как $|A + B|_0$. □

Определение 10.2.3. Аналогично, операция **кардинального умножения**

$$(- \cdot -) : \mathsf{Card} \to \mathsf{Card} \to \mathsf{Card}$$

определяется индукцией по усечению:

$$|A|_0 \cdot |B|_0 :\equiv |A \times B|_0 \ .$$

Лемма 10.2.4. Card — *коммутативное полукольцо, т.е. для* $\alpha, \beta, \gamma : \mathsf{Card}$ *имеем следующее:*

$$\begin{aligned}
(\alpha + \beta) + \gamma &= \alpha + (\beta + \gamma) \\
\alpha + 0 &= \alpha \\
\alpha + \beta &= \beta + \alpha \\
(\alpha \cdot \beta) \cdot \gamma &= \alpha \cdot (\beta \cdot \gamma) \\
\alpha \cdot 1 &= \alpha \\
\alpha \cdot \beta &= \beta \cdot \alpha \\
\alpha \cdot (\beta + \gamma) &= \alpha \cdot \beta + \alpha \cdot \gamma
\end{aligned}$$

где $0 :\equiv |\mathbf{0}|_0$, *а* $1 :\equiv |\mathbf{1}|_0$.

Доказательство. Докажем коммутативность умножения, $\alpha \cdot \beta = \beta \cdot \alpha$; остальные равенства доказываются аналогично. Поскольку Card — это множество, тип $\alpha \cdot \beta = \beta \cdot \alpha$ — это простое высказывание и, в частности, множество. Таким образом, по индукции достаточно предположить, что α и β имеют вид $|A|_0$ и $|B|_0$, соответственно, для некоторых $A, B : \mathsf{Set}$. Теперь $|A|_0 \cdot |B|_0 :\equiv |A \times B|_0$ и $|B|_0 \cdot |A|_0 :\equiv |B \times A|_0$, так что достаточно показать, что $A \times B = B \times A$. Наконец, в силу унивалентности достаточно предъявить эквивалентность $A \times B \simeq B \times A$. Но это просто: возьмем $(a, b) \mapsto (b, a)$ и его очевидное обратное. □

Определение 10.2.5. Операция **кардинального возведения в степень** (потенцирования) также определяется индукцией по усечению:

$$|A|_0^{|B|_0} :\equiv |B \to A|_0 \ .$$

Лемма 10.2.6. *Для* $\alpha, \beta, \gamma : \mathsf{Card}$ *имеет место*

$$\begin{aligned} \alpha^0 &= 1 \\ 1^\alpha &= 1 \\ \alpha^1 &= \alpha \\ \alpha^{\beta+\gamma} &= \alpha^\beta \cdot \alpha^\gamma \\ \alpha^{\beta\cdot\gamma} &= (\alpha^\beta)^\gamma \\ (\alpha\cdot\beta)^\gamma &= \alpha^\gamma \cdot \beta^\gamma \end{aligned}$$

Доказательство. Точно такое же, как в лемме 10.2.4. □

Определение 10.2.7. Отношение **кардинального неравенства**

$$(- \leqslant -) : \mathsf{Card} \to \mathsf{Card} \to \mathsf{Prop}$$

определяется индукцией по усечению:

$$|A|_0 \leqslant |B|_0 :\equiv \|\mathsf{inj}(A, B)\|$$

где $\mathsf{inj}(A, B)$ — тип инъекций от A к B. Другими словами, $|A|_0 \leqslant |B|_0$ означает, что существует просто инъекция от A к B.

Лемма 10.2.8. *Кардинальное неравенство является предпорядком, т.е. для* $\alpha, \beta, \gamma : \mathsf{Card}$ *имеем*

$$\begin{gathered} \alpha \leqslant \alpha \\ (\alpha \leqslant \beta) \to (\beta \leqslant \gamma) \to (\alpha \leqslant \gamma) \end{gathered}$$

Доказательство. Как и выше, можно использовать индукцию по усечению. Например, поскольку $(\alpha \leqslant \beta) \to (\beta \leqslant \gamma) \to (\alpha \leqslant \gamma)$ является простым предложением, индукцией по 0-усечению можно предположить, что α, β и γ есть $|A|_0$, $|B|_0$ и $|C|_0$, соответственно. Тогда, поскольку $|A|_0 \leqslant |C|_0$ является простым высказыванием, индукцией по (-1)-усечению можно предполагать заданными инъекции $f : A \to B$ и $g : B \to C$. Но тогда $g \circ f$ является инъекцией от A к C, поэтому выполняется $|A|_0 \leqslant |C|_0$. Доказательство рефлексивности еще проще. □

Точно так же можно показать, что кардинальное неравенство совместимо с операциями полукольца.

Лемма 10.2.9. *Пусть даны следующие утверждения:*

(i) *существует инъекция* $A \to B$;

(ii) *существует сюръекция* $B \to A$.

Тогда, предполагая исключение третьего:

- *для* $a_0 : A$, *имеем* (i)$\to$(ii);
- *следовательно, если* A *просто обитаем, имеем* (i)$\to$ *просто* (ii);
- *в предположении аксиомы выбора, имеем* (ii)$\to$ *просто* (i).

Доказательство. Если $f : A \to B$ — инъекция, определим $g : B \to A$ при $b : B$ следующим образом. Поскольку f инъективна, слой f при b является простым высказыванием. Следовательно, по закону исключения третьего, либо существует $a : A$ с $f(a) = b$, либо нет. В первом случае определим $g(b) :\equiv a$; в противном случае установим $g(b) :\equiv a_0$. Тогда для любого $a : A$, имеем $a = g(f(a))$, так что g сюръективна.

Второй случай следует отсюда индукцией по усечению. Для третьего, если $g : B \to A$ сюръективна, то, по аксиоме выбора, просто существует функция $f : A \to B$ с $g(f(a)) = a$ для всех a. Но тогда f должна быть инъекцией. □

Теорема 10.2.10 (Schroeder-Bernstein)**.** *Предполагая исключение третьего, для множеств A и B имеем*

$$\mathsf{inj}(A, B) \to \mathsf{inj}(B, A) \to (A \cong B) .$$

Доказательство. Обычная аргументация «туда-обратно» может быть применена здесь без существенных изменений. Заметим, что при этом фактически порождает изоморфизм $A \cong B$ (при условии исключения третьего, чтобы можно было бы установить, принадлежит ли данный элемент циклу, бесконечной цепочке, цепочке, начинающейся в A, или цепочке, начинающейся в B). □

Следствие 10.2.11. *Предполагая исключение третьего, кардинальное неравенство является частичным порядком, т.е. для $\alpha, \beta : \mathsf{Card}$ имеем*

$$(\alpha \leqslant \beta) \to (\beta \leqslant \alpha) \to (\alpha = \beta) .$$

Доказательство. Поскольку $\alpha = \beta$ — простое высказывание, индукцией по усечению можно считать, что α и β есть $|A|_0$ и $|B|_0$, соответственно, и что имеются инъекции $f : A \to B$ и $g : B \to A$. Но тогда, по теореме Шредера-Бернштейна, имеется изоморфизм $A \cong B$ и, следовательно, равенство $|A|_0 = |B|_0$. □

Наконец, можно воспроизвести теорему Кантора, утверждающую, что для каждого кардинала существует больший.

Теорема 10.2.12 (Cantor)**.** *Для $A : \mathsf{Set}$, сюръекция $A \to (A \to \mathbf{2})$ не существует.*

Доказательство. Пусть $A \to (A \to \mathbf{2})$ — произвольная функция. Определим $g : A \to \mathbf{2}$ как $g(a) :\equiv \neg f(a)(a)$. Если $g = f(a_0)$, то $g(a_0) = f(a_0)(a_0)$, но $g(a_0) = \neg f(a_0)(a_0)$, получено противоречие. Таким образом, f не сюръективна. □

Следствие 10.2.13. *Предполагая исключение третьего, для любого $\alpha : \mathsf{Card}$, существует кардинал β такой, что $\alpha \leqslant \beta$ и $\alpha \neq \beta$.*

Доказательство. Пусть $\beta = 2^\alpha$. Надо предъявить простое высказывание, поэтому по индукции можно предположить, что α есть $|A|_0$, так, что $\beta :\equiv |A \to \mathbf{2}|_0$. Используя исключение третьего, имеем функцию $f : A \to (A \to \mathbf{2})$, определенную как

$$f(a)(a') :\equiv \begin{cases} 1_{\mathbf{2}} & a = a' \\ 0_{\mathbf{2}} & a \neq a' \end{cases}$$

И, если $f(a) = f(a')$, то $f(a')(a) = f(a)(a) = 1_{\mathbf{2}}$, так, что $a = a'$; следовательно, f инъективна. Таким образом, $\alpha \equiv |A|_0 \leqslant |A \to \mathbf{2}|_0 \equiv 2^\alpha$.

С другой стороны, если $2^\alpha \leqslant \alpha$, то имеется инъекция $(A \to \mathbf{2}) \to A$. По лемме 10.2.9, поскольку есть $(\lambda x. 0_{\mathbf{2}}) : A \to \mathbf{2}$ и исключение третьего, то имеется сюръекция $A \to (A \to \mathbf{2})$, что противоречит теореме Кантора. □

10.3 Ординальные числа

Определение 10.3.1. Пусть A — множество и

$$(- < -) : A \to A \to \mathsf{Prop}$$

— бинарное отношение на A. Определим по индукции, что означает **доступность** элемента $a : A$ посредством $<$:

- если b доступен, для любого $b < a$, то доступен и a.

Будем писать $\mathsf{acc}(a)$ для указания доступности a.

Может показаться, что такое индуктивное определение никогда не сможет сработать, но, конечно, если a обладает свойством, что не существует b такого, что $b < a$, то доступность a является бессмысленной.

Обратите внимание, что это индуктивное определение семейства типов, подобных типу векторов, рассмотренному в §5.7. Точнее, это семейство имеет один конструктор, скажем $\mathsf{acc}_<$, с типом

$$\mathsf{acc}_< : \prod_{a:A} \left(\prod_{b:A} (b < a) \to \mathsf{acc}(b) \right) \to \mathsf{acc}(a)\,.$$

Принцип индукции для acc гласит, что для любого $P : \prod_{(a:A)} \mathsf{acc}(a) \to \mathcal{U}$, если имеет место

$$f : \prod_{(a:A)} \prod_{(h:\prod_{(b:A)}(b<a)\to\mathsf{acc}(b))} \left(\prod_{(b:A)} \prod_{(l:b<a)} P(b, h(b,l)) \right) \to P(a, \mathsf{acc}_<(a,h))\,,$$

то имеем $g : \prod_{(a:A)} \prod_{(c:\mathsf{acc}(a))} P(a,c)$, определенное по индукции, с

$$g(a, \mathsf{acc}_<(a,h)) \equiv f(a,\, h,\, \lambda b.\, \lambda l.\, g(b, h(b,l)))\,.$$

В этом не просто разобраться, но обычно применяется только более простой случай, когда $P : A \to \mathcal{U}$ зависит только от A. В этом случае второй и третий аргументы f могут быть объединены, так что остается доказать

$$f : \prod_{a:A} \left(\prod_{b:A} (b < a) \to \mathsf{acc}(b) \times P(b) \right) \to P(a)\,.$$

То есть, мы предполагаем, что каждый $b < a$ доступен и $g(b) : P(b)$ определено, а из них определяется $g(a) : P(a)$.

Пропуск второго аргумента в P оправдывается следующей леммой, доказательство которой — единственное место, где мы используем более общую форму принципа индукции.

Лемма 10.3.2. *Доступность является простым свойством.*

Доказательство. Мы должны показать, что для любых $a : A$ и $s_1, s_2 : \mathsf{acc}(a)$, имеем $s_1 = s_2$. Докажем это индукцией по s_1, с

$$P_1(a, s_1) :\equiv \prod_{s_2:\mathsf{acc}(a)} (s_1 = s_2)\,.$$

Таким образом, надо показать, что для любых $a : A$, $h_1 : \prod_{(b:A)}(b < a) \to \mathsf{acc}(b)$ и

$$k_1 : \prod_{(b:A)} \prod_{(l:b<a)} \prod_{(t:\mathsf{acc}(b))} h_1(b, l) = t\,,$$

имеем $\mathsf{acc}_<(a, h) = s_2$ для любого $s_2 : \mathsf{acc}(a)$. Мы рассматриваем это как $\prod_{(a:A)} \prod_{(s_2:\mathsf{acc}(a))} P_2(a, s_2)$, где

$$P_2(a, s_2) :\equiv \prod_{(h_1:\cdots)} \prod_{(k_1:\cdots)} (\mathsf{acc}_<(a, h_1) = s_2)\,;$$

таким образом, можно доказать это индукцией по s_2. Следовательно, мы предполагаем $h_2 : \prod_{(b:A)}(b < a) \to \mathsf{acc}(b)$, и соответствующий k_2 с огромным, но неприменимым типом, и должны показать, что для любых h_1 и k_1, с типами, указанными выше, имеем $\mathsf{acc}_<(a, h_1) = \mathsf{acc}_<(a, h_2)$. Исходя из функциональной экстенсиональности, достаточно показать, что $h_1(b, l) = h_2(b, l)$ для всех $b : A$ и $l : b < a$. А это следует из k_1. $\square$

Определение 10.3.3. Бинарное отношение $<$ на множестве A является отношением **(вполне) обоснованности**, если каждый элемент A является доступным.

Суть обоснованности состоит в том, что для $P : A \to \mathcal{U}$, можно использовать принцип индукции acc, чтобы вывести $\prod_{(a:A)} \mathsf{acc}(a) \to P(a)$, а затем применить обоснованность для заключения $\prod_{(a:A)} P(a)$. Другими словами, если из $\forall(b : A).\,(b < a) \to P(b)$ можно доказать $P(a)$, то $\forall(a : A).\,P(a)$. Это называется **индукцией обоснованности**.

Лемма 10.3.4. *Обоснованность — это просто свойство.*

Доказательство. Обоснованность $<$ является типом $\prod_{(a:A)} \mathsf{acc}(a)$, который есть простое высказывание, поскольку таковым является каждый $\mathsf{acc}(a)$. $\square$

Пример 10.3.5. Возможно, наиболее знакомым отношением обоснованности является обычный строгий порядок на $\mathbb{N}$. Чтобы показать эту обоснованность, надо показать, что n доступно для каждого $n : \mathbb{N}$. Это обычное доказательство «сильной индукции» из обычной индукции на $\mathbb{N}$.

В частности, докажем индукцией по $n : \mathbb{N}$, что k является доступным для всех $k \leqslant n$. Базовый случай заключается в доступности 0, что бессодержательно истинно, поскольку нет ничего, что строго меньше 0. Для индуктивного шага предположим, что k доступен для всех $k \leqslant n$, то есть для всех $k < n + 1$; следовательно, по определению доступен также и $n + 1$.

Другое отношение на $\mathbb{N}$, которое также является обоснованным, получается, если для всех $n : \mathbb{N}$ задано только $n < \mathsf{suc}(n)$. Обоснованность этого отношения — почти в точности обычный принцип индукции $\mathbb{N}$.

Пример 10.3.6. Пусть $A : \mathsf{Set}$, а $B : A \to \mathsf{Set}$ — семейство множеств. Напомним из §5.3, что W-тип $\mathsf{W}_{(a:A)}B(a)$ индуктивно генерируется единственным конструктором

- $\mathsf{sup} : \prod_{(a:A)}(B(a) \to \mathsf{W}_{(x:A)}B(x)) \to \mathsf{W}_{(x:A)}B(x)$.

Определим отношение $<$ на $\mathsf{W}_{(x:A)}B(x)$ рекурсией по второму аргументу:

- для любого $a : A$ и $f : B(a) \to \mathsf{W}_{(x:A)}B(x)$, определим $w < \mathsf{sup}(a, f)$, означающее существование $b : B(a)$ такого, что $w = f(b)$.

Теперь докажем, что каждый $w : \mathsf{W}_{(x:A)}B(x)$ доступен для этого отношения, используя обычный принцип индукции для $\mathsf{W}_{(x:A)}B(x)$. Это означает, что мы предполагаем задание $a : A$ и $f : B(a) \to \mathsf{W}_{(x:A)}B(x)$, а также поднятия $f' : \prod_{(b:B(a))} \mathsf{acc}(f(b))$. Но тогда, по определению $<$, имеем $\mathsf{acc}(w)$ для всех $w < \mathsf{sup}(a, f)$; следовательно, $\sup(a, f)$ доступен.

Обоснованность позволяет определять функции с помощью рекурсии и доказывать утверждения по индукции, такие как, например, приведенные далее. Напомним из §3.5, что $\mathcal{P}(B)$ обозначает *степенное множество* $\mathcal{P}(B) :\equiv (B \to \mathsf{Prop})$.

Лемма 10.3.7. *Пусть B — множество и имеется функция*

$$g : \mathcal{P}(B) \to B\,.$$

Тогда, если $<$ есть отношение обоснованности на A, то существует функция $f : A \to B$ такая, что для всех $a : A$ имеет место

$$f(a) = g\Big(\,\{f(a') \mid a' < a\}\,\Big)\,.$$

(Мы используем обозначения для образов подмножеств из §10.1.2)

Доказательство. Сначала определим, для каждого $a : A$ и $s : \mathsf{acc}(a)$, элемент $\bar{f}(a, s) : B$. По индукции достаточно предположить, что s является функцией, присваивающей каждому $a' < a$ свидетельство $s(a') : \mathsf{acc}(a')$, и что, кроме того, для каждого такого a' имеется элемент $\bar{f}(a', s(a')) : B$. В этом случае определяем

$$\bar{f}(a, s) :\equiv g\Big(\,\{\bar{f}(a', s(a')) \mid a' < a\}\,\Big)\,.$$

Теперь, поскольку $<$ вполне обоснован, имеется функция $w : \prod_{(a:A)} \mathsf{acc}(a)$. Таким образом, можно определить $f(a) :\equiv \bar{f}(a, w(a))$. ☐

В классической логике обоснованность имеет более известную формулировку. Далее, мы говорим, что подмножество $B : \mathcal{P}(A)$ **непусто**, если оно не равно пустому подмножеству $(\lambda(x).\,\bot) : \mathcal{P}(X)$. Читателю предлагается убедиться, что в предположении исключения третьего это эквивалентно просто заселению, т.е. условию $\exists(x : A).\, x \in B$.

Лемма 10.3.8. *В предположении исключения третьего, $<$ является вполне обоснованным тогда и только тогда, когда каждое непустое подмножество $B : \mathcal{P}(A)$ просто имеет минимальный элемент.*

Доказательство. Сначала, пусть $<$ является вполне обоснованным, и предположим, что $B \subseteq A$ — подмножество без минимального элемента, то есть для любого $a : A$ с $a \in B$, просто существует $b : A$ с $b < a$ и $b \in B$.

Можно утверждать, что для любого $a : A$ и $s : \mathsf{acc}(a)$, имеем $a \notin B$, поскольку, предполагая по индукции, что s является функцией, назначающей каждому $a' < a$ доказательство $s(a') : \mathsf{acc}(a')$, и, кроме того, что для каждого такого a' имеется $a' \notin B$, получаем, что если $a \in B$, то по предположению, просто существовало бы $b < a$ с $b \in B$, что противоречит этому предположению. Таким образом, $a \notin B$; это завершает индукцию. Поскольку $<$ весьма обосновано, имеет место $a \notin B$ для всех $a : A$, т.е. B пусто.

Теперь предположим, что каждое непустое подмножество просто имеет минимальный элемент. Пусть $B = \{a : A \mid \neg\mathsf{acc}(a)\}$. Тогда, если B непусто, у него просто имеется минимальный элемент. Таким образом, просто существует $a : A$ с $a \in B$такой, что для всех $b < a$, имеем $\mathsf{acc}(b)$. Но тогда, по определению (и индукции по усечению), a просто доступен, а значит, доступен, что противоречит $a \in B$. Таким образом, B пусто, поэтому $<$ является вполне обоснованным. ☐

Определение 10.3.9. Отношение обоснованности $<$ на множестве A является **экстенсиональным**, если для любых $a, b : A$, верно

$$(\forall (c : A). (c < a) \Leftrightarrow (c < b)) \to (a = b).$$

Заметим, поскольку A — множество, экстенсиональность — простое высказывание. Это понятие «экстенсиональности» не связано с экстенсиональным характером функции, а также не связано с экстенсиональным характером тождественных типов. Скорее, это «локальный» аналог аксиомы экстенсиональности в классической теории множеств.

Теорема 10.3.10. *Тип экстенсиональных отношений обоснованности является множеством.*

Доказательство. По аксиоме унивалентности, достаточно показать, что если $(A, <)$ экстенсионально и обосновано, и $f : (A, <) \cong (A, <)$, то $f = \mathsf{id}_A$. Индукцией по $<$ докажем, что $f(a) = a$ для всех $a : A$. Индуктивная гипотеза заключается в том, что для всех $a' < a$, имеем $f(a') = a'$.

Теперь, поскольку A экстенсионально, чтобы заключить, что $f(a) = a$, достаточно показать

$$\forall (c : A). (c < f(a)) \Leftrightarrow (c < a).$$

Снова, поскольку f — автоморфизм, имеем $(c < a) \Leftrightarrow (f(c) < f(a))$. Но $c < a$ влечет $f(c) = c$, по предположению индукции, поэтому $(c < a) \to (c < f(a))$. С другой стороны, если $c < f(a)$, то $f^{-1}(c) < a$, и поэтому $c = f(f^{-1}(c)) = f^{-1}(c)$, снова по индуктивному предположению; таким образом, $c < a$. Следовательно, имеем $(c < a) \Leftrightarrow (c < f(a))$ для любого $c : A$, поэтому $f(a) = a$. □

Определение 10.3.11. Если $(A, <)$ и $(B, <)$ экстенсиональны и вполне обоснованы, **имитация** — это функция $f : A \to B$ такая, что

(i) если $a < a'$, то $f(a) < f(a')$, и

(ii) для всех $a : A$ и $b : B$, если $b < f(a)$, то просто существует $a' < a$ с $f(a') = b$.

Лемма 10.3.12. *Любая имитация инъективна.*

Доказательство. Докажем двойной индукцией обоснованности, что для любых $a, b : A$, если $f(a) = f(b)$, то $a = b$. Индуктивная гипотеза для $a : A$ гласит, что для любого $a' < a$, и любого $b : B$, если $f(a') = f(b)$, то $a = b$. Внутренняя индуктивная гипотеза для $b : A$ утверждает, что для любого $b' < b$, если $f(a) = f(b')$, то $a = b'$.

Предположим, что $f(a) = f(b)$; надо показать, что $a = b$. По экстенсиональности, достаточно показать, что для любого $c : A$ выполняется $(c < a) \Leftrightarrow (c < b)$. Если $c < a$, то $f(c) < f(a)$, по определению 10.3.11 (i). Следовательно, $f(c) < f(b)$, поэтому, согласно определению 10.3.11(ii), просто существует $c' : A$ с $c' < b$ и $f(c) = f(c')$. По предположению индукции для a, имеем $c = c'$, следовательно, $c < b$. Двойственная аргументация симметрична. □

В частности, это означает, что в определении 10.3.11 (ii) слово «просто» может быть опущено без изменения смысла.

Следствие 10.3.13. *Если $f : A \to B$ — имитация, то для всех $a : A$ и $b : B$, если $b < f(a)$, просто существует $a' < a$ с $f(a') = b$.*

Доказательство. Поскольку f инъективна, $\sum_{(a:A)}(f(a) = b)$ — простое высказывание. □

Скажем, что подмножество $C : \mathcal{P}(B)$ является **начальным сегментом**, если $c \in C$ и $b < c$ подразумевают $b \in C$. Образ имитации должен быть начальным сегментом, в то время как включение любого начального сегмента — это имитация. Таким образом, по унивалентности, каждая имитация $A \to B$ *эквивалентна* включению некоторого начального сегмента B.

Теорема 10.3.14. *Для множества A, пусть $P(A)$ будет типом экстенсиональных отношений обоснованности на A. Если $<_A : P(A)$, $<_B : P(B)$ and $f : A \to B$, то пусть $H_{<_A <_B}(f)$ будет простым высказыванием, что f является имитацией. Тогда (P, H) — стандартная нотация структуры над $\mathcal{S}et$ в смысле* §9.8.

Доказательство. Мы предоставляем читателю возможность убедиться, что тождественности — это имитации и композиции имитаций — имитации. Таким образом, получаем понятие структуры. Для доказательства стандартности, мы должны показать, что если $<$ и $\prec$ являются экстенсиональными отношениями обоснованности на A, а id_A — имитация в обоих направлениях, то $<$ и $\prec$ эквивалентны. Поскольку экстенсиональность и обоснованность — простые высказывания, для этого достаточно иметь $\forall (a, b : A).\, (a < b) \Leftrightarrow (a \prec b)$, что, в свою очередь, следует из определения 10.3.11(i) для id_A. □

Следствие 10.3.15. *Существует категория, объекты которой представляют собой множества, снабженные экстенсиональными отношениями обоснованности, а морфизмы являются имитациями.*

По сути, такой категорией является частично упорядоченное множество.

Лемма 10.3.16. *Для экстенсиональных отношений обоснованности $(A, <)$ и $(B, <)$ существует не более одной имитации $f : A \to B$.*

Доказательство. Пусть $f, g : A \to B$ — имитации. Поскольку имитация — это простое свойство, достаточно показать, что $\forall (a : A).\, (f(a) = g(a))$. Индукцией по $<$ можно предположить, что $f(a') = g(a')$ для всех $a' < a$. И, в силу экстенсиональности B, чтобы иметь $f(a) = g(a)$, достаточно, чтобы выполнялось $\forall (b : B).\, (b < f(a)) \Leftrightarrow (b < g(a))$.

Но, поскольку f является имитацией, если $b < f(a)$, то имеем $a' < a$ с $f(a') = b$. По предположению индукции, $g(a') = b$, следовательно, $b < g(a)$. Двойственная аргументация симметрична. □

Таким образом, если A и B оснащены экстенсиональными отношениями обоснованности, можно писать $A \leqslant B$ для обозначения существования имитации $f : A \to B$. Следствие 10.3.15 выражает, что если $A \leqslant B$ и $B \leqslant A$, то $A = B$.

Определение 10.3.17. **Ординалом** называется множество A с экстенсиональным отношением обоснованности, которое *транзитивно*, т.е. удовлетворяет $\forall (a, b, c : A).\, (a < b) \to (b < c) \to (a < c)$.

Пример 10.3.18. Конечно, обычный строгий порядок на $\mathbb{N}$ транзитивен. Понятно, что он также экстенсионален; таким образом, это ординал. Как обычно, мы обозначаем этот ординал как ω.

Пусть Ord обозначает тип ординалов. Согласно предыдущим результатам, Ord является множеством и имеет естественный частичный порядок. Покажем, что Ord также допускает отношение обоснованности.

Если A — ординал и $a : A$, пусть $A_{/a} :\equiv \{ b : A \mid b < a \}$ обозначает начальный сегмент. Заметим, что если $A_{/a} = A_{/b}$ как ординалы, то этот изоморфизм должен учитывать их включения в A (поскольку имитации образуют частично упорядоченное множество) и, следовательно, они равны как подмножества A. Следовательно, поскольку A является экстенсиональным, $a = b$. Таким образом, функция $a \mapsto A_{/a}$ является инъекцией $A \to \mathsf{Ord}$.

Определение 10.3.19. Для ординалов A и B, имитация $f : A \to B$ называется **ограниченной**, если существует $b : B$ такое, что $A = B_{/b}$.

Из приведенных выше замечаний следует, что такой b единственный, если он существует, так что ограниченность — это простое свойство.

Будем писать $A < B$, если существует ограниченная имитация от A к B. Поскольку имитации уникальны, $A < B$, к тому же, является простым высказыванием.

Теорема 10.3.20. $(\mathsf{Ord}, <)$ *является ординалом.*

Точнее, эта теорема гласит, что тип $\mathsf{Ord}_{\mathcal{U}_i}$ ординалов в некотором универсуме является ординалом в следующем универсуме, более высокого уровня, т.е. $(\mathsf{Ord}_{\mathcal{U}_i}, <) : \mathsf{Ord}_{\mathcal{U}_{i+1}}$.

Доказательство. Пусть A — ординал; сначала покажем, что $A_{/a}$ доступен (в Ord) для всех $a : A$. Используя индукцию обоснованности по A, предположим, что $A_{/b}$ доступен для всех $b < a$. По определению доступности надо показать, что B доступен в Ord для всех $B < A_{/a}$. Однако, если $B < A_{/a}$, то существует некоторый $b < a$ такой, что $B = (A_{/a})_{/b} = A_{/b}$, что доступно по индуктивной гипотезе. Таким образом, $A_{/a}$ доступен для всех $a : A$.

Теперь, чтобы показать, что A доступен в Ord, по определению мы должны показать, что B доступен для всех $B < A$. Но, как и выше, $B < A$ означает, что $B = A_{/a}$ для некоторого $a : A$, что доступно, как только что было доказано. Таким образом, Ord вполне обоснован.

Для экстенсиональности, предположим, что A и B — ординалы такие, что $\prod_{(C:Ord)}(C < A) \Leftrightarrow (C < B)$. Тогда, для каждого $a : A$, поскольку $A_{/a} < A$, имеем $A_{/a} < B$, следовательно, существует $b : B$ с $A_{/a} = B_{/b}$. Определим $f : A \to B$ так, чтобы каждое a соответствовало b; несложно проверить, что f - изоморфизм. Таким образом, $A \cong B$, следовательно, $A = B$, по унивалентности.

Наконец, ясно, что $<$ транзитивно. □

Рассмотрение Ord как порядкового номера часто очень удобно, но содержит свои подводные камни. Например, в следующей лемме мы обращаем внимание на то, как используются универсумы.

Лемма 10.3.21. *Пусть $\mathcal{U}$ — универсум. Для любого A : $\mathsf{Ord}_{\mathcal{U}}$, существует B : $\mathsf{Ord}_{\mathcal{U}}$ такой, что $A < B$.*

Доказательство. Пусть $B = A + \mathbf{1}$, с элементом $\star : \mathbf{1}$, большим каждого элемента из A. Тогда B — ординал, и легко видеть, что $A \cong B_{/\star}$. □

Ординал B, введенный при доказательстве леммы 10.3.21, называется **преемником** A.

Эта лемма иллюстрирует потенциальную ловушку «типово-неоднозначного» стиля использования $\mathcal{U}$ для обозначения произвольного, неопределенного универсума. Рассмотрим следующее альтернативное доказательство этой леммы.

Другое предполагаемое доказательство леммы 10.3.21. Заметим, что $C < A$ тогда и только тогда, когда $C = A_{/a}$ для некоторого $a : A$. Это дает изоморфизм $A \cong \mathsf{Ord}_{/A}$, так что $A < \mathsf{Ord}$. Таким образом, можно выбрать $B :\equiv \mathsf{Ord}$. □

Второе доказательство было бы справедливым, если бы мы сформулировали лемму 10.3.21 в типово-неоднозначном стиле. Но полученная лемма была бы менее полезной, потому что второе доказательство ограничило бы второй «Ord» в формулировке леммы, ссылкой на более высокий уровень универсума, чем первый. Первое доказательство позволяет обоим универсумам быть одинаковыми.

Аналогичные замечания применимы и к следующей лемме, которую можно было бы доказать менее полезным способом, заметив, что $A \leqslant \mathsf{Ord}$ для любого $A : \mathsf{Ord}$.

Лемма 10.3.22. *Пусть $\mathcal{U}$ - универсум. Для любых $X : \mathcal{U}$ и $F : X \to \mathsf{Ord}_\mathcal{U}$, существует $B : \mathsf{Ord}_\mathcal{U}$ такое, что $Fx \leqslant B$, для всех $x : X$.*

Доказательство. Пусть B — фактор-множество (см. замечание 6.10.1) отношения эквивалентности $\sim$ на $\sum_{(x:X)} Fx$, определенного как

$$(x, y) \sim (x', y') :\equiv \Big((Fx)_{/y} \cong (Fx')_{/y'}\Big).$$

Определим $(x, y) < (x', y')$, если $(Fx)_{/y} < (Fx')_{/y'}$. Это явно переходит к частному и, как видно, превращает B в ординал. Более того, для каждого $x : X$ индуцированное отображение $Fx \to B$ является имитацией. ☐

10.4 Классические порядки

Теперь покажем эквивалентность введенных ординалов и более знакомого классического полного упорядочивания.

Лемма 10.4.1. *При предположении исключения третьего, каждый ординал является трихотомическим:*

$$\forall (a, b : A).\, (a < b) \vee (a = b) \vee (b < a).$$

Доказательство. Индукцией по a, можно предположить, что для любого $a' < a$ и любого $b' : A$, имеем $(a' < b') \vee (a' = b') \vee (b' < a')$. Далее, индукцией по b, можно предположить, что для любого $b' < b$, выполняется $(a < b') \vee (a = b') \vee (b' < a)$.

Согласно закона исключения третьего, либо просто существует $b' < b$ такое, что $a < b'$, либо просто существует $b' < b$ такое, что $a = b'$, либо для каждого $b' < b$ имеем $b' < a$. В первом случае, просто $a < b$, по транзитивности, следовательно, $a < b$, поскольку это простое высказывание. Аналогично, во втором случае $a < b$, по транспортированию. Итак, предположим, что $\forall (b' : A).\, (b' < b) \to (b' < a)$.

Аналогично, либо просто существует $a' < a$ такое, что $b < a'$, либо просто существует $a' < a$ такое, что $a' = b$, или для каждого $a' < a$ имеем $a' < b$. В первом и втором случаях, $b < a$, поэтому можно предположить, что $\forall (a' : A).\, (a' < a) \to (a' < b)$. Однако, теперь из двух этих предположений следует, что $a = b$. ☐

Лемма 10.4.2. *Отношение обоснованности не содержит циклов, т.е.*

$$\forall (n : \mathbb{N}).\, \forall (a : \mathbb{N}_n \to A).\, \neg\Big((a_0 < a_1) \wedge \cdots \wedge (a_{n-1} < a_n) \wedge (a_n < a_0)\Big).$$

Доказательство. Докажем индукцией по $a : A$, что не существует цикла, содержащего a. Итак, предположим по индукции, что для всех $a' < a$, не существует цикла, содержащего a'. Но в любом цикле, содержащем a, есть некоторый элемент, меньший, чем a, и содержащийся в том же цикле. ☐

В частности, отношение обоснованности должно быть **иррефлексивным**, т.е. $\neg(a < a)$, для всех a.

Теорема 10.4.3. *Предполагая закон исключения третьего, $(A, <)$ является ординалом тогда и только тогда, когда каждое непустое подмножество $B \subseteq A$ имеет наименьший элемент.*

Доказательство. Если A — ординал, то по лемме 10.3.8 каждое непустое подмножество просто имеет минимальный элемент. Но трихотомия подразумевает, что любой минимальный элемент является наименьшим элементом. Более того, наименьшие элементы уникальны, если они существуют, поэтому, *просто иметь* один — это значит *иметь* один.

Наоборот, если каждое непустое подмножество имеет наименьший элемент, то по лемме 10.3.8, A является обоснованным. Также имеется трихотомия, поскольку для любых a, b подмножество $\{a, b\} :\equiv \{x : A \mid x = a \vee x = b\}$ просто имеет наименьший элемент, которым должен быть, либо a, либо b. Это подразумевает транзитивность, так как, если $a < b$ и $b < c$, то, либо $a = c$, либо $c < a$, будет порождать цикл. Точно так же, это подразумевает экстенсиональность, поскольку если $\forall(c : A).\,(c < a) \Leftrightarrow (c < b)$, то $a < b$ влечет (пусть c будет a), что $a < a$, , что является циклом, и аналогично при $b < a$; следовательно, $a = b$. $\square$

В классической математике характеризация теоремы 10.4.3 принимается за определение полного упорядочения, при этом ординалы являются каноническим множеством представителей классов изоморфизма для такого упорядочения. В нашем контексте принцип структурной тождественности означает, что таких представителей не нужно долго искать: любое полное упорядочение ничем не хуже любого другого.

Теперь перейдем к рассмотрению следствий аксиомы выбора. Для любого множества X, пусть $\mathcal{P}_+(X)$ обозначает тип просто обитаемых подмножеств из X:

$$\mathcal{P}_+(X) :\equiv \{Y : \mathcal{P}(X) \mid \exists(x : X).\, x \in Y\}\,.$$

Предполагая исключение третьего, это эквивалентно типу *непустых* подмножеств из X, и тогда $\mathcal{P}(X) \simeq (\mathcal{P}_+(X)) + \mathbf{1}$.

Теорема 10.4.4. *При действии закона исключения третьего, следующие утверждения эквивалентны.*

(i) Для каждого множества X, просто существует функция $f : \mathcal{P}_+(X) \to X$ такая, что $f(Y) \in Y$, для всех $Y : \mathcal{P}_+(X)$.

(ii) Каждое множество просто допускает структуру ординала.

Конечно, (i) — это стандартная классическая версия аксиомы выбора; см. упражнение 10.10.

Доказательство. Одно направление простое: предположим (ii). Поскольку требуется доказать только утверждение (i), можно считать A ординалом. Но тогда можно определить $f(B)$ как наименьший элемент в B.

Теперь предположим (i). Как и ранее, поскольку (ii) — простое высказывание, то можно считать, что задана требуемая f. Расширим f до функции

$$\bar{f} : \mathcal{P}(X) \simeq (\mathcal{P}_+(X)) + \mathbf{1} \longrightarrow X + \mathbf{1}$$

очевидным образом. Теперь, для любого ординала A, можно определить $g_A : A \to X + \mathbf{1}$ с помощью рекурсии обоснования:

$$g_A(a) :\equiv \bar{f}\Big(X \setminus \{g_A(b) \mid (b < a) \wedge (g_A(b) \in X)\}\Big)$$

(рассматривая X как подмножество в $X + \mathbf{1}$ очевидным образом).

Пусть $A' :\equiv \{a : A \mid g_A(a) \in X\}$ будет прообразом $X \subseteq X + \mathbf{1}$; тогда можно утверждать, что ограничение $g'_A : A' \to X$ инъективно. Ведь если $a, a' : A$ с $a \neq a'$, то по трихотомии и без ограничения общности можно считать, что $a' < a$. Таким образом, $g_A(a') \in \{g_A(b) \mid b < a\}$, поэтому, поскольку $f(Y) \in Y$, для всех Y, то имеем $g_A(a) \neq g_A(a')$.

Более того, A' является начальным сегментом A. Поскольку $g_A(a)$ лежит в $\mathbf{1}$ тогда и только тогда, когда $\{g_A(b) \mid b < a\} = X$, и если это верно, то это верно и для любого $a' > a$. Таким образом, A' само по себе является ординалом.

Наконец, поскольку Ord — ординал, то можно взять $A :\equiv \mathsf{Ord}$. Пусть X' будет образом $g'_{\mathsf{Ord}} : \mathsf{Ord}' \to X$; тогда обратное к g'_{Ord} дает инъекцию $H : X' \to \mathsf{Ord}$. По лемме 10.3.22 существует ординал C такой, что $Hx \leqslant C$ для всех $x : X'$. Тогда по лемме 10.3.21 существует еще ординал D такой, что $C < D$, следовательно, $Hx < D$ для всех $x : X'$. Теперь имеем

$$\begin{aligned} g_{\mathsf{Ord}}(D) &= \bar{f}\Big(X \setminus \{g_{\mathsf{Ord}}(B) \mid B < D \wedge (g_{\mathsf{Ord}}(B) \in X)\}\Big) \\ &= \bar{f}\Big(X \setminus \{g_{\mathsf{Ord}}(B) \mid g_{\mathsf{Ord}}(B) \in X\}\Big) \end{aligned}$$

поскольку, если $B : \mathsf{Ord}$ и $(g_{\mathsf{Ord}}(B) \in X)$, то $B = Hx$ для некоторого $x : X'$, следовательно, $B < D$. Далее, если

$$\{g_{\mathsf{Ord}}(B) \mid g_{\mathsf{Ord}}(B) \in X\}$$

не является всем X, то $g_{\mathsf{Ord}}(D)$ будет лежать в X, но не в этом подмножестве, что было бы противоречием, поскольку D само является потенциальным значением для B. Таким образом, это множество должен полностью совпадать с X, и, следовательно, функция g'_{Ord} сюръективна так же, как и инъективна. Таким образом, можно перенести структуру ординала на Ord' в X. $\square$

Замечание 10.4.5. Если бы мы дали неправильное доказательство леммы 10.3.21 или 10.3.22, то получившееся доказательство теоремы 10.4.4 было бы неверным: не было бы способа последовательно назначать уровни универсумам. На самом деле, мы требуем пропозиционального изменения размера (что следует из LEM), чтобы гарантировать, что X' обитает в том же универсуме, что и X (с точностью до эквивалентности).

Следствие 10.4.6. *Исходя из аксиомы выбора, функция* $\mathsf{Ord} \to \mathsf{Set}$ *(которая забывает порядковую структуру) является сюръекцией.*

Обратите внимание, что Ord — это множество, а Set — 1-тип. В общем, нет причин для 1-типа допускать любую сюръективную функцию из множества. Даже аксиома выбора не означает, что *каждый* 1-тип делает это (хотя, см. упражнение 7.9), но подразумевает, что это так для 1-типов, построенных из Set, таких как типы объектов категорий структур, как в §9.8. Следующее следствие также применимо к подобным категориям.

Следствие 10.4.7. *Предполагая* AC*,* $\mathcal{S}et$ *допускает функтор слабой эквивалентности из строгой категории.*

Доказательство. Пусть $X_0 :\equiv \mathsf{Ord}$, а для $A, B : X_0$ пусть $\mathsf{hom}_X(A, B) :\equiv (A \to B)$. Тогда X — строгая категория, поскольку Ord — это множество, а указанная выше сюръекция $X_0 \to \mathsf{Set}$ продолжается до функтора слабой эквивалентности $X \to \mathcal{S}et$. $\square$

Напомним из §10.2, что имеется, кроме того, сюръекция $|-|_0 : \mathsf{Set} \to \mathsf{Card}$, и, следовательно, скомпонованная сюръекция $\mathsf{Ord} \to \mathsf{Card}$, которая связывает каждый ординал с его мощностью.

Теорема 10.4.8. *При предположении* AC, *сюръекция* $\mathsf{Ord} \to \mathsf{Card}$ *имеет сечение.*

Доказательство. Существует простое и неправильное доказательство: поскольку Ord и Card являются множествами, AC подразумевает, что любая сюръекция между ними *просто* имеет сечение. Однако, на самом деле, имеется каноническое *специфицированное* сечение: поскольку Ord является ординалом, каждое его непустое подмножество имеет однозначно специфицированный наименьший элемент. Таким образом, мы можем отобразить каждый кардинал в наименьший элемент в соответствующем слое. $\square$

В теории множеств традиционно отождествляют кардиналы с их образом в Ord: наименьшим порядковом числом, имеющем эту мощность.

Отсюда следует, что Card также канонически допускает структуру ординала: фактически, он изоморфен Ord. В частности, определим с помощью рекурсии обоснованности функцию $\aleph : \mathsf{Ord} \to \mathsf{Ord}$ такую, что $\aleph(A)$ является наименьшим порядковым номером, имеющим мощность, большую чем $\aleph(A_{/a})$, для всех $a : A$. Тогда (при условии AC) образ $\aleph$ в точности совпадает с образом Card.

10.5 Кумулятивная иерархия

Можно определить совокупную (или, кумулятивную) иерархию V всех множеств в заданном универсуме $\mathcal{U}$ как высший индуктивный тип, таким образом, что V снова будет множеством (в большем универсуме $\mathcal{U}'$), снабженным бинарным отношением «принадлежности» $x \in y$, которое удовлетворяет обычным законам теории множеств.

Определение 10.5.1. Кумулятивная иерархия V относительно универсума типов $\mathcal{U}$ — это высший индуктивный тип, порождаемый следующими конструкторами.

(i) Для каждого $A : \mathcal{U}$ и $f : A \to V$, существует элемент $\mathsf{set}(A, f) : V$.

(ii) Для всех $A, B : \mathcal{U}$, $f : A \to V$ и $g : B \to V$ таких, что
$$\big(\forall (a : A).\, \exists (b : B).\, fa =_V gb\big) \wedge \big(\forall (b : B).\, \exists (a : A).\, fa =_V gb\big) \tag{10.5.2}$$
существует путь $\mathsf{set}(A, f) =_V \mathsf{set}(B, g)$.

(iii) Конструктор 0-усечения: для всех $x, y : V$ и $p, q : x = y$, имеет место $p = q$.

На теоретико-множественном языке $\mathsf{set}(A, f)$ можно понимать как множество (в смысле классической теории множеств), которое является образом A при f, т.е. $\{f(a) \mid a \in A\}$. Однако мы будем избегать этой нотации, поскольку она будет противоречить нашей нотации для подтипов (но, см. (10.5.3) и определение 10.5.7).

Иерархия V (изначально) создается из пустого отображения $\mathsf{rec}_{\mathbf{0}}(V) : \mathbf{0} \to V$, что дает пустое множество в виде $\varnothing = \mathsf{set}(\mathbf{0}, \mathsf{rec}_{\mathbf{0}}(V))$. Затем синглетон $\{\varnothing\}$ пополняет V посредством $\mathbf{1} \to V$, определяемый как $\star \mapsto \varnothing$, и т.д. (эта процедура также может быть адаптирована для включения произвольного множества «атомов» или «элементов», путем добавления дополнительного точечного конструктора). Тип V обитает в том же универсуме, что и базовый универсум $\mathcal{U}$.

Второй конструктор V имеет форму, отличную от всех, с которыми мы имели дело раньше: он включает в себя не только пути в V (которые в §6.9, как отмечалось, были слегка подозрительными), но и их усечения. Это определенно не соответствует общей схеме, описанной в §6.13, и поэтому может быть неочевидно, каким должен быть его принцип индукции. К счастью, как и наше первое определение 0-усечения в §6.9, его можно «перевыразить» с помощью вспомогательных высших индуктивных типов. Читателю предоставляется проработка деталей (см. упражнение 10.11).

В конце концов, принцип индукции для V (выраженный на языке сопоставления с образцом) гласит, что для данного $P : V \to \mathsf{Set}$, чтобы построить $h : \prod_{(x:V)} P(x)$, достаточно сделать следующее.

(i) Для любой $f : A \to V$, создаем $h(\mathsf{set}(A, f))$, считая заданным $h(f(a))$, для всех $a : A$.

(ii) Проверяем, что если $f : A \to V$ и $g : B \to V$ удовлетворяют (10.5.2), то $h(\mathsf{set}(A, f)) =^P_q h(\mathsf{set}(B, g))$, где q — путь, возникающий из второго конструктора V и (10.5.2), при индуктивном предположении, что $h(f(a))$ и $h(g(b))$ определены для всех $a : A$ и $b : B$, и что выполняется следующее условие:

$$\begin{aligned}&\big(\forall (a : A).\, \exists (b : B).\, \exists (p : f(a) = g(b)).\, h(f(a)) =^P_p h(g(b))\big)\\ \wedge\quad &\big(\forall (b : B).\, \exists (a : A).\, \exists (p : f(a) = g(b)).\, h(f(a)) =^P_p h(g(b))\big)\end{aligned}$$

Второе условие проверяет, что определяемое отображение должно соответствовать путям, указанным в (10.5.2). Как обычно, когда мы формулируем принципы высшей индукции, используя сопоставление с образцом, результат может показаться тавтологичным, но это не так. Дело в том, что «$h(f(a))$», по сути, является формальным символом, в который мы не можем заглянуть, и через который должен быть определен $h(\mathsf{set}(A, f))$. Таким образом, во втором условии мы предполагаем равенство этих формальных символов, когда это необходимо, и проверяем, что элементы, полученные в результате построения первого условия, также равны. Конечно, если P — семейство простых высказываний, то второе условие является автоматическим.

Заметим, что по индукции, для каждого $v : V$, просто существуют $A : \mathcal{U}$ и $f : A \to V$ такие, что $v = \mathsf{set}(A, f)$. Таким образом, разумно попытаться определить **отношение принадлежности** $x \in v$ на V, задавая:

$$(x \in \mathsf{set}(A, f)) :\equiv (\exists (a : A).\, x = f(a))\,.$$

Чтобы убедиться в правильности этого определения, необходимо использовать принцип рекурсии для V. Итак, предположим, что у нас есть путь $\mathsf{set}(A, f) = \mathsf{set}(B, g)$, построенный с помощью (10.5.2). Если $x \in \mathsf{set}(A, f)$, то просто существует $a : A$ такое, что $x = f(a)$, но согласно (10.5.2) просто существует $b : B$ такое, что $f(a) = g(b)$, следовательно, $x = g(b)$ и $x \in \mathsf{set}(B, g)$. Обратное симметрично.

Отношение «быть подмножеством» $x \subseteq y$ определяется на V, как обычно:

$$(x \subseteq y) :\equiv \forall (z : V).\, z \in x \Rightarrow z \in y\,.$$

Класс может быть принят как простой предикат на V. Можно сказать, что класс $C : V \to \mathsf{Prop}$ является V**-множеством**, если просто существует $v : V$ такое, что

$$\forall (x : V).\, C(x) \Leftrightarrow x \in v\,.$$

Также можно использовать обычную нотацию для классов, которая соответствует нашей стандартной нотации для подтипов:

$$\{x \mid C(x)\} :\equiv \lambda x.\, C(x). \tag{10.5.3}$$

Класс $C : V \to \mathsf{Prop}$ будет называться **$\mathcal{U}$-малым**, если все его значения $C(x)$ лежат в $\mathcal{U}$, а именно $C : V \to \mathsf{Prop}_{\mathcal{U}}$. Поскольку V обитает в том же универсуме $\mathcal{U}'$, что и базовый универсум $\mathcal{U}$, из которого он построен, то же самое верно и для тождественных типов $v =_V w$, для любых $v, w : V$. Поэтому, для получения теории с хорошим поведением, в отсутствие пропозиционального изменения размера, будет удобно иметь $\mathcal{U}$-малое «изменение размера» отношения тождественности, которое можно определить по индукции следующим образом.

Определение 10.5.4. Определим отношение **би-имитации**

$$\sim : V \times V \longrightarrow \mathsf{Prop}_{\mathcal{U}}$$

двойной индукцией по V, где для $\mathsf{set}(A, f)$ и $\mathsf{set}(B, g)$, мы допускаем:

$$\mathsf{set}(A, f) \sim \mathsf{set}(B, g) :\equiv \big(\forall (a : A).\, \exists (b : B).\, f(a) \sim g(b)\big) \wedge \big(\forall (b : B).\, \exists (a : A).\, f(a) \sim g(b)\big).$$

Чтобы проверить правильность определения, просто нужно убедиться, что оно учитывает пути $\mathsf{set}(A, f) = \mathsf{set}(B, g)$, построенные с помощью (10.5.2), но это очевидно, и что $\mathsf{Prop}_{\mathcal{U}}$ — множество, которое таковым и является. Заметим, что $u \sim v$, по построению, находится в $\mathsf{Prop}_{\mathcal{U}}$.

Лемма 10.5.5. *Для любых $u, v : V$, имеем $(u =_V v) = (u \sim v)$.*

Доказательство. Несложная индукция показывает, что $\sim$ рефлексивна, поэтому, по транспортированию, имеем $(u =_V v) \to (u \sim v)$. Таким образом, остается показать, что $(u \sim v) \to (u =_V v)$. Индукцией по u и v можно предположить, что они есть $\mathsf{set}(A, f)$ и $\mathsf{set}(B, g)$, соответственно (можно игнорировать конструкторы путей V, поскольку $(u \sim v) \to (u =_V v)$ — простое высказывание). Тогда, по определению, $\mathsf{set}(A, f) \sim \mathsf{set}(B, g)$ влечет $(\forall (a : A).\, \exists (b : B).\, f(a) \sim g(b))$, и наоборот. Но индуктивная гипотеза тогда влечет $(\forall (a : A).\, \exists (b : B).\, f(a) = g(b))$, и наоборот. Итак, с помощью конструктора пути для V имеем $\mathsf{set}(A, f) =_V \mathsf{set}(B, g)$. □

Мы могли бы предположить, что можно было бы игнорировать конструктор V и *доказать*, что V является 0-усеченным, применяя теорему 7.2.2 к би-имитации. Однако, в доказательстве леммы 10.5.5 был использован тот факт, что V является 0-усеченным, для заключения, что $(u \sim v) \to (u =_V v)$ является простым высказыванием, так что, по индукции, достаточно предположить, что u и v есть $\mathsf{set}(A, f)$ и $\mathsf{set}(B, g)$.

Теперь можно использовать отношение идентичности с изменяемостью размера для получения следующего полезного принципа.

Лемма 10.5.6. *Для каждого $u : V$ существует $A_u : \mathcal{U}$ и мономорфизм $m_u : A_u \rightarrowtail V$ такие, что $u = \mathsf{set}(A_u, m_u)$.*

Доказательство. Возьмем произвольное представление $u = \mathsf{set}(A, f)$ и фактор $f : A \to V$, как сюръекцию, с последующей инъекцией:

$$f = m_u \circ e_u : A \twoheadrightarrow A_u \rightarrowtail V\,.$$

Ясно, что $u = \mathsf{set}(A_u, m_u)$, если только A_u все еще находится в $\mathcal{U}$, что верно, если ядро of $e_u : A \twoheadrightarrow A_u$ находится в $\mathcal{U}$. Но ядро $e_u : A \twoheadrightarrow A_u$ — это обратный образ вдоль $f : A \to V$

тождественности на V, который, как мы только что показали, $\mathcal{U}$-малый, с точностью до эквивалентности. Теперь эта конструкция пары (A_u, m_u) с $m_u : A_u \rightarrowtail V$ и $u = \mathsf{set}(A_u, m_u)$ из $u : V$ единственно с точностью до эквивалентности над V, а значит, с точностью до тождественности, по унивалентности. Таким образом, по принципу однозначного выбора (3.9.2) существует отображение $c : V \to \sum_{(A:\mathcal{U})}(A \to V)$ такое, что $c(u) = (A_u, m_u)$, с $m_u : A_u \rightarrowtail V$ и $u = \mathsf{set}(c(u))$, что и требовалось. □

Определение 10.5.7. Для $u : V$, только что построенное моническое представление $m_u : A_u \rightarrowtail V$ такое, что $u = \mathsf{set}(A_u, m_u)$, может называться **типом членов** u и обозначается $m_u : [u] \rightarrowtail V$, или даже $[u] \rightarrowtail V$. Можно думать об $[u]$ как о «подклассе V, состоящем из членов u».

Теорема 10.5.8. *Для $(V, \in)$ выполняется следующее:*

(i) экстенсиональность:
$$\forall (x, y : V).\, x \subseteq y \wedge y \subseteq x \Leftrightarrow x = y\,;$$

(ii) пустое множество: *для всех $x : V$, имеем $\neg(x \in \emptyset)$*;

(iii) спаривание: *для всех $u, v : V$, класс $\{u, v\} :\equiv \{x \mid x = u \vee x = v\}$ является V-множеством*;

(iv) бесконечность: *существует $v : V$ с $\emptyset \in v$ и $x \in v$ подразумевает $x \cup \{x\} \in v$*;

(v) объединение: *для всех $v : V$, класс $\cup v :\equiv \{x \mid \exists (u : V).\, x \in u \in v\}$ is a V-set*;

(vi) множество функций: *for all $u, v : V$, the class $v^u :\equiv \{x \mid x : u \to v\}$ является V-множеством*;[1]

(vii) $\in$-индукция: *если $C : V \to \mathsf{Prop}$ — класс такой, что $C(a)$ имеет место всякий раз, когда $C(x)$ для всех $x \in a$, то $C(v)$ для всех $v : V$*;

(viii) замена: *для любых $r : V \to V$ и $x : V$, класс*
$$\{y \mid \exists (z : V).\, z \in x \wedge y = r(z)\}$$
является V-множеством;

(ix) отделимость: *для любых $a : V$ и $\mathcal{U}$-малого $C : V \to \mathsf{Prop}_{\mathcal{U}}$, класс*
$$\{x \mid x \in a \wedge C(x)\}$$
является V-множеством.

Набросок доказательства.

(i) Экстенсиональность: если $\mathsf{set}(A, f) \subseteq \mathsf{set}(B, g)$, то $f(a) \in \mathsf{set}(B, g)$, для каждого $a : A$, поэтому, для каждого $a : A$, просто существует $b : B$ такое, что $f(a) = g(b)$. Предположение $\mathsf{set}(B, g) \subseteq \mathsf{set}(A, f)$ дает другую половину (10.5.2), поэтому $\mathsf{set}(A, f) = \mathsf{set}(B, g)$.

(ii) Пустое множество: предположим, что $x \in \emptyset = \mathsf{set}(\mathbf{0}, \mathsf{rec}_{\mathbf{0}}(V))$. Тогда $\exists (a : \mathbf{0}).\, x = \mathsf{rec}_{\mathbf{0}}(V, a)$, что абсурдно.

[1] Здесь $x : u \to v$ означает, что x — подходящее множество упорядоченных пар в соответствии с обычным способом кодирования функций в теории множеств.

(iii) Спаривание: для u и v, пусть $w = \mathsf{set}(\mathbf{2}, \mathsf{rec}_{\mathbf{2}}(V, u, v))$.

(iv) Бесконечность: возьмем $w = \mathsf{set}(\mathbb{N}, I)$, где $I : \mathbb{N} \to V$ задается рекурсией $I(0) :\equiv \varnothing$ и $I(n+1) :\equiv I(n) \cup \{I(n)\}$.

(v) Объединение: возьмем любой $v : V$ и любое представление $f : A \to V$ с $v = \mathsf{set}(A, f)$. Тогда, пусть $\tilde{A} :\equiv \sum_{(a:A)}[fa]$, где $m_{fa} : [fa] \rightarrowtail V$ — тип членов из определения 10.5.7. $\tilde{A}$ явно является $\mathcal{U}$-малым и имеется $\cup v :\equiv \mathsf{set}(\tilde{A}, \lambda x.\, m_{f(\mathsf{pr}_1(x))}(\mathsf{pr}_2(x)))$.

(vi) Множество функций: для заданных $u, v : V$, возьмем типы членов $[u] \rightarrowtail V$ и $[v] \rightarrowtail V$, а также функциональный тип $[u] \to [v]$. Надо определить отображение

$$r : ([u] \to [v]) \longrightarrow V$$

с «$r(f) = \{(x, f(x)) \mid x : [u]\}$», но для того, чтобы это имело смысл, необходимо сначала определить упорядоченную пару (x, y), взять отображение $r' : x \mapsto (x, f(x))$, а затем можно положить $r(f) :\equiv \mathsf{set}([u], r')$. Упорядоченная пара может быть определена в терминах неупорядоченного спаривания, как обычно.

(vii) $\in$-индукция: пусть $C : V \to \mathsf{Prop}$ — класс такой, что $C(a)$ выполняется всякий раз, когда $C(x)$, для всех $x \in a$, и возьмем любое $v = \mathsf{set}(B, g)$. Чтобы показать, что $C(v)$ верно по индукции, предположим, что $C(g(b))$, для всех $b : B$. Для любого $x \in v$ просто существует некоторый $b : B$ с $x = g(b)$, и поэтому выполняется $C(x)$. Таким образом, верно и $C(v)$.

(viii) Замена: пусть C обозначает рассматриваемый класс. Утверждение «C является V-множеством» является простым высказыванием, поэтому можно действовать по индукции следующим образом. Предположим, что x есть $\mathsf{set}(A, f)$, можно утверждать, что $w :\equiv \mathsf{set}(A, r \circ f)$ — множество, которое мы ищем. Если $C(y)$, то просто существуют $z : V$ и $a : A$, так что, $z = f(a)$ и $y = r(z)$, следовательно, $y \in w$. Наоборот, если $y \in w$,то просто существует $a : A$ такое, что $y = r(f(a))$, поэтому, если возьмем $z :\equiv f(a)$, то увидим, что $C(y)$ выполняется.

(ix) Отделимость: скажем, что класс $C : V \to \mathsf{Prop}$ является **отделимым** , если, для любого $a : V$, класс

$$a \cap C :\equiv \{x \mid x \in a \wedge C(x)\}$$

является V-множеством. Надо показать, что любой $\mathcal{U}$-малый $C : V \to \mathsf{Prop}_{\mathcal{U}}$ является отделимым. В самом деле, для $a = \mathsf{set}(A, f)$, пусть $A' = \sum_{(x:A)} C(fx)$, и возьмем $f' = f \circ i$, где $i : A' \to A$ — очевидное включение. Затем можно взять $a' = \mathsf{set}(A', f')$ и получим $x \in a \wedge C(x) \Leftrightarrow x \in a'$, как заявлено. Для этого нужно было предположить, что C попадает в $\mathcal{U}$, чтобы $A' = \sum_{(x:A)} C(fx)$ находилось в $\mathcal{U}$.

□

Также удобно иметь строго синтаксический критерий отделимости, чтобы можно было понять из выражения для класса, что он создает V-множество. Одним из таких знакомых условий является «Δ_0», означающее, что выражение строится из тождественности $x =_V y$ и принадлежности $x \in y$, используя только просто-пропозициональные связки $\neg$, $\wedge$, $\vee$, $\Rightarrow$ и кванторы $\forall$, $\exists$ над определенными множествами, то есть имеет форму $\exists(x \in a)$ и $\forall(y \in b)$ (они называются **ограниченными** кванторами).

Следствие 10.5.9. *Если класс* $C : V \to \mathsf{Prop}$ *есть* Δ_0 *в указанном выше смысле, то он является отделимым.*

Доказательство. Напомним, что имеется $\mathcal{U}$-малое изменение размера $x \sim y$ тождественности $x = y$. Поскольку $x \in y$ определяется в терминах $x = y$, также имеется $\mathcal{U}$-малое изменение размера принадлежности

$$x \mathrel{\widetilde{\in}} \mathsf{set}(A, f) :\equiv \exists (a : A).\, x \sim f(a)\,.$$

Теперь, пусть Φ будет Δ_0-выражением для C, так что, как классы, $\Phi = C$ (строго говоря, мы должны различать выражения по их значениям, но мы размывем различие). Пусть $\widetilde{\Phi}$ будет результатом замены всех вхождений $=$ и $\in$ их эквивалентами с изменением размера $\sim$ и $\widetilde{\in}$. Очевидно, что $\widetilde{\Phi}$ также выражает C, в том смысле, что, для всех $x : V$, $\widetilde{\Phi}(x) \Leftrightarrow C(x)$, и, следовательно, $\widetilde{\Phi} = C$ в силу унивалентности. Теперь, достаточно показать, что $\widetilde{\Phi}$ есть $\mathcal{U}$-малый, так как тогда он будет отделимым по предыдущей теореме.

Покажем, что $\widetilde{\Phi}$ есть $\mathcal{U}$-малый, индукцией по построению выражения. Базовыми случаями являются $x \sim y$ и $\widetilde{\in}$, размер которых уже изменен в $\mathcal{U}$. Также ясно, что $\mathcal{U}$ замкнут относительно чисто пропозициональных операций (и (-1)-усечения), поэтому остается только проверить ограниченные кванторы $\exists (x \in a)$ и $\forall (y \in b)$. По определению,

$$\exists (x \in a).\, P(x) :\equiv \left\| \sum_{x:V} (x \mathrel{\widetilde{\in}} a \wedge P(x)) \right\|,$$
$$\forall (y \in b).\, P(x) :\equiv \prod_{x:V} (x \mathrel{\widetilde{\in}} a \to P(x))\,.$$

Рассмотрим $\left\|\sum_{(x:V)} (x \mathrel{\widetilde{\in}} a \wedge P(x))\right\|$. Хотя основная часть, $(x \mathrel{\widetilde{\in}} a \wedge P(x))$, является $\mathcal{U}$-малой, поскольку $P(x)$ таково по индуктивной гипотезе, количественная оценка V не обязательно должна оставаться внутри $\mathcal{U}$. Однако, в данном случае можно заменить это количественной оценкой по типу $[a] \rightarrowtail V$ членов a, и легко показать, что

$$\sum_{x:V} (x \mathrel{\widetilde{\in}} a \wedge P(x)) = \sum_{x:[a]} P(x)\,.$$

Правая часть остается в $\mathcal{U}$, поскольку и $[a]$, и $P(x)$, находятся в $\mathcal{U}$. Случай $\prod_{(x:V)} (x \mathrel{\widetilde{\in}} a \to P(x))$ аналогичен, используя $\prod_{(x:V)} (x \mathrel{\widetilde{\in}} a \to P(x)) = \prod_{(x:[a])} P(x)$. □

Мы показали, что в теории типов с универсумом $\mathcal{U}$, кумулятивная иерархия V является моделью «конструктивной теории множеств» со многими стандартными аксиомами. Однако, насколько нам известно, в нем отсутствуют аксиома *сильной совокупности* и аксиома *совокупности подмножеств* , которые включены в конструктивную теорию множеств Цермело-Френкеля [Acz78]. В обычной интерпретации этой теории множеств в теории типов эти две аксиомы являются следствиями сетоидного определения равенства; в то время как в других построенных моделях теории множеств сильная совокупность может выполняться по другим причинам. Мы не знаем, выполняется ли какая-либо из этих аксиом в нашей модели $(V, \in)$, но это кажется маловероятным. Поскольку V представляет собой высший индуктивный тип *внутри* системы, а не является *внешней* конструкцией, неудивительно, что она в некоторых отношениях отличается от предшествующих интерпретаций.

Наконец, рассмотрим результат добавления аксиомы выбора для множеств в нашу теорию типов в виде AC из §10.1.5. Отсюда следует, что LEM тогда также выполняется по теореме 10.1.14, и поэтому Set является топосом с классификатором подобъектов $\mathbf{2}$, по теореме 10.1.12. В этом случае $\mathsf{Prop} = 2 : \mathcal{U}$, поэтому *все классы являются отделимыми*. Таким образом, показано:

Лемма 10.5.10. *В теории типов с* AC*, закон* ***(полного) отделения*** *выполняется для V: для любого класса $C : V \to \mathsf{Prop}$ и $a : V$, класс $a \cap C$ является V-множеством.*

Теорема 10.5.11. *В теории типов с* AC *и универсумом $\mathcal{U}$, кумулятивная иерархия V является моделью теории множеств Цермело-Френкеля с выбором (ZFC).*

Доказательство. У нас имеются все аксиомы, перечисленные в теореме 10.5.8, плюс полное отделение, поэтому просто нужно показать, что существуют степенные множества $\mathcal{P}(a) : V$, для всех $a : V$. Но поскольку имеется LEM, это просто функциональные типы $\mathcal{P}(a) = (a \to \mathbf{2})$. Таким образом, V является моделью теории множеств Цермело-Френкеля ZF (мы оставляем проверку теоретико-множественной аксиомы выбора из AC в качестве легкого упражнения). □

Примечания

Основные свойства, которые можно ожидать от категории множеств, восходят к начальному периоду теории элементарных топосов. *Элементарная теория категории множеств*, упоминаемая в §10.1.5, была введена Ловером (Lawvere) в [Law05], как теоретико-категорная аксиоматизация теории множеств. Понятие ΠW-предтопоса, рассматриваемого как предикативная версия элементарного топоса, было введено в [MP02]; см. также [Pal09].

Изучение категории множеств в §10.1 примерно следует содержимому из [RS13]. Тот факт, что эпиморфизмы сюръективны (лемма 10.1.4), хорошо известен в классической математике, но не так тривиален, как может показаться, для *предикативного* доказательства. Доказательство в [MRR88] использует операцию образования степенного множества (которая является непредикативной), хотя ее также можно рассматривать как предикативное доказательство более слабого утверждения, что отображение в универсуме $\mathcal{U}_i$ является сюръективным, если оно является эпиморфизмом в следующем универсуме $\mathcal{U}_{i+1}$. Предикативное доказательство для сетоидов было дано Виландером (Wilander) [Wil10]. Наше доказательство аналогично доказательству Виландера, но избегает сетоидов за счет использования амальгам и унивалентности.

Импликация в теореме 10.1.14 от AC к LEM является адаптацией к гомотопической теории типов теоремы из теории топосов, принадлежащей Диаконеску (Diaconescu) [Dia75]; это было позиционировано как проблема Бишопом (Bishop) [Bis67, Problem 2].

Об интуиционистской теории ординальных чисел см. [Tay96, Tay99], а также [JM95]. Определения обоснованности в теории типов с помощью принципа индукции, включая индуктивный предикат доступности, изучались в [Hue80, Pau86, Nor88], хотя идея восходит к доказательству непротиворечивости арифметики Генценом (Gentzen) [Gen36].

Идея алгебраической теории множеств, которая лежит в основе нашего развития кумулятивной иерархии в §10.5, изложена в [JM95], но происходит от более ранней работы [Acz78].

Упражнения

Упражнение 10.1. Следуя образцу $\mathcal{S}et$, хотелось бы создать категорию $\mathcal{T}ype$ всех типов и отображений между ними (в заданном универсуме $\mathcal{U}$). Однако, чтобы это была категория в смысле §9.1, сначала надо объявить $\hom(X, Y) :\equiv \|X \to Y\|_0$, причем композиция определяется индукцией по усечению из обычной композиции $(Y \to Z) \to (X \to Y) \to (X \to Z)$. Это было определено как *гомотопическая предкатегория типов* в примере 9.1.18. Однако это все еще не категория, а только предкатегория (ее тип объектов $\mathcal{U}$ даже не является 0-типом). Она становится категорией после пополнения Резка (см. пример 9.9.7), а его тип объектов может быть идентифицирован как $\|\mathcal{U}\|_1$ из упражнения 9.9. Покажите, что полученная категория $\mathcal{T}ype$, в отличие от $\mathcal{S}et$, не является предтопосом.

Упражнение 10.2. Покажите, что если каждая сюръекция имеет сечение в категории $\mathcal{S}et$, то аксиома выбора верна.

Упражнение 10.3. Покажите, что с помощью LEM, категория $\mathcal{S}et$ является точной, в том смысле, что выполняется следующее утверждение: для любых $f, g : A \to B$, если $f \neq g$, то существует функция $a : 1 \to A$ такая, что $f(a) \neq g(a)$. Покажите, что категория срезов $\mathcal{S}et/\mathbf{2}$, состоящая из функций $A \to \mathbf{2}$ и коммутативных треугольников, не имеет этого свойства (подсказка: терминальный объект в $\mathcal{S}et/\mathbf{2}$ — это тождественная функция $\mathbf{2} \to \mathbf{2}$, поэтому в этой категории имеются объекты X, которые не имеют элементов $1 \to X$).

Упражнение 10.4. Докажите, что если $(A, <_A)$ и $(B, <_B)$ являются обоснованными, экстенсиональными или ординалами, то таково же и $A + B$, с отношением $<$, определенным как

$$\begin{aligned}
(a < a') &:\equiv (a <_A a') && \text{для } a, a' : A \\
(b < b') &:\equiv (b <_B b') && \text{для } b, b' : B \\
(a < b) &:\equiv \mathbf{1} && \text{для } (a : A), (b : B) \\
(b < a) &:\equiv \mathbf{0} && \text{для } (a : A), (b : B).
\end{aligned}$$

Упражнение 10.5. multiplication!of ordinal numbersДокажите, что если $(A, <_A)$ и $(B, <_B)$ являются обоснованными, экстенсиональными или ординалами, то таково же и $A \times B$, с отношением $<$, определенным как

$$((a, b) < (a', b')) :\equiv (a <_A a') \vee ((a = a') \wedge (b <_B b')).$$

Упражнение 10.6. Определите обычные алгебраические операции над ординалами и докажите, что они удовлетворяют обычным свойствам.

Упражнение 10.7. Отметим, что $\mathbf{2}$ является ординалом под очевидным отношением $<$ таким, что $0_{\mathbf{2}} < 1_{\mathbf{2}}$.

(i) Определите отношение $<$ на Prop, которое превращает его в ординал.

(ii) Покавжите, что $\mathbf{2} =_{\mathsf{Ord}} \mathsf{Prop}$ тогда и только тогда, когда имеет место LEM.

Упражнение 10.8. Напомним, что $\mathbb{N}$ обозначено через ω, когда рассматривается как ординал; так что, также имеем ординал $\omega + 1$. С другой стороны, определим

$$\mathbb{N}_\infty :\equiv \left\{ a : \mathbb{N} \to \mathbf{2} \mid \forall (n : \mathbb{N}).\, (a_n \leqslant a_{\mathsf{succ}(n)}) \right\} ,$$

где $\leqslant$ обозначает очевидный частичный порядок на $\mathbf{2}$, с $0_\mathbf{2} < 1_\mathbf{2}$.

(i) Определите отношение $<$ на $\mathbb{N}_\infty$, которое превращает его в ординал.

(ii) Покавжите, что $\omega + 1 =_{\mathsf{Ord}} \mathbb{N}_\infty$ тогда и только тогда, когда действует ограниченный принцип всезнания (11.5.8).

Упражнение 10.9. Покажите, что если $(A, <)$ хорошо обосновано и экстенсионально, и $A : \mathcal{U}$, то существует имитация $A \to V$, где $(V, \in)$ — кумулятивная иерархия из §10.5, построенная из универсума $\mathcal{U}$.

Упражнение 10.10. Покажите, что теорема 10.4.4(i) эквивалентна аксиоме выбора (3.8.1).

Упражнение 10.11. Для типов A и B определим **биполное отношение** $R : A \to B \to \mathsf{Prop}$ такое, что

$$\Big(\forall (a : A).\, \exists (b : B).\, R(a, b)\Big) \wedge \Big(\forall (b : B).\, \exists (a : A).\, R(a, b)\Big) .$$

Для таких A, B, R, пусть $A \sqcup^R B$ — высший индуктивный тип, порожденный:

- $i : A \to A \sqcup^R B$,
- $j : B \to A \sqcup^R B$,
- для любых $a : A$ и $b : B$ таких, что $R(a, b)$, путем $i(a) = j(b)$.

Покажите, что кумулятивная иерархия V может быть определена с помощью следующего более простого списка конструкторов, и что результирующий принцип индукции — тот, который указан в §10.5.

- Для каждого $A : \mathcal{U}$ и $f : A \to V$, существует элемент $\mathsf{set}(A, f) : V$.
- Для любых $A, B : \mathcal{U}$ и биполного отношения $R : A \to B \to \mathsf{Prop}$, любого отображения $h : A \sqcup^R B \to V$, существует путь $\mathsf{set}(A, h \circ i) = \mathsf{set}(B, h \circ j)$.
- Конструктор 0-усечения.

Упражнение 10.12. В конструктивной теории множеств Цермело-Френкеля **аксиома сильной совокупности** имеет вид:

$$\Big((\forall (x \in v).\, \exists (y).\, R(x, y)\Big) \Rightarrow$$
$$\exists (w).\, \Big[(\forall (x \in v).\, \exists (y \in w).\, R(x, y)) \wedge (\forall (y \in w).\, \exists (x \in v).\, R(x, y)) \Big] .$$

Сохраняется ли это в кумулятивной иерархии V? (Ответ на этот вопрос не известен.)

Упражнение 10.13. Убедитесь, что если предположить AC, то кумулятивная иерархия V удовлетворяет обычной теоретико-множественной аксиоме выбора, которую можно выразить в форме:

$$\forall (x : V).\, \Big((\forall (y \in x).\, \exists (z : V).\, z \in y) \Rightarrow \exists (c \in (\cup x)^x).\, \forall (y \in x).\, c(y) \in y\Big) .$$

Упражнение 10.14. Предполагая пропозициональное изменение размера, покажите, что существует простой предикат $\mathsf{isPlump} : \mathsf{Ord} \to \mathsf{Prop}$ такой, что для любого $A : \mathsf{Ord}$, имеем

$$\mathsf{isPlump}(A) = \\ \Big(\forall (B < A).\, \mathsf{isPlump}(B)\Big) \wedge \Big(\forall (C, B : \mathsf{Ord}).\, C \leqslant B < A \wedge \mathsf{isPlump}(C) \Rightarrow C < A\Big)\,.$$

Заметим, что $\mathsf{isPlump}$ не может быть определен простой хорошо обоснованной индукцией по Ord; необходимо использовать другое отношение обоснованности. Говорят, что ординал A является **гладким** [Tay96, Tay99], если верно $\mathsf{isPlump}(A)$.

Упражнение 10.15. Покажите, что LEM эквивалентен утверждению «все порядковые числа гладкие».

Упражнение 10.16. Определим **гладкого преемника** ординала A как

$$t(A) :\equiv \{B : \mathsf{Ord} \mid (B \leqslant A) \wedge \mathsf{isPlump}(B)\}\ .$$

(i) По определению, $t(A)$ принадлежит последующему высшему универсуму. Покажите, что в предположении пропозиционального изменения размера он равен ординалу в том же универсуме, что и A.

(ii) Снова предполагая пропозициональное изменение размера, покажите, что если A — гладкий (упражнение 10.14), то $t(A)$ таков же.

Упражнение 10.17. **ZF-алгебра**[JM95] относительно универсума $\mathcal{U}_i$ является частично упорядоченным множеством (см. пример 9.1.14) $V : \mathcal{U}_{i+1}$, все супремумы которого индексированы типами из $\mathcal{U}_i$, снабженный функцией «преемник» $s : V \to V$ (не обязательно в каждом случае удовлетворяя $\leqslant$).

(i) Покажите, что кумулятивная иерархия $(V_{\mathcal{U}_i}, \subseteq, s)$ является инициальной ZF-алгеброй, где $s(x)$ — синглетон $\{x\}$.

(ii) Покажите, что $(\mathsf{Ord}_{\mathcal{U}_i}, \leqslant, s)$ является инициальной ZF-алгеброй со свойством $x \leqslant s(x)$, для всех x, где $s(A) = A + \mathbf{1}$ является преемником из леммы 10.3.21.

(iii) Предполагая пропозициональное изменение размера, покажите, что $(\{A : \mathsf{Ord}_{\mathcal{U}_i} \mid \mathsf{isPlump}(A)\}, \leqslant, t)$ является инициальной ZF-алгеброй со свойством $(x \leqslant y) \Rightarrow (t(x) \leqslant t(y))$, для всех x, y, где t — гладкий преемник из упражнения 10.16.

Упражнение 10.18. Для категории A, морфизм $f : \hom_A(a, b)$ называется **расщепляемым мономорфизмом**, если существует морфизм $g : \hom_A(b, a)$ такой, что $g \circ f = 1_a$ (такой g называется **ретракцией** f). Докажите, что следующие свойства логически эквивалентны.

(i) LEM.

(ii) Для любых множеств A и B, если A населено, то для любого мономорфизма $f : A \to B$ в $\mathcal{S}et$, f также является расщепляемым мономорфизмом в $\mathcal{S}et$.

Глава 11

Действительные числа

Любое обоснование математики, достойное своего названия, должно, в конечном итоге, обратиться к построению действительных чисел, как они понимаются в математическом анализе, а именно как полное архимедово упорядоченное поле. Существует два понятия полноты. Одно из них, по Коши, требует, чтобы вещественные числа были замкнуты в пределах последовательностей Коши, в то время как более сильное понятие полноты, по Дедекинду, требует замыкания в рамках дедекиндова сечения. Это приводит к двум способам построения вещественных чисел, которые мы изучаем, соответственно, в §11.2 и §11.3. В теоремах 11.2.14 и 11.3.50 характеризуются две конструкции с точки зрения универсальных свойств: вещественные числа Дедекинда являются финальным архимедово упорядоченным полем, а вещественные числа Коши — первоначальным полным архимедово упорядоченным полем Коши.

В традиционной конструктивной математике действительные числа всегда требуют определенных компромиссов. Например, действительные числа Дедекинда лучше работают со степенными множествами или какой-либо другой формой непредикативности, в то время как действительные числа Коши хорошо работают при наличии счетного выбора. Несмотря на это, мы предлагаем новую конструкцию вещественных чисел Коши высшего индуктивно-индуктивного типа, который представляется третьей возможностью, не требующей, ни степенных множеств, ни счетного выбора.

В §11.4 мы сравниваем две конструкции действительных чисел. Действительные числа Коши включены в действительные числа Дедекинда. Они совпадают, если выполняется исключение третьего или счетный выбор, но в целом включение корректно.

В §11.5 мы рассматриваем три понятия компактности отрезка [0,1]. Сначала мы покажем, что [0,1] метрически компактно в том смысле, что оно полно и везде ограничено, и что равномерно непрерывные отображения на метрически компактных пространствах ведут себя ожидаемо. Напротив, свойство Больцано-Вейерштрасса, согласно которому каждая последовательность имеет сходящуюся подпоследовательность, подразумевает ограниченный принцип всезнания, который является примером исключения третьего. Наконец, мы обсуждаем компактность Гейне-Бореля. Наивная формулировка свойства конечного подпокрытия не работает, в отличие от понятия индуктивного покрытия, имеющего отношение к доказательству. Этот раздел в основном представляет собой стандартный конструктивный анализ.

Исследование действительных чисел и анализа в теории гомотопического типа можно легко совместить с классической математикой. Допуская закон исключения третьего и аксиому выбора, получаем стандартный классический анализ: действительные числа Дедекинда и Коши совпадают, фундаментальные вопросы о непредикативной природе действительных чисел Дедекинда исчезают, а интервал становится настолько компактным, насколько это возможно.

Завершается глава построением сюрреалистических чисел Конвея как высшего индуктивно-индуктивного типа, в §11.6; эта конструкция в унивалентной теории типов более естественна, чем в классической теории множеств.

В дополнение к основной теории глав 2 и 3 мы используем, как отмечалось выше, «высшие индуктивно-индуктивные типы» для действительных чисел Коши и сюрреалистических чисел: они объединяют идеи главы 6 с понятием индуктивно-индуктивного типа, упомянутым в §5.7. Мы также будем часто использовать традиционные логические обозначения, описанные в §3.7, и тот факт (доказанный в §10.1),что наши «множества» ведут себя так, как мы и ожидали.

Заметим, что пространство расслоения универсального покрытия круга, которое в §8.1.5 играло роль, аналогичную «действительным числам» в классической алгебраической топологии, *не* является типом вещественных чисел, которые мы ищем. Этот тип является стягиваемым и, таким образом, эквивалентен типу синглетона, так что его нельзя оснастить нетривиальной алгебраической структурой.

11.1 Поле рациональных чисел

Сначала сконструируем рациональные числа $\mathbb{Q}$, так как действительные числа затем можно рассматривать как завершение $\mathbb{Q}$. Известно, что $\mathbb{Q}$ можно заменить любым аппроксимирующим полем, т.е. подкольцом $\mathbb{Q}$, в котором существуют произвольно точные аппроксимирующие обратные. Примером может служить кольцо двоично рациональных чисел, которые имеют вид $n/2^k$. Если бы мы реализовывали конструктивную математику на компьютере, аппроксимирующее поле было бы более подходящим, но мы оставляем такую тонкость для тех, кому важны цифры π.

Мы построили целые числа $\mathbb{Z}$ в §6.10 как частное от $\mathbb{N}\times\mathbb{N}$, и заметили, что оно порождается идемпотентом. В §6.11 мы выяснили, что $\mathbb{Z}$ — свободная группа на $\mathbf{1}$; аналогичным образом можно показать, что это свободное коммутативное кольцо на $\mathbf{0}$. Поле рациональных чисел $\mathbb{Q}$ строится по тем же принципам, а именно как частное

$$\mathbb{Q} :\equiv (\mathbb{Z} \times \mathbb{N})/{\approx}\,,$$

где

$$(u,a) \approx (v,b) :\equiv (u(b+1) = v(a+1))\,.$$

Другими словами, пара (u,a) представляет рациональное число $u/(1+a)$. Деления на ноль быть не может, потому что мы предусмотрительно добавили единицу к знаменателю a. Здесь тоже есть канонический выбор представителей, а именно дроби в низших термах. Таким образом, можно применить лемму 6.10.8, чтобы получить множество $\mathbb{Q}$, которое снова имеет разрешимое равенство.

Мы не утруждаем себя записью арифметических операций над $\mathbb{Q}$, поскольку уверены, что наши читатели умеют производить вычисления с дробями, даже в том случае, когда к знаменателю добавляется единица. Сделаем лишь вывод о том, что существует совершенно беспроблемное построение упорядоченного поля рациональных чисел $\mathbb{Q}$, с разрешимым равенством и разрешимым порядком. Его также можно охарактеризовать как начальное упорядоченное поле.

Наконец, обозначим через $\mathbb{Q}_+ :\equiv \{q : \mathbb{Q} \mid q > 0\}$ тип положительных рациональных чисел.

11.2 Действительные числа Дедекинда

Напомним основную идею конструкции Дедекинда. Будем использовать двухсторонние сечения Дедекинда, а не одностороннюю версию, поскольку симметрия делает конструкции более элегантными и работает конструктивно, а также классически. *Сечение Дедекинда* состоит из пары (L, U) подмножеств $L, U \subseteq \mathbb{Q}$, называемых, соответственно, *нижним* и *верхним сечением*, которые:

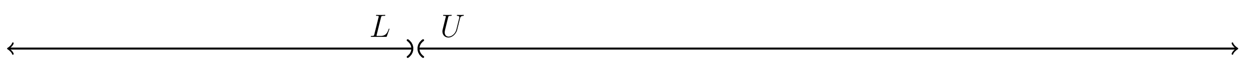
(i) *обитаемы*: существуют $q \in L$ и $r \in U$,

(ii) *округлены*: $q \in L \Leftrightarrow \exists (r \in \mathbb{Q}).\, q < r \wedge r \in L$ и $r \in U \Leftrightarrow \exists (q \in \mathbb{Q}).\, q \in U \wedge q < r$,

(iii) *не пересекаются*: $\neg(q \in L \wedge q \in U)$ и

(iv) *локализованы*: $q < r \Rightarrow q \in L \vee r \in U$.

Рассмотрение условия округленности слева-направо говорит о том, что сечения являются *открытыми*, а справа-налево — о том, что они являются *нижними*, соответственно, *верхними*, множествами. Условие локализации утверждает, что между L и U нет большого интервала. Поскольку сечения открыты, они никогда не включают «точку между ними», даже когда это целесообразно. Типичное сечение Дедекинда выглядит следующим образом:



Можно было бы наивно перевести неформальное определение в теорию типов, положив, что сечение — это пара отображений $L, U : \mathbb{Q} \to \mathsf{Prop}$. Но, как было отмечено в §3.5, Prop является двусмысленной нотацией для $\mathsf{Prop}_{\mathcal{U}_i}$ где $\mathcal{U}_i$ — это универсум. Как только мы используем конкретный $\mathcal{U}_i$ для определения сечений, тип действительных чисел будет находиться в следующем универсуме $\mathcal{U}_{i+1}$, свойство действительных чисел — двумя уровнями выше, в $\mathcal{U}_{i+2}$, свойство подмножеств действительных чисел — в $\mathcal{U}_{i+3}$, и т.д. В принципе, должна существовать возможность отслеживать уровни универсумов, особенно с помощью компьютерных программ, но здесь мы просто завязнем в деталях, чего предпочитаем избегать. Поэтому сделаем упрощающее предположение, что для всех наших целей достаточно одного типа высказываний Ω.

На самом деле, конструкция действительных чисел Дедекинда достаточно устойчива к логическим манипуляциям. Есть несколько способов использования одного типа Ω.

(i) Можно идентифицировать Ω с неоднозначным Prop и отслеживать все универсумы, которые фигурируют в определениях и конструкциях.

(ii) Можно принять аксиому пропозиционального изменения размера, как в §3.5, которая, по существу, сворачивает все $\mathsf{Prop}_{\mathcal{U}_i}$ до самого нижнего уровня , который обозначен как Ω.

(iii) Классический математик, не интересующийся тонкостями теоретико-типовых универсумов или вычислений, может просто принять закон исключенного третьего (3.4.1) для простых высказываний, так что $\Omega \equiv \mathbf{2}$. Это не только устранит проблемы с уровнями Prop, но и превратит все, что мы делаем, в стандартную классическую конструкцию действительных чисел.

(iv) На другом конце этого спектра можно было бы задать минимальные требования, которые заставят конструкции работать. Условие того, что простой предикат является сечением Дедекинда, выражается с использованием только конъюнкций, дизъюнкций и кванторов существования над $\mathbb{Q}$, которое является счетным множеством. Таким образом, можно принять Ω за исходный *σ-каркас*, т.е. решетку со счетными соединениями, в которой двоичные числа встречаются с распределением по счетным соединениям (исходный σ-каркас не может быть двухточечной решеткой $\mathbf{2}$, потому что $\mathbf{2}$ не замкнута при счетных соединениях, если мы не предполагаем исключение третьего). Это привело бы к построению Ω как высшего индуктивно-индуктивного типа, но одного опыта такого рода из §11.3 достаточно.

Во всех приведенных выше случаях Ω — множество. Без дальнейших церемоний мы переводим это неформальное определение в теорию типов. На протяжении всей текущей главы будут использоваться логические обозначения из определения 3.7.1.

Определение 11.2.1. Сечение Дедекинда — это пара (L, U) простых предикатов $L : \mathbb{Q} \to \Omega$ и $U : \mathbb{Q} \to \Omega$, которые являются:

(i) *обитаемыми* (i.e., *связанными*): $\exists(q : \mathbb{Q}).\, L(q)$ и $\exists(r : \mathbb{Q}).\, U(r)$,

(ii) *округленными*: для всех $q, r : \mathbb{Q}$,

$$L(q) \Leftrightarrow \exists(r : \mathbb{Q}).\, (q < r) \wedge L(r) \qquad \text{и}$$
$$U(r) \Leftrightarrow \exists(q : \mathbb{Q}).\, (q < r) \wedge U(q)\,,$$

(iii) *не пересекающимися*: $\neg(L(q) \wedge U(q))$ для всех $q : \mathbb{Q}$,

(iv) *локализованными*: $(q < r) \Rightarrow L(q) \vee U(r)$ для всех $q, r : \mathbb{Q}$.

Обозначим через $\mathsf{isCut}(L, U)$ конъюнкцию приведенных условий. Типом **действительных чисел Дедекинда** является

$$\mathbb{R}_d :\equiv \{(L, U) : (\mathbb{Q} \to \Omega) \times (\mathbb{Q} \to \Omega) \mid \mathsf{isCut}(L, U)\}\,.$$

Очевидно, что $\mathsf{isCut}(L, U)$ — простое высказывание, а поскольку $\mathbb{Q} \to \Omega$ является множеством, действительные числа Дедекинда тоже образуют множество. См. в упражнениях 11.2–11.4 варианты сечений Дедекинда, которые приводят к расширенным действительным числам, нижним и верхним действительным числам и интервальной области.

Существует вложение $\mathbb{Q} \to \mathbb{R}_d$, которое ставит в соответствие каждому рациональному числу $q : \mathbb{Q}$ сечение (L_q, U_q), где

$$L_q(r) :\equiv (r < q) \qquad \text{and} \qquad U_q(r) :\equiv (q < r)\,.$$

Будем просто записывать q для сечения (L_q, U_q), ассоциированного с рациональным числом.

11.2.1 Алгебраическая структура действительных чисел Дедекинда

Построение алгебраической и теоретико-порядковой структуры вещественных чисел Дедекинда происходит, как принято в интуиционистской логике. Вместо того, чтобы останавливаться на деталях, укажем на различия между классической и интуиционистской установками. Записывая

L_x и U_x для нижнего и верхнего сечения действительного числа $x : \mathbb{R}_d$, определим сложение как

$$\begin{aligned} L_{x+y}(q) &:\equiv \exists(r, s : \mathbb{Q}).\, L_x(r) \wedge L_y(s) \wedge q = r + s\,, \\ U_{x+y}(q) &:\equiv \exists(r, s : \mathbb{Q}).\, U_x(r) \wedge U_y(s) \wedge q = r + s\,, \end{aligned}$$

а аддитивную инверсию, как

$$\begin{aligned} L_{-x}(q) &:\equiv \exists(r : \mathbb{Q}).\, U_x(r) \wedge q = -r\,, \\ U_{-x}(q) &:\equiv \exists(r : \mathbb{Q}).\, L_x(r) \wedge q = -r\,. \end{aligned}$$

С этими операциями $(\mathbb{R}_d, 0, +, -)$ является абелевой группой. Определение умножения несколько более громоздко:

$$\begin{aligned} L_{x\cdot y}(q) &:\equiv \exists(a, b, c, d : \mathbb{Q}).\, L_x(a) \wedge U_x(b) \wedge L_y(c) \wedge U_y(d) \wedge q < \min(a\cdot c, a\cdot d, b\cdot c, b\cdot d)\,, \\ U_{x\cdot y}(q) &:\equiv \exists(a, b, c, d : \mathbb{Q}).\, L_x(a) \wedge U_x(b) \wedge L_y(c) \wedge U_y(d) \wedge \max(a\cdot c, a\cdot d, b\cdot c, b\cdot d) < q\,. \end{aligned}$$

Эти формулы связаны с умножением интервалов в интервальной арифметике, где умножение интервалов $[a, b]$ и $[c, d]$ с рациональными концами есть интервал

$$[a, b] \cdot [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]\,.$$

Например, формул для нижнего сечения можно воспринимать так, что $q < x\cdot y$, когда существуют интервалы $[a, b]$ и $[c, d]$, содержащие x и y, соответственно, такие, что q находится слева от $[a, b] \cdot [c, d]$. Полезно думать об интервале $[a, b]$ так, что $L_x(a)$ и $U_x(b)$ являются аппроксимацией x, см. упражнение 11.4.

Теперь, имеем коммутативное кольцо с единицей $(\mathbb{R}_d, 0, 1, +, -, \cdot)$. Чтобы рассматривать мультипликативные обратные, надо сначала ввести порядок. Определим $\leqslant$ и $<$ как

$$\begin{aligned} (x \leqslant y) &:\equiv \forall(q : \mathbb{Q}).\, L_x(q) \Rightarrow L_y(q)\,, \\ (x < y) &:\equiv \forall(q : \mathbb{Q}).\, U_x(q) \wedge L_y(q)\,. \end{aligned}$$

Лемма 11.2.2. *Для всех $x : \mathbb{R}_d$ и $q : \mathbb{Q}$, $L_x(q) \Leftrightarrow (q < x)$ и $U_x(q) \Leftrightarrow (x < q)$.*

Доказательство. Если $L_x(q)$, то, в силу округления, просто существует $r > q$ такое, что $L_x(r)$, а поскольку $U_q(r)$, то отсюда следует, что $q < x$. И наоборот, если $q < x$, то существует $r : \mathbb{Q}$ такое, что $U_q(r)$ и $L_x(r)$, следовательно, $L_x(q)$ поскольку L_x является нижним множеством. Другая половина доказательства симметрична. □

Отношение $\leqslant$ является частичным порядком, а $<$ — транзитивно и нерефлексивно. Линейность

$$(x < y) \vee (y \leq x)$$

Линейность имеет силу, если предполагается исключение третьего, но без этого получим слабую линейность

$$(x < y) \Rightarrow (x < z) \vee (z < y)\,. \tag{11.2.3}$$

На первый взгляд может быть неясно, какое отношение (11.2.3) имеет к линейному порядку. Но если взять $x \equiv u - \epsilon$ и $y \equiv u + \epsilon$, для $\epsilon > 0$, то получим

$$(u - \epsilon < z) \vee (z < u + \epsilon)\,.$$

Это является линейностью «с точностью до небольшой числовой ошибки», т.е., поскольку неразумно ожидать, что действительно можно вычислять с бесконечной точностью, мы не должны удивляться тому, что можем допустить использование $<$ только с той конечной точностью, с которой проводили вычисление.

Чтобы убедиться в справедливости (11.2.3), предположим, что $x < y$. Тогда просто существует $q : \mathbb{Q}$ такое, что $U_x(q)$ и $L_y(q)$. В силу округления, просто существуют $r, s : \mathbb{Q}$ такие, что $r < q < s$, $U_x(r)$ и $L_y(s)$. Тогда, по расположению, верно $L_z(r)$ или $U_z(s)$. В первом случае получаем $x < z$, а во втором — $z < y$.

С точки зрения классической теории, мультипликативные обратные существуют для всех чисел, отличных от нуля. Однако, без исключения третьего требуется более сильное условие. Скажем, что $x, y : \mathbb{R}_d$ **отделены** друг от друга, что обозначается как $x \# y$, если $(x < y) \vee (y < x)$:

$$(x \# y) :\equiv (x < y) \vee (y < x)\,.$$

Если $x \# y$, то $\neg(x = y)$. Обратное верно, если предполагается исключение третьего, но это конструктивно недоказуемо. В самом деле, если $\neg(x = y)$ подразумевает $x \# y$, то отсюда следует «немного» исключения третьего; см. упражнение 11.10.

Теорема 11.2.4. *Действительное число обратимо тогда и только тогда, когда оно отделено от* 0.

Замечание 11.2.5. Действительное число обратимо тогда и только тогда, когда оно просто обратимо. В самом деле, то же самое верно в любом кольце, так как кольцо есть множество, и мультипликативные обратные единственны, если они существуют. См. обсуждение после следствия 3.9.2.

Доказательство. Предположим, $x \cdot y = 1$. Тогда просто существуют $a, b, c, d : \mathbb{Q}$ такие, что $a < x < b$, $c < y < d$ и $0 < \min(ac, ad, bc, bd)$. Из $0 < ac$ и $0 < bc$ следует, что a, b и c, либо все положительны, либо все отрицательны. Следовательно, либо $0 < a < x$, либо $x < b < 0$, так что, $x \# 0$.

И наоборот, если $x \# 0$, то

$$L_{x^{-1}}(q) :\equiv \exists (r : \mathbb{Q}).\, U_x(r) \wedge ((0 < r \wedge qr < 1) \vee (r < 0 \wedge 1 < qr))$$
$$U_{x^{-1}}(q) :\equiv \exists (r : \mathbb{Q}).\, L_x(r) \wedge ((0 < r \wedge qr > 1) \vee (r < 0 \wedge 1 > qr))$$

определяет желаемое обратное. Действительно, $L_{x^{-1}}$ и $U_{x^{-1}}$ обитаемы, поскольку $x \# 0$. □

Принцип Архимеда можно сформулировать несколькими способами. Мы находим его наиболее показательным в форме, в которой говорится, что $\mathbb{Q}$ плотно в $\mathbb{R}_d$.

Теорема 11.2.6 (Принцип Архимеда для $\mathbb{R}_d$)**.** Для всех $x, y : \mathbb{R}_d$, если $x < y$, то просто существует $q : \mathbb{Q}$ такое, что $x < q < y$.

Доказательство. По определению $<$. □

Прежде чем заняться полнотой вещественных чисел Дедекинда, точно сформулируем, какой алгебраической структурой они обладают. В следующем определении мы стремимся не к минимальной аксиоматизации, а к полезному количеству структуры и свойств.

Определение 11.2.7. Упорядоченное поле представляет собой множество F вместе с константами 0 и 1, операциями $+$, $-$, $\cdot$, $\min$, $\max$, и простыми отношениями $\leqslant$, $<$, $\#$ такими, что:

(i) $(F, 0, 1, +, -, \cdot)$ — коммутативное кольцо с единицей;

(ii) $x : F$ обратима тогда и только тогда, когда $x \# 0$;

(iii) $(F, \leqslant, \min, \max)$ представляет собой решетку;

(iv) строгий порядок $<$ является транзитивным, нерефлексивным и слабо линейным ($x < y \Rightarrow x < z \vee z < y$);

(v) отделенность $\#$ нерефлексивна, симметрична и ко-транзитивна ($x \# y \Rightarrow x \# z \vee y \# z$);

(vi) для всех $x, y, z : F$:

$$\begin{aligned} x \leqslant y &\Leftrightarrow \neg(y < x), & x < y \leqslant z &\Rightarrow x < z, \\ x \# y &\Leftrightarrow (x < y) \vee (y < x), & x \leqslant y < z &\Rightarrow x < z, \\ x \leqslant y &\Leftrightarrow x + z \leq y + z, & x \leqslant y \wedge 0 \leqslant z &\Rightarrow xz \leqslant yz, \\ x < y &\Leftrightarrow x + z < y + z, & 0 < z &\Rightarrow (x < y \Leftrightarrow xz < yz), \\ 0 < x + y &\Rightarrow 0 < x \vee 0 < y, & & 0 < 1. \end{aligned}$$

Каждое такое поле имеет каноническое вложение $\mathbb{Q} \to F$. Упорядоченное поле является **архимедовым**, когда для всех $x, y : F$, если $x < y$, то просто существует $q : \mathbb{Q}$ такое, что $x < q < y$.

Теорема 11.2.8. *Действительные числа Дедекинда образуют упорядоченное архимедово поле.*

Доказательство. Мы опускаем доказательство в надежде, что продемонстрированное до сих пор, делает теорему правдоподобной. □

11.2.2 Действительные числа Дедекинда являются полными в смысле Коши

Напомним, что $x : \mathbb{N} \to \mathbb{Q}$ является *последовательностью Коши*, если она удовлетворяет условию

$$\prod_{(\epsilon:\mathbb{Q}_+)} \sum_{(n:\mathbb{N})} \prod_{(m,k \geqslant n)} |x_m - x_k| < \epsilon \,. \tag{11.2.9}$$

Отметим, что внутренняя экзистенциальная функция не усечена, потому что, на самом деле, требуется вычислить скорость сходимости — аппроксимация без оценки ошибки несет мало полезной информации. По теореме 2.15.7, (11.2.9) дает функцию $M : \mathbb{Q}_+ \to \mathbb{N}$, называемую *модулем сходимости*, такую, что $m, k \geqslant M(\epsilon)$ подразумевает $|x_m - x_k| < \epsilon$. Отсюда получаем $\left|x_{M(\delta/2)} - x_{M(\epsilon/2)}\right| < \delta + \epsilon$ для всех $\delta, \epsilon : \mathbb{Q}_+$. На самом деле, отображение $(\epsilon \mapsto x_{M(\epsilon/2)}) : \mathbb{Q}_+ \to \mathbb{Q}$ несет ту же информацию о пределе, что и исходное условие Коши (11.2.9). Мы будем работать с этими аппроксимирующими функциями, а не с последовательностями Коши.

Определение 11.2.10. Аппроксимация Коши — это отображение $x : \mathbb{Q}_+ \to \mathbb{R}_d$, которое удовлетворяет условию

$$\forall(\delta, \epsilon : \mathbb{Q}_+).\, |x_\delta - x_\epsilon| < \delta + \epsilon \,. \tag{10.2.11}$$

Пределом аппроксимации Коши $x : \mathbb{Q}_+ \to \mathbb{R}_d$ является число $\ell : \mathbb{R}_d$ такое, что

$$\forall(\epsilon, \theta : \mathbb{Q}_+).\, |x_\epsilon - \ell| < \epsilon + \theta \,.$$

Теорема 11.2.12. *Любая* аппроксимации Коши *в* $\mathbb{R}_d$ *имеет предел.*

Доказательство. Заметим, что мы показываем существование, а не просто существование, предела. Для аппроксимацию Коши $x : \mathbb{Q}_+ \to \mathbb{R}_d$ определим

$$L_y(q) :\equiv \exists(\epsilon, \theta : \mathbb{Q}_+).\, L_{x_\epsilon}(q + \epsilon + \theta)\,,$$
$$U_y(q) :\equiv \exists(\epsilon, \theta : \mathbb{Q}_+).\, U_{x_\epsilon}(q - \epsilon - \theta).$$

Ясно, что L_y и U_y обитаемы, являются округленными и не пересекаются. Чтобы установить локализованность, рассмотрим любые $q, r : \mathbb{Q}$ такие, что $q < r$. Существует $\epsilon : \mathbb{Q}_+$ такое, что $5\epsilon < r - q$. Поскольку $q + 2\epsilon < r - 2\epsilon$, то просто $L_{x_\epsilon}(q + 2\epsilon)$ или просто $U_{x_\epsilon}(r - 2\epsilon)$. В первом случае имеем $L_y(q)$, а во втором — $U_y(r)$.

Чтобы показать, что y является пределом x, рассмотрим любые $\epsilon, \theta : \mathbb{Q}_+$. Поскольку $\mathbb{Q}$ плотно в $\mathbb{R}_d$, просто существуют $q, r : \mathbb{Q}$ такие, что

$$x_\epsilon - \epsilon - \theta/2 < q < x_\epsilon - \epsilon - \theta/4 < x_\epsilon < x_\epsilon + \epsilon + \theta/4 < r < x_\epsilon + \epsilon + \theta/2\,,$$

и, таким образом, $q < y < r$. Теперь, либо $y < x_\epsilon + \theta/2$, либо $x_\epsilon - \theta/2 < y$. В первом случае имеем

$$x_\epsilon - \epsilon - \theta/2 < q < y < x_\epsilon + \theta/2\,,$$

а во втором,

$$x_\epsilon - \theta/2 < y < r < x_\epsilon + \epsilon + \theta/2\,.$$

В любом случае, отсюда следует, что $|y - x_\epsilon| < \epsilon + \theta$. □

Для завершенности, приведем и классическую формулировку.

Следствие 11.2.13. *Пусть $x : \mathbb{N} \to \mathbb{R}_d$ удовлетворяет условию Коши* (11.2.9). *Тогда существует $y : \mathbb{R}_d$ такой, что*

$$\prod_{(\epsilon:\mathbb{Q}_+)} \sum_{(n:\mathbb{N})} \prod_{(m,k \geqslant n)} |x_m - y| < \epsilon\,.$$

Доказательство. По теореме 2.15.7, существует $M : \mathbb{Q}_+ \to \mathbb{N}$ такое, что $\bar{x}(\epsilon) :\equiv x_{M(\epsilon/2)}$ есть аппроксимация Коши. Пусть y — ее предел, существующий по теореме 11.2.12. Для любого $\epsilon : \mathbb{Q}_+$, пусть $n :\equiv M(\epsilon/4)$, и заметим, что для любого $m \geqslant n$,

$$|x_m - y| \leqslant |x_m - x_n| + |x_n - y| = |x_m - x_n| + |\bar{x}(\epsilon/2) - y| < \epsilon/4 + \epsilon/2 + \epsilon/4 = \epsilon\,.$$

□

11.2.3 Действительные числа Дедекинда являются полными в смысле Дедекинда

Мы получили $\mathbb{R}_d$ как тип сечений Дедекинда на $\mathbb{Q}$. Но вместо этого можно было бы начать с любого архимедова упорядоченного поля F и построить дедекиндовы сечения на F. Они снова образовали бы архимедово упорядоченное поле $\bar{F}$, **дедекиндово пополнение** F, с F, содержащимся в качестве подполя. Что произойдет, если применить эту конструкцию к $\mathbb{R}_d$, получим ли мы еще больше действительных чисел? Ответ отрицательный. На самом деле мы докажем более сильный результат: $\mathbb{R}_d$ является заключительным.

Скажем, что упорядоченное поле F **допустимо для** Ω когда строгий порядок $<$ на F является отображением $<: F \to F \to \Omega$.

Теорема 11.2.14. *Любое архимедово упорядоченное поле, допустимое для Ω, является подполем в $\mathbb{R}_d$.*

Доказательство. Пусть F — архимедово упорядоченное поле. Для каждого $x : F$ определим $L_x, U_x : \mathbb{Q} \to \Omega$ с помощью

$$L_x(q) :\equiv (q < x) \qquad \text{и} \qquad U_x(q) :\equiv (x < q)$$

(использовано предположение о допустимости F для Ω). Тогда (L_x, U_x) — сечение Дедекинда. Действительно, эти сечения обитаемы и округлены, поскольку F архимедово, а $<$ — транзитивно, дизъюнктивно, поскольку $<$ нерефлексивно и локализовано, поскольку $<$ — слабый линейный порядок. Пусть $e : F \to \mathbb{R}_d$ будет отображением $e(x) :\equiv (L_x, U_x)$.

Мы утверждаем, что e — вложение полей, сохраняющее и отражающее порядок. Прежде всего, обратите внимание, что $e(q) = q$, для рационального числа q. Далее, имеем эквивалентности, для всех $x, y : F$,

$$x < y \Leftrightarrow (\exists(q : \mathbb{Q}).\, x < q < y) \Leftrightarrow (\exists(q : \mathbb{Q}).\, U_x(q) \wedge L_y(q)) \Leftrightarrow e(x) < e(y)\,,$$

так что, e действительно сохраняет и отражает порядок. $e(x + y) = e(x) + e(y)$ выполняется, потому что, для всех $q : \mathbb{Q}$,

$$q < x + y \Leftrightarrow \exists(r, s : \mathbb{Q}).\, r < x \wedge s < y \wedge q = r + s\,.$$

Импликация справа налево очевидна. Для другого направления, если $q < x + y$, то просто существует $r : \mathbb{Q}$ такое, что $q - y < r < x$, и, взяв $s :\equiv q - r$, получим желаемые r и s. Мы оставляем доказательство сохранения умножения на e в качестве упражнения. □

Чтобы установить, что сечения Дедекинда на $\mathbb{R}_d$ не привносят ничего нового, понадобится еще одна лемма.

Лемма 11.2.15. *Если F допустимо для Ω, то допустимо и его дедекиндово пополнение.*

Доказательство. Пусть $\bar{F}$ — дедекиндово пополнение F. Строгий порядок на $\bar{F}$ определяется как

$$((L, U) < (L', U')) :\equiv \exists(q : \mathbb{Q}).\, U(q) \wedge L'(q)\,.$$

Поскольку $U(q)$ и $L'(q)$ — элементы Ω, лемма верна, пока Ω замкнут относительно конъюнкций и счетных экзистенциалов, что и предполагалось с самого начала. □

Следствие 11.2.16. *Действительные числа Дедекинда полны по Дедекинду: для каждого действительно-значного сечения Дедекинда (L, U) существует единственный $x : \mathbb{R}_d$ такой, что $L(y) = (y < x)$ и $U(y) = (x < y)$, для всех $y : \mathbb{R}_d$.*

Доказательство. По лемме 11.2.15, пополнение Дедекинда $\bar{\mathbb{R}}_d$ для $\mathbb{R}_d$ допустимо для Ω, поэтому, по теореме 11.2.14, имеем вложение $\bar{\mathbb{R}}_d \to \mathbb{R}_d$, а также вложение $\mathbb{R}_d \to \bar{\mathbb{R}}_d$. Но эти вложения должны быть изоморфизмами, так как их композиции — это, сохраняющие порядок, гомоморфизмы полей, фиксирующие плотное подполе $\mathbb{Q}$, а значит, являющиеся тождественными. Данное следствие теперь немедленно следует из того факта, что $\bar{\mathbb{R}}_d \to \mathbb{R}_d$ является изоморфизмом. □

11.3 Действительные числа Коши

Действительные числа Коши, по замыслу, являются пополнением $\mathbb{Q}$ под пределами последовательностей Коши. В классической конструкции этих чисел Коши рассматривается множество $\mathcal{C}$ всех последовательностей Коши в $\mathbb{Q}$, а затем формируется подходящее частное $\mathcal{C}/\approx$. Далее, для демонстрации того, что $\mathcal{C}/\approx$ является полным по Коши, рассматривается последовательность Коши $x : \mathbb{N} \to \mathcal{C}/\approx$, которая поднимается до последовательности последовательностей $\bar{x} : \mathbb{N} \to \mathcal{C}$, и строится предел x, используя $\bar{x}$. Однако, поднятие x до $\bar{x}$ использует аксиому счетного выбора (пример (3.8.1), где $X = \mathbb{N}$) или закон исключения третьего, которого можно избежать. Каждая конструкция действительных чисел, последним шагом которой является частное, страдает этим недостатком. В конструктивной математике есть три распространенных выхода из этой ситуации.

(i) Предположим, что действительные числа представляют собой сетоид $(\mathcal{C}, \approx)$, т.е. тип последовательностей Коши $\mathcal{C}$ с отношением совпадения.Тогда последовательность действительных чисел представляет собой просто последовательность представляющих их последовательностей Коши.

(ii) Можно принять аксиому счетного выбора. В конце концов, эта аксиома верна в большинстве моделей конструктивной математики, основанных на вычислительной точке зрения, таких как модели реализуемости.

(iii) Объявить действительные числа Коши неприемлемыми и использовать вместо них действительные числа Дедекинда. Такое решение совершенно справедливо в определенных контекстах, например, в теоретико-пучковых моделях конструктивной математики. Однако, как отмечалось в §11.2, конструктивные действительные числа Дедекинда имеют свои проблемы.

Однако, при использовании высших индуктивных типов, существует четвертое решение, которое мы считаем предпочтительным по сравнению с любым из вышеперечисленных и интересным даже для классического математика. Идея состоит в том, что действительные числа Коши должны быть *свободным полным метрическим пространством*, порожденным $\mathbb{Q}$. Вообще, создание любой новинки, предназначенной для широкого использования, требует многократных применений ее операций к генераторам. Например, элементы свободной группы на множестве X — это не просто бинарные произведения и инверсии элементов из X, а слова, полученные в результате повторения произведения и инверсий. Таким образом, можно было бы, естественно, ожидать, что то же самое верно и для пополнения Коши, с соответствующей «операцией» — «взять предел последовательности Коши» (в этом случае итерация должна была бы происходить трансфинитно, поскольку даже после бесконечного множества шагов будут появляться новые последовательности Коши, предел которых, однако, конечен).

Приведенное выше рассуждение показывает, что если выполняется исключение третьего или счетный выбор, то пополнение по Коши имеет особое значение: при построении пополнения пространства достаточно прекратить применение операции после первого шага. Это можно рассматривать как аналог того факта, что свободным моноидам и свободным группам можно дать явное описание в терминах (редуцированных) слов. Однако, в §6.11 показано, что высшие индуктивные типы позволяют создавать подобные доступные новинки *напрямую*, независимо от того, доступно ли их явное описание. В этом разделе будет показано, что то же самое верно и для действительных чисел Коши (аналогичный метод позволяет построить пополнение Коши для любого метрического пространства; см. упражнение 11.9). В частности, высшие индуктивные типы позволяют *одновременно* добавлять пределы последовательностей Коши и частное по отношению совпадения, так что можно избежать проблемы поднятия последовательности действительных чисел до последовательности представителей.

11.3.1 Построение действительных чисел Коши

Конструкция действительных чисел Коши $\mathbb{R}_c$ как высшего индуктивного типа немного более тонкая, чем конструкция свободных алгебраических структур, рассмотренных в §6.11. Мы намерены включить конструктор «взять предел», входом которого является последовательность Коши действительных чисел, но понятие «последовательности Коши действительных чисел» зависит от того, существует ли способ измерить «расстояние» между действительными числами. В общем, конечно, расстояние между двумя действительными числами будет другим действительным числом, что может привести к потенциально проблематичной цикличности.

Однако то, что на самом деле нужно для понятия последовательности действительных чисел Коши, — это не общее понятие «расстояние», а способ выразить, что «расстояние между двумя действительными числами меньше, чем ϵ», для любого $\epsilon : \mathbb{Q}_+$. Это может быть представлено семейством бинарных отношений, которые будут обозначаться $\sim_\epsilon : \mathbb{R}_c \to \mathbb{R}_c \to \mathsf{Prop}$. Предполагаемое значение $x \sim_\epsilon y$ есть $|x - y| < \epsilon$, но поскольку у нас пока нет понятий вычитания, абсолютного значения или неравенства (в конце концов, мы только определяем $\mathbb{R}_c$), нам нужно будет определить эти отношения $\sim_\epsilon$ одновременно с определением $\mathbb{R}_c$. А поскольку $\sim_\epsilon$ является семейством типов, индексируемым двумя копиями $\mathbb{R}_c$, мы не можем сделать это с помощью обычного взаимного (высшего) индуктивного определения; вместо этого мы должны использовать *высшее индуктивно-индуктивное определение*.

Напомним из §5.7, что обычное понятие индуктивно-индуктивного определения позволяет определять тип и индексируемое им семейство типов одновременной индукцией. Конечно, «высшая» версия этого позволяет, как типу, так и семейству, иметь конструкторы путей, а также конструкторы точек. Мы не будем пытаться сформулировать какую-либо общую теорию высших индуктивно-индуктивных определений, но будем надеяться, что описание, которое мы дадим для $\mathbb{R}_c$ и $\sim_\epsilon$, сделает идею прозрачной.

Замечание 11.3.1. Можно было бы также рассмотреть высшее *индуктивно-рекурсивное определение*, в котором $\sim_\epsilon$ определяется с использованием принципа *рекурсии* $\mathbb{R}_c$, одновременно с *индуктивным* определением $\mathbb{R}_c$. Вместо этого выбирается индуктивно-индуктивный путь, по двум причинам. Во-первых, высшие индуктивно-рекурсивные определения кажутся более трудными для обоснования в гомотопической семантике. Во-вторых, и это более важно, индуктивно-индуктивное определение дает более мощный принцип индукции, который понадобится для разработки даже базовой теории действительных чисел Коши.

Наконец, как и при обсуждении полноты по Коши вещественных чисел Дедекинда в §11.2.2, мы будем иметь дело с *аппроксимациями Коши* (определение 11.2.10) вместо последовательностей Коши. Конечно, наши аппроксимации Коши теперь будут состоять из действительных чисел Коши, а не из действительных чисел Дедекинда или рациональных чисел.

Определение 11.3.2. Пусть $\mathbb{R}_c$ и отношение $\sim : \mathbb{Q}_+ \times \mathbb{R}_c \times \mathbb{R}_c \to U$ — следующее семейство высших индуктивно-индуктивных типов. Тип $\mathbb{R}_c$ **действительных чисел Коши** генерируется следующими конструкторами:

- *рациональные точки*: для любого $q : \mathbb{Q}$ существует действительное число $\mathsf{rat}(q)$;
- *предельные точки*: для любого $x : \mathbb{Q}_+ \to \mathbb{R}_c$ такого, что
$$\forall(\delta, \epsilon : \mathbb{Q}_+).\, x_\delta \sim_{\delta+\epsilon} x_\epsilon\,, \tag{11.3.3}$$
существует точка $\mathsf{lim}(x) : \mathbb{R}_c$. Мы называем x **аппроксимацией Коши**;
- *пути*: для $u, v : \mathbb{R}_c$ таких, что
$$\forall(\epsilon : \mathbb{Q}_+).\, u \sim_\epsilon v\,, \tag{11.3.4}$$
существует путь $\mathsf{eq}_{\mathbb{R}_c}(u, v) : u =_{\mathbb{R}_c} v$.

Одновременно, семейство типов $\sim$: $\mathbb{R}_c \to \mathbb{R}_c \to \mathbb{Q}_+ \to \mathcal{U}$ генерируется следующими конструкторами. Здесь q и r обозначают рациональные числа; δ, ϵ и η обозначают положительные рациональные числа; u и v обозначают действительные числа Коши; а x и y — аппроксимации Коши:

- для любых q, r, ϵ, если $-\epsilon < q - r < \epsilon$, то $\mathsf{rat}(q) \sim_\epsilon \mathsf{rat}(r)$,
- для любых q, y, ϵ, δ, если $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$, то $\mathsf{rat}(q) \sim_\epsilon \mathsf{lim}(y)$,
- для любых x, r, ϵ, δ, если $x_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$, то $\mathsf{lim}(x) \sim_\epsilon \mathsf{rat}(r)$,
- для любых $x, y, \epsilon, \delta, \eta$, если $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$, то $\mathsf{lim}(x) \sim_\epsilon \mathsf{lim}(y)$,
- для любых u, v, ϵ, если $\xi, \zeta : u \sim_\epsilon v$, то $\xi = \zeta$ (пропозициональное усечение).

Первый конструктор $\mathbb{R}_c$ выражает то, что любое рациональное число можно рассматривать как действительное число. Второй гласит, что из любой аппроксимации Коши к действительному числу можно получить новое действительное число, называемое его «пределом». А третий отражает то, что если две аппроксимации Коши совпадают, то их пределы равны.

Первые четыре конструктора $\sim$ определяют, когда два рациональных числа близки, когда рациональное число близко к пределу и когда два предела близки. В случае двух рациональных чисел это просто обычное понятие ϵ-близости для рациональных чисел, тогда как другие случаи можно вывести, заметив, что каждая аппроксимация x_δ должна находиться в рамках δ предела $\mathsf{lim}(x)$.

Напомним о релевантности доказательства: действительное число, полученное из lim, представлено не только аппроксимацией Коши x, но и доказательством p выражения (11.3.3), поэтому технически мы должны были бы написать $\mathsf{lim}(x, p)$ вместо просто $\mathsf{lim}(x)$. Аналогичное наблюдение относится к $\mathsf{eq}_{\mathbb{R}_c}$ и (11.3.4), но мы будем писать просто $\mathsf{eq}_{\mathbb{R}_c} : u = v$ вместо $\mathsf{eq}_{\mathbb{R}_c}(u, v, p) : u = v$. Эти злоупотребления обозначениями смягчаются тем фактом, что мы опускаем простые предложения и информацию, которая легко угадывается. Точно так же, последний конструктор $\sim_\epsilon$ оправдывает то, что мы оставили остальные четыре безымянными.

Сразу можно заполнить $\mathbb{R}_c$ многими действительными числами. Предположим, что $x : \mathbb{N} \to \mathbb{Q}$ — традиционная последовательность Коши рациональных чисел, и пусть $M : \mathbb{Q}_+ \to \mathbb{N}$ — ее модуль конвергенции. Тогда $\mathsf{rat} \circ x \circ M : \mathbb{Q}_+ \to \mathbb{R}_c$ является аппроксимацией Коши с использованием первого конструктора $\sim$ для создания нужного свидетельства. Таким образом, $\mathsf{lim}(\mathsf{rat} \circ x \circ m)$ - действительное число. Различные известные действительные числа, такие как $\sqrt{2}$, π, e, … являются пределами таких последовательностей Коши рациональных чисел.

11.3.2 Индукция и рекурсия на действительных числах Коши

Чтобы сделать что-нибудь полезное с $\mathbb{R}_c$, конечно, нужно указать его принцип индукции. Как и в случае, когда индуктивно определяются два или более объектов одновременно, базовый принцип индукции для $\mathbb{R}_c$ и $\sim$ требует одновременной индукции по обоим сразу. В этом случае, можно ожидать, предполагая два семейства типов над $\mathbb{R}_c$ и $\sim$, соответственно (вместе с данными, соответствующими каждому конструктору), что существуют сечения обоих этих семейств. Однако, поскольку $\sim$ индексируется на двух копиях $\mathbb{R}_c$, точные зависимости этих семейств немного неуловимы. Принцип индукции применим к любой паре семейств типов:

$$\begin{aligned}
&A : \mathbb{R}_c \to \mathcal{U} \\
&B : \prod_{x,y:\mathbb{R}_c} A(x) \to A(y) \to \prod_{\epsilon:Q_+} (x \sim_\epsilon y) \to \mathcal{U}\,.
\end{aligned}$$

Тип A очевиден, но тип B требует небольшого размышления. Поскольку B должен зависеть от $\sim$, а $\sim$, в свою очередь, зависит от двух копий $\mathbb{R}_c$ и одной копии $\mathbb{Q}_+$, совершенно очевидно, что B также должен зависеть от переменных $x, y : \mathbb{R}_c$ и $\epsilon : \mathbb{Q}_+$, а также от элемента из $(x \sim_\epsilon y)$. Что немного менее очевидно, так это то, что B также должен зависеть от $A(x)$ и $A(y)$.

Это может стать более очевидным, если рассмотреть независимый случай (принцип рекурсии), где A — простой тип (а не семейство типов). В этом случае ожидаем, что B не будет зависеть от $x, y : \mathbb{R}_c$ или $x \sim_\epsilon y$. Но принцип рекурсии (вместе со связанным с ним принципом уникальности) должен отражать то, что $\mathbb{R}_c$ с $\sim_\epsilon$ является «инициальным объектом» в некоторой категории, поэтому в этом случае структура зависимостей A и B должна отражать структуру зависимостей $\mathbb{R}_c$ и $\sim_\epsilon$, то есть должно быть $B : A \to A \to \mathbb{Q}_+ \to \mathcal{U}$. Комбинируя это наблюдение с тем фактом, что в зависимом случае B также должен зависеть от $x, y : \mathbb{R}_c$ и $x \sim_\epsilon y$, неизбежно приходим к типу, указанному выше для B.

Полезно думать о B как об ϵ-индексированном семействе отношений между типами $A(x)$ и $A(y)$. Тогда можно записать $B(x, y, a, b, \epsilon, \xi)$ как $(x, a) \frown_\epsilon^\xi (y, b)$. Поскольку $\xi : x \sim_\epsilon y$ уникален, когда он существует, мы обычно опускаем его в обозначениях и пишем $(x, a) \frown_\epsilon (y, b)$; это некритично, пока мы помним, что это отношение определено только тогда, когда $x \sim_\epsilon y$. Мы также можем иногда применить еще большее упрощение и записывать $a \frown_\epsilon b$, где x и y выведены из типов a и b, но иногда будет необходимо указывать их, для ясности.

Теперь, для семейства типов $A : \mathbb{R}_c \to \mathcal{U}$ и семейства отношений $\frown$, как указано выше, гипотезы принципа индукции состоят из следующих положений, по одному для каждого конструктора $\mathbb{R}_c$ или $\sim$.

- Для любого $q : \mathbb{Q}$, имется элемент $f_q : A(\mathsf{rat}(q))$.
- Для любой аппроксимации Коши x и любого $a : \prod_{(\epsilon:\mathbb{Q}_+)} A(x_\epsilon)$ таких, что
$$\forall(\delta, \epsilon : \mathbb{Q}_+).\, (x_\delta, a_\delta) \frown_{\delta+\epsilon} (x_\epsilon, a_\epsilon)\,, \tag{11.3.5}$$
имеется элемент $f_{x,a} : A(\mathsf{lim}(x))$. Назовем такое a **зависимой аппроксимацией Коши** по x.
- Для $u, v : \mathbb{R}_c$ таких, что $h : \forall(\epsilon : \mathbb{Q}_+).\, u \sim_\epsilon v$, и всех $a : A(u)$ и $b : A(v)$ таких, что $\forall(\epsilon : Q_+).\, (u, a) \frown_\epsilon (v, b)$, имеется зависимый путь $a =^A_{\mathsf{eq}_{\mathbb{R}_c}(u,v)} b$.
- Для $q, r : \mathbb{Q}$ и $\epsilon : \mathbb{Q}_+$, если $-\epsilon < q - r < \epsilon$, то имеем $(\mathsf{rat}(q), f_q) \frown_\epsilon (\mathsf{rat}(r), f_r)$.

- Для $q : \mathbb{Q}$, $\delta, \epsilon : \mathbb{Q}_+$, y — аппроксимации Коши и b — зависимой аппроксимации Коши по y, если $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$, то

$$(\mathsf{rat}(q), f_q) \frown_{\epsilon-\delta} (y_\delta, b_\delta) \;\Rightarrow\; (\mathsf{rat}(q), f_q) \frown_\epsilon (\mathsf{lim}(y), f_{y,b}) .$$

- Точно так же, для $r : \mathbb{Q}$, $\delta, \epsilon : \mathbb{Q}_+$, x — аппроксимации Коши и a — зависимой аппроксимации Коши по x, если $x_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$, то

$$(x_\delta, a_\delta) \frown_{\epsilon-\delta} (\mathsf{rat}(r), f_r) \;\Rightarrow\; (\mathsf{lim}(x), f_{x,a}) \frown_\epsilon (\mathsf{rat}(q), f_r) .$$

- Для $\epsilon, \delta, \eta : \mathbb{Q}_+$, x, y — аппроксимаций Коши, a и b — зависимых аппроксимаций Коши по x и по y, соответственно, если имеет место $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$, то

$$(x_\delta, a_\delta) \frown_{\epsilon-\delta-\eta} (y_\eta, b_\eta) \;\Rightarrow\; (\mathsf{lim}(x), f_{x,a}) \frown_\epsilon (\mathsf{lim}(y), f_{y,b}) .$$

- Для $\epsilon : \mathbb{Q}_+$, $x, y : \mathbb{Q}_+$, $\xi, \zeta : x \sim_\epsilon y$, $a : A(x)$ и $b : A(y)$, любые два элемента из $(x,a) \frown^\xi_\epsilon (y,b)$ и $(x,a) \frown^\zeta_\epsilon (y,b)$ зависимо равны по $\xi = \zeta$. Отметим, что, как обычно, это равносильно тому, что $\frown$ принимает значения в простых высказываниях.

При этих предположениях выводим функции

$$\begin{aligned} f &: \prod_{x:\mathbb{R}_c} A(x) \\ g &: \prod_{(x,y:\mathbb{R}_c)} \; \prod_{(\epsilon:\mathbb{Q}_+)} \; \prod_{(\xi : x\sim_\epsilon y)} (x, f(x)) \frown^\xi_\epsilon (y, f(y)) \end{aligned}$$

которые вычисляются так, как и ожидалось:

$$f(\mathsf{rat}(q)) :\equiv f_q , \tag{11.3.6}$$
$$f(\mathsf{lim}(x)) :\equiv f_{x,(f,g)[x]} . \tag{11.3.7}$$

Здесь, $(f,g)[x]$ обозначает результат применения f и g к аппроксимации Коши x для получения зависимой аппроксимации Коши по x. То есть мы определяем $(f,g)[x]_\epsilon :\equiv f(x_\epsilon) : A(x_\epsilon)$, а затем, для любых $\epsilon, \delta : \mathbb{Q}_+$, имеем $g(x_\epsilon, x_\delta, \epsilon+\delta, \xi)$, чтобы засвидетельствовать тот факт, что $(f,g)[x]$ является зависимой аппроксимацией Коши , где $\xi : x_\epsilon \sim_{\epsilon+\delta} x_\delta$ возникает из предположения, что x является аппроксимацией Коши.

Далее это обозначение не будет использоваться, так что не стоит прилагать усилий для его запоминания. Обычно мы используем соглашение о сопоставлении с образцом, где f определяется такими уравнениями, как (11.3.6) и (11.3.7), в которых правая часть (11.3.7) может включать символы $f(x_\epsilon)$ и предположение, что они образуют зависимую аппроксимацию Коши.

Тем не менее, этот принцип индукции, по общему признанию, все еще довольно громоздкий. Чтобы помочь разобраться в этом, отметим, что в качестве особых случаев он содержит два отдельных принципа индукции для $\mathbb{R}_c$ и для $\sim$. Во-первых, предположим, что задано только семейство типов $A : \mathbb{R}_c \to \mathcal{U}$, и определим $\frown$ как константу в $\mathbf{1}$. Тогда большая часть требуемых данных становится тривиальной, и у нас остается:

- для любого $q : \mathbb{Q}$, имеется элемент $f_q : A(\mathsf{rat}(q))$,
- для любой аппроксимайии Коши x и любого $a : \prod_{(\epsilon:\mathbb{Q}_+)} A(x_\epsilon)$, имеется элемент $f_{x,a} : A(\mathsf{lim}(x))$,

- для $u, v : \mathbb{R}_\mathsf{c}$, $h : \forall(\epsilon : \mathbb{Q}_+).\, u \sim_\epsilon v$, $a : A(u)$ и $b : A(v)$, имеем $a =^A_{\mathsf{eq}_{\mathbb{R}_\mathsf{c}}(u,v)} b$.

Для этих данных, принцип индукции дает функцию $f : \prod_{(x:\mathbb{R}_\mathsf{c})} A(x)$ такую, что

$$f(\mathsf{rat}(q)) :\equiv f_q\,,$$
$$f(\mathsf{lim}(x)) :\equiv f_{x,f(x)}\,.$$

Мы называем этот принцип $\mathbb{R}_\mathsf{c}$**-индукцией**; по сути, это говорит о том, что если мы принимаем $\sim_\epsilon$ как данное, то $\mathbb{R}_\mathsf{c}$ индуктивно генерируется его конструкторами.

Заметим, что если A — простое свойство, то третья гипотеза $\mathbb{R}_\mathsf{c}$-индукции является автоматической (сейчас мы увидим, что на самом деле это эквивалентные утверждения). Таким образом, мы можно доказать простые свойства действительных чисел, просто доказав их для рациональных чисел и для пределов аппроксимаций Коши. Приведем пример.

Лемма 11.3.8. *Для любых $u : \mathbb{R}_c$ и $\epsilon : \mathbb{Q}_+$, we have $u \sim_\epsilon u$.*

Доказательство. Определим $A(u) :\equiv \forall(\epsilon : \mathbb{Q}_+).\,(u \sim_\epsilon u)$. Поскольку это просто высказывание (по последнему конструктору $\sim$), по $\mathbb{R}_\mathsf{c}$-индукции, достаточно доказать его, когда u есть $\mathsf{rat}(q)$ и когда u есть $\mathsf{lim}(x)$. В первом случае, очевидно, имеем $|q-q| < \epsilon$ для любого ϵ, следовательно, $\mathsf{rat}(q) \sim_\epsilon \mathsf{rat}(q)$ по первому конструктору $\sim$. А во втором случае можно индуктивно предположить, что $x_\delta \sim_\epsilon x_\delta$, для всех $\delta, \epsilon : \mathbb{Q}_+$. Тогда, в частности, имеем $x_{\epsilon/3} \sim_{\epsilon/3} x_{\epsilon/3}$, откуда $\mathsf{lim}(x) \sim_\epsilon \mathsf{lim}(x)$ по четвертому конструктору $\sim$. □

Из леммы 11.3.8 заключаем, что прямое применение $\mathbb{R}_\mathsf{c}$-индукции имеет шанс на успех, только если семействоиндукции имеет шанс на успех, только если семейство $A : \mathbb{R}_\mathsf{c} \to \mathcal{U}$ — это простое свойство. Чтобы увидеть это, зафиксируем $u : \mathbb{R}_\mathsf{c}$. Принимая v за u, третья гипотеза $\mathbb{R}_\mathsf{c}$-индукции говорит, что, для любого $a : A(u)$, имеем $a =^A_{\mathsf{eq}_{\mathbb{R}_\mathsf{c}}(u,u)} a$. Для точки $b : A(u)$, кроме того, также получаем $a =^A_{\mathsf{eq}_{\mathbb{R}_\mathsf{c}}(u,u)} b$. Из определения типа зависимого пути заключаем, что комбинация этих двух путей влечет $a = b$, т.е. все точки в $A(u)$ равны.

Теорема 11.3.9. $\mathbb{R}_\mathsf{c}$ — *множество.*

Доказательство. Только что было показано, что простое отношение $P(u,v) :\equiv \forall(\epsilon : \mathbb{Q}_+).\,(u \sim_\epsilon v)$ рефлексивно. Поскольку это подразумевает тождество, с помощью конструктора пути $\mathbb{R}_\mathsf{c}$, результат следует из теоремы 7.2.2. □

Также можно показать, что хотя $\mathbb{R}_\mathsf{c}$ может и не быть частным множества последовательностей Коши *рациональных чисел*, тем не менее, оно является частным множества последовательностей Коши *действительных чисел* (конечно, это недопустимое *определение* $\mathbb{R}_\mathsf{c}$, но является полезным свойством). Определим тип аппроксимаций Коши как

$$\mathcal{C} :\equiv \{x : \mathbb{Q}_+ \to \mathbb{R}_\mathsf{c} \mid \forall(\epsilon, \delta : \mathbb{Q}_+).\, x_\delta \sim_{\delta+\epsilon} x_\epsilon\}\ .$$

Второй конструктор $\mathbb{R}_\mathsf{c}$ дает функцию $\mathsf{lim} : \mathcal{C} \to \mathbb{R}_\mathsf{c}$.

Лемма 11.3.10. *Каждое действительное число просто является предельной точкой:* $\forall(u : \mathbb{R}_\mathsf{c}).\,\exists(x : \mathcal{C}).\, u = \mathsf{lim}(x)$. *Другими словами,* $\mathsf{lim} : \mathcal{C} \to \mathbb{R}_\mathsf{c}$ *сюръективен.*

Доказательство. По $\mathbb{R}_{\mathsf{c}}$-индукции можно разделить случаи по u. Конечно, если u — предел $\mathsf{lim}(x)$, утверждение тривиально. Предположим, что u — рациональная точка $\mathsf{rat}(q)$; мы утверждаем, что u равно $\mathsf{lim}(\lambda\epsilon.\,\mathsf{rat}(q))$. С помощью конструктора пути $\mathbb{R}_{\mathsf{c}}$, достаточно показать, что $\mathsf{rat}(q) \sim_\epsilon \mathsf{lim}(\lambda\epsilon.\,\mathsf{rat}(q))$, для всех $\epsilon : \mathbb{Q}_+$. А с помощью второго конструктора $\sim$, для этого достаточно найти $\delta : \mathbb{Q}_+$ такое, что $\mathsf{rat}(q) \sim_{\epsilon-\delta} \mathsf{rat}(q)$. Но с помощью первого конструктора $\sim$, можно взять любой $\delta : \mathbb{Q}_+$ с $\delta < \epsilon$. □

Лемма 11.3.11. *Если A — множество и $f : \mathcal{C} \to A$ учитывает совпадение аппроксимаций Коши в том смысле, что*

$$\forall (x, y : \mathcal{C}).\,\mathsf{lim}(x) = \mathsf{lim}(y) \Rightarrow f(x) = f(y)\,,$$

то f однозначно факторизуется посредством $\mathsf{lim} : \mathcal{C} \to \mathbb{R}_{\mathsf{c}}$.

Доказательство. Поскольку lim сюръективен, по теореме 10.1.5 $\mathbb{R}_{\mathsf{c}}$ является частным $\mathcal{C}$ по ядерной паре lim. Но это и является утверждением леммы. □

Для второго частного случая принципа индукции предположим, вместо этого, что A постоянно в точке $\mathbf{1}$. В этом случае, $\frown$ — это просто ϵ-индексированное семейство отношений на ϵ-близких парах действительных чисел, поэтому можно использовать $u \frown_\epsilon v$ вместо $(u, \star) \frown_\epsilon (v, \star)$. Тогда требуемые данные сводятся к следующему, где q, r обозначают рациональные числа, ϵ, δ, η — положительные рациональные числа, а x, y — аппроксимации Коши:

- если $-\epsilon < q - r < \epsilon$, то $\mathsf{rat}(q) \frown_\epsilon \mathsf{rat}(r)$,
- если $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$ и $\mathsf{rat}(q) \frown_{\epsilon-\delta} y_\delta$, то $\mathsf{rat}(q) \frown_\epsilon \mathsf{lim}(y)$,
- если $x_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$ и $x_\delta \frown_{\epsilon-\delta} \mathsf{rat}(r)$, то $\mathsf{lim}(y) \frown_\epsilon \mathsf{rat}(q)$,
- если $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$ и $x_\delta \frown_{\epsilon-\delta-\eta} y_\eta$, то $\mathsf{lim}(x) \frown_\epsilon \mathsf{lim}(y)$.

В результате получается $\forall (u, v : \mathbb{R}_{\mathsf{c}}).\,\forall (\epsilon : \mathbb{Q}_+).\,(u \sim_\epsilon v) \to (u \frown_\epsilon v)$. Мы называем этот принцип **$\sim$-индукцией**; по сути, это говорит о том, что если принять $\mathbb{R}_{\mathsf{c}}$ как данное, то $\sim_\epsilon$ индуктивно генерируется (как семейство типов) *его* конструкторами. Например, это можно использовать, чтобы показать, что $\sim$ симметрично.

Лемма 11.3.12. *Для любых $u, v : \mathbb{R}_{\mathsf{c}}$ и $\epsilon : \mathbb{Q}_+$, имеет место $(u \sim_\epsilon v) = (v \sim_\epsilon u)$.*

Доказательство. Поскольку обе части доказываемого равенства являются простыми высказываниями, по симметрии достаточно показать одну импликацию. Итак, пусть $(u \frown_\epsilon v) :\equiv (v \sim_\epsilon u)$. По $\sim$-индукции эту возможность можно свести к случаю, когда $u \sim_\epsilon v$ получается от одного из четырех интересных конструкторов $\sim$. В первом случае, когда u и v оба рациональны, результат тривиален (можно снова применить первый конструктор). В остальных трех случаях индуктивная гипотеза (вместе с коммутативностью сложения в $\mathbb{Q}$) дает в точности входные данные для другого конструктора $\sim$ (второй и третий конструкторы меняются местами, а четвертый остается на месте). □

Таким образом, общий принцип индукции, который можно назвать **$(\mathbb{R}_{\mathsf{c}}, \sim)$-индукцией**, является своего рода совместной $\mathbb{R}_{\mathsf{c}}$-индукцией и $\sim$-индукцией. Рассмотрим, к примеру, ее независимую версию, которую назовем **$(\mathbb{R}_{\mathsf{c}}, \sim)$-рекурсией**, и которая может больше всего пригодиться. Обычная $\mathbb{R}_{\mathsf{c}}$-рекурсия обеспечивает то, что для определения функции $f : \mathbb{R}_{\mathsf{c}} \to A$ достаточно:

(i) для каждого $q : \mathbb{Q}$ построить $f(\mathsf{rat}(q)) : A$,

(ii) для каждой аппроксимации Коши $x : \mathbb{Q}_+ \to \mathbb{R}_\mathsf{c}$, построить $f(x) : A$, предполагая, что $f(x_\epsilon)$ уже определена для всех $\epsilon : \mathbb{Q}_+$,

(iii) доказать, что $f(u) = f(v)$ для всех $u, v : \mathbb{R}_\mathsf{c}$, удовлетворяющих $\forall(\epsilon : \mathbb{Q}_+).\, u \sim_\epsilon v$.

Однако, как правило, довольно трудно показать (iii), не зная кое-что о том, как f действует на ϵ-близкие действительные числа Коши. Усовершенствованный принцип $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии исправляет этот недостаток, позволяя указать *произвольный* «способ, которым f действует на ϵ-близкие числа Коши», который тогда можно доказать по одновременной индукции с определением f. Это семейство отношений $\frown$. Поскольку A не зависит от $\mathbb{R}_\mathsf{c}$, можно для простоты считать, что $\frown$ зависит только от A и $\mathbb{Q}_+$, и, таким образом, нет никакой двусмысленности в записи $a \frown_\epsilon b$ вместо $(u, a) \frown_\epsilon (v, b)$. В этом случае, определение функции $f : \mathbb{R}_\mathsf{c} \to A$ с помощью $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии требует учета следующих случаев (которые записываются, используя соглашение о сопоставлении с образцом).

- Для каждого $q : \mathbb{Q}$, конструируется $f(\mathsf{rat}(q)) : A$.

- Для каждой аппроксимации Коши $x : \mathbb{Q}_+ \to \mathbb{R}_\mathsf{c}$, конструируется $f(\mathsf{lim}(x)) : A$, индуктивно предполагая, что $f(x_\epsilon)$ уже определена для всех $\epsilon : \mathbb{Q}_+$ и формируется «аппроксимация Коши относительно $\frown$», т.е., что $\forall(\epsilon, \delta : \mathbb{Q}_+).\, (f(x_\epsilon) \frown_{\epsilon+\delta} f(x_\delta))$.

- Доказывается, что отношения $\frown$ являются *разделяемыми*, т.е., что, для любых $a, b : A$, верно $(\forall(\epsilon : \mathbb{Q}_+).\, a \frown_\epsilon b) \Rightarrow (a = b)$.

- Доказывается, что если $-\epsilon < q - r < \epsilon$ для $q, r : \mathbb{Q}_+$, то $f(\mathsf{rat}(q)) \frown_\epsilon f(\mathsf{rat}(r))$.

- Для любого $q : \mathbb{Q}$ и любой аппроксимации Коши y, доказывается, что $f(\mathsf{rat}(q)) \frown_\epsilon f(\mathsf{lim}(y))$, индуктивно предполагая, что $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$ и $f(\mathsf{rat}(q)) \frown_{\epsilon-\delta} f(y_\delta)$ для некоторого $\delta : \mathbb{Q}_+$, и что $\eta \mapsto f(x_\eta)$ является аппроксимацией Коши относительно $\frown$.

- Для любой аппроксимации Коши x и любого $r : \mathbb{Q}$, доказывается, что $f(\mathsf{lim}(x)) \frown_\epsilon f(\mathsf{rat}(r))$, индуктивно предполагая, что $x_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$ и $f(x_\delta) \frown_{\epsilon-\delta} f(\mathsf{rat}(r))$ для некоторого $\delta : \mathbb{Q}_+$, и что $\eta \mapsto f(x_\eta)$ является аппроксимацией Коши относительно $\frown$.

- Для любых аппроксимаций Коши x, y, доказывается, что $f(\mathsf{lim}(x)) \frown_\epsilon f(\mathsf{lim}(y))$, индуктивно предполагая, что $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$ и $f(x_\delta) \frown_{\epsilon-\delta-\eta} f(y_\eta)$ для некоторых $\delta, \eta : \mathbb{Q}_+$, и, что $\theta \mapsto f(x_\theta)$ и $\theta \mapsto f(y_\theta)$ являются аппроксимациями Коши относительно $\frown$.

Заметим, что в последних четырех доказательствах можно использовать конкретные определения $f(\mathsf{rat}(q))$ и $f(\mathsf{lim}(x))$ из первых двух пунктов. Однако доказательство разделяемости должно применяться к *любым* двум элементам из A без всякого отношения к f: это своего рода условие «допустимости» для семейства отношений $\frown$. Таким образом, мы часто сначала проверяем его, сразу после определения $\frown$, прежде чем переходить к определению $f(\mathsf{rat}(q))$ и $f(\mathsf{lim}(x))$.

При указанных выше гипотезах, $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсия дает функцию $f : \mathbb{R}_\mathsf{c} \to A$ такую, что $f(\mathsf{rat}(q))$ и $f(\mathsf{lim}(x))$ субъективно равны определениям, данным для них в первых двух пунктах. Кроме того, также можно заключить

$$\forall(u, v : \mathbb{R}_\mathsf{c}).\, \forall(\epsilon : \mathbb{Q}_+).\, (u \sim_\epsilon v) \to (f(u) \frown_\epsilon f(v))\,. \tag{11.3.13}$$

Парадигматический пример: $(\mathbb{R}_{\mathsf{c}}, \sim)$-рекурсия позволяет расширить функции, определенные на $\mathbb{Q}$, на все $\mathbb{R}_{\mathsf{c}}$, если они достаточно непрерывны.

Определение 11.3.14. Функция $f : \mathbb{Q} \to \mathbb{R}_{\mathsf{c}}$ является **функцией Лифшица** если существует $L : \mathbb{Q}_+$ (**постоянная Лифшица**) такая, что

$$|q - r| < \epsilon \Rightarrow (f(q) \sim_{L\epsilon} f(r))$$

для всех $\epsilon : \mathbb{Q}_+$ и $q, r : \mathbb{Q}$. Аналогично, $g : \mathbb{R}_{\mathsf{c}} \to \mathbb{R}_{\mathsf{c}}$ является **функцией Лифшица**, если существует $L : \mathbb{Q}_+$ такая, что

$$(u \sim_\epsilon v) \Rightarrow (g(u) \sim_{L\epsilon} g(v))$$

для всех $\epsilon : \mathbb{Q}_+$ и $u, v : \mathbb{R}_{\mathsf{c}}$.

В частности, обратите внимание, что по первому конструктору $\frown$, если $f : \mathbb{Q} \to \mathbb{Q}$ является лифшицевой в очевидном смысле, то такой же является и композиция $\mathbb{Q} \xrightarrow{f} \mathbb{Q} \to \mathbb{R}_{\mathsf{c}}$.

Лемма 11.3.15. *Предположим, что $f : \mathbb{Q} \to \mathbb{R}_{\mathsf{c}}$ является функцией Лифшица с постоянной $L : \mathbb{Q}_+$. Тогда существует отображение Лифшица $\bar{f} : \mathbb{R}_{\mathsf{c}} \to \mathbb{R}_{\mathsf{c}}$, также с постоянной L, такое, что $\bar{f}(\mathsf{rat}(q)) \equiv f(q)$ для всех $q : \mathbb{Q}$.*

Доказательство. Определим $\bar{f}$ по $(\mathbb{R}_{\mathsf{c}}, \sim)$-рекурсии, с ко-областью $A :\equiv \mathbb{R}_{\mathsf{c}}$. Определим отношение $\frown : \mathbb{R}_{\mathsf{c}} \to \mathbb{R}_{\mathsf{c}} \to \mathbb{Q}_+ \to \mathsf{Prop}$ как

$$(u \frown_\epsilon v) :\equiv (u \sim_{L\epsilon} v).$$

Для $q : \mathbb{Q}$, определим

$$\bar{f}(\mathsf{rat}(q)) :\equiv \mathsf{rat}(f(q)).$$

Для аппроксимации Коши $x : \mathbb{Q}_+ \to \mathbb{R}_{\mathsf{c}}$, определим

$$\bar{f}(\mathsf{lim}(x)) :\equiv \mathsf{lim}(\lambda\epsilon.\, \bar{f}(x_{\epsilon/L})).$$

Чтобы это имело смысл, надо убедиться, что $y :\equiv \lambda\epsilon.\, \bar{f}(x_{\epsilon/L})$ является аппроксимацией Коши. Однако, индуктивная гипотеза для этого шага состоит в том, что, для любых $\delta, \epsilon : \mathbb{Q}_+$ имеет место $\bar{f}(x_\delta) \frown_{\delta+\epsilon} \bar{f}(x_\epsilon)$, т.е. $\bar{f}(x_\delta) \sim_{L\delta+L\epsilon} \bar{f}(x_\epsilon)$. Таким образом,

$$y_\delta \equiv f(x_{\delta/L}) \sim_{\delta+\epsilon} f(x_{\epsilon/L}) \equiv y_\epsilon.$$

Для доказательства разделяемости мы просто замечаем, что $\forall(\epsilon : \mathbb{Q}_+).\, a \frown_\epsilon b$ означает $\forall(\epsilon : \mathbb{Q}_+).\, a \sim_{L\epsilon} b$, откуда следует $\forall(\epsilon : \mathbb{Q}_+).\, a \sim_\epsilon b$ и, следовательно, $a = b$.

Для завершения $(\mathbb{R}_{\mathsf{c}}, \sim)$-рекурсии остается проверить четыре условия на $\frown$. По сути, это сводится к доказательству того, что $\bar{f}$ является функцией Лифшица для всех четырех конструкторов $\sim$.

(i) Когда u есть $\mathsf{rat}(q)$, а v есть $rat(r)$ с $-\epsilon < |q - r| < \epsilon$, предположение, что f является функцией Лифшица, приводит к $f(q) \sim_{L\epsilon} f(r)$, следовательно, $\bar{f}(\mathsf{rat}(q)) \frown_\epsilon \bar{f}(\mathsf{rat}(r))$ по определению.

(ii) Когда u есть $\mathsf{lim}(x)$, а v есть $\mathsf{rat}(q)$ с $x_\eta \sim_{\epsilon-\eta} \mathsf{rat}(q)$, тогда индуктивная гипотеза есть $\bar{f}(x_\eta) \sim_{L\epsilon-L\eta} \mathsf{rat}(f(q))$, что доказывает $\bar{f}(\mathsf{lim}(x)) \sim_{L\epsilon} \bar{f}(\mathsf{rat}(q))$, по третьему конструктору $\sim$.

(iii) Симметричный случай, когда u рационально, а v — предел, по существу, идентичен.

(iv) Когда u есть $\mathsf{lim}(x)$, а v есть $\mathsf{lim}(y)$, с $\delta, \eta : \mathbb{Q}_+$ такими, что $x_\delta \sim_{\epsilon-\delta-\eta} y_\eta$, тогда индуктивная гипотеза есть $\bar{f}(x_\delta) \sim_{L\epsilon-L\delta-L\eta} \bar{f}(y_\eta)$, что доказывает $\bar{f}(\mathsf{lim}(x)) \sim_{L\epsilon} \bar{f}(\mathsf{lim}(y))$ использованием четвертого конструктора $\sim$.

Это завершает $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсию и, следовательно, построение $\bar{f}$. Искомое равенство $\bar{f}(\mathsf{rat}(q)) \equiv f(q)$ является в точности первым правилом вычисления для $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии, а дополнительное условие (11.3.13) точно выражает, что $\bar{f}$ является функцией Лифшица с константой L. □

Здесь мы зашли так далеко, как только могли. Мы указали в конструкторах $\sim$ условия, при которых действительные числа Коши двух разных форм были бы ϵ-близкими. Однако, откуда известно, что в получившемся семействе индуктивно-индуктивных типов они являются единственным свидетельством этого факта? Мы видели, что семейства индуктивных типов (например, тождественные типы, см. §5.8) и высшие индуктивные типы имеют тенденцию содержать «больше, чем в них было вложено», так что это не праздный вопрос.

Чтобы более точно охарактеризовать $\sim$, *рекурсивно* определим семейство отношений $\approx_\epsilon$ на $\mathbb{R}_\mathsf{c}$, так что они будут вычисляться на конструкторах, и докажем, что это семейство эквивалентно $\sim_\epsilon$.

Теорема 11.3.16. *Существует семейство простых отношений* $\approx : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c} \to \mathbb{Q}_+ \to \mathsf{Prop}$ *такое, что*

$$(\mathsf{rat}(q) \approx_\epsilon \mathsf{rat}(r)) :\equiv (-\epsilon < q - r < \epsilon) \tag{11.3.17}$$
$$(\mathsf{rat}(q) \approx_\epsilon \mathsf{lim}(y)) :\equiv \exists(\delta : \mathbb{Q}_+).\, \mathsf{rat}(q) \approx_{\epsilon-\delta} y_\delta \tag{11.3.18}$$
$$(\mathsf{lim}(x) \approx_\epsilon \mathsf{rat}(r)) :\equiv \exists(\delta : \mathbb{Q}_+).\, x_\delta \approx_{\epsilon-\delta} \mathsf{rat}(r) \tag{11.3.19}$$
$$(\mathsf{lim}(x) \approx_\epsilon \mathsf{lim}(y)) :\equiv \exists(\delta, \eta : \mathbb{Q}_+).\, x_\delta \approx_{\epsilon-\delta-\eta} y_\eta\,. \tag{11.3.20}$$

Более того, имеет место

$$(u \approx_\epsilon v) \Leftrightarrow \exists(\theta : \mathbb{Q}_+).\, (u \approx_{\epsilon-\theta} v) \tag{11.3.21}$$
$$(u \approx_\epsilon v) \to (v \sim_\delta w) \to (u \approx_{\epsilon+\delta} w) \tag{11.3.22}$$
$$(u \sim_\epsilon v) \to (v \approx_\delta w) \to (u \approx_{\epsilon+\delta} w) \tag{11.3.23}$$

Дополнительные условия (11.3.21)–(11.3.23) оказываются необходимыми для выполнения индуктивного определения. Условие (11.3.21) называется **округлением**. Рассмотрение его справа налево отражает **монотонность** $\approx$,

$$(\delta < \epsilon) \wedge (u \approx_\delta v) \Rightarrow (u \approx_\epsilon v)$$

а слева направо — **открытость** $\approx$,

$$(u \approx_\epsilon v) \Rightarrow \exists(\delta : \mathbb{Q}_+).\, (\delta < \epsilon) \wedge (u \approx_\delta v)\,.$$

Условия (11.3.22) и (11.3.23) являются формами неравенства треугольника, которые выражают то, что $\approx$ является «модулем» над $\sim$ с обеих сторон.

Доказательство. Определим $\approx : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c} \to \mathbb{Q}_+ \to \mathsf{Prop}$ с помощью двойной $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии. Сначала применим $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсию с ко-областью в подмножестве $\mathbb{R}_\mathsf{c} \to \mathbb{Q}_+ \to \mathsf{Prop}$, состоящий из тех семейств предикатов, которые округлены и удовлетворяют одной подходящей форме неравенства треугольника. Думая об этих предикатах как о половине бинарного отношения, запишем их как $(u, \epsilon) \mapsto (\diamondsuit \approx_\epsilon u)$, где символ $\diamondsuit$ относится ко всему отношению. Теперь можно записать A точно так, как

$$\begin{aligned} A :\equiv \Bigg\{\diamondsuit : \mathbb{R}_\mathsf{c} \to \mathbb{Q}_+ \to \mathsf{Prop} \Bigg| \\ &\Big(\forall (u : \mathbb{R}_\mathsf{c}).\, \forall (\epsilon : \mathbb{Q}_+).\, \big((\diamondsuit \approx_\epsilon u) \Leftrightarrow \exists (\theta : \mathbb{Q}_+).\, (\diamondsuit \approx_{\epsilon-\theta} u)\big)\Big) \\ &\wedge \Big(\forall (u, v : \mathbb{R}_\mathsf{c}).\, \forall (\eta, \epsilon : \mathbb{Q}_+).\, (u \sim_\epsilon v) \to \\ &\big((\diamondsuit \approx_\eta u) \to (\diamondsuit \approx_{\eta+\epsilon} v)\big) \wedge \big((\diamondsuit \approx_\eta v) \to (\diamondsuit \approx_{\eta+\epsilon} u)\big)\Big)\Bigg\} \end{aligned}$$

Как обычно, при работе с подмножествами будем использовать те же обозначения для обитателя в A и его первого компонента $\diamondsuit$. В качестве семейства соотношений, необходимых для $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии, рассмотрим следующее (что обеспечит другую форму неравенства треугольника):

$$(\diamondsuit \frown_\epsilon \heartsuit) :\equiv \forall (u : \mathbb{R}_\mathsf{c}).\, \forall (\eta : \mathbb{Q}_+).\, ((\diamondsuit \approx_\eta u) \to (\heartsuit \approx_{\epsilon+\eta} u)) \wedge ((\heartsuit \approx_\eta u) \to (\diamondsuit \approx_{\epsilon+\eta} u)).$$

Заметим, что эти отношения разделены. Для предположения $\forall (\epsilon : \mathbb{Q}_+).\, (\diamondsuit \frown_\epsilon \heartsuit)$, чтобы показать, что $\diamondsuit = \heartsuit$, достаточно показать $(\diamondsuit \approx_\epsilon u) \Leftrightarrow (\heartsuit \approx_\epsilon u)$, для всех $u : \mathbb{R}_\mathsf{c}$. Но $\diamondsuit \approx_\epsilon u$ подразумевает $\diamondsuit \approx_{\epsilon-\theta} u$, для некоторого θ, по округленности, что вместе с $\diamondsuit \frown_\epsilon \heartsuit$ влечет $\heartsuit \approx_\epsilon u$; обратное идентично.

Теперь, первыми двумя данными, требуемыми принципом рекурсии, являются следующие.

- Для любого $q : \mathbb{Q}$, надо указать элемент в A, который обозначим $(\mathsf{rat}(q) \approx_{(-)} -)$.
- Для любой аппроксимации Коши x, если предположить, что задана функция $\mathbb{Q}_+ \to A$, которую обозначим как $\epsilon \mapsto (x_\epsilon \approx_{(-)} -)$, с тем свойством, что

$$\forall (u : \mathbb{R}_\mathsf{c}).\, \forall (\delta, \epsilon, \eta : \mathbb{Q}_+).\, (x_\delta \approx_\eta u) \to (x_\epsilon \approx_{\eta+\delta+\epsilon} u), \tag{11.3.24}$$

  мы должны указать элемент в A, который обозначим как $(\mathsf{lim}(x) \approx_{(-)} -)$.

В обоих случаях мы даем требуемое определение, используя вложенную $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсию с ко-областью, являющейся подмножеством в $\mathbb{Q}_+ \to \mathsf{Prop}$, состоящего из округленных семейств простых высказываний. Представляя эти высказывания как нулевые половинки бинарного отношения, запишем их как $\epsilon \mapsto (\bullet \approx_\epsilon \triangle)$, где символ $\triangle$ относится ко всему семейству. Теперь можно записать ко-область этих внутренних рекурсий как

$$C :\equiv \left\{\triangle : \mathbb{Q}_+ \to \mathsf{Prop} \,\middle|\, \forall (\epsilon : \mathbb{Q}_+).\, \Big((\bullet \approx_\epsilon \triangle) \Leftrightarrow \exists (\theta : \epsilon : \mathbb{Q}_+).\, (\bullet \approx_{\epsilon-\theta} \triangle)\Big)\right\}.$$

Примем искомое семейство отношений оставшейся частью неравенства треугольника:

$$(\triangle \smile_\epsilon \square) :\equiv \forall (\eta : \mathbb{Q}_+).\, ((\bullet \approx_\eta \triangle) \to (\bullet \approx_{\epsilon+\eta} \square)) \wedge ((\bullet \approx_\eta \square) \to (\bullet \approx_{\epsilon+\eta} \triangle)).$$

Эти отношения разделяются тем же аргументом, что и для $\frown$, с использованием округления всех элементов C.

Отметим, что если такая внутренняя рекурсия завершится успешно, она даст семейство предикатов $\diamondsuit : \mathbb{R}_\mathsf{c} \to \mathbb{Q}_+ \to \mathsf{Prop}$, которые округлены (поскольку их образ в $\mathbb{Q}_+ \to \mathsf{Prop}$ лежит в C) и удовлетворяют условию

$$\forall (u, v : \mathbb{R}_\mathsf{c}).\, \forall (\epsilon : \mathbb{Q}_+).\, (u \sim_\epsilon v) \to \big((\diamondsuit \approx_{(-)} u) \smile_\epsilon (\diamondsuit \approx_{(-)} u) \big)\,.$$

Расширяя определение $\smile$, получаем в точности третье условие принадлежности $\diamondsuit$ к A; так что, это именно то, что требовалось.

Именно здесь можно дать определения (11.3.17)–(11.3.20) в качестве первых двух предложений каждой из двух внутренних рекурсий, соответствующих рациональным точкам и пределам. В каждом случае необходимо проверить, что отношение округлено и, следовательно, лежит в C. В рационально-рациональном случае (11.3.17) это ясно, а в других случаях следует из индуктивной гипотезы (в (11.3.18) соответствующая индуктивная гипотеза состоит в том, что $(\mathsf{rat}(q) \approx_{(-)} y_\delta) : C$, тогда как в (11.3.19) и (11.3.20) это $(x_\delta \approx_{(-)} -) : A$).

Остальные данные под-рекурсий призваны показать, что (11.3.17)–(11.3.20) удовлетворяют неравенству треугольника справа относительно конструкторов $\sim$. Имеются восемь случаев — по четыре в каждой подрекурсии — соответствующих восьми возможным способам выбора u, v и w в (11.3.22) в качестве рациональных точек или пределов. Сначала рассмотрим случаи, когда u есть $\mathsf{rat}(q)$.

(i) Предполагая $\mathsf{rat}(q) \approx_\phi \mathsf{rat}(r)$ и $-\epsilon < |r - s| < \epsilon$, надо показать, что $\mathsf{rat}(q) \approx_{\phi+\epsilon} \mathsf{rat}(s)$. Но, по определению $\approx$, это сводится к неравенству треугольника для рациональных чисел.

(ii) Предположим, что $\phi, \epsilon, \delta : \mathbb{Q}_+$ такие, что $\mathsf{rat}(q) \approx_\phi \mathsf{rat}(r)$ и $rat(r) \sim_{\epsilon-\delta} y_\delta$, и, по индукции, что

$$\forall (\psi : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi \mathsf{rat}(r)) \to (\mathsf{rat}(q) \approx_{\psi+\epsilon-\delta} y_\delta)\,. \tag{11.3.25}$$

Также предполагаем, что $\psi, \delta \mapsto (\mathsf{rat}(q) \approx_\psi y_\delta)$ является аппроксимацией Коши относительно $\smile$, т.е.

$$\forall (\psi, \xi, \zeta : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi y_\xi) \to (\mathsf{rat}(q) \approx_{\psi+\xi+\zeta} y_\zeta)\,, \tag{11.3.26}$$

хотя нам не нужно это предположение в данном случае. В самом деле, (11.3.25) с $\psi :\equiv \phi$ немедленно дает $\mathsf{rat}(q) \approx_{\phi+\epsilon-\delta} y_\delta$, и, следовательно, $\mathsf{rat}(q) \approx_{\phi+\epsilon} \mathsf{lim}(y)$, по определению $\approx$.

(iii) Предположим, что $\phi, \epsilon, \delta : \mathbb{Q}_+$ такие, что $\mathsf{rat}(q) \approx_\phi \mathsf{lim}(y)$ и $y_\delta \sim_{\epsilon-\delta} \mathsf{rat}(r)$, и, по индукции, что

$$\forall (\psi : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi y_\delta) \to (\mathsf{rat}(q) \approx_{\psi+\epsilon-\delta} \mathsf{rat}(r))\,. \tag{11.3.27}$$

$$\forall (\psi, \xi, \zeta : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi y_\xi) \to (\mathsf{rat}(q) \approx_{\psi+\xi+\zeta} y_\zeta)\,. \tag{11.3.28}$$

По определению, $\mathsf{rat}(q) \approx_\phi \mathsf{lim}(y)$ означает, что мы имеем $\theta : \mathbb{Q}_+$ с $rat(q) \approx_{\phi-\theta} y_\theta$. Следовательно, по предположению (11.3.28) имеем также $\mathsf{rat}(q) \approx_{\phi+\delta} y_\delta$, и тогда из (11.3.27) следует, что $\mathsf{rat}(q) \approx_{\phi+\epsilon} \mathsf{rat}(r)$, что и требовалось.

(iv) Предположим, что $\phi, \epsilon, \delta, \eta : \mathbb{Q}_+$ такие, что $\mathsf{rat}(q) \approx_\phi \mathsf{lim}(y)$ и $y_\delta \sim_{\epsilon-\delta-\eta} z_\eta$, и, по индукции, что

$$\forall (\psi : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi y_\delta) \to (\mathsf{rat}(q) \approx_{\psi+\epsilon-\delta-\eta} z_\eta)\,, \tag{11.3.29}$$

$$\forall (\psi, \xi, \zeta : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi y_\xi) \to (\mathsf{rat}(q) \approx_{\psi+\xi+\zeta} y_\zeta)\,, \tag{11.3.30}$$

$$\forall (\psi, \xi, \zeta : \mathbb{Q}_+).\, (\mathsf{rat}(q) \approx_\psi z_\xi) \to (\mathsf{rat}(q) \approx_{\psi+\xi+\zeta} z_\zeta)\,. \tag{11.3.31}$$

Опять же, $\mathsf{rat}t(q) \approx_\phi \lim(y)$ означает, что имеет место $\xi : \mathbb{Q}_+$ с $\mathsf{rat}(q) \approx_{\phi-\xi} y_\xi$, тогда как (11.3.30) подразумевает $\mathsf{rat}(q) \approx_{\phi+\delta} y_\delta$ а (11.3.29) подразумевает $\mathsf{rat}(q) \approx_{\phi+\epsilon-\eta} z_\eta$. Но по определению $\approx$, это подразумевает $\mathsf{rat}(q) \approx_{\phi+\epsilon} \lim(z)$, что и требовалось.

Теперь перейдем к случаям, когда u есть $\lim(x)$, где x — аппроксимация Коши. В этом случае гипотеза объемлющей индуктивности определения $(\lim(x) \approx_{(-)} -) : A$ состоит в том, что имеем $(x_\delta \approx_{(-)} -) : A$, так что помимо того, что они округлены, они удовлетворяют неравенству треугольника справа.

(v) Предполагая $\lim(x) \approx_\phi \mathsf{rat}(r)$ и $-\epsilon < |r - s| < \epsilon$, надо показать, что $\lim(x) \approx_{\phi+\epsilon} \mathsf{rat}(s)$. По определению $\approx$, первое означает, что $x_\delta \approx_{\phi-\delta} \mathsf{rat}(r)$, так что приведенное выше неравенство треугольника подразумевает $x_\delta \approx_{\epsilon+\phi-\delta} \mathsf{rat}(s)$, следовательно, $\lim(x) \approx_{\phi+\epsilon} \mathsf{rat}(s)$, что и требовалось.

(vi) Предположим, что имеются $\phi, \epsilon, \delta : \mathbb{Q}_+$ такие, что $\lim(x) \approx_\phi \mathsf{rat}(r)$ и $\mathsf{rat}(r) \sim_{\epsilon-\delta} y_\delta$, и две необязательные индуктивные гипотезы. По определению, имеется $\eta : \mathbb{Q}_+$ такое, что $x_\eta \approx_{\phi-\eta} \mathsf{rat}(r)$, поэтому индуктивное неравенство треугольника дает $x_\eta \approx_{\phi+\epsilon-\eta-\delta} y_\delta$. Тогда определение $\approx$ сразу дает $\lim(x) \approx_{\phi+\epsilon} \lim(y)$.

(vii) Предположим, что имеются $\phi, \epsilon, \delta : \mathbb{Q}_+$ такие, что $\lim(x) \approx_\phi \lim(y)$ и $y_\delta \sim_{\epsilon-\delta} rat(r)$, и две необязательные индуктивные гипотезы. По определению, имеются $\xi, \theta : \mathbb{Q}_+$ такие, что $x_\xi \approx_{\phi-\xi-\theta} y_\theta$. Поскольку y является приближением Коши, имеем $y_\theta \sim_{\theta+\delta} y_\delta$, поэтому индуктивное неравенство треугольника дает $x_\xi \approx_{\phi+\delta-\xi} y_\delta$, а затем $x_\xi \sim_{\phi+\epsilon-\xi} \mathsf{rat}(r)$. Определение $\approx$ дает $\lim(x) \approx_{\phi+\epsilon} \mathsf{rat}(r)$, что и требовалось.

(viii) Наконец, предположим, что имеются $\phi, \epsilon, \delta, \eta : \mathbb{Q}_+$ такие, что $\lim(x) \approx_\phi \lim(y)$ и $y_\delta \sim_{\epsilon-\delta-\eta} z_\eta$. Тогда, как и выше, имеются $\xi, \theta : \mathbb{Q}_+$ с $x_\xi \approx_{\phi-\xi-\theta} y_\theta$, и два применения неравенства треугольника, как выше.

На этом завершаются две внутренние рекурсии и, таким образом, определения семейств отношений $(\mathsf{rat}(q) \approx_{(-)} -)$ и $(\lim(x) \approx_{(-)} -)$. Поскольку все они являются элементами в A, они округлены и удовлетворяют неравенству треугольника справа относительно $\sim$. Остается проверить условия, относящиеся к $\frown$, то есть, что эти отношения удовлетворяют неравенству треугольника *слева* относительно конструкторов $\sim$. Четыре случая соответствуют четырем выборам рациональных или предельных точек для u и v в (11.3.23), и, поскольку все они являются простыми высказываниями, можно применить $\mathbb{R}_\mathsf{c}$-индукцию и предположить, что w также является либо рациональным, либо предельным. Это дает еще восемь случаев, доказательства которых по существу идентичны только что приведенным. □

Теперь можно доказать следующее.

Теорема 11.3.32. *Для любых $u, v : \mathbb{R}_\mathsf{c}$ и $\epsilon : \mathbb{Q}_+$ имеет место $(u \sim_\epsilon v) = (u \approx_\epsilon v)$.*

Доказательство. Поскольку обе части равенства являются простыми высказываниями, достаточно доказать двунаправленную импликацию. Для направления слева направо используем $\sim$-индукцию, примененную к $C(u, v, \epsilon) :\equiv (u \approx_\epsilon v)$. Таким образом, достаточно рассмотреть четыре конструктора $\sim$. В каждом случае, u и v специализированы либо к рациональным точкам, либо к пределам, так что определение $\approx$ обеспечивает оценивание, и всегда применяется индуктивная гипотеза.

Для направления справа налево используем $\mathbb{R}_c$-индукцию, чтобы предположить, что u и v являются рациональными точками или пределами, что позволяет использовать $\approx$ для оценивания. Но теперь определения $\approx$, и индуктивные гипотезы предоставляют именно те данные, которые требуются для соответствующих конструкторов $\sim$. □

С натяжкой, можно назвать $\approx$ расслоением «кодов» для $\sim$, причем двумя направлениями приведенного выше доказательства являются encode и decode, соответственно. По определению $\approx$, из теоремы 11.3.32 получаем эквивалентности

$$\begin{aligned}
(\mathsf{rat}(q) \approx_\epsilon \mathsf{rat}(r)) &:\equiv (-\epsilon < q - r < \epsilon)\\
(\mathsf{rat}(q) \approx_\epsilon \mathsf{lim}(y)) &:\equiv \exists(\delta : \mathbb{Q}_+).\, \mathsf{rat}(q) \approx_{\epsilon-\delta} y_\delta\\
(\mathsf{lim}(x) \approx_\epsilon \mathsf{rat}(r)) &:\equiv \exists(\delta : \mathbb{Q}_+).\, x_\delta \approx_{\epsilon-\delta} \mathsf{rat}(r)\\
(\mathsf{lim}(x) \approx_\epsilon \mathsf{lim}(y)) &:\equiv \exists(\delta, \eta : \mathbb{Q}_+).\, x_\delta \approx_{\epsilon-\delta-\eta} y_\eta\,.
\end{aligned}$$

Наше доказательство также предоставляет следующую дополнительную информацию.

Следствие 11.3.33. *Отношение $\sim$ округлено и удовлетворяет неравенству треугольника*:

$$(u \sim_\epsilon v) \simeq \exists(\theta : \mathbb{Q}_+).\, u \sim_{\epsilon-\theta} v \tag{11.3.34}$$

$$(u \sim_\epsilon v) \to (v \sim_\delta w) \to (u \sim_{\epsilon+\delta} w)\,. \tag{11.3.35}$$

Имея в качестве инструмента неравенство треугольника, можно показать, что «пределы» аппроксимаций Коши на самом деле ведут себя как пределы.

Лемма 11.3.36. *Для любых $u : \mathbb{R}_c$, аппроксимации Коши y, и $\epsilon, \delta : \mathbb{Q}_+$, если $u \sim_\epsilon y_\delta$, то $u \sim_{\epsilon+\delta} \mathsf{lim}(y)$.*

Доказательство. Мы используем $\mathbb{R}_c$-индукцию по u. Если u есть $\mathsf{rat}(q)$, то это в точности второй конструктор $\sim$. Теперь предположим, что u есть $\mathsf{lim}(x)$, и каждый x_η обладает тем свойством, что для любых y, ϵ, δ, если $x_\eta \sim_\epsilon y_\delta$, то $x_\eta \sim_{\epsilon+\delta} \mathsf{lim}(y)$. В частности, принимая в этом предположении $y :\equiv x$ и $\delta :\equiv \eta$, заключаем, что $x_\eta \sim_{\eta+\theta} \mathsf{lim}(x)$, для любых $\eta, \theta : \mathbb{Q}_+$.

Пусть теперь y, ϵ, δ произвольны, и пусть $\mathsf{lim}(x) \sim_\epsilon y_\delta$. По округленности, существует θ такое, что $\mathsf{lim}(x) \sim_{\epsilon-\theta} y_\delta$. Тогда, согласно приведенному выше наблюдению, для любого η, имеем $x_\eta \sim_{\eta+\theta/2} \mathsf{lim}(x)$, и, следовательно, $x_\eta \sim_{\epsilon+\eta-\theta/2} y_\delta$, по неравенству треугольника. Следовательно, четвертый конструктор $\sim$ дает $\mathsf{lim}(x) \sim_{\epsilon+2\eta+\delta-\theta/2} \mathsf{lim}(y)$. Таким образом, если выбрать $\eta :\equiv \theta/4$, то результат будет, как и ожидалось. □

Лемма 11.3.37. *Для любой аппроксимации Коши y и любых $\delta, \eta : \mathbb{Q}_+$, имеем $y_\delta \sim_{\delta+\eta} \mathsf{lim}(y)$.*

Доказательство. Возьмем $u :\equiv y_\delta$ и $\epsilon :\equiv \eta$ из предыдущей леммы. □

Замечание 11.3.38. Можно было бы ожидать, что $y_\delta \sim_\delta \mathsf{lim}(y)$, но в примерах это не работает. Например, рассмотрим x, определенный как $x_\epsilon :\equiv \epsilon$. Его предел явно равен 0, но $|\epsilon - 0| < \epsilon$ не верно, а работает только с $\leqslant$.

В качестве приложения, лемма 11.3.37 позволяет показать, что расширения функций Лифшица из леммы 11.3.15 единственны.

Лемма 11.3.39. *Пусть $f, g : \mathbb{R}_c \to \mathbb{R}_c$ и f непрерывна в том смысле, что*

$$\forall(u : \mathbb{R}_c).\, \forall(\epsilon : \mathbb{Q}_+).\, \exists(\delta : \mathbb{Q}_+).\, \forall(v : \mathbb{R}_c).\, (u \sim_\delta v) \to (f(u) \sim_\epsilon f(v))\,,$$

и аналогично для g. Если $f(\mathsf{rat}(q)) = g(\mathsf{rat}(q))$, для всех $q : \mathbb{Q}$, то $f = g$.

Доказательство. Докажем $f(u) = g(u)$, для всех u, по $\mathbb{R}_\mathsf{c}$-индукции. Рациональный случай — это всего лишь гипотеза. Итак, предположим, что $f(x_\delta) = g(x_\delta)$, для всех δ. Покажем, что $f(\mathsf{lim}(x)) \sim_\epsilon g(\mathsf{lim}(x))$, для всех ϵ, так что применяется конструктор пути $\mathbb{R}_\mathsf{c}$.

Поскольку f и g непрерывны, существуют θ, η такие, что, для всех v, имеем

$$(\mathsf{lim}(x) \sim_\theta v) \to (f(\mathsf{lim}(x)) \sim_{\epsilon/2} f(v))$$
$$(\mathsf{lim}(x) \sim_\eta v) \to (g(\mathsf{lim}(x)) \sim_{\epsilon/2} g(v))\,.$$

Выбирая $\delta < \min(\theta, \eta)$, по лемме 11.3.37 имеем, как $\mathsf{lim}(x) \sim_\theta y_\delta$, так и $\mathsf{lim}(x) \sim_\eta y_\delta$. Следовательно

$$f(\mathsf{lim}(x)) \sim_{\epsilon/2} f(y_\delta) = g(y_\delta) \sim_{\epsilon/2} g(\mathsf{lim}(x))$$

и, таким образом, $f(\mathsf{lim}(x)) \sim_\epsilon g(\mathsf{lim}(x))$, по неравенству треугольника. □

11.3.3 Алгебраическая структура действительных чисел Коши

Сначала определим аддитивную структуру $(\mathbb{R}_\mathsf{c}, 0, +, -)$. Ясно, что аддитивный единичный элемент 0 — это просто $\mathsf{rat}(0)$, а аддитивный обратный $- : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$ получается как расширение аддитивного обратного $- : \mathbb{Q} \to \mathbb{Q}$, используя лемму 11.3.15 с постоянной Лифшица 1. Для определения сложения придется приложить определенные усилия.

Лемма 11.3.40. *Предположим, что $f : \mathbb{Q} \times \mathbb{Q} \to \mathbb{Q}$, для всех $q, r, s : \mathbb{Q}$,*

$$|f(q, s) - f(r, s)| \leqslant |q - r| \qquad \text{и} \qquad |f(q, r) - f(q, s)| \leqslant |r - s|\,.$$

Тогда существует функция $\bar f : \mathbb{R}_\mathsf{c} \times \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$ такая, что $\bar f(\mathsf{rat}(q), \mathsf{rat}(r)) = f(q, r)$, для всех $q, r : \mathbb{Q}$. Кроме того, для всех $u, v, w : \mathbb{R}_\mathsf{c}$ и $q : \mathbb{Q}_+$, верно

$$u \sim_\epsilon v \Rightarrow \bar f(u, w) \sim_\epsilon \bar f(v, w) \quad \text{и} \quad v \sim_\epsilon w \Rightarrow \bar f(u, v) \sim_\epsilon \bar f(u, w)\,.$$

Доказательство. Будем использовать $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсию для построения каррированной формы $\bar f$ в виде отображения $\mathbb{R}_\mathsf{c} \to A$, где A — пространство нерасширяющихся действительных функций:

$$A :\equiv \{h : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c} \mid \forall(\epsilon : \mathbb{Q}_+).\, \forall(u, v : \mathbb{R}_\mathsf{c}).\, u \sim_\epsilon v \Rightarrow h(u) \sim_\epsilon h(v)\}\,.$$

Нам также понадобится подходящее $\frown_\epsilon$ на A, которую определим как

$$(h \frown_\epsilon k) :\equiv \forall(u : \mathbb{R}_\mathsf{c}).\, h(u) \sim_\epsilon k(u)\,.$$

Ясно, что если $\forall(\epsilon : \mathbb{Q}_+).\, h \frown_\epsilon k$, то $h(u) = k(u)$, для всех all $u : \mathbb{R}_\mathsf{c}$, поэтому $\frown$ является отделимым.

Для базового случая определим $\bar f(\mathsf{rat}(q)) : A$, где $q : \mathbb{Q}$, как расширение отображения Лифшица $\lambda r.\, f(q, r)$ от $\mathbb{Q} \to \mathbb{Q}$ к $\mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$, как построено в лемме 11.3.15 с постоянной Лифшица 1. Далее, для аппроксимации Коши x определим $\bar f(\mathsf{lim}(x)) : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$ как

$$\bar f(\mathsf{lim}(x))(v) :\equiv \mathsf{lim}(\lambda\epsilon.\, \bar f(x_\epsilon)(v))\,.$$

Чтобы это определение было корректным, $\lambda\epsilon.\, \bar f(x_\epsilon)(v)$ должно быть аппроксимацией Коши, поэтому возьмем произвольные $\delta, \epsilon : \mathbb{Q}$. Тогда, по предположению, $\bar f(x_\delta) \frown_{\delta+\epsilon} \bar f(x_\epsilon)$, следовательно,

$\bar{f}(x_\delta)(v) \sim_{\delta+\epsilon} \bar{f}(x_\epsilon)(v)$. Кроме того, $\bar{f}(\mathsf{lim}(x))$ не является расширяемым, потому что таково $\bar{f}(x_\epsilon)$, по предположению индукции. В самом деле, если $u \sim_\epsilon v$, то, для всех $\epsilon : \mathbb{Q}$,

$$\bar{f}(x_{\epsilon/3})(u) \sim_{\epsilon/3} \bar{f}(x_{\epsilon/3})(v)\,,$$

поэтому $\bar{f}(\mathsf{lim}(x))(u) \sim_\epsilon \bar{f}(\mathsf{lim}(x))(v)$, в связи с четвертым конструктором $\sim$.

Осталось проверить еще четыре условия, проиллюстрируем это только на одном. Предположим, что $\epsilon : \mathbb{Q}_+$, и для некоторого $\delta : \mathbb{Q}_+$ имеем $\mathsf{rat}(q) \sim_{\epsilon-\delta} y_\delta$ и $\bar{f}(\mathsf{rat}(q)) \frown_{\epsilon-\delta} \bar{f}(y_\delta)$. Чтобы показать, что $\bar{f}(\mathsf{rat}(q)) \frown_\epsilon \bar{f}(\mathsf{lim}(y))$, возьмем произвольный $v : \mathbb{R}_\mathsf{c}$ и заметим, что

$$\bar{f}(\mathsf{rat}(q))(v) \sim_{\epsilon-\delta} \bar{f}(y_\delta)(v)\,.$$

Следовательно, по второму конструктору $\sim$, имеем $\bar{f}(\mathsf{rat}(q))(v) \sim_\epsilon \bar{f}(\mathsf{lim}(y))(v)$,что и требовалось. □

Лемму 11.3.40 можно применить к любой рациональной функции двух переменных, которая не расширяется отдельно по каждой переменной. Сложение является такой функцией, поэтому имеем $+ : \mathbb{R}_\mathsf{c} \times \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$. Кроме того, расширение уникально, пока требуется, чтобы оно не касалось отдельной переменной, и, как в случае с одной переменной, тождества на рациональных числах расширяются до тождеств на действительных числах. Поскольку композиция не-расширяющихся отображений снова является не-расширяющейся, можно заключить, что сложение удовлетворяет обычным свойствам, таким как коммутативность и ассоциативность. Следовательно, $(\mathbb{R}_\mathsf{c}, 0, +, -)$ — коммутативная группа.

Также можно применить лемму 11.3.40 к функциям $\min : \mathbb{R}_\mathsf{c}\times\mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$ и $\max : \mathbb{R}_\mathsf{c}\times\mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$, что превращает $\mathbb{R}_\mathsf{c}$ в решетку. Частичный порядок $\leqslant$ на $\mathbb{R}_\mathsf{c}$ определяется через $\max$ как

$$(u \leqslant v) :\equiv (\max(u,v) = v)\,.$$

Отношение $\leqslant$ является частичным порядком, поскольку оно таково на $\mathbb{Q}$, а аксиомы частичного порядка выражаются в виде уравнений через $\min$ и $\max$, поэтому они переносятся в $\mathbb{R}_\mathsf{c}$.

Другая функция, расширяемая до $\mathbb{R}_\mathsf{c}$ тем же методом, — это абсолютное значение, $|{-}|$. Опять же, оно обладает ожидаемыми свойствами, поэтому что они переходят из $\mathbb{Q}$ в $\mathbb{R}_\mathsf{c}$.

Используя $\leqslant$, можно получить строгий порядок $<$ посредством

$$(u < v) :\equiv \exists(q, r : \mathbb{Q}).\, (u \leqslant \mathsf{rat}(q)) \wedge (q < r) \wedge (\mathsf{rat}(r) \leqslant v)\,.$$

То есть, $u < v$ выполняется, когда просто существует пара рациональных чисел q, r с $q < r$ такая, что $x \leqslant \mathsf{rat}(q)$ и $\mathsf{rat}(r) \leqslant v$. Нетрудно проверить, что $<$ является не-рефлексивным и транзитивным и имеет другие свойства, ожидаемые для упорядоченного поля. Принцип Архимеда следует непосредственно из определения $<$.

Теорема 11.3.41 (Принцип Архимеда для $\mathbb{R}_\mathsf{c}$)**.** *Для любых $u, v : \mathbb{R}_\mathsf{c}$ таких, что $u < v$, просто существует $q : \mathbb{Q}$ такое, что $u < q < v$.*

Доказательство. Из $u < v$ мы просто получаем $r, s : \mathbb{Q}$ такое, что $u \leqslant r < s \leqslant v$, и можно взять $q :\equiv (r+s)/2$. □

Теперь достаточно структурированности на $\mathbb{R}_\mathsf{c}$, чтобы выразить $u \sim_\epsilon v$ со стандартными понятиями.

Лемма 11.3.42. *Если $q : \mathbb{Q}$ и $u : \mathbb{R}_\mathsf{c}$ удовлетворяют $u \leqslant \mathsf{rat}(q)$, то для любых $v : \mathbb{R}_\mathsf{c}$ и $\epsilon : \mathbb{Q}_+$, если $u \sim_\epsilon v$, то $v \leqslant \mathsf{rat}(q+\epsilon)$.*

Доказательство. Заметим, что функция $\max(\mathsf{rat}(q), -) : \mathbb{R}_c \to \mathbb{R}_c$ является функцией Лифшица с константой 1. Сначала рассмотрим случай, когда $u = \mathsf{rat}(r)$ рационально. Для этого воспользуемся индукцией по v. Если v рационально, то утверждение очевидно. Если v есть $\mathsf{lim}(y)$, индуктивно предполагаем, что для любых ϵ, δ, если $\mathsf{rat}(r) \sim_\epsilon y_\delta$, то $y_\delta \leqslant \mathsf{rat}(q+\epsilon)$, т.е. $\max(\mathsf{rat}(q+\epsilon), y_\delta) = \mathsf{rat}(q+\epsilon)$.

Теперь, предполагая ϵ и $\mathsf{rat}(r) \sim_\epsilon \mathsf{lim}(y)$, имеем θ такое, что $\mathsf{rat}(r) \sim_{\epsilon-\theta} \mathsf{lim}(y)$, следовательно, $\mathsf{rat}(r) \sim_\epsilon y_\delta$ всякий раз, когда $\delta < \theta$. Таким образом, индуктивная гипотеза дает $\max(\mathsf{rat}(q+\epsilon), y_\delta) = \mathsf{rat}(q+\epsilon)$, для таких δ. Но по определению,

$$\max(\mathsf{rat}(q+\epsilon), \mathsf{lim}(y)) \equiv \mathsf{lim}(\lambda\delta.\ \max(\mathsf{rat}(q+\epsilon), y_\delta))\,.$$

Поскольку предел постоянной, в конечном счете, аппроксимации Коши является константой, имеем

$$\max(\mathsf{rat}(q+\epsilon), \mathsf{lim}(y)) = \mathsf{rat}(q+\epsilon)\,,$$

следовательно, $\mathsf{lim}(y) \leqslant \mathsf{rat}(q+\epsilon)$.

Теперь рассмотрим общий случай с $u : \mathbb{R}_c$. Поскольку $u \leqslant \mathsf{rat}(q)$ означает $\max(\mathsf{rat}(q), u) = \mathsf{rat}(q)$, предположение $u \sim_\epsilon v$ и свойство Лифшица $\max(\mathsf{rat}(q), -)$ подразумевают $\max(\mathsf{rat}(q), v) \sim_\epsilon \mathsf{rat}(q)$. Таким образом, поскольку $\mathsf{rat}(q) \leqslant \mathsf{rat}(q)$, первый случай подразумевает $\max(\mathsf{rat}(q), v) \leqslant \mathsf{rat}(q+\epsilon)$, и, следовательно, $v \leqslant \mathsf{rat}(q+\epsilon)$, в силу транзитивности $\leqslant$. □

Лемма 11.3.43. *Предположим, что $q : \mathbb{Q}$ и $u : \mathbb{R}_c$ удовлетворяют $u < \mathsf{rat}(q)$. Тогда*:

(i) *для любых $v : \mathbb{R}_c$ и $\epsilon : \mathbb{Q}_+$, если $u \sim_\epsilon v$, то $v < \mathsf{rat}(q+\epsilon)$,*

(ii) *существует $\epsilon : \mathbb{Q}_+$ такое, что для любого $v : \mathbb{R}_c$, если $u \sim_\epsilon v$, то имеем $v < \mathsf{rat}(q)$.*

Доказательство. По определению, $u < \mathsf{rat}(q)$ означает, что существуют $r : \mathbb{Q}$ с $r < q$ и $u \leqslant \mathsf{rat}(r)$. Тогда по лемме 11.3.42, для любого ϵ, если $u \sim_\epsilon v$, то $v \leqslant \mathsf{rat}(r+\epsilon)$. Заключение (i) следует сразу, так как $r + \epsilon < q + \epsilon$, а для (ii) можно взять любое $\epsilon < q - r$. □

Теперь можно показать, что вспомогательное отношение $\sim$ является именно тем, чем мы его считаем.

Теорема 11.3.44. $(u \sim_\epsilon v) \simeq (|u - v| < \mathsf{rat}(\epsilon))$ *для всех $u, v : \mathbb{R}_c$ и $\epsilon : \mathbb{Q}_+$.*

Доказательство. Свойства Лифшица для вычитания и абсолютного значения подразумевают, что если $u \sim_\epsilon v$, то $|u - v| \sim_\epsilon |u - u| = 0$. Таким образом, для направления слева направо достаточно показать, что если $u \sim_\epsilon 0$, то $|u| < \mathsf{rat}(\epsilon)$. Сделаем это $\mathbb{R}_c$-индукцией по u.

Если u рационально, утверждение следует сразу же, поскольку абсолютное значение и порядок расширяются на $\mathbb{Q}_+$. Если u есть $\mathsf{lim}(x)$, то, в силу округления, имеем $\theta : \mathbb{Q}_+$ с $\mathsf{lim}(x) \sim_{\epsilon-\theta} 0$. Таким образом, по неравенству треугольника имеем $x_{\theta/3} \sim_{\epsilon-2\theta/3} 0$, поэтому индуктивная гипотеза дает $\left|x_{\theta/3}\right| < \mathsf{rat}(\epsilon - 2\theta/3)$. Но $x_{\theta/3} \sim_{2\theta/3} \mathsf{lim}(x)$, следовательно, $\left|x_{\theta/3}\right| \sim_{2\theta/3} |\mathsf{lim}(x)|$, по свойству Лифшица, поэтому из леммы 11.3.43(i) следует $\mathsf{lim}(x) < \mathsf{rat}(\epsilon)$.

В другом направлении используем $\mathbb{R}_c$-индукцию по u и v. Если обе переменные рациональны, это есть первый конструктор $\sim$.

Если u есть $\mathsf{rat}(q)$, а v — $\mathsf{lim}(y)$, индуктивно предполагаем, что для любых ϵ, δ, если $|\mathsf{rat}(q) - y_\delta| < \mathsf{rat}(\epsilon)$, то $\mathsf{rat}(q) \sim_\epsilon y_\delta$. Зафиксируем ϵ так, чтобы было $|\mathsf{rat}(q) - \mathsf{lim}(y)| < \mathsf{rat}(\epsilon)$. Поскольку $\mathbb{Q}$ порядково-плотно в $\mathbb{R}_c$, существует $\theta < \epsilon$ с $|\mathsf{rat}(q) - \mathsf{lim}(y)| < \mathsf{rat}(\theta)$. Теперь, для любых δ, η, имеем $\mathsf{lim}(y) \sim_{2\delta} y_\delta$, следовательно, по свойству Лифшица

$$|\mathsf{rat}(q) - \mathsf{lim}(y)| \sim_{\delta+\eta} |\mathsf{rat}(q) - y_\delta|\ .$$

Таким образом, по лемме 11.3.43(i) имеем $|\mathsf{rat}(q) - y_\delta| < \mathsf{rat}(\theta + 2\delta)$. Так что, по индуктивной гипотезе, $\mathsf{rat}(q) \sim_{\theta+2\delta} y_\delta$, и, следовательно, $\mathsf{rat}(q) \sim_{\theta+4\delta} \mathsf{lim}(y)$, по неравенству треугольника. Тогда, достаточно выбрать $\delta :\equiv (\epsilon - \theta)/4$.

Доказательства остальных двух случаев полностью аналогичны. □

Далее, хотелось бы снабдить $\mathbb{R}_\mathsf{c}$ мультипликативной структурой. Для каждого $q : \mathbb{Q}$ отображение $r \mapsto q \cdot r$ является отображением Липшица с константой[1] $|q| + 1$, поэтому можем расширить его до умножения по q на действительные числа. Поэтому $\mathbb{R}_\mathsf{c}$ является векторным пространством над $\mathbb{Q}$. В общем, можно определить умножение действительных чисел как

$$u \cdot v :\equiv \tfrac{1}{2} \cdot \left((u + v)^2 - u^2 - v^2\right), \tag{11.3.45}$$

поэтому просто требуется операция «возведение в квадрат» $u \mapsto u^2$ как отображение $\mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$. Возведение в квадрат не является отображением Лифшица, но оно является таковым на каждой ограниченной области, что позволяет соединять их вместе. Определим открытые и закрытые интервалы как

$$[u, v] :\equiv \{x : \mathbb{R}_\mathsf{c} \mid u \leqslant x \leqslant v\} \qquad \text{и} \qquad (u, v) :\equiv \{x : \mathbb{R}_\mathsf{c} \mid u < x < v\}\ .$$

Хотя технически элемент $[u, v]$ или (u, v) является действительным числом Коши вместе с доказательством, поскольку последнее обитает в простом высказывании, оно неинтересно. Таким образом, как это обычно бывает с типами подмножеств, обычно записывают просто $x : [u, v]$ всякий раз, когда $x : \mathbb{R}_\mathsf{c}$ таково, что $u \leqslant x \leqslant v$, и аналогично для открытых и смешанных интервалов.

Теорема 11.3.46. *Существует единственная функция $(-)^2 : \mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{c}$, которая расширяет возведение в квадрат $q \mapsto q^2$ для рациональных чисел и удовлетворяет условию*

$$\forall (n : \mathbb{N}).\, \forall (u, v : [-n, n]).\ \left|u^2 - v^2\right| \leqslant 2 \cdot n \cdot |u - v|\ .$$

Доказательство. Сначала заметим, что, для любого $u : \mathbb{R}_\mathsf{c}$, просто существует $n : \mathbb{N}$ такое, что $-n \leqslant u \leqslant n$, см. упражнение 11.7, поэтому отображение

$$e : \left(\sum_{n:\mathbb{N}} [-n, n]\right) \to \mathbb{R}_\mathsf{c} \qquad \text{определенное как} \qquad e(n, x) :\equiv x$$

сюръективно. Далее, для каждого $n : \mathbb{N}$, отображение возведения в квадрат

$$s_n : \{q : \mathbb{Q} \mid -n \leqslant q \leqslant n\} \to \mathbb{Q} \qquad \text{определенное как} \qquad s_n(q) :\equiv q^2$$

является отображением Лифшица с постоянной $2n$, поэтому можно использовать лемму 11.3.15, чтобы расширить его до отображения $\bar{s}_n : [-n, n] \to \mathbb{R}_\mathsf{c}$ с постоянной Липшица $2n$, детали см. в упражнении 11.8. Отображения $\bar{s}_n$ совместимы: если $m < n$, для некоторых $m, n : \mathbb{N}$, то s_n, ограниченное $[-m, m]$, должно совпадать с s_m, потому что оба являются Лифшицевыми и, следовательно, непрерывны в смысле леммы 11.3.39. Следовательно, по теореме 10.1.5 отображение

$$\left(\sum_{n:\mathbb{N}} [-n, n]\right) \to \mathbb{R}_\mathsf{c}, \qquad \text{заданное как} \qquad (n, x) \mapsto s_n(x)\,,$$

факторизуется однозначно через $\mathbb{R}_\mathsf{c}$, для получения желаемой функции. □

[1] Мы определили константы Лифшица как положительные рациональные числа.

На данный момент имеются, кольцевая структура вещественных чисел и архимедов порядок. Чтобы обосновать $\mathbb{R}_c$ как архимедово упорядоченное поле, все еще требуются обратные.

Теорема 11.3.47. *Действительное число Коши обратимо тогда и только тогда, когда оно отлично от нуля.*

Доказательство. Во-первых, предположим, что $u : \mathbb{R}_c$ имеет обратный $v : \mathbb{R}_c$. По принципу Архимеда, существует $q : \mathbb{Q}$ такое, что $|v| < q$. Тогда $1 = |uv| < |u| \cdot v < |u| \cdot q$ и, следовательно, $|u| > 1/q$, то есть, что $u \# 0$.

Для обратного построения построим обратное отображение

$$(-)^{-1} : \{u : \mathbb{R}_c \mid u \# 0\} \to \mathbb{R}_c$$

путем склеивания функций, аналогично построению возведения в квадрат в теореме 11.3.46. Здесь наметим лишь основные шаги. Для каждого $q : \mathbb{Q}$ пусть

$$[q, \infty) :\equiv \{u : \mathbb{R}_c \mid q \leqslant u\} \qquad \text{и} \qquad (-\infty, q] :\equiv \{u : \mathbb{R}_c \mid u \leqslant -q\}\ .$$

Тогда, поскольку q ранжировано над $\mathbb{Q}_+$, типы $(-\infty, q]$ и $[q, \infty)$ совместно охватывают $\{u : \mathbb{R}_c \mid u \# 0\}$. На каждом таком $[q, \infty)$ и $(-\infty, q]$ обратная функция получается применением леммы 11.3.15 с постоянной Лифшица $1/q^2$. Наконец, теорема 10.1.5 гарантирует, что обратная функция однозначно факторизуется через $\{u : \mathbb{R}_c \mid u \# 0\}$. □

Мы резюмируем алгебраическую структуру $\mathbb{R}_c$ с помощью теоремы.

Теорема 11.3.48. *Действительные числа Коши образуют архимедово упорядоченное поле.*

11.3.4 Действительные числа Коши являются полными в смысле Коши

Мы построили $\mathbb{R}_c$, замыкая $\mathbb{Q}$ в пределах аппроксимаций Коши, поэтому лучше иметь случай, когда $\mathbb{R}_c$ является Коши-полным. Благодаря теореме 11.3.44 нет никакой разницы между аппроксимацией Коши $x : \mathbb{Q}_+ \to \mathbb{R}_c$, определенной при построении $\mathbb{R}_c$, и аппроксимацией Коши в смысле определения 11.2.10 (адаптированное к $\mathbb{R}_c$).

Таким образом, для данной аппроксимации Коши $x : \mathbb{Q}_+ \to \mathbb{R}_c$ вполне естественно ожидать, что $\mathsf{lim}(x)$ является ее пределом , где понятие предела определено, как в определении 11.2.10. Но это так, по теореме 11.3.44 и лемме 11.3.37. Так что, мы доказали:

Теорема 11.3.49. *Каждая аппроксимация Коши в $\mathbb{R}_c$ имеет предел.*

Архимедово упорядоченное поле, в котором каждая аппроксимация Коши имеет предел, называется **полным по Коши**. Действительные числа Коши — наименьшее такое поле.

Теорема 11.3.50. *Действительные числа Коши встраиваются в каждое полное архимедово упорядоченное поле Коши.*

Доказательство. Предположим, что F — полное по Коши архимедово упорядоченное поле. Поскольку пределы единственны, существует оператор lim, который переводит аппроксимации Коши из F в их пределы. Определим вложение $e : \mathbb{R}_c \to F$ с помощью $(\mathbb{R}_c, \sim)$-рекурсии как

$$e(\mathsf{rat}(q)) :\equiv q \qquad \text{и} \qquad e(\mathsf{lim}(x)) :\equiv \lim(e \circ x)\,.$$

Подходящим $\frown$ на F является

$$(a \frown_\epsilon b) :\equiv |a - b| < \epsilon\,.$$

Это — разделяемое отношение, поскольку F архимедово. Остальные пункты для $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии легко проверяются. Также нужно было бы проверить, что e является вложением упорядоченных полей, которое фиксирует рациональные числа. □

11.4 Сравнение действительных чисел Коши и Дедекинда

Теперь, немного о взаимоотношениях между действительными числами Коши и Дедекинда. По теореме 11.3.48 поле, $\mathbb{R}_\mathsf{c}$ является архимедовым упорядоченным полем, что также допустимо для Ω, и в этом легко убедиться (в случае, когда Ω является исходным σ-фреймом, требуется простая индукция, а в остальных случаях все получается непосредственно). Следовательно, по теореме 11.2.14, существует вложение упорядоченных полей

$$\mathbb{R}_\mathsf{c} \to \mathbb{R}_\mathsf{d}\,,$$

фиксирующее рациональные числа (мы могли бы получить это также из теорем 11.2.12 и 11.3.50). Вообще, мы не ожидаем, что $\mathbb{R}_\mathsf{c}$ и $\mathbb{R}_\mathsf{d}$ совпадут без дальнейших предположений.

Лемма 11.4.1. *Если для каждого $x : \mathbb{R}_\mathsf{d}$ просто существует*

$$c : \prod_{q,r:\mathbb{Q}} (q < r) \to (q < x) + (x < r)\,, \tag{11.4.2}$$

то действительные числа Коши и Дедекинда совпадают.

Доказательство. Заметим, что тип в (11.4.2) — это неусеченный вариант (11.2.3), в котором утверждается, что $<$ — слабый линейный порядок. Мы уже знаем, что $\mathbb{R}_\mathsf{c}$ вкладывается в $\mathbb{R}_\mathsf{d}$, поэтому достаточно показать, что каждое действительное число Дедекинда просто является пределом последовательности Коши рациональных чисел.

Возьмем произвольный $x : \mathbb{R}_\mathsf{d}$. По условию, просто существует c, как в утверждении леммы, а по заселенности сечений просто существуют $a, b : \mathbb{Q}$ такие, что $a < x < b$. Построим последовательность $f : \mathbb{N} \to \{(q,r) \in \mathbb{Q} \times \mathbb{Q} \mid q < r\}$, используя рекурсию.

(i) Установим, $f(0) :\equiv (a, b)$.

(ii) Предположим, что $f(n)$ уже определено как (q_n, r_n), так что $q_n < r_n$. Определим $s :\equiv (2q_n + r_n)/3$ и $t :\equiv (q_n + 2r_n)/3$. Тогда $c(s,t)$ выбирает между $s < x$ и $x < t$. Если $s < x$, то установим $f(n+1) :\equiv (s, r_n)$, иначе — $f(n+1) :\equiv (q_n, t)$.

Запишем (q_n, r_n) для n-го члена последовательности f. Тогда ясно, что $q_n < x < r_n$ и $|q_n - r_n| \leqslant (2/3)^n \cdot |q_0 - r_0|$, для всех $n : \mathbb{N}$. Следовательно, $q_0, q_1, \ldots$ и $r_0, r_1, \ldots$ являются последовательностями Коши, сходящимися к сечению Дедекинда x. Таким образом, показано, что для любого $x : \mathbb{R}_\mathsf{d}$, просто существует последовательность Коши, сходящаяся к x. □

Из этой леммы следует, что для совпадения $\mathbb{R}_\mathsf{c}$ и $\mathbb{R}_\mathsf{d}$ достаточно либо счетного выбора, либо исключения третьего..

Следствие 11.4.3. *Если исключение третьего или счетный выбор применимы, то $\mathbb{R}_\mathsf{c}$ и $\mathbb{R}_\mathsf{d}$ эквивалентны.*

Доказательство. Если имеет место исключение третьего, то можно доказать $(x < y) \to (x < z) + (z < y)$: либо $x < z$, либо $\neg(x < z)$. В первом случае доказательство завершается, а во втором получаем $z < y$, потому что $z \leqslant x < y$. Таким образом, получаем (11.4.2), так что можно применить лемму 11.4.1.

Предположим, что имеет место счетный выбор. Множество $S = \{(q, r) \in \mathbb{Q} \times \mathbb{Q} \mid q < r\}$ эквивалентно $\mathbb{N}$, поэтому можно применить счетный выбор к утверждению, что x локализовано,

$$\forall((q, r) : S).\, (q < x) \vee (x < r)\,.$$

Заметим, что $(q < x) \vee (x < r)$ выражено как экзистенциальное утверждение $\exists(b : \mathbf{2}).\, (b = 0_{\mathbf{2}} \to q < x) \wedge (b = 1_{\mathbf{2}} \to x < r)$. (Каррированная форма) функции выбора тогда в точности есть (11.4.2), так что лемма 11.4.1 снова применима. □

11.5 Компактность интервала

Мы уже указывали, что наши конструкции вещественных чисел полностью совместимы с классической логикой. Таким образом, приняв закон исключения третьего (3.4.1) и аксиому выбора (3.8.1), мы могли бы разработать классический анализ, который, по сути, был бы равносилен копированию любой стандартной книги по анализу.

Тем не менее, любому, кто интересуется вычислениями, например, численному аналитику, должно быть любопытно разработать анализ в значимой для вычислений среде. То, что анализ в конструктивной обстановке даже возможен, было продемонстрировано в [Bis67]. В качестве образца различий и сходств между классическим и конструктивным анализом мы кратко обсудим только одну тему — компактность интервала $[0, 1]$ и пару теорем, связанных с этим понятием.

Компактность не является исключением из общего феномена в конструктивной математике, когда классически эквивалентные понятия раздваиваются. Отметим три наиболее часто используемых понятия компактности:

(i) **метрическая компактность**: «полное по Коши и вполне ограниченное»,

(ii) **компактность Больцано-Вейерштрасса**: «каждая последовательность имеет сходящуюся подпоследовательность»,

(iii) **компактность Гейне-Бореля**: «каждое открытое покрытие имеет конечное подпокрытие».

Все они эквивалентны в классической математике. Посмотрим, как они проявляются в гомотопической теории типов. Мы можем использовать действительные числа Дедекинда или Коши, поэтому будем обозначать действительные числа просто как $\mathbb{R}$. Сначала напомним несколько основных определений.

Определение 11.5.1. Метрическое пространство (M, d) — это множество M с отображением $d : M \times M \to \mathbb{R}$, удовлетворяющим, для всех $x, y, z : M$, следующим свойствам:

$$d(x, y) \geqslant 0\,, \qquad d(x, y) = d(y, x)\,,$$
$$d(x, y) = 0 \Leftrightarrow x = y\,, \qquad d(x, z) \leqslant d(x, y) + d(y, z)\,.$$

Определение 11.5.2. Аппроксимация Коши в M — это последовательность $x : \mathbb{Q}_+ \to M$, удовлетворяющая условию:

$$\forall(\delta, \epsilon).\, d(x_\delta, x_\epsilon) < \delta + \epsilon\,.$$

Предел аппроксимации Коши $x : \mathbb{Q}_+ \to M$ есть точка $\ell : M$, удовлетворяющая условию:

$$\forall(\epsilon, \theta : \mathbb{Q}_+).\, d(x_\epsilon, \ell) < \epsilon + \theta\,.$$

Полное метрическое пространство — это пространство, в котором каждая аппроксимация Коши имеет предел.

Определение 11.5.3. Для положительного рационального числа ϵ, **ϵ-сеть** в метрическом пространстве (M, d) есть элемент из

$$\sum_{(n:\mathbb{N})} \sum_{(x_1,\ldots,x_n:M)} \forall(y : M).\, \exists(k \leqslant n).\, d(x_k, y) < \epsilon\,.$$

Дословно, это конечная последовательность точек $x_1, \ldots, x_n$ такая, что каждая точка в M просто находится в пределах ϵ некоторого x_k.

Метрическое пространство (M, d) является **вполне ограниченным**, если оно имеет ϵ-сети всех размеров:

$$\prod_{(\epsilon:\mathbb{Q}_+)} \sum_{(n:\mathbb{N})} \sum_{(x_1,\ldots,x_n:M)} \forall(y : M).\, \exists(k \leqslant n).\, d(x_k, y) < \epsilon\,.$$

Замечание 11.5.4. В определении вполне ограниченности мы использовали некорректную нотацию $\sum_{(n:\mathbb{N})} \sum_{(x_1,\ldots,x_n:M)}$. Формально, надо было бы использовать $\sum_{(x:\mathsf{List}(M))}$, где $\mathsf{List}(M)$ — индуктивный тип конечных списков из §5.1. Однако, это сделало бы остальную часть выражения немного более громоздкой.

Заметим, что в определении вполне ограниченности требуется чистое существование ϵ-net, а не просто существование. Таким образом, получаем функцию, которая присваивает каждому $\epsilon : \mathbb{Q}_+$ определенную ϵ-сеть. Такую функцию можно назвать «модулем вполне ограниченности». Вообще, при переносе классических метрических понятий в гомотопическую теорию типов необходимо экономно использовать пропозициональное усечение, обычно для того, чтобы не запрашивать непостоянное отображение от $\mathbb{R}$ к $\mathbb{Q}$ или к $\mathbb{N}$. К примеру, приведем «корректное» определение равномерной непрерывности.

Определение 11.5.5. Отображение $f : M \to \mathbb{R}$ на метрическом пространстве **равномерно непрерывно**, если

$$\prod_{(\epsilon:\mathbb{Q}_+)} \sum_{(\delta:\mathbb{Q}_+)} \forall(x, y : M).\, d(x, y) < \delta \Rightarrow |f(x) - f(y)| < \epsilon\,.$$

В частности, равномерно непрерывное отображение имеет модуль равномерной непрерывности , который является функцией, сопоставляющей каждому ϵ соответствующее δ.

Покажем, что $[0, 1]$ компактно в первом смысле.

Теорема 11.5.6. *Закрытый интервал* $[0, 1]$ *полон и вполне ограничен.*

Доказательство. Для $\epsilon : \mathbb{Q}_+$, существует $k : \mathbb{N}$ такое, что $2/k < \epsilon$, поэтому можно принять ϵ-сеть $x_i = i/k$ для $i = 0, \ldots, k$. Это именно ϵ-сеть, потому что, для каждого $y : [0,1]$, просто существует i такое, что $0 \leqslant i \leqslant k$ и $(i-1)/k < y < (i+1)/k$, и поэтому $|y - x_i| < 2/k < \epsilon$.

Для доказательства полноты $[0,1]$, рассмотрим аппроксимацию Коши $x : \mathbb{Q}_+ \to [0,1]$ и пусть ℓ будет ее пределом в $\mathbb{R}$. Поскольку $\max$ и $\min$ являются отображениями Лифшица, стягивание $r : \mathbb{R} \to [0,1]$, определяемое выражением $r(x) :\equiv \max(0, \min(1,x))$, коммутирует с пределами аппроксимаций Коши, поэтому

$$r(\ell) = r(\lim x) = \lim(r \circ x) = \lim x = \ell\,,$$

что означает $0 \leqslant \ell \leqslant 1$, как и требовалось. □

Таким образом, имеется, по крайней мере, одно хорошее понятие компактности в гомотопической теории типов. К сожалению, оно ограничено метрическими пространствами, потому что вполне ограниченность — это метрическое понятие. Вскоре мы рассмотрим два других понятия, но сначала докажем, что равномерно непрерывное отображение на вполне ограниченном пространстве имеет **супремум**, т.е. верхнюю границу, меньшую или равную всем другим верхним границам.

Теорема 11.5.7. *Равномерно непрерывное отображение $f : M \to \mathbb{R}$ на вполне ограниченном метрическом пространстве (M,d) имеет супремум $m : \mathbb{R}$. Для каждого $\epsilon : \mathbb{Q}_+$ существует $u : M$ такое, что $|m - f(u)| < \epsilon$.*

Доказательство. Пусть $h : \mathbb{Q}_+ \to \mathbb{Q}_+$ — модуль равномерной непрерывности f. Определим аппроксимацию $x : \mathbb{Q}_+ \to \mathbb{R}$ следующим образом: для любого $\epsilon : \mathbb{Q}_+$ вполне ограниченность M дает $h(\epsilon)$-сеть $y_0, \ldots, y_n$. Определим

$$x_\epsilon :\equiv \max(f(y_0), \ldots, f(y_n))\,.$$

Мы утверждаем, что x является аппроксимацией Коши. Возьмем любые $\epsilon, \eta : \mathbb{Q}_+$, так что

$$x_\epsilon \equiv \max(f(y_0), \ldots, f(y_n)) \quad \text{и} \quad x_\eta \equiv \max(f(z_0), \ldots, f(z_m))$$

для некоторых, $h(\epsilon)$-сети $y_0, \ldots, y_n$ и $h(\eta)$-сети $z_0, \ldots, z_m$. Каждый z_i просто $h(\epsilon)$-близок к некоторому y_j, поэтому $|f(z_i) - f(y_j)| < \epsilon$, из чего можно заключить, что

$$f(z_i) < \epsilon + f(y_j) \leqslant \epsilon + x_\epsilon\,,$$

так что, $x_\eta < \epsilon + x_\epsilon$. Симметрично получаем $x_\eta < \eta + x_\eta$, поэтому $|x_\eta - x_\epsilon| < \eta + \epsilon$.

Мы утверждаем, что $m :\equiv \lim x$ является супремумом функции f. Чтобы доказать, что $f(x) \leqslant m$, для всех $x : M$, достаточно показать $\neg(m < f(x))$. Предположим противное, что $m < f(x)$. Существует $\epsilon : \mathbb{Q}_+$ такое, что $m + \epsilon < f(x)$. Но теперь, просто для некоторого y_i участвующего в определении x_ϵ, получаем $|f(x) - f(y_i)| < \epsilon$, поэтому $m < f(x) - \epsilon < f(y_i) \leqslant m$, получили противоречие.

Завершим доказательство, показав, что m удовлетворяет второй части теоремы, потому что тогда оно автоматически является точной верхней границей. Для любого $\epsilon : \mathbb{Q}_+$, с одной стороны, $\left|m - f(x_{\epsilon/2})\right| < 3\epsilon/4$, а с другой, $\left|f(x_{\epsilon/2}) - f(y_i)\right| < \epsilon/4$, просто для некоторого y_i, участвующего в определении $x_{\epsilon/2}$, поэтому, взяв $u :\equiv y_i$, получаем $|m - f(u)| < \epsilon$, из неравенства треугольника. □

Теперь, если бы в теореме 11.5.7 мы также знали, что M полны, то могли бы надеяться ослабить предположение о равномерной непрерывности до непрерывности и усилить вывод о существовании точки, в которой достигается супремум. Обычные доказательства этих улучшений опираются на тот факт, что в полном вполне ограниченном пространстве

(i) непрерывность подразумевает равномерную непрерывность, и

(ii) каждая последовательность имеет сходящуюся подпоследовательность.

Первое утверждение легко следует из компактности Гейне-Бореля, а второе — просто компактность Больцано-Вейерштрасса. К сожалению, и то, и другое, несколько проблематично. Сначала покажем, что компактность Больцано-Вейерштрасса подразумевает экземпляр исключения третьего, известный как **ограниченный принцип всеведения**: для каждого $\alpha : \mathbb{N} \to \mathbf{2}$,

$$\left(\sum_{n:\mathbb{N}} \alpha(n) = 1_{\mathbf{2}}\right) + \left(\prod_{n:\mathbb{N}} \alpha(n) = 0_{\mathbf{2}}\right). \tag{11.5.8}$$

Говоря вычислительным языком, мы не ожидаем, что этот принцип будет выполняться, потому что он требует разрешения ситуации с бесконечным множеством значений функции $0_{\mathbf{2}}$.

Теорема 11.5.9. *Компактность Больцано-Вейерштрасса для* $[0, 1]$ влечет за собой ограниченный принцип всеведения.

Доказательство. Для любого $\alpha : \mathbb{N} \to \mathbf{2}$, определим последовательность $x : \mathbb{N} \to [0, 1]$ посредством

$$x_n :\equiv \begin{cases} 0 & \text{если } \alpha(k) = 0_{\mathbf{2}} \text{ для всех } k < n, \\ 1 & \text{если } \alpha(k) = 1_{\mathbf{2}} \text{ для всех } k < n. \end{cases}$$

Если выполняется свойство Больцано-Вейерштрасса, существует строго возрастающая функция $f : \mathbb{N} \to \mathbb{N}$ такая, что $x \circ f$ является последовательностью Коши. При достаточно большом $n : \mathbb{N}$ n-ый член $x_{f(n)}$ находится в пределах $1/6$ своего предела. Либо $x_{f(n)} < 2/3$, либо $x_{f(n)} > 1/3$. Если $x_{f(n)} < 2/3$, то x_n сходится к 0, поэтому $\prod_{(n:\mathbb{N})} \alpha(n) = 0_{\mathbf{2}}$. Если $x_{f(n)} > 1/3$, то $x_{f(n)} = 1$, поэтому $\sum_{(n:\mathbb{N})} \alpha(n) = 1_{\mathbf{2}}$. □

Хотя мы не можем слишком печалиться о компактности Больцано-Вейерштрасса, нам кажется труднее жить без компактности Гейне-Бореля, о чем свидетельствует тот факт, что и классическая математика, и интуитивизм Брауэра приняли ее. Поскольку мы не хотим слишком углубляться в общую топологию, будем работать с базовыми открытыми множествами. В случае $\mathbb{R}$ это открытые интервалы с рациональными концами. Семейство таких интервалов, индексированное типом I, было бы отображением

$$\mathcal{F} : I \to \{(q, r) : \mathbb{Q} \times \mathbb{Q} \mid q < r\},$$

с идеей, что пара рациональных чисел (q, r), с $q < r$, определяет тип $\{x : \mathbb{R} \mid q < x < r\}$. Несколько удобнее допустить и вырожденные интервалы, поэтому в качестве отображения возьмем **семейство базовых интервалов**

$$\mathcal{F} : I \to \mathbb{Q} \times \mathbb{Q}.$$

Если быть точным, семейство — это зависимая пара $(I, \mathcal{F})$, а не просто $\mathcal{F}$. **Конечное семейство базовых интервалов** — это интервал, индексируемый $\{m : \mathbb{N} \mid m < n\}$, для некоторого n :

$\mathbb{N}$. Обычно он представляется конечным списком $[(q_0, r_0), \ldots, (q_{n-1}, r_{n-1})]$. Наконец, **конечное подсемейство** семейства $(I, \mathcal{F})$ задается списком индексов $[i_1, \ldots, i_n]$, которые затем определяют конечное семейство $[\mathcal{F}(i_1), \ldots, \mathcal{F}(i_n)]$.

Пока мы знаем о различии между парой (q, r) и соответствующим интервалом $\{x : \mathbb{R} \mid q < x < r\}$, можно безопасно использовать одну и ту же нотацию (q, r) для обоих. Пересечения и включения интервалов выражаются в терминах их концов:

$$(q, r) \cap (s, t) :\equiv (\max(q, s), \min(r, t)) \,,$$
$$(q, r) \subseteq (s, t) :\equiv (q < r \Rightarrow s \leq q < r \leq t) \,.$$

Скажем, что $(I, \lambda i. (q_i, r_i))$ **(поточечно) покрывает** $[a, b]$ когда

$$\forall (x : [a, b]). \exists (i : I). q_i < x < r_i \,. \tag{11.5.10}$$

Компактность Гейне-Бореля для $[0, 1]$ утверждает, что каждое покрывающее семейство $[0, 1]$ просто имеет конечное подсемейство, которое по-прежнему покрывает $[0, 1]$.

Теорема 11.5.11. *Если имеет место исключение третьего, то* $[0, 1]$ *является компактностью Гейне-Бореля.*

Доказательство. Предположим, для достижения противоречия, что семейство $(I, \lambda i. (a_i, b_i))$ покрывает $[0, 1]$, но никакое конечное подсемейство не покрывает. Построим последовательность замкнутых интервалов $[q_n, r_n]$, которые вложены друг в друга, их размеры сокращаются до 0, и ни один из них не покрывается конечным подсемейством $(I, \lambda i. (a_i, b_i))$.

Положим $[q_0, r_0] :\equiv [0, 1]$. Предполагая, что $[q_n, r_n]$ построено, пусть $s :\equiv (2q_n + r_n)/3$ и $t :\equiv (q_n + 2r_n)/3$. Оба, $[q_n, t]$ и $[s, r_n]$, покрываются $(I, \lambda i. (a_i, b_i))$, но они не могут оба иметь конечное подпокрытие, иначе $[q_n, r_n]$. Либо $[q_n, t]$ имеет конечное подпокрытие, либо нет. Если это так, то установим $[q_{n+1}, r_{n+1}] :\equiv [s, r_n]$, иначе установим $[q_{n+1}, r_{n+1}] :\equiv [q_n, t]$.

Последовательности $q_0, q_1, \ldots$ и $r_0, r_1, \ldots$ являются последовательностями Коши и сходятся к точке $x : [0, 1]$, которая содержится в каждом $[q_n, r_n]$. Просто существует $i : I$ такое, что $a_i < x < b_i$. Поскольку размеры интервалов $[q_n, r_n]$ уменьшаются до нуля, существует $n : \mathbb{N}$ такое, что $a_i < q_n \leqslant x \leqslant r_n < b_i$, но это означает, что $[q_n, r_n]$ покрывается одним интервалом (a_i, b_i), и в то же время не имеет конечного подпокрытия. Получено противоречие. $\square$

Без исключения третьего или щепотки брауэровского интуиционизма мы, похоже, застряли. Тем не менее компактность Гейне-Бореля $[0, 1]$ *может* быть восстановлена в конструктивной постановке способом, который все еще совместим с классической математикой! Для этого нужно пересмотреть понятие покрытия. Проблема с (11.5.10) заключается в том, что усеченное экзистенциальное пространство позволяет покрывать пространство любым случайным образом, и поэтому с вычислительной точки зрения у нас нет шансов просто извлечь конечное подпокрытие. Убрав усечение, получим

$$\prod_{(x:[0,1])} \sum_{(i:I)} q_i < x < r_i \,, \tag{11.5.12}$$

что могло бы помочь, если бы оно не было слишком требовательным к покрытиям. С этим определением мы не смогли бы даже показать, что $(0, 3)$ и $(2, 5)$ покрывают $[1, 4]$, потому что это означало бы демонстрацию непостоянного отображения $[1, 4] \to \mathbf{2}$, см. упражнение 11.6. Здесь мы можем извлечь урок из «бесточечной топологии» (т.е. теории локалей): понятие покрытия должно быть выражено в терминах открытых множеств без ссылки на точки. Такой «целостный» взгляд на пространство позволит нам затем проанализировать понятие покрытия, и мы сможем восстановить гейне-борелевскую компактность. Теория локалей использует степенные множества, которые мы могли бы получить, предполагая пропозициональное изменение размера; но вместо этого мы можем позаимствовать идеи у предикативной родственницы теории локалей, которая называется «формальной топологией».

Предположим, что имеется семейство $(I, \mathcal{F})$ и интервал (a, b). Как можно выразить тот факт, что (a, b) покрывается семейством, не обращаясь к точкам? Вот один из способов: если (a, b) равно некоторому $\mathcal{F}(i)$, то оно покрывается семейством. И еще: если (a, b) покрывается каким-то другим семейством $(J, \mathcal{G})$, а каждое $\mathcal{G}(j)$, в свою очередь, покрывается $(I, \mathcal{F})$, то (a, b) покрывается $(I, \mathcal{F})$. Обратите внимание, что мы перечисляем *правила*, которые можно использовать для *вывода* о том, что $(I, \mathcal{F})$ покрывает (a, b). Мы должны найти достаточно хорошие правила и превратить их в индуктивное определение.

Определение 11.5.13. Индуктивное покрытие $\lhd$ — это простое отношение

$$\lhd\colon (\mathbb{Q} \times \mathbb{Q}) \to \left(\sum_{I:\mathcal{U}} (I \to \mathbb{Q} \times \mathbb{Q}) \right) \to \mathsf{Prop}\,,$$

определяемое индуктивно следующими правилами, где q, r, s, t — рациональные числа, а $(I, \mathcal{F})$, $(J, \mathcal{G})$ — семейства основных интервалов:

(i) *рефлексивность*: $\mathcal{F}(i) \lhd (I, \mathcal{F})$, для всех $i : I$,

(ii) *транзитивность*: если $(q, r) \lhd (J, \mathcal{G})$ и $\forall (j : J).\, \mathcal{G}(j) \lhd (I, \mathcal{F})$, то $(q, r) \lhd (I, \mathcal{F})$,

(iii) *монотонность*: если $(q, r) \subseteq (s, t)$ и $(s, t) \lhd (I, \mathcal{F})$, то $(q, r) \lhd (I, \mathcal{F})$,

(iv) *локализация*: если $(q, r) \lhd (I, \mathcal{F})$, то $(q, r) \cap (s, t) \lhd (I, \lambda i.\, (\mathcal{F}(i) \cap (s, t)))$,

(v) если $q < s < t < r$, то $(q, r) \lhd [(q, t), (r, s)]$,

(vi) $(q, r) \lhd (\{(s, t) : \mathbb{Q} \times \mathbb{Q} \mid q < s < t < r\}, \lambda u.\, u)$.

Это определение следует воспринимать как высший индуктивный тип, в котором перечисленные правила являются точечными конструкторами, а тип является (-1)-усеченным. Первые четыре пункта носят общий характер и должны быть интуитивно понятны. Последние два относятся к вещественной оси: в одном говорится, что интервал может быть покрыт двумя интервалами, если они перекрываются, а в другом — что интервал может быть покрыт изнутри. В данном случае, если $r \leqslant q$, то (q, r) покрывается пустым семейством (по последнему пункту).

Индуктивные покрытия обладают свойством Гейне-Бореля, для доказательства которого требуется лемма.

Лемма 11.5.14. *Предположим, что $q < s < t < r$ и $(q, r) \lhd (I, \mathcal{F})$. Тогда просто существует конечное подсемейство в $(I, \mathcal{F})$, которое индуктивно покрывает (s, t).*

Доказательство. Докажем утверждение индукцией по $(q, r) \lhd (I, \mathcal{F})$. Имеются шесть случаев.

(i) Рефлексивность: если $(q, r) = \mathcal{F}(i)$, то, по монотонности, (s, t) покрывается конечным подсемейством $[\mathcal{F}(i)]$.

(ii) Транзитивность: предположим, что $(q,r) \lhd (J,\mathcal{G})$ и $\forall (j : J).\, \mathcal{G}(j) \lhd (I,\mathcal{F})$. По индуктивному предположению, просто существуют $[\mathcal{G}(j_1),\ldots,\mathcal{G}(j_n)]$, которые покрывают (s,t). Снова, по индуктивному предположению, каждое из $\mathcal{G}(j_k)$ покрывается конечным подсемейством $(I,\mathcal{F})$, и можно собрать их в конечное подсемейство, которое покрывает (s,t).

(iii) Монотонность: если $(q,r) \subseteq (u,v)$ и $(u,v) \lhd (I,\mathcal{F})$, то можно применить индуктивное предположение к $(u,v) \lhd (I,\mathcal{F})$, поскольку $u < s < t < v$.

(iv) Локализация: предположим, что $(q',r') \lhd (I,\mathcal{F})$ и $(q,r) = (q',r') \cap (a,b)$. Поскольку $q' < s < t < r'$, то, по предположению индукции, существует конечное подпокрытие $[\mathcal{F}(i_1),\ldots,\mathcal{F}(i_n)]$ из (s,t). Также, $a < s < t < b$, поэтому $(s,t) = (s,t) \cap (a,b)$ покрывается $[\mathcal{F}(i_1) \cap (a,b),\ldots,\mathcal{F}(i_n) \cap (a,b)]$, которое является конечным подсемейством $(I, \lambda i.\, (\mathcal{F}(i) \cap (a,b))$.

(v) Если $(q,r) \lhd [(q,v),(u,r)]$ для некоторых $q < u < v < r$, то, по монотонности, $(s,t) \lhd [(q,v),(u,r)]$.

(vi) Ну, и наконец, $(s,t) \lhd (\{(u,v) : \mathbb{Q} \times \mathbb{Q} \mid q < u < v < r\}, \lambda z.\, z)$, по рефлексивности.

□

Скажем, что $(I,\mathcal{F})$ **индуктивно покрывает** $[a,b]$, когда просто существует $\epsilon : \mathbb{Q}_+$ такое, что $(a-\epsilon, b+\epsilon) \lhd (I,\mathcal{F})$.

Следствие 11.5.15. *Закрытый интервал является компактным по Гейне-Борелю для индуктивных покрытий.*

Доказательство. Предположим, что $[a,b]$ индуктивно покрывается $(I,\mathcal{F})$, поэтому просто существует $\epsilon : \mathbb{Q}_+$ такое, что $(a-\epsilon, b+\epsilon) \lhd (I,\mathcal{F})$. По лемме 11.5.14 существует конечное подпокрытие $(a-\epsilon/2, b+\epsilon/2)$, которое, следовательно, является конечным подпокрытием $[a,b]$. □

Опыт формальной топологии показывает, что правил индуктивных накрытий достаточно для конструктивного развития бесточечной топологии. Но мы также можем представить собственные доказательства того, что это разумное понятие.

Теорема 11.5.16.

(i) *Индуктивное покрытие является также и точечным покрытием.*

(ii) *В предположении исключения третьего, точечное покрытие является также и индуктивным покрытием.*

Доказательство.

(i) Рассмотрим семейство базовых интервалов $(I,\mathcal{F})$, где $(q_i,r_i) :\equiv \mathcal{F}(i)$, интервал (a,b), индуктивно покрытый семейством $(I,\mathcal{F})$, и x такой, что $a < x < b$. Докажем, индукцией по $(a,b) \lhd (I,\mathcal{F})$, что просто существует $i : I$ такое, что $q_i < x < r_i$. В большинство случаев доказательства довольно очевидны, поэтому приведем только два. Если $(a,b) \lhd (I,\mathcal{F})$, по рефлексивности, то просто существует некоторое $i : I$ такое, что $(a,b) = (q_i,r_i)$, и, поэтому, $q_i < x < r_i$. Если $(a,b) \lhd (I,\mathcal{F})$, по транзитивности, посредством $(J, \lambda j.\, (s_j,t_j)$, то, по индуктивному предположению, просто существует $j : J$ такое, что $s_j < x < t_j$, а затем, поскольку $(s_j,t_j) \lhd (I,\mathcal{F})$, снова, по индуктивному предположению, просто существует $i : I$ такое, что $q_i < x < r_i$. Другие случаи не менее интересны.

(ii) Предположим, что $(I, \lambda i.\,(q_i, r_i))$ поточечно покрывает (a, b). По пункту (vi) определения 11.5.13 достаточно показать, что $(I, \lambda i.\,(q_i, r_i))$ индуктивно покрывает (c, d), когда $a < c < d < b$, поэтому будем использовать такие c и d. По теореме 11.5.11, существует конечное подсемейство $[i_1, \ldots, i_n]$, которое, также поточечно, покрывает $[c, d]$, а значит, и (c, d). Пусть $\epsilon : \mathbb{Q}_+$ — число Лебега для $(q_{i_1}, r_{i_1}), \ldots, (q_{i_n}, r_{i_n})$, как в упражнении 11.12. Существует положительное $k : \mathbb{N}$ такое, что $2(d-c)/k < \min(1, \epsilon)$. Для $0 \leqslant i \leqslant k$, пусть

$$c_k :\equiv ((k-i)c + id)/k\,.$$

Интервалы (c_0, c_2), (c_1, c_3), …, (c_{k-2}, c_k) индуктивно покрывают (c, d) по (повторному использованию) транзитивности и пункта (v) в определении 11.5.13. Поскольку их ширина меньше ϵ, каждый из них содержится в некотором (q_i, r_i), и можно использовать транзитивность и монотонность, чтобы заключить, что $(I, \lambda i.\,(q_i, r_i))$ индуктивно покрывает (c, d).

□

Результатом предыдущей теоремы является то, что с точки зрения классической математики нет никакой разницы между точечным и индуктивным накрытием. В частности, поскольку предположение об исключении третьего в гомотопической теории типов непротиворечиво, не получится предъявить индуктивное покрытие, которое не будет являться точечным покрытием. Или, другими словами, разница между точечными и индуктивными покрытиями не в том, что они покрывают, а в их *доказательствах*.

Мы могли бы написать еще одну книгу, продолжая в том же духе, но давайте остановимся здесь и будем надеяться, что мы дали достаточное обоснование утверждению того, что анализ может быть развит в гомотопической теории типов. Заинтересованный читатель может обратиться к упражнению 11.13 за конструктивными версиями теоремы о промежуточном значении.

11.6 Сюрреалистические числа

В этом разделе рассмотрим еще один пример высшего индуктивно-индуктивного типа, который объединяет многие из наших тем: поле Конвея No *сюрреалистических чисел* [Con76]. Сюрреалистические числа являются естественным общим обобщением действительных чисел (Дедекинда) (§11.2) и порядковых чисел (§11.3). Конвей, работающий в области классической математики (с исключением третьего и выбором) определяет сюрреалистическое число как пару множеств сюрреалистических чисел, обозначаемое как $\{L \mid R\}$, таких, что каждый элемент из L строго меньше, любого элемента из R. Очевидно, такое определение похоже на индуктивное, но имеются три проблемы, связанные с рассмотрением сюрреалистического числа как такового.

Во-первых, это определение требует отношения (строгого) неравенства между сюрреалистами числами, так что это отношение должно быть определено одновременно с типом No этих чисел (Конвей избегает этой проблемы, сначала определяя *геймы*, которые подобны сюрреалистическим числам, но без условия совместимости на L и R). Как и в случае отношения $\sim$ для действительных чисел Коши, это одновременное определение может быть *априори*, либо индуктивно-индуктивным, либо индуктивно-рекурсивным. Мы сделаем его индуктивно-индуктивным по тем же причинам, что и для $\sim$.

Более того, мы определим строгое неравенство $<$ и нестрогое неравенство $\leqslant$ для сюрреалистических чисел раздельно (и взаимно индуктивно). Конвей определяет $<$ в терминах $\leqslant$, что разумно с классической точки зрения, но не является конструктивным. Кроме того, определение $<$ посредством отрицания, сделало бы его неприемлемым в качестве гипотезы конструктора высшего индуктивного типа (см. §5.6).

Во-вторых, Конвей заявляет, что L и R в $\{L \mid R\}$ должны быть «множествами сюрреалистических чисел», но наивное значение этого, как предиката $\mathsf{No} \to \mathsf{Prop}$, не является положительным, поэтому его нельзя использовать в качестве входных данных для индуктивного конструктора. Однако, в любом случае это не будет хорошим теоретико-типовым переводом того, что имеет в виду Конвей, потому что в теории множеств сюрреалистические числа образуют собственный класс, тогда как множества L и R являются истинными (малыми) множествами, а не произвольными подклассами для No. В теории типов это означает, что No будет определен относительно универсума $\mathcal{U}$, но сам будет принадлежать следующему, более высокому, универсуму $\mathcal{U}'$, подобно множествам Ord и Card ординалов и кардиналов, кумулятивной иерархии V или даже действительным числам Дедекинда при отсутствии пропозиционального изменения размера. Тогда мы потребуем, чтобы «множества» сюрреалистических чисел L и R были $\mathcal{U}$-малыми, поэтому естественно представить их *семействами* сюрреалистических чисел, индексируемыми некоторым $\mathcal{U}$-малым типом (это все точно то же, что мы сделали с кумулятивной иерархией в §10.5). То есть, конструктор сюрреалистических чисел будет иметь тип

$$\prod_{\mathcal{L},\mathcal{R}:\mathcal{U}} (\mathcal{L} \to \mathsf{No}) \to (\mathcal{R} \to \mathsf{No}) \to (\text{какое-то условие}) \to \mathsf{No}\,,$$

что действительно строго положительно.

Наконец, приводя взаимные определения No и порядка, Конвей объявляет два сюрреалистических числа x и y *равными*, если $x \leqslant y$ и $y \leqslant x$. Естественно, это можно понимать как переход к частному множества «пред-сюрреалистических чисел» по отношению эквивалентности. Однако в отсутствие аксиомы выбора такое частное представляет ту же проблему, что и частное в обычном построении действительных чисел Коши: больше не будет иметь место то, что пара семейств *сюрреалистических чисел* дает новое сюрреалистическое число $\{L \mid R\}$, так как мы не можем обязательно «поднять» L и R в семейства пред-сюрреалистических чисел. Конечно, можно решить эту проблему так же, как мы это было сделано для действительных чисел Коши, используя высшее индуктивно-индуктивное определение.

Определение 11.6.1. Тип No **сюрреалистических чисел**, вместе с отношениями $< : \mathsf{No} \to \mathsf{No} \to \mathcal{U}$ и $\leqslant : \mathsf{No} \to \mathsf{No} \to \mathcal{U}$, определяются индуктивно-индуктивно (высшего порядка) следующим образом. Тип No имеет конструкторы:

- для любых $\mathcal{L}, \mathcal{R} : \mathcal{U}$, и функций $\mathcal{L} \to \mathsf{No}$, $\mathcal{R} \to \mathsf{No}$, значения которых обозначим как x^L и x^R, для $L : \mathcal{L}$ и $R : \mathcal{R}$, соответственно, если $\forall (L : \mathcal{L}).\, \forall (R : \mathcal{R}).\, x^L < x^R$, то существует сюрреалистическое число x;
- для любых $x, y : \mathsf{No}$ таких, что $x \leqslant y$ и $y \leqslant x$, имеем $\mathsf{eq}_{\mathsf{No}}(x, y) : x = y$.

Будем ссылаться на входные данные первого конструктора как на **сечение**. Если x — сюрреалистическое число, построенное из сечения, то обозначение x^L будет неявно предполагать $L : \mathcal{L}$, и аналогично, x^R предполагает $R : \mathcal{R}$. Таким образом, обычно можно избежать именования типов индексации $\mathcal{L}$ и $\mathcal{R}$, что удобно, когда обсуждается множество различных сечений. Следуя Конвею, мы называем x^L *левой опцией* x, а x^R — *правой опцией*.

Конструктор пути подразумевает, что разные сечения могут определять одно и то же сюрреалистическое число. Таким образом, нет смысла говорить о левой или правой опциях произвольного сюрреалистического числа x, если не учитывать, что x определяется конкретным сечением. Поэтому в дальнейшем будем говорить, например, «дано сечение, определяющий сюрреалистическое число x» в отличие от «дано сюрреалистическое число x».

Отношение $\leqslant$ имеет следующие конструкторы.

- Для сечений, определяющих два сюрреалистических числа x и y, если $x^L < y$, для всех L, и $x < y^R$, для всех R, то $x \leqslant y$.
- Пропозициональное усечение: для любых $x, y : \mathsf{No}$, если $p, q : x \leqslant y$, то $p = q$.

Отношение $<$ имеет следующие конструкторы.

- Для сечений, определяющих два сюрреалистических числа x и y, если существует L такое, что $x \leqslant y^L$, то $x < y$.
- Для сечений, определяющих два сюрреалистических числа x и y, если существует R такое, что $x^R \leqslant y$, то $x < y$.
- Пропозициональное усечение: для любых $x, y : \mathsf{No}$, если $p, q : x < y$, то $p = q$.

Сравним это с определением Конвея:

– если L, R — любые два множества чисел, и ни один элемент из L не является «$\geqslant$ любому элементу из R», то существует число $\{L \mid R\}$; все числа построены таким образом и обладают свойствами:

– $x \geqslant y$ тогда и только тогда, когда (нет $x^R \leqslant y$ и $x \leqslant$ нет y^L);

– $x = y$ тогда и только тогда, когда ($x \geqslant y$ и $y \geqslant x$);

– $x > y$ тогда и только тогда, когда ($x \geqslant y$ и $y \not\geqslant x$).

Включение «$x \geqslant y$» в определение «$x > y$» является излишним, если все объекты являются (сюрреалистическими) числами, а не «геймами». Таким образом, «$<$» у Конвея есть просто отрицание его «$\geqslant$», так что его условие того, чтобы $\{L \mid R\}$ было сюрреалистичным, такое же, как и у нас. Отрицая «$\leqslant$» у Конвея и отменяя двойные отрицания, приходим к нашему определению «$<$», а затем можем переформулировать его «$\leqslant$» в терминах «$<$» без отрицаний.

Мы можем сразу «заселить» No многими сюрреалистическими числами. В стиле Конвея, будем записывать

$$\{x, y, z, \cdots \mid u, v, w, \ldots\}$$

для сюрреалистического числа, определенного сечением, где $\mathcal{L} \to \mathsf{No}$ и $\mathcal{R} \to \mathsf{No}$ — это семейства, записываемые как $x, y, z, \ldots$ и $u, v, w, \ldots$. Конечно, если $\mathcal{L}$ или $\mathcal{R}$ есть $\mathbf{0}$, то соответствующая часть обозначения оставляется пустой. При этом, существует досадное пересечение со стандартной записью $\{x : A \mid P(x)\}$ для подмножеств, но мы не будем использовать последнюю в этом разделе.

- Определим $\iota_{\mathbb{N}} : \mathbb{N} \to \mathsf{No}$ рекурсивно, как

$$\iota_{\mathbb{N}}(0) :\equiv \{\,\mid\,\}\ ,$$
$$\iota_{\mathbb{N}}(\mathsf{succ}(n)) :\equiv \{\iota_{\mathbb{N}}(n) \mid \,\}\ .$$

  То есть, $\iota_{\mathbb{N}}(0)$ определяется сечением, состоящим из $\mathbf{0} \to \mathsf{No}$ и $\mathbf{0} \to \mathsf{No}$. Точно так же, $\iota_{\mathbb{N}}(\mathsf{succ}(n))$ определяется как $\mathbf{1} \to \mathsf{No}$ (выбирая $\iota_{\mathbb{N}}(n)$) и $\mathbf{0} \to \mathsf{No}$.

- Аналогично, определим $\iota_{\mathbb{Z}} : \mathbb{Z} \to \mathsf{No}$, используя принцип «знакопеременной» рекурсии (лемма 6.10.12):

$$\begin{aligned} \iota_{\mathbb{Z}}(0) &:\equiv \{\,|\,\}\,, \\ \iota_{\mathbb{Z}}(n+1) &:\equiv \{\iota_{\mathbb{Z}}(n)\,|\,\} && n \geqslant 0\,, \\ \iota_{\mathbb{Z}}(n-1) &:\equiv \{\,|\,\iota_{\mathbb{Z}}(n)\} && n \leqslant 0\,. \end{aligned}$$

- Под **диадическим рациональным числом** понимается пара (a,n), где $a : \mathbb{Z}$, $n : \mathbb{N}$, такая, что если $n > 0$, то a нечетно. Будем записывать такое число как $a/2^n$, и отождествим его с соответствующим рациональным числом. Если $\mathbb{Q}_D$ обозначает множество диадических рациональных чисел, то определим $\iota_{\mathbb{Q}_D} : \mathbb{Q}_D \to \mathsf{No}$ индукцией по n:

$$\begin{aligned} &\iota_{\mathbb{Q}_D}(a/2^0) :\equiv \iota_{\mathbb{Z}}(a)\,, \\ &\iota_{\mathbb{Q}_D}(a/2^n) :\equiv \{\iota_{\mathbb{Q}_D}(a/2^n - 1/2^n \mid \iota_{\mathbb{Q}_D}(a/2^n + 1/2^n)\}\,, \quad \text{for } n > 0. \end{aligned}$$

  Здесь используется тот факт, что если $n > 0$ и a нечетно, то $a/2^n \pm 1/2^n$ является диадическим рациональным числом с меньшим знаменателем, чем $a/2^n$.

- Определим $\iota_{\mathbb{R}_d} : \mathbb{R}_d \to \mathsf{No}$, где $\mathbb{R}_d$ — версия действительных чисел Дедекинда из §11.2, как

$$\iota_{\mathbb{R}_d}(x) :\equiv \{q \in \mathbb{Q}_D \text{ такое, что } q < x \mid q \in \mathbb{Q}_D \text{ такое, что } x < q\}\,.$$

  В отличие от предыдущих случаев, неочевидно, что это есть расширение $\iota_{\mathbb{Q}_D}$, когда рассматриваются диадические рациональные числа как действительные числа Дедекинда. Это следует из теоремы простоты (теорема 11.6.2).

- Напомним тип Ord *ординальных чисел* из §10.3, который упорядочен отношением $<$, где $A < B$ означает, что $A = B_{/b}$, для некоторого $b : B$. Определим $\iota_{\mathsf{Ord}} : \mathsf{Ord} \to \mathsf{No}$ обоснованной рекурсией (лемма 10.3.7) на Ord:

$$\iota_{\mathsf{Ord}}(A) :\equiv \left\{\iota_{\mathsf{Ord}}(A_{/a}) \text{ для всех } a : A \,\middle|\, \right\}\,.$$

  Из теоремы простоты также следует, что ι_{Ord} ограниченный конечными ординальными числами, согласуется с $\iota_{\mathbb{N}}$ (однако, предупреждаем читателя, что, в отличие от приведенных выше примеров, ι_{Ord} не является конструктивно инъективным, если только не ограничить его меньшим классом ординальных чисел (см. упражнения 11.16 и 11.17).

- Еще несколько интересных примеров, заимствованных у Конвея:

$$\begin{aligned} \omega &:\equiv \{0,1,2,3,\cdots \mid\} \qquad \text{(также, ординальное число)} \\ -\omega &:\equiv \{\mid \ldots,-3,-2,-1,0\} \\ 1/\omega &:\equiv \{0 \mid 1, \tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{8}, \ldots\} \\ \omega - 1 &:\equiv \{0,1,2,3,\cdots \mid \omega\} \\ \omega/2 &:\equiv \{0,1,2,3,\cdots \mid \ldots, \omega-2, \omega-1, \omega\}\,. \end{aligned}$$

При идентифицировании сюрреалистических чисел, представленных разными сечениями, полезно следующее простое наблюдение.

Теорема 11.6.2 (теорема простоты, Конвей)**.** *Пусть x и z — сюрреалистические числа, определяемые сечениями, и верно следующее.*

- $x^L < z < x^R$, *для всех L и R.*
- *Для каждой левой опции z^L для z, существует левая опция $x^{L'}$ с $z^L \leqslant x^{L'}$.*
- *Для каждой правой опции z^R для z, существует правая опция $x^{R'}$ с $x^{R'} \leqslant z^R$.*

Тогда $x = z$.

Доказательство. Применяя конструктор пути для No, надо показать, что $x \leqslant z$ и $z \leqslant x$. Первое влечет $x^L < z$, для всех L, что предполагалось, и $x < z^R$, для всех R. Но, по предположению, для любого z^R, существует $x^{R'}$ с $x^{R'} \leqslant z^R$, следовательно, $x < z^R$, что и требовалось. Таким образом, $x \leqslant z$; доказательство $z \leqslant x$ симметрично. □

Однако, чтобы сообщать о сюрреалистических числах гораздо больше, необходим их принцип индукции. Принцип взаимной индукции для $(\mathsf{No}, \leqslant, <)$ применяется к трем семействам типов:

$$
\begin{aligned}
&A : \mathsf{No} \to \mathcal{U}\,,\\
&B : \prod_{(x,y:\mathsf{No})} \prod_{(a:A(x))} \prod_{(b:A(y))} (x \leqslant y) \to \mathcal{U}\,,\\
&C : \prod_{(x,y:\mathsf{No})} \prod_{(a:A(x))} \prod_{(b:A(y))} (x < y) \to \mathcal{U}\,.
\end{aligned}
$$

Как и в случае с принципом индукции для действительных чисел Коши, полезно думать о B и C как о семействах отношений между типами $A(x)$ и $A(y)$. Таким образом, мы записываем $B(x,y,a,b,\xi)$ как $(x,a) \trianglelefteqslant^{\xi} (y,b)$, а $C(x,y,a,b,\xi)$ как $(x,a) \triangleleft^{\xi} (y,b)$. Точно так же, обычно будем опускать ξ, так как он содержится в простом высказывании, и поэтому не представляет интереса, и часто можно также опускать x и y, записывая просто $a \trianglelefteqslant b$ или $a \triangleleft b$. С этими обозначениями гипотезы принципа индукции таковы.

- Для любого сечения, определяющего сюрреалистическое число x, вместе с

  (i) для каждого L, элементом $a^L : A(x^L)$, и

  (ii) для каждого R, элементом $a^R : A(x^R)$, такими, что

  (iii) для всех L и R, имеем $(x^L, a^L) \triangleleft (x^R, a^R)$,

  существует точно определенный элемент $f_a : A(x)$. Мы называем такие данные **зависимым сечением** над сечением, определяющим x.

- Для любых $x, y : \mathsf{No}$ с $a : A(x)$ и $b : A(y)$, если $x \leqslant y$ и $y \leqslant x$, а также $(x,a) \trianglelefteqslant (y,b)$ и $(y,b) \trianglelefteqslant (x,a)$, то $a =^A_{\mathsf{eq}_{\mathsf{No}}} b$.

- Для сечений, определяющих два сюрреалистических числа x и y, и зависимых сечений a над x и b над y, таких, что, для всех L, имеем $x^L < y$ и $(x^L, a^L) \triangleleft (y, f_b)$, а, для всех R, имеем $x < y^R$ и $(x, f_a) \triangleleft (y^R, b^R)$, имеет место $(x, f_a) \trianglelefteqslant (y, f_b)$.

- $\trianglelefteqslant$ принимает значения в простых высказываниях.

- Для сечений, определяющих два сюрреалистических числа x и y, зависимых сечений a над x и b над y, а также L_0, таких, что $x \leqslant y^{L_0}$ и $(x, f_a) \trianglelefteq (y^{L_0}, b^{L_0})$, имеет место $(x, f_a) \lhd (y, f_b)$.

- Для сечений, определяющих два сюрреалистических числа x и y, зависимых сечений a над x и b над y, а также R_0, таких, что $x^{R_0} \leqslant y$, вместе с $(x^{R_0}, a^{R_0}), \trianglelefteq (y, f_b)$, имеет место $(x, f_a) \lhd (y, f_b)$.

- $\lhd$ принимает значения в простых высказываниях.

При этих гипотезах мы выводим функцию $f : \prod_{(x:\mathsf{No})} A(x)$ такую, что

$$\begin{aligned} f(x) &\equiv f_{f[x]} \\ (x \leqslant y) &\Rightarrow (x, f(x)) \trianglelefteq (y, f(y)) \\ (x < y) &\Rightarrow (x, f(x)) \lhd (y, f(y)). \end{aligned} \tag{11.6.3}$$

В правиле вычисления (11.6.3) для точечного конструктора, x — сюрреалистическое число, определяемое сечением, а $f[x]$ обозначает зависимое сечение над x, определяемое применением f (и использованием того факта, что f переводит $<$ в $\lhd$). Как обычно, будем использовать нотацию сопоставления с образцом, где определение f на сечении $\left\{x^L \mid x^R\right\}$ на сечении $f(x^L)$, $f(x^R)$, и предположение, что они образуют зависимое сечение.

Как и в случае с действительными числами Коши, имеются особые случаи, возникающие в результате упрощения некоторых из A, $\trianglelefteq$ и $\lhd$. Полагая $\trianglelefteq$ и $\lhd$ постоянными в $\mathbf{1}$, имеем No-**индукцию**, которую для простоты сформулируем только для простых свойств:

- С учетом $P : \mathsf{No} \to \mathsf{Prop}$, если $P(x)$ выполняется всякий раз, когда x является сюрреалистическим числом, определяемым сечением таким, что $P(x^L)$ и $P(x^R)$ выполняются для всех L и R, то $P(x)$ выполняется для всех $x : \mathsf{No}$.

Это следует сравнить с замечанием Конвея:

> Обычно, когда надо установить истинность высказывания $P(x)$, для всех чисел x, оно доказывается индуктивно, выводя $P(x)$ из истинности всех высказываний $P(x^L)$ and $P(x^R)$. Мы рассматриваем фразу «все числа построены таким образом» как оправдание правомерности этой процедуры.

Используя No-индукцию, можно доказать

Теорема 11.6.4 (Теорема Конвея, 0)**.**

(i) *Для любого* $x : \mathsf{No}$*, верно* $x \leqslant x$.

(ii) *Для любого* $x : \mathsf{No}$*, определенного сечением, имеем* $x^L < x$ *и* $x < x^R$*, для всех* L *и* R.

Доказательство. Сначала заметим, что если $x \leqslant x$, то всякий раз, когда x встречается как левая опция некоторого сечения y, имеем $x < y$, по первому конструктору $<$, и аналогично всякий раз, когда x встречается как правая опция сечения y, имеем $y < x$, по второму конструктору $<$. В частности, (i)⇒(ii).

Докажем (i) с помощью No-индукции по x. Предположим, что x определяется сечением таким, что $x^L \leqslant x^L$ и $x^R \leqslant x^R$, для всех L и R. Но, согласно наблюдению, отмеченному выше, эти предположения подразумевают $x^L < x$ и $x < x^R$, для всех L и R, что приводит к $x \leqslant x$ конструктором для $\leqslant$. □

Следствие 11.6.5. No *есть* 0*-тип*.

Доказательство. Простое отношение $R(x, y) :\equiv (x \leqslant y) \wedge (y \leqslant x)$ подразумевает тождественность, по конструктору пути No, и содержит диагональ, по теореме 11.6.4(i). Таким образом, применима теорема 7.2.2. $\square$

В отличие от этого, теорему Конвея 1 (транзитивность $\leqslant$) несколько сложнее установить с помощью нашего определения; см. следствие 11.6.17.

Нам также понадобится принцип совместной рекурсии, $(\mathsf{No}, \leqslant, <)$**-рекурсии**. Его удобно сформулировать следующим образом. Предположим, что тип A снабжен отношениями $\trianglelefteqslant : A \to A \to \mathsf{Prop}$ и $\triangleleft : A \to A \to \mathsf{Prop}$. Тогда можно определить $f : \mathsf{No} \to A$, выполнив следующие действия.

(i) Для любого x, определенного сечением, предполагая, что $f(x^L)$ и $f(x^R)$ определены таким образом, что $f(x^L) \triangleleft f(x^R)$, для всех L и R, необходимо определить $f(x)$ (мы называем это *начальным предложением* рекурсии).

(ii) Доказать, что $\trianglelefteqslant$ *антисимметрично*: если $a \trianglelefteqslant b$ и $b \trianglelefteqslant a$, то $a = b$.

(iii) Для x, y, определенных сечениями, такими, что $x^L < y$, для всех L, и $x < y^R$, для всех R, и предполагая по индукции, что $f(x^L) \triangleleft f(y)$, для всех L, $f(x) \triangleleft f(y^R)$, для всех R, а также, что $f(x^L) \triangleleft f(x^R)$ и $f(y^L) \triangleleft f(y^R)$, для всех L и R, надо доказать, что $f(x) \trianglelefteqslant f(y)$.

(iv) Для x, y, определяемых сечениями, и L_0, такими, что $x \leqslant y^{L_0}$, и при индуктивном предположении, что $f(x) \trianglelefteqslant f(y^{L_0})$, а также, что $f(x^L) \triangleleft f(x^R)$ и $f(y^L) \triangleleft f(y^R)$, для всех L и R, надо доказать, что $f(x) \triangleleft f(y)$.

(v) Для x, y, определенных сечениями, и R_0, такими, что $x^{R_0} \leqslant y$, и при индуктивном предположении, что $f(x^{R_0}) \trianglelefteqslant f(y)$, а также, что $f(x^L) \triangleleft f(x^R)$ и $f(y^L) \triangleleft f(y^R)$, для всех L и R, надо доказать, что $f(x) \triangleleft f(y)$.

Последние три предложения можно описать короче, отметив необходимость доказательства того, что f (как определено в первом предложении) переводит $\leqslant$ в $\trianglelefteqslant$, а $<$ — в $\triangleleft$. Мы будем ссылаться на эти свойства, говоря, что f *сохраняет неравенства*. Более того, при доказательстве того, что f сохраняет неравенства, можно предположить, что конкретный экземпляр $\leqslant$ или $<$ получен из одного из его конструкторов, а также можно использовать индуктивные гипотезы о том, что f сохраняет все неравенства, появляющиеся на входе этого конструктора.

Если будут достигнуты цели в (i)–(v), то мы получим $f : \mathsf{No} \to A$, которая вычисляется на сечениях, как указано в (i), и которая сохраняет все неравенства:

$$\forall (x, y : \mathsf{No}). \Big((x \leq y) \to (f(x) \trianglelefteqslant f(y))\Big) \wedge \Big((x < y) \to (f(x) \triangleleft f(y))\Big).$$

Подобно $(\mathbb{R}_\mathsf{c}, \sim)$-рекурсии для действительных чисел Коши, этот принцип рекурсии необходим для определения функций на No, поскольку мы не можем сначала определить функцию на «пред-сюрреалистических числах» и только потом доказать, что она не нарушает понятия тождественности.

Пример 11.6.6. Определим функцию *отрицания* $\mathsf{No} \to \mathsf{No}$. Применим принцип совместной рекурсии с $A :\equiv \mathsf{No}$, $(x \trianglelefteqslant y) :\equiv (y \leqslant x)$ и $(x \triangleleft y) :\equiv (y < x)$. Ясно, что этот $\trianglelefteqslant$ антисимметричен.

Для основного предложения в приведенном определении предполагается, что x определяется сечением, где $-x^L$ и $-x^R$ определены так, что $-x^L \lhd -x^R$, для всех L и R. По определению, это означает, что $-x^R < -x^L$, для всех L и R, поэтому можно определить $-x$ сечением $\{-x^R \mid -x^L\}$. Эта нотация, следуя Конвею, относится к сечению, левые опции которого индексируются типом $\mathcal{R}$, индексирующим правые опции x, и чьи правые опции индексируются типом $\mathcal{L}$, индексирующим левые опции x с соответствующими семействами $\mathcal{R} \to \mathsf{No}$ и $\mathcal{L} \to \mathsf{No}$, определенными путем их композиции для x с отрицанием.

Теперь нужно проверить, что f сохраняет неравенства.

- Для $x \leqslant y$, можно принять $x^L < y$, для всех L, и $x < y^R$, для всех R, и показать, что $-y \leqslant -x$. Но, индуктивно можно предположить, что $-y < -x^L$ и $-y^R < -x$, которое дает желаемый результат, по определению $-y$, $-x$, и конструктору для $\leqslant$.

- Для $x < y$, в первом случае, когда оно возникает из некоторого $x \leqslant y^{L_0}$, можно индуктивно предположить, что $-y^{L_0} \leqslant -x$, и, в этом случае, $-y < -x$ следует по конструктору для $<$.

- Точно так же, если $x < y$ возникает из $x^{R_0} \trianglelefteq y$, индуктивной гипотезой будет $-y \trianglelefteq -x^R$, что снова дает $-y < -x$.

Однако, чтобы сделать гораздо больше, потребуется охарактеризовать отношения $\leqslant$ и $<$ более явно, как это было сделано для действительных чисел Коши в теореме 11.3.32. Так же, как и там, нам придется одновременно доказать пару существенных свойств этих отношений, чтобы была применена индукция.

Теорема 11.6.7. *Существуют отношения $\preccurlyeq : \mathsf{No} \to \mathsf{No} \to \mathsf{Prop}$ и $\prec : \mathsf{No} \to \mathsf{No} \to \mathsf{Prop}$ такие, что если x and y — сюрреалистические числа, определенные сечениями, то*

$$
\begin{aligned}
(x \preccurlyeq y) &:\equiv \big(\forall (L).\, x^L \prec y\big) \wedge \big(\forall (R).\, x \prec y^R\big) \\
(x \prec y) &:\equiv \big(\exists (L).\, x \preccurlyeq y^L\big) \vee \big(\exists (R).\, x^R \preccurlyeq y\big).
\end{aligned}
$$

Более того, имеет место

$$
(x \prec y) \to (x \preccurlyeq y) \tag{11.6.8}
$$

а также, и все приемлемые свойства транзитивности, превращающие $\prec$ и $\preccurlyeq$ в «бимодуль» над $\leqslant$ и $<$:

$$
\begin{array}{ll}
(x \leqslant y) \to (y \preccurlyeq z) \to (x \preccurlyeq z) & \quad (x \preccurlyeq y) \to (y \leqslant z) \to (x \preccurlyeq z) \\
(x \leqslant y) \to (y \prec z) \to (x \prec z) & \quad (x \preccurlyeq y) \to (y < z) \to (x \prec z) \\
(x < y) \to (y \preccurlyeq z) \to (x \prec z) & \quad (x \prec y) \to (y \leqslant z) \to (x \prec z).
\end{array} \tag{11.6.9}
$$

Доказательство. Определим $\preccurlyeq$ и $\prec$ двойной $(\mathsf{No}, \leqslant, <)$-индукцией по x, y. Первая индукция представляет собой простую рекурсию, ко-область которой представляет собой подмножество A из $(\mathsf{No} \to \mathsf{Prop}) \times (\mathsf{No} \to \mathsf{Prop})$, состоящее из пар предикатов, из которых одно влечет другое, и которые удовлетворяют «транзитивности справа», т.е. (11.6.8) и правому столбцу (11.6.9) с заменой $(x \preceq -)$ и $(x \prec -)$ двумя данными предикатами. Как и в доказательстве теоремы 11.3.16, мы рассматриваем эти предикаты как половину бинарных отношений, записывая их как $y \mapsto (\Diamond \preccurlyeq y)$ и $y \mapsto (\Diamond \prec y)$, где $\Diamond$ обозначает пару отношений. Снабдим A следующими двумя отношениями:

$$\begin{aligned}(\Diamond \trianglelefteq \heartsuit) &:\equiv \forall(y : \mathsf{No}). \Big((\heartsuit{\preccurlyeq}y) \to (\Diamond{\preccurlyeq}y)\Big) \wedge \Big((\heartsuit{\prec}y) \to (\Diamond{\prec}y)\Big), \\ (\Diamond \triangleleft \heartsuit) &:\equiv \forall(y : \mathsf{No}). \Big((\heartsuit{\preccurlyeq}y) \to (\Diamond{\preccurlyeq}y)\Big).\end{aligned}$$

Отметим, что $\trianglelefteq$ является антисимметричным, так как если $\Diamond \trianglelefteq \heartsuit$ и $\heartsuit \trianglelefteq \Diamond$, то $(\heartsuit{\preccurlyeq}y) \Leftrightarrow (\Diamond{\preccurlyeq}y)$ и $(\heartsuit{\prec}y) \Leftrightarrow (\Diamond{\prec}y)$, для всех y, следовательно, $\Diamond = \heartsuit$, по унивалентности для простых высказываний и функциональной экстенсиональности. Более того, утверждение, что функция $\mathsf{No} \to A$ сохраняет неравенства, в точности означает, что, рассматриваемая как пара бинарных отношений на No, она удовлетворяет «транзитивности слева» (левый столбец (11.6.9)).

Теперь, для начального предложения рекурсии мы предполагаем, что данный x определяется сечением, и предполагаем отношения $(x^L \prec -)$, $(x^R \prec -)$, $(x^L \preccurlyeq -)$ и $(x^R \preccurlyeq -)$, для всех L и R, из которых строгие отношения влекут нестрогие, удовлетворяющие транзитивности справа, и такие, что

$$\forall(L,R). \forall(y : \mathsf{No}). \Big((x^R \preccurlyeq y) \to (x^L \prec y)\Big). \tag{11.6.10}$$

Теперь нужно определить $(x \prec y)$ и $(x \preccurlyeq y)$, для всех y. Здесь, в отличие от теоремы 11.3.16, вместо вложенной рекурсии используется вложенная индукция, чтобы можно было индуктивно использовать транзитивность слева относительно неравенств $x^L < x$ и $x < x^R$. Определим $A' : \mathsf{No} \to \mathcal{U}$, приняв $A'(y)$ за подмножество A' множества $\mathsf{Prop} \times \mathsf{Prop}$, состоящее из двух простых высказываний, обозначенных $\triangle \preccurlyeq y$ и $\triangle \prec y$ (с $\triangle : A'(y)$), таких, что

$$(\triangle \prec y) \to (\triangle \preccurlyeq y) \tag{11.6.11}$$

$$\forall(L). (\triangle \preccurlyeq y) \to (x^L \prec y) \tag{11.6.12}$$

$$\forall(R). (x^R \preccurlyeq y) \to (\triangle \prec y). \tag{11.6.13}$$

Используя обозначения, аналогичные $\trianglelefteq$ и $\triangleleft$, снабдим A' двумя отношениями, определенными для $\triangle : A'(y)$ и $\square : A'(z)$ формулами

$$\begin{aligned}(\triangle \sqsubseteq \square) &:\equiv ((\triangle \preccurlyeq y) \to (\square \preccurlyeq z)) \wedge ((\triangle \prec y) \to (\square \prec z)) \\ (\triangle \sqsubset \square) &:\equiv ((\triangle \preccurlyeq y) \to (\square \prec z))\ .\end{aligned}$$

Опять же, $\sqsubseteq$, очевидно, антисимметрично в соответствующем смысле. Более того, функция $\prod_{(y:\mathsf{No})} A'(y)$, сохраняющая неравенства, есть в точности пара предикатов, один из которых влечет другой, удовлетворяющих транзитивности справа и транзитивности слева относительно неравенств $x^L < x$ и $x < x^R$. Таким образом, эта внутренняя индукция обеспечит то, что нужно для завершения начального предложения внешней рекурсии.

Для начального предложения внутренней индукции также предположим, что даны, y, определяемая сечением, и свойства $(x \prec y^L)$, $(x \prec y^R)$, $(x \preceq y^L)$, $(x \preceq y^R)$, для всех L и R, причем из строгих отношений следуют нестрогие, а также, транзитивность слева, относительно $x^L < x$ и $x < x^R$, и справа, относительно $y^L < y^R$. Теперь можно дать определения, указанные в формулировке теоремы:

$$(x \preccurlyeq y) :\equiv (\forall(L). x^L \prec y) \wedge (\forall(R). x \prec y^R), \tag{11.6.14}$$

$$(x \prec y) :\equiv (\exists(L). x \preceq y^L) \vee (\exists(R). x^R \preceq y). \tag{11.6.15}$$

Чтобы определить элемент $A'(y)$, необходимо сначала показать, что $(x \prec y) \to (x \preccurlyeq y)$. Предположение $x \prec y$ имеет два случая. С одной стороны, если существует L_0 с $x \preccurlyeq y^{L_0}$, то, в силу транзитивности справа относительно $y^{L_0} < y^R$, имеем $x \prec y^R$, для всех R. Более того, в силу транзитивности слева относительно $x^L < x$, имеем $x^L \prec y^{L_0}$, для любого L, следовательно, $x^L \prec y$, по транзитивности справа. Таким образом, $x \preccurlyeq y$.

С другой стороны, если существует R_0 с $x^{R_0} \preccurlyeq y$, то, в силу (11.6.10), имеем $x^L \prec y$, для всех L. И, по транзитивности слева и справа относительно $x < x^{R_0}$ и $y < y^R$, имеем $x \prec y^R$, для любого R. Таким образом, $x \preccurlyeq y$.

Также нужно показать, что эти определения транзитивны слева относительно $x^L < x$ и $x < x^R$. Но, если $x \preccurlyeq y$, то $x^L \prec y$, для всех L , по определению; а если имеет место $x^R \preccurlyeq y$, то $x \prec y$, также по определению.

Таким образом, (11.6.14) и (11.6.15) определяют элемент из $A'(y)$. Теперь нужно проверить, что это определение сохраняет неравенства как зависимую функцию на A', т.е. что эти отношения транзитивны справа. Напомним, что в каждом случае можно индуктивно предположить, что они транзитивны справа по отношению ко всем неравенствам, возникающим в конструкторе неравенств.

- Предположим, что $x \preccurlyeq y$ и $y \leqslant z$, последнее следует из $y^L < z$ и $y < z^R$, для всех L и R. Тогда гипотеза индукции (внутренней рекурсии), примененная к $y < z^R$, дает $x \prec z^R$, для любого R. Кроме того, по определению, $x \preccurlyeq y$ подразумевает, что $x^L \prec y$, для любого L, поэтому, по индуктивному предположению внешней рекурсии, имеем $x^L \prec z$. Таким образом, $x \preccurlyeq z$.

- Пусть $x \preccurlyeq y$ и $y < z$. Во-первых, предположим, что $y < z$ следует из $y \leqslant z^{L_0}$. Тогда гипотеза внутренней индукции, примененная к $y \leqslant z^{L_0}$, дает $x \preccurlyeq z^{L_0}$, следовательно, $x \prec z$. Во-вторых, предположим, что $y < z$ следует из $y^{R_0} \leqslant z$. Тогда, по определению, из $x \preccurlyeq y$ следует $x \prec y^{R_0}$, а затем, по гипотезе внутренней индукции для $y^{R_0} \leqslant z$, имеем $x \prec z$.

- Предположим, что $x \prec y$ и $y \leqslant z$, последнее следует из $y^L < z$ и $y < z^R$, для всех L и R. По определению, $x \prec y$ подразумевает, что просто существует R_0 с $x^{R_0} \preccurlyeq y$ или L_0 с $x \preccurlyeq y^{L_0}$. Если $x^{R_0} \preccurlyeq y$, то гипотеза внешней индукции дает $x^{R_0} \preccurlyeq z$, следовательно, $x \prec z$. Если $x \preccurlyeq y^{L_0}$, то внутренняя индуктивная гипотеза для $y^{L_0} < z$ (которая выполняется конструктором для $y \leqslant z$) дает $x \prec z$.

Этим завершается внутренняя индукция. Таким образом, для любого x, определяемого сечением, имеем $(x \prec -)$ и $(x \preccurlyeq -)$, определенные по (11.6.14) и (11.6.15), и транзитивные справа.

Для завершения внешней рекурсии, нужно убедиться, что эти определения транзитивны слева. После No-индукции по z, получаем три случая, по существу идентичных только что описанным выше для транзитивности справа. Поэтому мы их опускаем. □

Теорема 11.6.16. *Для любых $x, y : \mathsf{No}$, имеем $(x < y) = (x \prec y)$ и $(x \leqslant y) = (x \preccurlyeq y)$.*

Доказательство. Слева направо используем $(\mathsf{No}, \leqslant, <)$-индукцию, где $A(x) :\equiv \mathbf{1}$ с $\preccurlyeq$ и $\prec$ задают отношения $\trianglelefteq$ и $\triangleleft$. Во всех случаях конструкторов, x и y определяются сечениями, поэтому определения $\preccurlyeq$ и $\prec$ однозначно формируются, и применяются индуктивные гипотезы.

Справа налево используем No-индукцию, для предположения, что x и y определяются сечениями. Но теперь определения для $\preccurlyeq$ и $\prec$, а также индуктивные гипотезы, предоставляют именно те данные, которые требуются соответствующим конструкторам для $\leqslant$ и $<$. □

Следствие 11.6.17. *Отношения* $\leqslant$ *и* $<$ *на* No *удовлетворяют условию*

$$\forall(x, y : \mathsf{No}).\,(x < y) \to (x \leqslant y)$$

и являются транзитивными:

$$(x \leqslant y) \to (y \leqslant z) \to (x \leqslant z)$$
$$(x \leqslant y) \to (y < z) \to (x < z)$$
$$(x < y) \to (y \leqslant z) \to (x < z)$$

Как и в случае с действительными числами Коши, принцип совместной $(\mathsf{No}, \leqslant, <)$-рекурсии остается важным при определении всех операций над No.

Пример 11.6.18. Определим $+ : \mathsf{No} \to \mathsf{No} \to \mathsf{No}$ рекурсией по первому аргументу с последующей индукцией по второму аргументу. Для внешней рекурсии будем воспринимать ко-область как подмножество $\mathsf{No} \to \mathsf{No}$, состоящее из функций g таких, что $(x < y) \to (g(x) < g(y))$ и $(x \leqslant y) \to (g(x) \leqslant g(y))$, для всех x, y. Для такого g, а также h (с подобными условиями), определим $(g \trianglelefteq h) :\equiv \forall(x : \mathsf{No}).\, g(x) \leqslant h(x)$ и $(g \lhd h) :\equiv \forall(x : \mathsf{No}).\, g(x) < h(x)$. Ясно, что $\trianglelefteq$ антисимметрично.

Для начального предложения рекурсии предполагаем, что x определяется сечением, что функции (x^L+-) и (x^R+-) определены, сохраняют неравенства и удовлетворяют $x^L+y < x^R+y$, и определим $(x + -)$. Как и в теореме 11.6.7, вместо внутренней рекурсии будем использовать внутреннюю индукцию в семейство $A : \mathsf{No} \to \mathcal{U}$, где $A(y)$ — подмножество $z : \mathsf{No}$ таких, что $x^L + y < z$ и $x^R + y > z$. Снабдим A отношениями $\leqslant$ и $<$, индуцированными из No, так что антисимметрия очевидна. Для начального предложения внутренней рекурсии также предполагаем, что y определяется сечением, причем каждый из $x + y^L$ и $x + y^R$ определен и удовлетворяет отношениям $x^L + y^L < x + y^L$, $x^L + y^R < x + y^R$, $x + y^L < x^R + y^L$ и $x + y^R < x^R + y^R$ (они вытекают из дополнительных условий, наложенных на элементы $A(y)$), а также $x + y^L < x + y^R$ (поскольку элементы $x + y^L$ и $x + y^R$ множества $A(y)$ образуют зависимое сечение). Теперь приведем определение Конвея:

$$x + y :\equiv \left\{x^L + y, x + y^L \mid x^R + y, x + y^R\right\} .$$

Другими словами, левыми опциями $x + y$ являются все числа вида $x^L + y$ для некоторой левой опции x^L, или $x + y^L$ для некоторой левой опции y^L. Надо показать, что каждая из этих левых опций меньше, чем каждая из следующих правых опций:

- $x^L + y < x^R + y$, по гипотезе внешней индукции.
- $x^L + y < x^L + y^R < x + y^R$, первое верно, поскольку $(x^L + -)$ сохраняет неравенства, а второе верно, поскольку имеет место $x + y^R : A(y^R)$.
- $x+y^L < x^R+y^L < x^R+y$, первое верно, поскольку $x+y^L : A(y^L)$, а второе верно, поскольку $(x^R + -)$ сохраняет неравенства.
- $x + y^L < x + y^R$, по внутренней индуктивной гипотезе (в частности, по тому факту, что имеется зависимое сечение).

Также надо показать, что $x+y$, определенное таким образом, находится в $A(y)$, т.е. что $x^L+y < x + y$ и $x + y < x^R + y$; но это верно по теореме 11.6.4(ii).

Далее, следует проверить, что определение $(x + -)$ сохраняет неравенства:

- если $y \leqslant z$ возникает из того, что $y^L < z$ и $y < z^R$, для всех L и R, то внутренняя индуктивная гипотеза дает $x+y^L < x+z$ и $x+y < x+z^R$, тогда как внешние индуктивные гипотезы дают $x^L + y \leqslant x^L + z$ и $x^R + y \leqslant x^R + z$. Более того, поскольку $x^R + y$ является, по определению, правой опцией $x + y$, имеем $x + y < x^R + y$. Точно так же находим, что $x^L + z$ является левой опцией $x + z$, так что, $x^L + z < x + z$. Таким образом, используя транзитивность, имеем $x^L + y < x + z$ и $x + y < x^R + z$; поэтому можно заключить, что $x + y \leqslant x + z$, используя конструктор для $\leqslant$;

- если $y < z$ следует из L_0 с $y \leqslant z^{L_0}$, то, индуктивно, $x + y \leqslant x + z^{L_0}$, следовательно, $x + y < x + z$, поскольку $x + z^{L_0}$ является правой опцией для $x + z$;

- аналогично, если $y < z$ следует из $y^{R_0} \leqslant z$, то $x + y < x + z$, поскольку $x + y^{R_0} \leqslant x + z$.

Этим завершается внутренняя индукция. Для внешней рекурсии требуется убедиться, что $+$ также сохраняет неравенство слева. Использованием No-индукции это подтверждается.

В Приложении к части 0 из [Con76], Конвей обсуждает, как сюрреалистические числа могут быть формализованы в теории множеств ZFC: путем итерации по ординальным числам и перехода к множествам представителей низшего ранга для каждого класса эквивалентности или путем представления чисел с помощью «знаков-расширений». Затем он замечает, что

> Необычайно сложный характер этих конструкций больше говорит нам о характере формализации внутри ZF, чем о нашей системе чисел ...

и продолжает отстаивать общую теорию «допустимых видов построения», которая должна включать

(i) Объекты могут быть созданы из уже существующих объектов любым разумно конструктивным способом.

(ii) Равенство между созданными объектами может быть любым желаемым отношением эквивалентности.

Условие (i) может быть естественно прочитано как оправдывающее общие принципы *индуктивного определения*, подобные тем, которые представлены в §§ 5.6 and 5.7. В частности, условие строгой позитивности для конструкторов можно рассматривать как формализацию того, что значит быть «разумно конструктивным». Затем, условие (ii) предполагает, что его необходимо расширить до *высших* индуктивных определений всех видов, в которых можно вводить конструкторы путей, делающие объекты равными любым разумным образом. Например, в следующем абзаце Конвей говорит:

> ... мы также могли бы, например, свободно создать новый объект (x, y) и назвать его упорядоченной парой для x и y. Мы также могли бы создать упорядоченную пару $[x, y]$, отличную от (x, y), но сосуществующую с ней ... Если, вместо этого, мы захотели бы превратить (x, y) в неупорядоченную пару, то могли бы определить равенство с помощью отношения эквивалентности $(x, y) = (z, t)$ тогда и только тогда, когда $x = z, y = t$ *или* $x = t, y = z$.

Свобода вводить новые объекты с новыми именами, порожденными определенными формами конструкторов, — это как раз то, что имеется в теории индуктивных определений. Как и в случае с двумя нашими копиями натуральных чисел $\mathbb{N}$ и $\mathbb{N}'$ в §5.2, если мы запишем идентичное определение декартова типа произведения $A \times B$, то получим другой тип произведения $A \times' B$, канонические элементы которого можно было бы свободно записать как $[x, y]$. И можно было бы сделать один из них типом неупорядоченных пар, добавив подходящий конструктор пути.

Конечно, Конвей хотел бы не отвергать конкретно ZF, а выступить сразу против всех основополагающих теорий:

> ... это предложение не относится к какой-либо конкретной теории в качестве альтернативы ZF ... Вместо этого предлагается дать себе свободу создавать произвольные математические теории такого рода, но доказать метатеорему, которая гарантирует раз и навсегда, что любая такая теория может быть формализована в терминах любой из стандартных основополагающих теорий.

Можно было бы возразить, что на самом деле унивалентные основания — это не одна из «стандартных фундаментальных теорий», которую имел в виду Конвей, а скорее *метатеория*, в которой можно выразить нашу способность создавать новые теории и относительно которой можно доказать метатеорему Конвея. Например, сюрреалистические числа — это одна из «математических теорий», которую имеет в виду Конвей, и мы видели, что их можно сконструировать и обосновать в рамках унивалентных оснований. Точно так же, Конвей ранее заметил, что

> ... теория множеств была бы такой теорией, где множества конструируются из уже созданных с помощью способов, соответствующих обычным аксиомам, а отношение равенства соответствует тому, что они имеют одни и те же члены.

Это описание близко соответствует более индуктивной конструкции кумулятивной иерархии теории множеств в §10.5. Тогда метатеорема Конвея соответствовала бы тому факту, на который мы уже несколько раз ссылались, что мы можем построить модель унивалентных оснований внутри ZFC (что выходит за рамки этой книги).

Однако унивалентные основания настолько богаты и сильны сами по себе, что было бы безрассудно относить их только к метатеории, в которой можно строить теории, подобные множествам. Мы видели, что даже на уровне множеств (0-типов) высшие индуктивные типы в унивалентных основаниях дают прямые конструкции объектов по их универсальным свойствам (§6.11), такие как конструктивная теория пополнения Коши (§11.3). Но самое главное, возможность моделировать гомотопическую теорию и теорию категорий непосредственно в фундаментальной системе (главы 8 и 9) дает унивалентным основаниям преимущество, с которым не может сравниться никакой теоретико-множественный фундамент.

Примечания

Определение алгебраических операций над действительными числами Дедекинда, особенно умножения, несколько сложно и утомительно. Есть несколько способов заставить работать арифметику: у каждого есть свои преимущества, но все они, похоже, требуют некоторой технической работы. Например, Ричман (Richman) [Ric08] определяет умножение действительных чисел Дедекинда сначала на положительных сечениях, а затем алгебраически распространяет его на все сечения Дедекинда, в то время как Конвей (Conway) [Con76] заметил, что определение умножения для сюрреалистических чисел хорошо работает для действительных чисел Дедекинда.

Наша трактовка действительных чисел Дедекинда заимствует многие идеи из [BT09], где числа Дедекинда построены в контексте ASD («абстрактной двойственности Стоуна (Stone)»).

Это (ограниченная) форма простого типизированного λ-исчисления с выделенным объектом Σ, которая классифицирует открытые множества, а по двойственности, и закрытые. В [BT09] также можно найти подробные доказательства основных свойств арифметических операций.

Тот факт, что $\mathbb{R}_\mathsf{c}$ является наименьшим полным по Коши архимедовым упорядоченным полем, как было доказано в теореме 11.3.50, указывает на то, что наши действительные числа Коши, вероятно, совпадают с полем действительных чисел Эскардо-Симпсона (Escardó-Simpson)[ES01] Было бы интересно проверить, так ли это на самом деле. Понятие действительных чисел Эскардо-Симпсона, или, точнее, соответствующего замкнутого интервала, интересно тем, что его можно сформулировать в любой категории с конечными произведениями.

В конструктивной теории множеств, дополненной «аксиомой регулярного расширения», можно также попытаться определить пополнение по Коши замыканием под пределами последовательностей Коши с трансфинитной итерацией. Было бы также интересно проверить, согласуется ли это построение с нашим.

Можно отнести к конструктивному фольклору то, что совпадение действительных чисел Коши и Дедекинда требует зависимого выбора, но менее известно, что достаточно исчисляемого выбора. Напомним, что **зависимый выбор** утверждает, что для полного отношения R на A, под которым мы понимаем $\forall (x : A).\, \exists (y : A).\, R(x, y)$, и для любого $a : A$, просто существует $f : \mathbb{N} \to A$ такая, что $f(0) = a$ и $R(f(n), f(n+1))$, для всех $n : \mathbb{N}$. Следствие 11.4.3 использует типичный прием для преобразования применения зависимого выбора в приложение, использующее исчисляемый выбор. А именно, мы используем исчисляемый выбор один раз, чтобы заранее произвести все возможные варианты выбора, а затем используем функцию выбора, чтобы избежать зависимых выборов.

Сложная взаимосвязь между различными понятиями компактности в конструктивном контексте обсуждается в [BIS02]. У Палмгрена (Palmgren) [Pal07] имеется хорошее сравнение поточечного анализа и бесточечной топологии.

Сюрреалистические числа были определены в [Con76], с использованием своего рода индуктивного определения, но без явного обоснования его с точки зрения какой-либо фундаментальной системы. По этой причине некоторые более поздние авторы стремились использовать расширения знаков или другие более явные представления, которые можно было бы более явно закодировать в теории множеств. Идею представления их в теории типов впервые рассмотрел Хэнкок (Hancock), а Сетцер (Setzer) и Форсберг (Forsberg) [FS12] отметили, что сюрреалистические числа и их отношения неравенства $<$ и $\leqslant$ естественным образом образуют индуктивно-индуктивное определение. Представленная у нас *высшая* индуктивно-индуктивная версия, которая строит правильное понятие равенства для сюрреалистических чисел, является новой.

Упражнения

Упражнение 11.1. Дайте альтернативное определение действительных чисел Дедекинда, сначала определив квадрат, а затем используя уравнение. (11.3.45). Убедитесь, что получается коммутативное кольцо.

Упражнение 11.2. Предположим, что условие ограниченности (i) в определении 11.2.1 удалено. Тогда мы получаем **расширенные действительные числа**, которые содержат $-\infty :\equiv (\mathbf{0}, \mathbb{Q})$ и $\infty :\equiv (\mathbb{Q}, \mathbf{0})$. Какие определения арифметических операций над сечениями все еще имеют смысл для расширенных действительных чисел? Какую алгебраическую структуру мы получили?

Упражнение 11.3. Рассматривая односторонние сечения, получаем **нижние** и **верхние** действительные числа Дедекинда, соответственно. Например, нижнее действительное число задается предикатом $L : \mathbb{Q} \to \Omega$, который

(i) *заселен*: $\exists(q : \mathbb{Q}).\, L(q)$ и

(ii) *округлен*: $L(q) = \exists(r : \mathbb{Q}).\, q < r \wedge L(r)$.

(можно было бы также потребовать $\exists(r : \mathbb{Q}).\, \neg L(r)$, чтобы исключить сечение $\infty :\equiv \mathbb{Q}$). Какие арифметические операции можно определить над нижними действительными числами? В частности, что происходит с противоположным элементом?

Упражнение 11.4. Предположим, что в определении 11.2.1 удалено условие локализации. Тогда получим **интервальную область** $\mathbb{I}$, поскольку сечения могут иметь «пустоты», которые являются просто интервалами. Определим частичный порядок $\sqsubseteq$ на $\mathbb{I}$ как

$$((L, U) \sqsubseteq (L', U')) :\equiv (\forall(q : \mathbb{Q}).\, L(q) \Rightarrow L'(q)) \wedge (\forall(q : \mathbb{Q}).\, U(q) \Rightarrow U'(q))\,.$$

Каковы максимальные элементы $\mathbb{I}$ относительно $\mathbb{I}$? Определите операции «концы», которые присваивают элементу интервальной области его нижнюю и верхнюю конечные точки. Являются ли концевые действительные числа нижними или верхними действительными числами (см. упражнение 11.3)? Какие определения арифметических операций над сечениями по-прежнему имеют смысл для интервальной области?

Упражнение 11.5. Покажите, что, для всех $x, y : \mathbb{R}_\mathsf{d}$,

$$\neg(x < y) \Rightarrow y \leqslant x$$

и

$$(x \leqslant y) \simeq \left(\prod_{\epsilon : \mathbb{Q}_+} x < y + \epsilon\right)\,.$$

Подразумевает ли $\neg(x \leqslant y)$, что $y < x$?

Упражнение 11.6.

(i) Предполагая исключение третьего, постройте непостоянное отображение $\mathbb{R}_\mathsf{d} \to \mathbb{Z}$.

(ii) Предположим, что $f : \mathbb{R}_\mathsf{d} \to \mathbb{Z}$ — это отображение такое, что $f(0) = 0$ и $f(x) \neq 0$, для всех $x > 0$. Выведите отсюда ограниченный принцип всеведения (11.5.8).

Упражнение 11.7. Покажите, что в упорядоченном поле F, плотность $\mathbb{Q}$ и традиционная аксиома Архимеда эквивалентны:

$$(\forall(x, y : F).\, x < y \Rightarrow \exists(q : \mathbb{Q}).\, x < q < y) \Leftrightarrow (\forall(x : F).\, \exists(k : \mathbb{Z}).\, x < k).$$

Упражнение 11.8. Предположим, что $a, b : \mathbb{Q}$ и $f : \{q : \mathbb{Q} \mid a \leqslant q \leqslant b\} \to \mathbb{R}_\mathsf{c}$ являются лифшицевыми с константой L. Покажите, что существует единственное расширение $\bar{f} : [a, b] \to \mathbb{R}_\mathsf{c}$ для f, являющееся расширением Лифшица с константой L. Подсказка: вместо изменения леммы 11.3.15 для замкнутых интервалов, заметим, что существует ретракция $r : \mathbb{R}_\mathsf{c} \to [-n, n]$, и применим лемму 11.3.15 к $f \circ r$.

Упражнение 11.9. Обобщите конструкцию $\mathbb{R}_\mathsf{c}$, чтобы построить пополнение Коши любого метрического пространства. Во-первых, подумайте, какое понятие действительных чисел является наиболее естественным в качестве ко-области для функции расстояния метрического пространства. Это важно? Далее, проработайте детали двух конструкций:

(i) Следуя построению действительных чисел Коши, определите пополнение метрического пространства как индуктивно-индуктивный тип, замкнутый относительно пределов последовательностей Коши.

(ii) Воспользуйтесь следующей конструкцией Ловера (Lawvere) [Law74] и Рихмана (Richman) [Ric00], где в качестве типа **местоположений** задано пополнение метрического пространства (M, d). Местоположение — это функция $f : M \to \mathbb{R}$ такая, что

(a) $f(x) \geqslant |f(y) - d(x, y)|$ для всех $x, y : M$, и

(b) $\mathsf{inf}_{x \in M} f(x) = 0$, под которым подразумевается $\forall(\epsilon : \mathbb{Q}_+). \exists(x : M). \; |f(x)| < \epsilon$ и $\forall(x : M). \, f(x) \geqslant 0$.

Идея состоит в том, что f выглядит, как измеряющая расстояние от точки.

Наконец, докажите следующее универсальное свойство метрических пополнений: локально равномерно непрерывное отображение из метрического пространства в полное по Коши метрическое пространство однозначно продолжается до локально равномерно непрерывного отображения на пополнении (говорим, что отображение **локально равномерно непрерывно**, если оно равномерно непрерывно на открытых шарах).

Упражнение 11.10. **Принцип Маркова** гласит, что для всех $f : \mathbb{N} \to \mathbf{2}$,

$$(\neg\neg\exists(n : \mathbb{N}). \, f(n) = 1_\mathbf{2}) \Rightarrow \exists(n : \mathbb{N}). \, f(n) = 1_\mathbf{2} \, .$$

Это частный случай закона двойного отрицания (3.4.2). Покажите, что $\forall(x, y : \mathbb{R}_\mathsf{d}). \, x \neq y \Rightarrow x \# y$ подразумевает принцип Маркова. Верно ли и обратное?

Упражнение 11.11. Убедитесь, что для действительных чисел выполняется свойство «нет делителей нуля»: $xy \# 0 \Leftrightarrow x \# 0 \wedge y \# 0$.

Упражнение 11.12. Пусть $(q_1, r_1), \ldots, (q_n, r_n)$ поточечно покрывает (a, b). Тогда существует $\epsilon : \mathbb{Q}_+$ такое, что всякий раз, когда $a < x < y < b$ и $|x - y| < \epsilon$, просто существует i такое, что $q_i < x < r_i$ и $q_i < y < r_i$. Такое ϵ называется **числом Лебега** для данного покрытия.

Упражнение 11.13. Докажите следующую приближенную версию теоремы о промежуточном значении:

Если $f : [0, 1] \to \mathbb{R}$ равномерно непрерывна и $f(0) < 0 < f(1)$, то для любого $\epsilon : \mathbb{Q}_+$ просто существует $x : [0, 1]$ такой, что $|f(x)| < \epsilon$.

Подсказка: не пытайтесь использовать метод деления пополам, поскольку он приводит к аксиоме выбора. Вместо этого аппроксимируйте f кусочно-линейным отображением. Как построить такое отображение?

Упражнение 11.14. Проверьте, все ли в [Knu74] можно сделать с помощью высших индуктивно-индуктивных сюрреалистических чисел из §11.6.

Упражнение 11.15. Напомним, что функция $\iota_{\mathbb{R}_\mathsf{d}} : \mathbb{R}_\mathsf{d} \to \mathsf{No}$ определена на с. 426.

(i) Покажите, что $\iota_{\mathbb{R}_\mathsf{d}}$ инъективна.

(ii) Существуют очевидные расширения $\iota_{\mathbb{R}_\mathsf{d}}$ на расширенные действительные числа (упражнение 11.2}) и интервальную область (упражнение 11.4). Они инъективны?

Упражнение 11.16. Покажите, что определенная функция $\iota_{\mathsf{Ord}} : \mathsf{Ord} \to \mathsf{No}$ на с. 426 является инъективной тогда и только тогда, когда выполняется LEM.

Упражнение 11.17. Определите тип POrd, снабженный бинарными отношениями $\leqslant$ и $<$, имитируя определение No, но используя только левые опции.

(i) Постройте отображение $j : \mathsf{POrd} \to \mathsf{No}$ и покажите, что оно является вложением.

(ii) Покажите, что POrd является ординалом (в следующем, более высоком универсуме, как Ord) при отношении $<$.

(iii) Предполагая пропозициональное изменение размера, покажите, что POrd эквивалентно подмножеству
$$\{A : \mathsf{Ord} \mid \mathsf{isPlump}(A)\}$$
в Ord из упражнения 10.14. Заключите, что $\iota_{\mathsf{Ord}} : \mathsf{Ord} \to \mathsf{No}$ является инъективным при ограничении гладкими ординалами (см. упражнение 10.14).

В отсутствие пропозиционального изменения размера по-прежнему можно называть элементы POrd (или их образы в No) гладкими ординалами.

Упражнение 11.18. Определите сюрреалистическое число как **псевдо-ординальное**, если оно есть сечение $\{x^L \mid \}$ без правых опций (но его левые опции, сами по себе, могут иметь правые опции) . Покажите, что утверждение «каждый псевдо-ординал является гладким ординалом» эквивалентно LEM.

Упражнение 11.19. Заметим, что и в теореме 11.6.7, и в примере 11.6.18 используется аналогичный шаблон для определения функции $\mathsf{No} \to \mathsf{No} \to B$: внешняя No-рекурсия, ко-область которой является множеством функций, сохраняющих порядок, $\mathsf{No} \to B$, с последующей внутренней No-индукцией в семейство $A : \mathsf{No} \to \mathcal{U}$, где $A(y)$ — подмножество в Bc гарантией, что неравенства $x^L < x$ и $x < x^R$ также сохраняются. Сформулируйте и докажите общий принцип «двойной No-рекурсии», обобщающий эти доказательства.

Приложение A

Формальная теория типов

Подобно тому, как можно развивать математику в теории множеств без явного использования аксиом теории множеств Цермело-Френкеля, в этой книге мы развили математику на унивалентных основаниях без явного обращения к формальной системе гомотопической теории типов. Тем не менее, важно иметь точное описание гомотопического теории типов как формальной системы, чтобы, например,

- изложить и доказать его метатеоретические свойства, включая логическую непротиворечивость,
- строить модели, например в симплициальных множествах, модельных категориях, высших топосах и т.д., и
- реализовать его в помощниках по доказательствам, таких как Coq или Agda.

Даже логическая непротиворечивость гомотопической теории типов, а именно то, что в пустом контексте нет терма $a : \mathbf{0}$, не очевидна: если бы мы ошибочно выбрали определение эквивалентности, для которого $\mathbf{0} \simeq \mathbf{1}$, то унивалентность будет означать, что $\mathbf{0}$ содержит элемент, поскольку $\mathbf{1}$ обитаем. Также не очевидно, что, например, наше определение $\mathbb{S}^1$, как более высокого индуктивного типа, порождает тип, который ведет себя как обычная окружность.

Имеются два аспекта теории типов, которые необходимо уточнить, прежде чем решать такие вопросы. Напомним из введения, что теория типов включает набор правил, определяющих, когда суждения $a : A$ и $a \equiv a' : A$ выполняются — например, произведения характеризуются правилом, что всякий раз, когда $a : A$ и $b : B$, $(a, b) : A \times B$. Чтобы сделать это точным, сначала надо тщательно определить синтаксис термов — объектов $a, a', A, \ldots$, которые связаны с этими суждениями; затем мы должны точно определить суждения и правила их вывода — способ, которым суждения могут быть выведены из других суждений.

В этом приложении мы представляем две формулировки теории типов Мартина-Лёфа (Martin-Löf) и расширений, составляющих гомотопическую теорию типов. Первое представление (Приложение A.1) описывает синтаксис термов и формы суждений как расширение нетипизированного λ-исчисления, оставляя правила вывода неформальными. Во втором (Приложение A.2) представлении термины, суждения и правила вывода определяются индуктивно в стиле естественной дедукции, как это принято во многих теоретико-ти́повых публикациях.

Предисловие

В главе 1 были представлены два основных **суждения** теории типов. Первое, $a : A$, означает, что терм a является типом A. Второе, $a \equiv b : A$, утверждает, что два терма a и b **равны как суждения** для типа A. Эти суждения индуктивно определяются набором правил вывода, описанных в Приложении А.2.

Чтобы построить элемент a типа A, нужно вывести $a : A$; в этой книге мы приводим неформальные аргументы, которые описывают построение a, но формально необходимо указать определенный терм a и полный вывод того, что $a : A$.

Однако, основное различие между изложением теории типов в книге и в этом приложении состоит в том, что здесь суждения явно формулируются в окружающем **контексте**, или списке допущений, в виде

$$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n \,.$$

Элемент $x_i : A_i$ этого контекста выражает предположение, что переменная x_i имеет тип A_i. Переменные $x_1, \ldots, x_n$, появляющиеся в контексте, должны быть разными. Для указания контекстов будем использовать символы Γ и Δ.

То, что суждение $a : A$ находится в контексте Γ, записывается в виде

$$\Gamma \vdash a : A$$

и означает, что $a : A$ верно при предположениях, перечисленных в Γ. Когда список предположений пуст, мы просто пишем

$$\vdash a : A$$

или

$$\cdot \vdash a : A$$

где $\cdot$ обозначает пустой контекст. То же самое относится к равенству суждений

$$\Gamma \vdash a \equiv b : A \,.$$

Однако такие суждения имеют смысл только в **правильно построенных** контекстах, понятии, зафиксированном нашим третьим, заключительным суждением

$$(x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n) \ \mathsf{ctx}$$

выражающим то, что каждый A_i является типом в контексте $x_1 : A_1, x_2 : A_2, \ldots, x_{i-1} : A_{i-1}$. В частности, поэтому, если $\Gamma \vdash a : A$ и Γ ctx, то утверждается, что каждый A_i содержит только переменные $x_1, \ldots, x_{i-1}$, и что a и A содержат только переменные $x_1, \ldots, x_n$.

В неформальных математических представлениях контекст присутствует неявно. На каждом этапе доказательства математик знает, какие переменные доступны и какие типы они имеют, либо по общепринятому соглашению (n — обычно число, f — это функция и т.д.), либо потому, что переменные явно вводятся такими предложениями, как «пусть x будет действительным числом». Мы обсуждаем некоторые преимущества использования явных контекстов в Приложениях А.2.4 и А.2.5.

Будем обозначать через $B[a/x]$ **подстановку** вместо терма a свободных вхождений переменной x в терме B, с возможным переименованием связанных переменных, во избежание проблемы конфликта имен, как описано в §1.2. Общая форма подстановки

$$B[a_1, \ldots, a_n / x_1, \ldots, x_n]$$

заменяет выражения $a_1, \ldots, a_n$ на переменные $x_1, \ldots, x_n$ одновременно.

Связывание переменной x в выражении B означает включение их обеих в более крупное выражение, называемое **абстракцией**, цель которой — выразить тот факт, что x является «локальной» для B, т.е. ее не следует путать с другими вхождениями x, появляющимися где-либо еще. Связанные переменные знакомы программистам, но не математикам. Для связывания используются различные обозначения, такие как $x \mapsto B$, $\lambda x.\, B$, $x\,.\,B$, в зависимости от ситуации. Можно записать $C[a]$ для подстановки вместо терма a переменной в абстрактном выражении, то есть можно определить $(x.B)[a]$ как $B[a/x]$. Как обсуждалось в §1.2, изменение имени связанной переменной повсюду в выражении («α-конверсия») не меняет выражения. Таким образом, если быть очень точным, выражение — это класс эквивалентности синтаксических форм, которые различаются именами связанных переменных.

Можно также рассматривать каждую переменную x_i суждения

$$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n \vdash a : A$$

как связанную в своей **области**, состоящей из выражений $A_{i+1}, \ldots, A_n$, a и A.

А.1 Первая нотация

Объекты и типы нашей теории типов могут быть записаны как термы с использованием следующего синтаксиса, который является расширением λ-исчисления *переменными* $x, x', \ldots$, *примитивными константами* $c, c', \ldots$, *определенными константами* $f, f', \ldots$, и терм-образующими действиями, согласно рекурсивному определению

$$t ::= x \mid \lambda x.\, t \mid t(t') \mid c \mid f$$

Используемая здесь нотация означает, что терм t является, либо переменной x, либо имеет форму $\lambda x.\, t$, где x — переменная, а t — терм, или имеет вид $t(t')$, где t и t' являются термами, или это примитивная константа c, либо определенная константа f. Синтаксические маркеры «λ», «(«, «)» и «.» являются своеобразными знаками препинания.

Мы используем $t(t_1, \ldots, t_n)$ как сокращение для повторяющейся применимости $t(t_1)(t_2)\ldots(t_n)$. Также можем использовать *инфиксную* запись $t_1 \star t_2$ вместо $\star(t_1, t_2)$, когда $\star$ является примитивной или определенной константой.

Каждая определенная константа имеет ноль, одно или несколько **определяющих уравнений**. Существуют два вида определяемых констант. *Явная* определенная константа f имеет одно определяющее уравнение

$$f(x_1, \ldots, x_n) :\equiv t\,,$$

где t не включает в себя f. Например, можно было бы ввести явную определенную константу $\circ$ с определяющим уравнением

$$\circ(x, y)(z) :\equiv x(y(z))\,,$$

и использовать инфиксную запись $x \circ y$ вместо $\circ(x, y)$. Это, конечно, является просто композицией функций.

Второй тип определенных констант используется для указания (параметризованного) отображения $f(x_1, \ldots, x_n, x)$, где x ранжируется по типу, элементы которого генерируются нулем или несколькими примитивными константами. Для каждой такой примитивной константы c существует определяющее уравнение вида

$$f(x_1, \ldots, x_n, c(y_1, \ldots, y_m)) :\equiv t\,,$$

где f может входить в t, но только таким образом, чтобы было ясно, что уравнения задают всюду определенную функцию. Примерами парадигмы таких определенных функций являются функции, задаваемые примитивной рекурсией на натуральных числах. Мы можем назвать такое определение функции *общерекурсивным определением*. В информатике и логике такое определение функции для рекурсивного типа данных было названо **определением структурной рекурсией**.

Конвертируемость $t \downarrow t'$ между термами t и t' — это отношение эквивалентности, порожденное определяющими уравнениями для констант, правило вычисления

$$(\lambda x.\, t)(u) :\equiv t[u/x]\,,$$

и правила, которые делают его *конгруэнтностью* по отношению к применению (функции) и λ-абстракции:

- если $t \downarrow t'$ и $s \downarrow s'$, то $t(s) \downarrow t'(s')$, и
- если $t \downarrow t'$, то $(\lambda x.\, t) \downarrow (\lambda x.\, t')$.

Суждение о равенстве $t :\equiv u : A$ затем выводится по следующему единственному правилу:

- если $t : A$, $u : A$ и $t \downarrow u$, то $t :\equiv u : A$.

Тождественность суждений — это отношение эквивалентности.

Обратите внимание, что теория типов в этом представлении отличается от теории типов, использованной в основной части текста, в том, что она не включает принцип уникальности суждения $f :\equiv (\lambda x.\, f(x))$ для функций. Такое равенство требует, чтобы тождественность суждений была чувствительной к типу задействованных термов, поскольку это равенство имеет смысл только тогда, когда известно, что f является функцией, тогда как в этом представлении отношение конвертируемости не зависит от типа. Второе представление в Приложении A.2 включает принцип уникальности.

A.1.1 Универсумы типов

Мы постулируем иерархию **универсумов**, обозначаемых примитивными константами

$$\mathcal{U}_0, \quad \mathcal{U}_1, \quad \mathcal{U}_2, \quad \ldots$$

Первые два правила для универсумов говорят, что они образуют кумулятивную иерархию типов:

- $\mathcal{U}_m : \mathcal{U}_n$ для $m < n$,
- если $A : \mathcal{U}_m$ и $m \leqslant n$, то $A : \mathcal{U}_n$,

а третий выражает идею о том, что объект универсума может служить типом и стоять справа от двоеточия в суждениях:

- если $\Gamma \vdash A : \mathcal{U}_n$, и x — новая переменная,[1] то $\vdash (\Gamma, x : A)\ \mathsf{ctx}$.

В основной части книги суждение о равенстве $A :\equiv B : \mathcal{U}_n$ между типами A и B обычно сокращается до $A :\equiv B$. Это пример типовой двусмысленности, поскольку всегда можно переключиться на более крупный универсум, что, однако, не влияет на обоснованность суждения.

Следующее правило преобразования позволяет заменять тип на другой, равный первому, при типировании суждения:

- если $a : A$ и $A :\equiv B$, то $a : B$.

[1] Под «новой» подразумевается, что переменная не появляется в Γ или A.

A.1.2 Зависимые функциональные типы (Π-типы)

Мы вводим примитивную константу c_Π, но записываем $c_\Pi(A, \lambda x. B)$ как $\prod_{(x:A)} B$. Суждения относительно таких выражений и выражений вида $\lambda x. b$ вводятся по следующим правилам:

- если $\Gamma \vdash A : \mathcal{U}_n$ и $\Gamma, x : A \vdash B : \mathcal{U}_n$, то $\Gamma \vdash \prod_{(x:A)} B : \mathcal{U}_n$
- если $\Gamma, x : A \vdash b : B$, то $\Gamma \vdash (\lambda x. b) : (\prod_{(x:A)} B)$
- если $\Gamma \vdash g : \prod_{(x:A)} B$ и $\Gamma \vdash t : A$, то $\Gamma \vdash g(t) : B[t/x]$

Если x не входит свободно в B, мы сокращаем $\prod_{(x:A)} B$ как независимый функциональный тип $A \to B$ и выводим следующее правило:

- если $\Gamma \vdash g : A \to B$ и $\Gamma \vdash t : A$, то $\Gamma \vdash g(t) : B$

Используя независимые функциональные типы и оставляя неявным контекст Γ, приведенные выше правила можно записать в следующем альтернативном стиле, который мы используем в оставшейся части этого раздела приложения:

- если $A : \mathcal{U}_n$ и $B : A \to \mathcal{U}_n$, то $\prod_{(x:A)} B(x) : \mathcal{U}_n$
- если $x : A \vdash b : B(x)$, то $\lambda x. b : \prod_{(x:A)} B(x)$
- если $g : \prod_{(x:A)} B(x)$ и $t : A$, то $g(t) : B(t)$

A.1.3 Зависимые типы пар (Σ-типы)

Введем примитивные константы c_Σ и c_{pair}. Выражение вида $c_\Sigma(A, \lambda a. B)$ запишем как $\sum_{(a:A)} B$, а выражение вида $c_{\mathsf{pair}}(a, b)$ — как (a, b). Мы пишем $A \times B$ вместо $\sum_{(x:A)} B$, если x не входит свободно в B.

Суждения относительно таких выражений вводятся по следующим правилам:

- если $A : \mathcal{U}_n$ и $B : A \to \mathcal{U}_n$, то $\sum_{(x:A)} B(x) : \mathcal{U}_n$
- если, в дополнение, $a : A$ и $b : B(a)$, то $(a, b) : \sum_{(x:A)} B(x)$

Если имеются A и B, как выше, $C : (\sum_{(x:A)} B(x)) \to \mathcal{U}_m$, и

$$d : \prod_{(x:A)} \prod_{(y:B(x))} C((x, y)),$$

можно ввести определенную константу

$$f : \prod_{(p:\sum_{(x:A)} B(x))} C(p)$$

с определяющим уравнением

$$f((x, y)) :\equiv d(x, y).$$

Заметим, что C, d, x и y могут содержать дополнительные неявные параметры $x_1, \ldots, x_n$, если они были получены в некотором непустом контексте; поэтому, в целом, детальная схема рекурсии имеет вид

$$f(x_1, \ldots, x_n, (x(x_1, \ldots, x_n), y(x_1, \ldots, x_n))) :\equiv d(x_1, \ldots, x_n, (x(x_1, \ldots, x_n), y(x_1, \ldots, x_n))).$$

A.1.4 Типы копроизведений

Введем примитивные константы c_+, c_{inl} и c_{inr}. Будем записывать $A+B$ вместо $c_+(A,B)$, $\mathsf{inl}(a)$ вместо $c_{\mathsf{inl}}(a)$, и $\mathsf{inr}(a)$ вместо $c_{\mathsf{inr}}(a)$:

- если $A, B : \mathcal{U}_n$, то $A + B : \mathcal{U}_n$
- кроме того, $\mathsf{inl} : A \to A + B$ и $\mathsf{inr} : B \to A + B$

Если имеются A и B, как указано выше, $C : A{+}B \to \mathcal{U}_m$, $d : \prod_{(x:A)} C(\mathsf{inl}(x))$ и $e : \prod_{(y:B)} C(\mathsf{inr}(y))$, то можно ввести определенную константу $f : \prod_{(z:A+B)} C(z)$ с определяющими уравнениями

$$f(\mathsf{inl}(x)) :\equiv d(x) \qquad \text{и} \qquad f(\mathsf{inr}(y)) :\equiv e(y)\,.$$

A.1.5 Конечные типы

Введем примитивные константы $\star$, $\mathbf{0}$, $\mathbf{1}$, удовлетворяющие правилам:

- $\mathbf{0} : U_0$, $\mathbf{1} : U_0$
- $\star : \mathbf{1}$

Для $C : \mathbf{0} \to \mathcal{U}_n$ можно ввести определенную константу $f : \prod_{(x:\mathbf{0})} C(x)$ без определяющих уравнений.

Для $C : \mathbf{1} \to \mathcal{U}_n$ и $d : C(\star)$ введем определенную константу $f : \prod_{(x:\mathbf{1})} C(x)$ с определяющим уравнением $f(\star) :\equiv d$.

A.1.6 Натуральные числа

Тип натуральных чисел получается путем введения примитивных констант $\mathbb{N}$, 0 и suc по следующим правилам:

- $\mathbb{N} : \mathcal{U}_0$
- $0 : \mathbb{N}$
- $\mathsf{suc} : \mathbb{N} \to \mathbb{N}$

Кроме того, можно определять функции с помощью примитивной рекурсии. Если имеется $C : \mathbb{N} \to \mathcal{U}_k$, то можно ввести определенную константу $f : \prod_{(x:\mathbb{N})} C(x)$ всякий раз, когда имеются

$$\begin{aligned} &d : C(0) \\ &e : \textstyle\prod_{(x:\mathbb{N})}(C(x) \to C(\mathsf{suc}(x))) \end{aligned}$$

с определяющими уравнениями

$$f(0) :\equiv d \qquad \text{и} \qquad f(\mathsf{suc}(x)) :\equiv e(x, f(x))\,.$$

A.1.7 W-типы

Для W-типов введем примитивные константы c_{W} и c_{sup}. Выражение вида $c_{\mathsf{W}}(A, \lambda x.\, B)$ запишем как $\mathsf{W}_{(x:A)}B$, а $c_{\mathsf{sup}}(x,u)$ —как $\mathsf{sup}(x,u)$:

- если $A : \mathcal{U}_n$ и $B : A \to \mathcal{U}_n$, то $\mathsf{W}_{(x:A)}B(x) : \mathcal{U}_n$
- если, к тому же, $a : A$ и $u : B(a) \to \mathsf{W}_{(x:A)}B(x)$, то $\mathsf{sup}(a,u) : \mathsf{W}_{(x:A)}B(x)$

Здесь также можно определять функции с помощью полной рекурсии. Если имеются A и B, как указано выше, и $C : (\mathsf{W}_{(x:A)}B(x)) \to \mathcal{U}_m$, то можно ввести определенную константу $f : \prod_{(z:\mathsf{W}_{(x:A)}B(x))} C(z)$, если имеет место

$$d : \prod_{(a:A)} \prod_{(u:B(a)\to \mathsf{W}_{(x:A)}B(x))}((\prod_{(y:B(a))}C(u(y))) \to C(\mathsf{sup}(a,u)))$$

с определяющим уравнением

$$f(\mathsf{sup}(a,u)) :\equiv d(a,u,f\circ u)\,.$$

A.1.8 Типы тождественностей

Введем примитивные константы $c_=$ и c_{refl}. Будем записывать $a =_A b$ вместо $c_=(A,a,b)$ и refl_a вместо $c_{\mathsf{refl}}(A,a)$, когда $a : A$ понимается как:

- если $A : \mathcal{U}_n$, $a : A$ и $b : A$, то $a =_A b : \mathcal{U}_n$
- если $a : A$, то $\mathsf{refl}_a : a =_A a$

Для $a : A$, если $y : A,\ z : a =_A y \vdash C : \mathcal{U}_m$ и $\vdash d : C[a, \mathsf{refl}_a/y, z]$, то можно ввести определенную константу

$$f : \prod_{(y:A)} \prod_{(z:a=_A y)} C$$

с определяющим уравнением

$$f(a, \mathsf{refl}_a) :\equiv d\,.$$

A.2 Вторая нотация

В этом разделе используются три вида суждений

$$\Gamma\ \mathsf{ctx} \qquad\qquad \Gamma \vdash a : A \qquad\qquad \Gamma \vdash a \equiv a' : A$$

которые мы специфицируем предоставлением правил вывода. Типичное **правило вывода** имеет вид

$$\frac{\mathcal{J}_1 \qquad \cdots \qquad \mathcal{J}_k}{\mathcal{J}}\ \textsc{Name}$$

Оно гласит, что можно сделать **заключение** $\mathcal{J}$, при условии, что уже выведены **гипотезы** $\mathcal{J}_1, \ldots, \mathcal{J}_k$ (отметим, что, будучи суждениями, а не типами, они не являются *внутренними* гипотезами теории типов в смысле §1.1; вместо этого они являются гипотезами дедуктивной системы, т.е. метатеорией). Справа мы пишем название Name правила, также здесь могут присутствовать дополнительные побочные условия, которые необходимо проверить перед применением правила.

Выведение суждения — это дерево, построенное из таких правил вывода, с суждением в корне дерева. Например, с помощью правил, приведенных ниже, следующее является производным от $\cdot \vdash \lambda x.\, x : \mathbf{1} \to \mathbf{1}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdot\ \mathsf{ctx}}\ \text{ctx-EMP}}{\vdash \mathbf{1} : \mathcal{U}_0}\ \mathbf{1}\text{-FORM}}{x{:}\mathbf{1}\ \mathsf{ctx}}\ \text{ctx-EXT}}{x{:}\mathbf{1} \vdash x : \mathbf{1}}\ \mathsf{Vble}}{\cdot \vdash \lambda x.\, x : \mathbf{1} \to \mathbf{1}}\ \prod\text{-INTRO}$$

A.2.1 Контексты

Контекст — это список

$$x_1 : A_1, x_2 : A_2, \ldots, x_n : A_n$$

который указывает, что различные переменные $x_1, \ldots, x_n$, как предполагается, имеют, соответственно, типы $A_1, \ldots, A_n$. Список может быть и пустым. Можно именовать контексты, например, с помощью букв Γ и Δ, чтобы использовать их при формировании более крупных контекстов.

Суждение $\Gamma\ \mathsf{ctx}$ формально выражает тот факт, что Γ является правильно сформированным контекстом и регулируется правилами вывода

$$\dfrac{}{\cdot\ \mathsf{ctx}}\ \text{ctx-EMP} \qquad\qquad \dfrac{x_1 : A_1, \ldots, x_{n-1} : A_{n-1}}{(x_1 : A_1, \ldots, x_n : A_n)}\ \text{ctx-EXT}$$

с дополнительным условием для второго правила: переменная x_n должна отличаться от переменных $x_1, \ldots, x_{n-1}$. Заметим, что гипотеза и заключение ctx-EXT являются суждениями разных форм: гипотеза гласит, что в контексте переменных $x_1, \ldots, x_{n-1}$, выражение A_n имеет тип $\mathcal{U}_i$; а заключение выражает то, что расширенный контекст $(x_1 : A_1, \ldots, x_n : A_n)$ сформирован правильно.

Это мета-теоретическое свойство системы: если любое суждение вида $\Gamma \vdash a : A$ или $\Gamma \vdash a \equiv a' : A$ выводимо, то же относится к суждению $\Gamma\ \mathsf{ctx}$, то контекст Γ сформирован правильно. Посылки всех правил выбраны так, чтобы включать в себя ровно столько же гипотез правильности, чтобы это свойство можно было доказать, но не более того. Например, для ctx-EXT необязательно предполагать правильность формата $(x_1 : A_1, \ldots, x_{n-1} : A_{n-1})$, поскольку это будет следовать из выводимости его посылки; но это необходимо для правила Vble из следующего раздела, чтобы предположить правильность его контекста. Этот выбор — лишь один из многих возможных способов точной формулировки теории типов, но подробное исследование таких вопросов выходит за рамки данного приложения.

A.2.2 Структурные правила

Тот факт, что контекст поддерживает суждения, выражается правилом, которое гласит, что можно вывести те суждения, которые перечислены в контексте:

$$\dfrac{(x_1 : A_1, \ldots, x_n : A_n)\ \mathsf{ctx}}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\ \mathsf{Vble}$$

Как и в случае с ctx-EXT, гипотеза и заключение правила Vble являются суждениями разных форм, только теперь они поменялись местами: мы начинаем с правильно сформированного контекста и производим типирование суждения.

Следующие важные принципы, называемые **замещением** и **ослаблением**, не должны явно предполагаться. Скорее, можно показать индукцией по структуре всех возможных выводов, что всякий раз, когда гипотезы этих правил выводимы, их заключение также выводимо.[2] Для типирования суждений эти принципы проявляются как

$$\frac{\Gamma \vdash a : A \qquad \Gamma, x{:}A, \Delta \vdash b : B}{\Gamma, \Delta[a/x] \vdash b[a/x] : B[a/x]}\ \mathsf{Subst}_1 \qquad\qquad \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, \Delta \vdash b : B}{\Gamma, x{:}A, \Delta \vdash b : B}\ \mathsf{Wkg}_1$$

а для равенств суждений они становятся

$$\frac{\Gamma \vdash a : A \qquad \Gamma, x{:}A, \Delta \vdash b \equiv c : B}{\Gamma, \Delta[a/x] \vdash b[a/x] \equiv c[a/x] : B[a/x]}\ \mathsf{Subst}_2 \qquad\qquad \frac{\Gamma \vdash a \equiv b : A \qquad \Gamma, x{:}A, \Delta \vdash c : C}{\Gamma, \Delta[a/x] \vdash c[b/x] : C[a/x]}\ \mathsf{Subst}_3$$

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, \Delta \vdash b \equiv c : B}{\Gamma, x{:}A, \Delta \vdash b \equiv c : B}\ \mathsf{Wkg}_2$$

В дополнение к правилам равенств суждений, заданным для каждого формирователя типа, мы также предполагаем, что равенство суждений — это отношение эквивалентности, которое соблюдается при типировании.

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a \equiv a : A} \qquad\qquad \frac{\Gamma \vdash a \equiv b : A}{\Gamma \vdash b \equiv a : A} \qquad\qquad \frac{\Gamma \vdash a \equiv b : A \qquad \Gamma \vdash b \equiv c : A}{\Gamma \vdash a \equiv c : A}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash A \equiv B : \mathcal{U}_i}{\Gamma \vdash a : B} \qquad\qquad \frac{\Gamma \vdash a \equiv b : A \qquad \Gamma \vdash A \equiv B : \mathcal{U}_i}{\Gamma \vdash a \equiv b : B}$$

Наконец, мы предполагаем, что равенство суждений — это соответствие, которое соблюдается при типировании, т.е. что каждый конструктор сохраняет равенство суждений в каждом из своих аргументов. Например, наряду с Π-INTRO, предполагается правило

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash B : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash b \equiv b' : B}{\Gamma \vdash \lambda x.\, b \equiv \lambda x.\, b' : \prod_{(x:A)} B}\ \textstyle\prod\text{-INTRO-EQ}$$

Взятые вместе, эти локальные принципы подразумевают принципы глобальной конгруэнтности Subst_2 и Subst_3, указанные выше. Для краткости, опустим эти локальные правила.

A.2.3 Универсумы типов

Мы постулируем бесконечную иерархию типов универсумов

$$\mathcal{U}_0, \quad \mathcal{U}_1, \quad \mathcal{U}_2, \quad \ldots$$

Каждый универсум содержится в следующем, и любой тип из $\mathcal{U}_i$ также находится и в $\mathcal{U}_{i+1}$:

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathcal{U}_i : \mathcal{U}_{i+1}}\ \mathcal{U}\text{-INTRO} \qquad\qquad \frac{\Gamma \vdash A : \mathcal{U}_i}{\Gamma \vdash A : \mathcal{U}_{i+1}}\ \mathcal{U}\text{-CIMUL}$$

Установим правила теории типов таким образом, что $\Gamma \vdash a : A$ влечет $\Gamma \vdash A : \mathcal{U}_i$ для некоторого i. Другими словами, если A играет роль типа, то он находится в некотором универсуме. Еще одно свойство нашей системы типов состоит в том, что $\Gamma \vdash a \equiv b : A$ подразумевает $\Gamma \vdash a : A$ и $\Gamma \vdash b : A$.

[2]Такие правила называются **допустимыми**.

A.2.4 Зависимые функциональные типы (Π-типы)

В §1.2 были введены независимые функции $A \to B$, чтобы определить семейство типов как функцию $\lambda(x : A).\, B : A \to \mathcal{U}_i$, что затем привело к типу зависимых функций $\prod_{(x:A)} B$. Но с явным контекстом можно заменить $\lambda(x : A).\, B : A \to \mathcal{U}_i$ суждением

$$x{:}A \vdash B : \mathcal{U}_i$$

Следовательно, можно определять зависимые функции напрямую, без ссылки на независимые. Таким образом, мы следуем общему принципу, согласно которому каждый формирователь типов с его константами и правилами должен вводиться независимо от всех других формирователей типов. Фактически, отныне каждый такой формирователь систематически можно вводить как:

- **правило формирования**, указывающее, когда можно применять формирователь типа;
- некоторые **правила введения**, описывающие, как заселять тип;
- **правила исключения** или принцип индукции, устанавливающие, как использовать элемент конкретного типа;
- **правила вычисления**, которые представляют собой оценочные равенства, объясняющие, что происходит, когда правила исключения применяются к результатам правил введения;
- необязательные **принципы уникальности**, которые представляют собой равенства суждений, объясняющие, как каждый элемент типа однозначно определяется результатами примененных к нему правил исключения.

(См. также замечание 1.5.1).

Для зависимого функционального типа эти правила таковы:

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash B : \mathcal{U}_i}{\Gamma \vdash \prod_{(x:A)} B : \mathcal{U}_i}\ \Pi\text{-FORM} \qquad\qquad \frac{\Gamma, x{:}A \vdash b : B}{\Gamma \vdash \lambda(x : A).\, b : \prod_{(x:A)} B}\ \Pi\text{-INTRO}$$

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B[a/x]}\ \Pi\text{-ELIM} \qquad\qquad \frac{\Gamma, x{:}A \vdash b : B \qquad \Gamma \vdash a : A}{\Gamma \vdash (\lambda(x : A).\, b)(a) \equiv b[a/x] : B[a/x]}\ \Pi\text{-COMP}$$

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B}{\Gamma \vdash f \equiv (\lambda x.\, f(x)) : \prod_{(x:A)} B}\ \Pi\text{-UNIQ}$$

Выражение $\lambda(x : A).\, b$ связывает свободные вхождения x в b, как и $\prod_{(x:A)} B$ для B.

Когда x не входит свободно в B, так что B не зависит от A, получаем, как частный случай, обычный функциональный тип $A \to B :\equiv \prod_{(x:A)} B$. Мы принимаем это за *определение* $\to$.

Можно сократить выражение $\lambda(x : A).\, b$ до $\lambda x.\, b$, понимая, что пропущенный тип A должен быть соответствующим образом заселен перед проверкой типа.

A.2.5 Зависимые типы пар (Σ-типы)

В §1.6 нам потребовались типы $\to$ и $\prod$, чтобы определить правила введения и исключения для $\sum$; как и в случае с $\prod$, контексты позволяют независимо устанавливать правила для $\sum$.

Напомним, что правило исключения для позитивного типа, такого как Σ, называется *индукцией* и обозначается ind.

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash B : \mathcal{U}_i}{\Gamma \vdash \sum_{(x:A)} B : \mathcal{U}_i}\ \textstyle\sum\text{-FORM}$$

$$\frac{\Gamma, x{:}A \vdash B : \mathcal{U}_i \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B[a/x]}{\Gamma \vdash (a,b) : \sum_{(x:A)} B}\ \textstyle\sum\text{-INTRO}$$

$$\frac{\Gamma, z{:}\sum_{(x:A)} B \vdash C : \mathcal{U}_i \qquad \Gamma, x{:}A, y{:}B \vdash g : C[(x,y)/z] \qquad \Gamma \vdash p : \sum_{(x:A)} B}{\Gamma \vdash \mathsf{ind}_{\sum_{(x:A)} B}(z.C, x.y.g, p) : C[p/z]}\ \textstyle\sum\text{-ELIM}$$

$$\frac{\Gamma, z{:}\sum_{(x:A)} B \vdash C : \mathcal{U}_i \qquad \Gamma, x{:}A, y{:}B \vdash g : C[(x,y)/z] \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B[a/x]}{\Gamma \vdash \mathsf{ind}_{\sum_{(x:A)} B}(z.C, x.y.g, (a,b)) \equiv g[a,b/x,y] : C[(a,b)/z]}\ \textstyle\sum\text{-COMP}$$

Выражение $\sum_{(x:A)} B$ связывает свободные вхождения x в B. Кроме того, поскольку $\mathsf{ind}_{\sum_{(x:A)} B}$ имеет некоторые аргументы со свободными переменными помимо тех, что указаны в Γ, мы связываем (следуя именам переменных, использованных выше) z в C, а x и y — в g. Эти привязки записываются как $z.C$ и $x.y.g$, указывая имена связанных переменных. В частности, мы рассматриваем $\mathsf{ind}_{\sum_{(x:A)} B}$ как примитив, два аргумента которого содержат связующие элементы; это внешне похоже, но отличается от $\mathsf{ind}_{\sum_{(x:A)} B}$, являющейся функцией, которая принимает функции в качестве аргументов.

Когда B не содержит свободных вхождений x, мы получаем, как частный случай, декартово произведение $A \times B :\equiv \sum_{(x:A)} B$. Мы принимаем это за *определение* декартова произведения.

Отметим, что мы не постулируем принцип единственности суждения для $\sum$-типов, хотя мы могли бы; см. следствие 2.7.3 для доказательства соответствующего принципа пропозициональной единственности.

A.2.6 Типы копроизведений

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i}{\Gamma \vdash A + B : \mathcal{U}_i}\ +\text{-FORM}$$

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{inl}(a) : A + B}\ +\text{-INTRO}_1 \qquad \frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i \qquad \Gamma \vdash b : B}{\Gamma \vdash \mathsf{inr}(b) : A + B}\ +\text{-INTRO}_2$$

$$\frac{\Gamma, z{:}(A+B) \vdash C : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y{:}B \vdash d : C[\mathsf{inr}(y)/z] \qquad \Gamma \vdash e : A + B}{\Gamma \vdash \mathsf{ind}_{A+B}(z.C, x.c, y.d, e) : C[e/z]}\ +\text{-ELIM}$$

$$\frac{\Gamma, z{:}(A+B) \vdash C : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y{:}B \vdash d : C[\mathsf{inr}(y)/z] \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{ind}_{A+B}(z.C, x.c, y.d, \mathsf{inl}(a)) \equiv c[a/x] : C[\mathsf{inl}(a)/z]}\ +\text{-COMP}_1$$

$$\frac{\Gamma, z{:}(A+B) \vdash C : \mathcal{U}_i \qquad \Gamma, x{:}A \vdash c : C[\mathsf{inl}(x)/z] \qquad \Gamma, y{:}B \vdash d : C[\mathsf{inr}(y)/z] \qquad \Gamma \vdash b : B}{\Gamma \vdash \mathsf{ind}_{A+B}(z.C, x.c, y.d, \mathsf{inr}(b)) \equiv c[b/y] : C[\mathsf{inr}(b)/z]}\ +\text{-COMP}_2$$

В ind_{A+B}, z связан в C, x связан в c, а y связан в d.

A.2.7 Пустой тип $\mathbf{0}$

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbf{0} : \mathcal{U}_i}\ \mathbf{0}\text{-FORM} \qquad \frac{\Gamma, x{:}\mathbf{0} \vdash C : \mathcal{U}_i \qquad \Gamma \vdash a : \mathbf{0}}{\Gamma \vdash \mathsf{ind}_{\mathbf{0}}(c.C, c, a) : C[a/x]}\ \mathbf{0}\text{-ELIM}$$

В $\mathsf{ind}_{\mathbf{0}}$, x связан в C. Пустой тип не имеет правила введения и правила вычисления.

А.2.8 Единичный тип 1

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbf{1} : \mathcal{U}_i}\ \mathbf{1}\text{-FORM} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \star : \mathbf{1}}\ \mathbf{1}\text{-INTRO} \qquad \frac{\Gamma, x{:}\mathbf{1} \vdash C : \mathcal{U}_i \quad \Gamma \vdash c : C[\star/x] \quad \Gamma \vdash a : \mathbf{1}}{\Gamma \vdash \mathsf{ind}_{\mathbf{1}}(c.C, c, a) : C[a/x]}\ \mathbf{1}\text{-ELIM}$$

$$\frac{\Gamma, x{:}\mathbf{1} \vdash C : \mathcal{U}_i \quad \Gamma \vdash c : C[\star/x]}{\Gamma \vdash \mathsf{ind}_{\mathbf{1}}(c.C, c, \star) \equiv c : C[\star/x]}\ \mathbf{1}\text{-COMP}$$

В $\mathsf{ind}_{\mathbf{1}}$ переменная x связана в C.

Заметьте, что мы не постулируем принцип уникальности суждения для типа единицы; см. §1.5 для доказательства соответствующего утверждения пропозициональной единственности.

А.2.9 Тип натуральных чисел

Приведем правила для натуральных чисел, следуя §1.9.

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbb{N} : \mathcal{U}_i}\ \mathbb{N}\text{-FORM} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash 0 : \mathbb{N}}\ \mathbb{N}\text{-INTRO}_1 \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{succ}(n) : \mathbb{N}}\ \mathbb{N}\text{-INTRO}_2$$

$$\frac{\Gamma, x{:}\mathbb{N} \vdash C : \mathcal{U}_i \qquad \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, n{:}\mathbb{N}, y : C \vdash c_s C[\mathsf{succ}(x)/x] \qquad \Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(x.C, c_0, x.y.c_s, n) : C[n/x]}\ \mathbb{N}\text{-ELIM}$$

$$\frac{\Gamma, x{:}\mathbb{N} \vdash C : \mathcal{U}_i \qquad \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, n{:}\mathbb{N}, y : C \vdash c_s C[\mathsf{succ}(x)/x]}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(x.C, c_0, x.y.c_s, 0) \equiv c_0 : C[0/x]}\ \mathbb{N}\text{-COMP}_1$$

$$\frac{\Gamma, x{:}\mathbb{N} \vdash C : \mathcal{U}_i \qquad \Gamma \vdash c_0 : C[0/x] \qquad \Gamma, n{:}\mathbb{N}, y : C \vdash c_s C[\mathsf{succ}(x)/x] \qquad \Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{ind}_{\mathbb{N}}(x.C, c_0, x.y.c_s, \mathsf{succ}(n)) \equiv c_s[n, \mathsf{ind}_{\mathbb{N}}(x.C, c_0, x.y.c_s, n)/x, y] : C[\mathsf{succ}(n)/x]}\ \mathbb{N}\text{-COMP}_2$$

В $\mathsf{ind}_{\mathbb{N}}$, x связан в C, а x и y связаны в c_s.

Остальные индуктивно определенные типы следуют той же общей схеме.

А.2.10 Типы тождественностей

Приведенное здесь представление соответствует (необоснованному) принципу индукции по путям для тождественных типов из §1.12.

$$\frac{\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash a : A \quad \Gamma \vdash b : A}{\Gamma \vdash a =_A b : \mathcal{U}_i}\ {=}\text{-FORM} \qquad \frac{\Gamma \vdash A : \mathcal{U}_i \quad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}_a : a =_A a}\ {=}\text{-INTRO}$$

$$\frac{\begin{array}{c}\Gamma, x{:}A, y{:}A, p{:}x{=_A}y \vdash C : \mathcal{U}_i \quad \Gamma, z{:}A \vdash c : C[z, z, \mathsf{refl}_z/x, y, p] \\ \Gamma \vdash a : A \quad \Gamma \vdash b : A \quad \Gamma \vdash p' : a =_A b\end{array}}{\Gamma \vdash \mathsf{ind}_{=_A}(x.y.p.C, z.c, a, b, p') : C[a, b, p'/x, y, p]}\ {=}\text{-ELIM}$$

$$\frac{\Gamma, x{:}A, y{:}A, p{:}x{=_A}y \vdash C : \mathcal{U}_i \quad \Gamma, z{:}A \vdash c : C[z, z, \mathsf{refl}_z/x, y, p] \quad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{ind}_{=_A}(x.y.p.C, z.c, a, a, \mathsf{refl}_a) \equiv c[a/z] : C[a, a, \mathsf{refl}_a/x, y, p]}\ {=}\text{-COMP}$$

В $\mathsf{ind}_{=_A}$, x, y и p связаны в C, а z связана в c.

A.2.11 Определения

Хотя правила, которые были перечислены до сих пор, позволяют конструировать все, что нужно, напрямую, мы все же хотели бы иметь возможность использовать именованные константы, такие как isequiv, для удобства. Неформально, можно рассматривать эти константы просто как аббревиатуры, но с формализацией ситуация немного тоньше.

Например, рассмотрим композицию функций, которая принимает $f : A \to B$ и $g : B \to C$ для формирования $g \circ f : A \to C$. Несколько неожиданным является то, что формально $\circ$ должна принимать в качестве аргументов не только f и g, но также их типы A, B, C:

$$\circ :\equiv \lambda(A : \mathcal{U}_i).\, \lambda(B : \mathcal{U}_i).\, \lambda(C : \mathcal{U}_i).\, \lambda(g : B \to C).\, \lambda(f : A \to B).\, \lambda(x : A).\, g(f(x))\,.$$

С практической точки зрения, мы не хотим аннотировать каждое использование $\circ$ с помощью A, B и C, так как они обычно довольно легко подразумеваются из окружающей информации. Мы хотели бы просто использовать $g \circ f$. Тогда, строго говоря, $g \circ f$ не является сокращением для $\lambda(x : A).\, g(f(x))$, поскольку оно включает дополнительные **неявные аргументы**, которые мы хотим подавить.

Вывод неявных аргументов, типовая неоднозначность (§1.3), гарантия того, что символы определены только один раз, и т.д., в совокупности называется **уточнением**, которое должно проводиться до проверки вывода, и поэтому обычно не рассматривается как часть теории типов. Однако, практически, невозможно использовать какую-либо реализацию теории типов, не выполняющую уточнения; см. [Coq12, Nor07] для дальнейшего обсуждения.

A.3 Гомотопическая теория типов

В этом разделе мы сформулируем дополнительные аксиомы гомотопической теории типов, которые отличают ее от стандартной теории типов Мартина-Лёфа, включая: функциональную экстенсиональность, аксиому унивалентности и высшие индуктивные типы. Мы формулируем их в стиле второй нотации данного приложения, хотя и первая нотация может быть использована с тем же успехом.

A.3.1 Функциональная экстенсиональность и унивалентность

Есть два основных способа введения аксиом, которые не навязывают новые, синтаксис или равенства суждений (функциональная экстенсиональность и унивалентность обладают этим): либо добавить примитивную константу для аксиомы заселения, либо доказать все теоремы, которые зависят от аксиомы, выдвигая гипотезу о переменной, принадлежащей аксиоме, ср. §1.1. Хотя они по существу эквивалентны, мы выбираем первый подход, потому что считаем, что аксиомы гомотопической теории типов являются существенной частью основной теории типов.

Аксиома 2.9.3 формализована введением константы funext, которая утверждает, что happly является эквивалентностью:

$$\frac{\Gamma \vdash f : \prod_{(x:A)} B \qquad \Gamma \vdash g : \prod_{(x:A)} B}{\Gamma \vdash \mathsf{funext}(f, g) : \mathsf{usequiv}(\mathsf{happly}_{f,g})}\ \textstyle\prod\text{-EXT}$$

Определения happly и isequiv можно найти в (2.9.2) и §4.5, соответственно.

Аксиома 2.10.3 формализована аналогичным образом:

$$\frac{\Gamma \vdash A : \mathcal{U}_i \qquad \Gamma \vdash B : \mathcal{U}_i}{\Gamma \vdash \mathsf{univalence}(A, B) : \mathsf{usequiv}(\mathsf{idtoeqv}_{A,B})} \ \mathcal{U}_i\text{-UNIV}$$

Определение $\mathsf{idtoeqv}$ можно найти в (2.10.2).

A.3.2 Окружность

Здесь мы приводим пример базового высшего индуктивного типа; другие следуют той же общей схеме, хотя и с уточнениями.

Заметим, что приведенные ниже правила не точно следуют шаблону обычных индуктивных типов по второй нотации: правила относятся к понятиям транспортирования и функториальности отображений (§2.2), а второе правило вычислений является пропозициональным, не суждением, равенством. Эти различия обсуждаются в §6.2.

$$\frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathbb{S}^1 : \mathcal{U}_i}\ \mathbb{S}^1\text{-FORM} \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{base} : \mathbb{S}^1}\ \mathbb{S}^1\text{-INTRO}_1 \qquad \frac{\Gamma\ \mathsf{ctx}}{\Gamma \vdash \mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}}\ \mathbb{S}^1\text{-INTRO}_2$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b \qquad \Gamma \vdash p : \mathbb{S}^1}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.C, b, \ell, p) : C[p/x]}\ \mathbb{S}^1\text{-ELIM}$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b}{\Gamma \vdash \mathsf{ind}_{\mathbb{S}^1}(x.C, b, \ell, \mathsf{base}) \equiv b : C[\mathsf{base}/x]}\ \mathbb{S}^1\text{-COMP}_1$$

$$\frac{\Gamma, x{:}\mathbb{S}^1 \vdash C : \mathcal{U}_i \qquad \Gamma \vdash b : C[\mathsf{base}/x] \qquad \Gamma \vdash \ell : b =^{C}_{\mathsf{loop}} b}{\Gamma \vdash \mathbb{S}^1\text{-}\mathsf{loopcomp} : \mathsf{apd}_{(\lambda y.\, \mathsf{ind}_{\mathbb{S}^1}(x.C,b,\ell,y))}(\mathsf{loop}) = \ell}\ \mathbb{S}^1\text{-COMP}_2$$

В $\mathsf{ind}_{\mathbb{S}^1}$, x связан в C. Обозначение $b =^{C}_{\mathsf{loop}} b$ для зависимых путей было введено в §6.2.

A.4 Базовая метатеория

В этом разделе обсуждаются метатеоретические свойства теории типов, представленные в Приложении A.1, и аналогичные результаты справедливы для Приложения A.2. Выяснение того, какие из них все еще остаются актуальными, когда мы добавляем функции из Приложения A.3, быстро приводит к открытым вопросам, как обсуждается в конце данного раздела.

Напомним, что в Приложении A.1 термины теории типов определяются как расширение нетипизированного λ-исчисления, которое имеет собственное понятие вычислений, а именно правило вычислений

$$(\lambda x.\, t)(u) :\equiv t[u/x]\,.$$

Это правило вместе с определяющими уравнениями для определенных констант образуют *правила перезаписи* , которые определяют шаги редукции для системы перезаписи. Эти шаги дают понятие вычисления в том смысле, что каждое правило имеет естественное направление: оно упрощает $(\lambda(x).\, t)(u)$, оценивая функцию по ее аргументу.

Более того, эта система является *конфлюэнтной*, то есть, если a упрощается за некоторое количество шагов, и до a', и до a'', существует некоторый b, до которого в конечном итоге упрощаются, как a', так и a''. Таким образом, можно определить $t \downarrow u$, чтобы обозначить, что t и u упрощаются до одного и того же терма.

(Ситуация аналогична представленной в Приложении А.2: хотя там правила вычислений рассмотрены как неориентированные равенства $\equiv$, можно дать операционную семантику, сказав, что применение элиминатора к вводной форме упрощается до ей равной, а не другой, в обратном направлении).

Используя стандартные методы теории типов, можно показать, что система в Приложении А.1 обладает следующими свойствами:

Теорема А.4.1. *Если $A : \mathcal{U}$ и $A \downarrow A'$, то $A' : \mathcal{U}$. Если $t : A$ и $t \downarrow t'$, то $t' : A$.*

Мы говорим, что терм **нормализуем** (соответственно, **сильно нормализуем**), если некоторая (соответственно каждая) последовательность шагов переписывания терма, завершается.

Теорема А.4.2. *Если $A : \mathcal{U}$, то A сильно нормализуемо. Если $t : A$, то A и t сильно нормализуемы.*

Мы говорим, что терм имеет **нормальную форму**, если его нельзя упростить, и что терм **замкнут**, если он не содержит свободных переменных. Замкнутый нормальный тип должен быть примитивным типом, т.е. иметь форму $c(\vec{v})$ для некоторой примитивной константы c (где список $\vec{v}$ замкнутых нормальных термов может быть опущен, если он пуст, например , как и в случае с $\mathbb{N}$). Фактически, мы можем явно описать все нормальные формы:

Лемма А.4.3. *Термы в нормальной форме можно описать следующим синтаксисом:*

$$\begin{aligned} v &::= k \mid \lambda x.\, v \mid c(\vec{v}) \mid f(\vec{v})\,, \\ k &::= x \mid k(v) \mid f(\vec{v})(k)\,, \end{aligned}$$

где $f(\vec{v})$ представляет собой частичное применение определенной функции f. В частности, тип в нормальной форме имеет вид k или $c(\vec{v})$.

Теорема А.4.4. *Если A находится в нормальной форме, то суждение $A : \mathcal{U}$ разрешимо. Если $A : \mathcal{U}$ и t находятся в нормальной форме, то суждение $t : A$ разрешимо.*

Логическая непротиворечивость (системы в Приложении А.1) следует немедленно: если бы у нас было $a : \mathbf{0}$ в пустом контексте, то по теоремам А.4.1 и А.4.2, a упрощается до нормального терма $a' : \mathbf{0}$. Но, по лемме А.4.3, такого терма не существует.

Следствие А.4.5. *Система в Приложении А.1 логически непротиворечива.*

Так же, имеется свойство *каноничности*: если $a : \mathbb{N}$ в пустом контексте, то a упрощается до нормального терма $\mathsf{succ}^k(0)$, для некоторого числа k.

Следствие А.4.6. *Система в Приложении А.1 обладает свойством каноничности.*

Наконец, если a и A находятся в нормальной форме, то разрешима доказуемость $a : A$; другими словами, поскольку проверка типов сводится к проверке правильности доказательства, это означает, что мы всегда можем «распознать правильное доказательство, когда увидим его».

Следствие А.4.7. *Свойство быть доказательством в системе из приложения А.1 разрешимо.*

Приведенные выше результаты не применимы к расширенной системе гомотопической теории типов (т.е., к вышеупомянутой системе, расширенной Приложением А.3), поскольку вхождения аксиомы унивалентности и конструкторов высших индуктивных типов никогда не упрощаются, нарушая лемму А.4.3. Открытым является вопрос, можно ли упростить применение этих констант, чтобы восстановить каноничность. У нас также нет схемы, описывающей все допустимые высшие индуктивные типы, и мы не знаем, как правильно сформулировать их правила (например, должны ли правила вычислений на высших конструкторах быть равенствами суждений).

Непротиворечивость теории типов Мартина-Лёфа, расширенной за счет унивалентности и высших индуктивных типов, может быть продемонстрирована путем изобретения соответствующей процедуры нормализации, но в настоящее время единственные доказательства того, что эти системы непротиворечивы, получены с помощью семантических моделей: для унивалентности — это модель в комплексах Кана (Kan), созданная Воеводским [KLV12], а для высших индуктивных типов — модель Ламсдейна (Lumsdaine) и Шульмана (Shulman) [LS17].

Другие метатеоретические вопросы и краткое изложение наших текущих результатов более подробно обсуждаются в разделах «Конструктивность» и «Открытые проблемы» во введении к этой книге.

Примечания

Система правил с введением (примитивные константы) и правилами исключения и вычисления (определенная константа) вдохновлена естественным выводом Гентцена (Gentzen). Возможность усиления правила исключения для экзистенциальной количественной оценки была указана в [How80]. Усиление аксиом дизъюнкции обсуждается в [ML98], а аксиом исключения абсурда и тождественного типа — в [ML75]. W-типы были введены в [ML82]. Они обобщают нотацию деревьев из [Tai68].

Обобщенная форма примитивной рекурсии для натуральных чисел и ординалов представлена в [Hil26]. Это мотивировало появление системы T от Гёделя [Göd58], которая была проанализирована в [Tai67], где, следуя [Göd58], была использована терминология «дефинициальное равенство» для преобразования: два терма *равны с позиции суждений*, если они сводятся к общему терму посредством последовательного применения правил редукции. Эта терминология также использовалась де Брейном (de Bruijn) [dB73] в его презентации компьютерной системы *AUTOMATH*.

Наша вторая нотация в А.2 содержит довольно стандартное представление интенсиональной теории типов Мартина-Лёфа с некоторыми дополнительными особенностями, диктуемыми гомотопической теорией типов. По сравнению со справочным изложением [Hof97], теория типов в этой книге имеет несколько некритических отличий:

- универсумы по Расселу (Russell) в смысле [ML84]; и
- суждение η и функциональная экстенсиональность для типов $\prod$;

и особенностей, важных для гомотопической теории типов:

- аксиома унивалентности; и
- высшие индуктивные типы.

Для удобства, в этой книге, в первую очередь, функции определяются функции по индукции, используя *сопоставление с образцом*. Можно формализовать это понятие, как это сделано в Приложении А.1. Однако стандартное теоретико-типовое представление, принятое в Приложении А.2, состоит в том, чтобы ввести единственное *зависимое исключение* для каждого формирователя типа, из которого должны определяться функции вне этого типа. Этот подход легче формализовать как синтаксически, так и семантически, поскольку он составляет универсальное свойство формирователя типов. Эти два подхода эквивалентны; см. §1.10 для более глубокого осмысления.

Литература

[AB04] Steven Awodey and Andrej Bauer. Propositions as [types]. *Journal of Logic and Computation*, 14(4):447–471 , 2004.

[Acz78] Peter Aczel. The type theoretic interpretation of constructive set theory. In A. MacIntyre, L. Pacholski, and J. Paris, editors, *Logic Colloquium '77, volume 96 of Studies in Logic and the Foundations of Mathematics*, pages 55–66. North-Holland, Amsterdam, 1978.

[AG02] Peter Aczel and Nicola Gambino. Collection principles in dependent type theory. In Paul Callaghan, Zhaohui Luo, James McKinna, and Robert Pollack, editors, *Types for Proofs and Programs, International Workshop, TYPES 2000, Durham, UK, December 8-12, 2000, Selected Papers, volume 2277 of Lecture Notes in Computer Science*, pages 1–23. Springer, 2002.

[AGS12] Steve Awodey, Nicola Gambino, and Kristina Sojakova. Inductive types in homotopy type theory. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 95–104. IEEE Computer Society, 2012, arXiv:1201.3898.

[AKL13] Jeremy Avigad, Krzysztof Kapulkin, and Peter LeFanu Lumsdaine. Homotopy limits in Coq, 2013. arXiv:1304.0680.

[AKS13] Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. Univalent categories and the Rezk completion, 2013. arXiv:1303.0584.

[Alt99] Thorsten Altenkirch. Extensional equality in intensional type theory. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2–5, 1999*, pages 412–420, 1999.

[AMS07] Thorsten Altenkirch, Conor McBride, and Wouter Swierstra. Observational equality, now! In Aaron Stump and Hongwei Xi, editors, *Proceedings of the ACM Workshop Programming Languages meets Program Verification, PLPV 2007, Freiburg, Germany, October 5, 2007*, 2007.

[Ang13] Carlo Angiuli. The $(\infty, 1)$-accidentopos model of unintentional type theory. Sigbovik '13, April 1 2013.

[AW09] Steve Awodey and Michael A. Warren. Homotopy theoretic models of identity types. Mathematical Proceedings of the Cambridge Philosophical Society, 146:45–55, 2009.

[Bau13] Andrej Bauer. Five stages of accepting constructive mathematics, 2013. `http://video.ias.edu/members/1213/0318-AndrejBauer`.

[BCH13] Bruno Barras, Thierry Coquand, and Simon Huber. A generalization of Takeuti-Gandy interpretation. `http://uf-ias-2012.wikispaces.com/file/view/semi.pdf`, 2013.

[Bee85] Michael Beeson. *Foundations of Constructive Mathematics*. Springer, 1985.

[Ber09] Julia E. Bergner. A survey of $(\infty, 1)$-categories. In John C. Baez and J. Peter May, editors, *Towards Higher Categories*, volume 152 of *The IMA Volumes in Mathematics and its Applications*, pages 69–83. Springer, 2009, arXiv:math.CT/0610239.

[Bis67] Erret Bishop. *Foundations of constructive analysis*. McGraw-Hill Book Co., New York, 1967.

[BIS02] Douglas Bridges, Hajime Ishihara, and Peter Schuster. Compactness and continuity, constructively revisited. In Julian C. Bradfield, editor, *Computer Science Logic, 16th International Workshop, CSL 2002, 11th Annual Conference of the EACSL, Edinburgh, Scotland, UK, September 22-25, 2002, Proceedings, volume 2471 of Lecture Notes in Computer Science*, pages 89–102. Springer, 2002.

[Bla79] Georges Blanc. Équivalence naturelle et formules logiques en théorie des catégories. *Archiv für Mathematische Logik und Grundlagenforschung*, 19(3-4):131–137, 1978/79.

[Bou68] Nicolas Bourbaki. *Theory of Sets*. Hermann, Paris, 1968.

[BSP11] Clark Barwick and Christopher Schommer-Pries. On the unicity of the homotopy theory of higher categories, 2011. arXiv:1112.0040.

[BT09] Andrej Bauer and Paul Taylor. The Dedekind reals in abstract Stone duality. *Mathematical structures in computer science*, 19(4):757–838, 2009.

[Bun79] Marta Bunge. Stack completions and Morita equivalence for categories in a topos. *Cahiers de Topologie et G ´eom ´etrie Diff ´erentielle*, 20(4):401–436, 1979.

[CAB+86] Robert L. Constable, Stuart F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, Robert W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the NuPRL Proof Development System*. Prentice Hall, 1986.

[Car95] Aurelio Carboni. Some free constructions in realizability and proof theory. *Journal of Pure and Applied Algebra*, 103:117–148, 1995.

[Chu33] Alonzo Church. A set of postulates for the foundation of logic 2. *Annals of Mathematics*, 34:839–864, 1933.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[Chu41] Alonzo Church. *The Calculi of Lambda Conversation*. Princeton University Press, 1941.

[CM85] Robert L. Constable and N. P. Mendler. Recursive definitions in type theory. In Rohit Parikh, editor, *Logics of Programs, Conference, Brooklyn College, June 17–19, 1985, Proceedings, Vol. 193 of Lecture Notes in Comp. Sci.*, pages 61–78, 1985.

[Con76] John H. Conway. *On numbers and games*. A K Peters Ltd., 1976.

[Con85] Robert L. Constable. Constructive mathematics as a programming logic I: Some principles of theory. In *Annals of Mathematics*, volume 24, pages 21–37. Elsevier Science Publishers, B.V. (North-Holland), 1985. Reprinted from *Topics in the Theory of Computation*, Selected Papers of the International Conference on Foundations of Computation Theory, FCT '83.

[Coq92] Thierry Coquand. The paradox of trees in type theory. *BIT Numerical Mathematics*, 32(1):10–14, 1992.

[Coq12] Coq Development Team. *The Coq Proof Assistant Reference Manual*. INRIA-Rocquencourt, 2012.

[CP90] Thierry Coquand and Christine Paulin. Inductively defined types. In *COLOG-88 (Tallinn, 1988), volume 416 of Lecture Notes in Computer Science*, pages 50–66. Springer, 1990.

[dB73] Nicolaas Govert de Bruijn. *AUTOMATH, a language for mathematics*. Les Presses de l'Université de Montréal, Montreal, Quebec, 1973. Séminaire de Mathématiques Supérieures, No. 52 (Été 1971).

[Dia75] Radu Diaconescu. Axiom of choice and complementation. *Proceedings of the American Mathematical Society*, 51:176–178, 1975.

[dPGM04] Valeria de Paiva, Rajeev Goré, and Michael Mendler. Modalities in constructive logics and type theories. *Journal of Logic and Computation*, 14(4):439–446, 2004.

[Dyb91] Peter Dybjer. Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics. In Gerard Huet and Gordon Plotkin, editors, *Logical Frameworks*, pages 280–30. Cambridge University Press, 1991.

[Dyb00] Peter Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. *Journal of Symbolic Logic*, 65(2):525–549, 2000.

[ES01] Martín Hötzel Escardó and Alex K. Simpson. A universal characterization of the closed euclidean interval. In *16th Annual IEEE Symposium on Logic in Computer Science, Boston, Massachusetts, USA, June 16-19, 2001, Proceedings*, pages 115–125. IEEE Computer Society, 2001.

[EucBC] Euclid. *Elements*, Vols. 1–13. Elsevier, 300 BC.

[Fre76] Peter Freyd. Properties invariant within equivalence types of categories. In *Algebra, topology, and category theory (a collection of papers in honor of Samuel Eilenberg)*, pages 55–61. Academic Press, 1976.

[FS12] Fredrik Nordvall Forsberg and Anton Setzer. A finite axiomatisation of inductive-inductive definitions. `http://cs.swan.ac.uk/~csfnf/papers/indind_finite.pdf`, 2012.

[GAA+13] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, Françqis Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Thery. A machine-checked proof of the odd order theorem. In *Interactive Theorem Proving*, 2013.

[Gar09] Richard Garner. On the strength of dependent products in the type theory of Martin-Löf. *Annals of Pure and Applied Logic*, 160(1):1–12, 2009.

[Gen36] Gerhard Gentzen. Die widerspruchsfreiheit der reinen zahlentheorie. *Mathematische Annalen*, 112(1):493–565, 1936.

[Göd58] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica. International Journal of Philosophy*, 12:280–287, 1958.

[Hat02] Allen Hatcher. Algebraic Topology. Cambridge University Press, 2002. `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`

[Hed98] Michael Hedberg. A coherence theorem for Martin-Löf's type theory. *Journal of Functional Programming*, 8(4):413–436, 1998.

[Hey66] Arend Heyting. *Intuitionism: an introduction*. Studies in logic and the foundations of mathematics. North-Holland Pub. Co., 1966.

[Hil26] David Hilbert. Über das Unendliche. *Mathematische Annalen*, 95(1):161–190, 1926.

[Hof95] Martin Hofmann. *Extensional concepts in intensional type theory*. PhD thesis, University of Edinburgh, 1995.

[Hof97] Martin Hofmann. Syntax and semantics of dependent types. In Semantics and logics of computation, volume 14 of *Publictions of the Newton Institute*, pages 79–130. Cambridge University Press, Cambridge, 1997.

[How80] William A. Howard. The formulae-as-types notion of construction. In J. Roger Seldin, Jonathan P.; Hindley, editor, *To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism,* pages 479–490. Academic Press, 1980. original paper manuscript from 1969.

[HS98] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In Giovanni Sambin and Jan M. Smith, editors, *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford University Press, New York, 1998.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

[Jac99] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1999.

[JM95] A. Joyal and I. Moerdijk. *Algebraic set theory*, volume 220 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1995.

[Joh02] Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium: Vol. 1 and 2*. Num. 43 in Oxford Logic Guides. Oxford Science Publ., 2002.

[JT91] André Joyal and Myles Tierney. Strong stacks and classifying spaces. In *Category Theory. Proceedings of the International Conference held in Como, Italy, July* 22–28, 1990, volume 1488 of *Lecture Notes in Mathematics*, pages 213–236. Springer, Berlin, 1991.

[KECA13] Nicolai Kraus, Martın Escardó, Thierry Coquand, and Thorsten Altenkirch. Generalizations of hedberg's theorem. In Masahito Hasegawa, editor, *11th International Conference, Typed Lambda Calculus and Applications 2013, Eindhoven, The Netherlands, June 26–28, 2013. Proceedings*, vol. 7941 of *Lecture Notes in Comp. Sci.*, pages 173–188. Springer Berlin Heidelberg, 2013.

[KLN04] Fairouz Kamareddine, Twan Laan, and Rob Nederpelt. *A Modern Perspective on Type Theory: From its Origins until Today*. #29 in Applied Logic. Kluwer, 2004.

[KLV12] Chris Kapulkin, Peter LeFanu Lumsdaine, and Vladimir Voevodsky. The simplicial model of univalent foundations, 2012. arXiv:1211.2851.

[Knu74] Donald Ervin Knuth. *Surreal Numbers*. Addison-Wesley, 1974.

[Kol32] Andrej Kolmogorov. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift*, 35:58–65, 1932.

[Law74] F. William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43:135–166, 1974. Reprinted as Reprints in Theory and Applications of Categories 1:1–37, 2002.

[Law05] F. William Lawvere. An elementary theory of the category of sets (long version) with commentary. *Reprints in Theory and Applications of Categories*, 11:1–35, 2005. Reprinted and expanded from Proc. Nat. Acad. Sci. U.S.A. 52 (1964), With comments by the author and Colin McLarty.

[Law06] F. William Lawvere. Adjointness in foundations. *Reprints in Theory and Applications of Categories*, 16:1–16, 2006. Reprinted from Dialectica 23 (1969).

[LH12] Daniel R. Licata and Robert Harper. Canonicity for 2-dimensional type theory. In *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 337–348, New York, NY, USA, 2012. ACM.

[LS13] Daniel R. Licata and Michael Shulman. Calculating the fundamental group of the circle in homotopy type theory. In LICS 2013: *Proceedings of the Twenty-Eighth Annual ACM/IEEE Symposium on Logic in Computer Science*, 2013.

[LS17] Peter LeFanu Lumsdaine and Michael Shulman. Semantics of higher inductive types. arXiv:1705.07088, 2017. (Cited on pages 10, 163, 203, and 411.)

[Lum10] Peter LeFanu Lumsdaine. Weak omega-categories from intensional type theory. *Typed lambda calculi and applications*, 6:1–19, 2010. arXiv:0812.0409.

[Lur09] Jacob Lurie. Higher topos theory. Number 170 in Annals of Mathematics Studies. Princeton University Press, 2009. arXiv:math.CT/0608040.

[Mak95] Michael Makkai. First order logic with dependent sorts, with applications to category theory. http://www.math.mcgill.ca/makkai/folds/, 1995.

[Mak01] Michael Makkai. On comparing definitions of weak n-category. http://www.math.mcgill.ca/makkai/, August 2001.

[ML71] Per Martin-Löf. Hauptsatz for the intuitionistic theory of iterated inductive definitions. In *Proceedings of the Second Scandinavian Logic Symposium (University of Oslo 1970), volume 63 of Studies in Logic and the Foundations of Mathematics*, pages 179–216. North-Holland, 1971.

[ML75] Per Martin-Löf. An intuitionistic theory of types: predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73, Proceedings of the Logic Colloquium, volume 80 of Studies in Logic and the Foundations of Mathematics*, pages 73–118. North-Holland, 1975.

[ML82] Per Martin-Löf. Constructive mathematics and computer programming. In L. Jonathan Cohen, Jerzy Łoś, Helmut Pfeiffer, and Klaus-Peter Podewski, editors, *Logic, Methodology and Philosophy of Science VI, Proceedings of the Sixth International Congress of Logic, Methodology and Philosophy of Science, Hannover 1979*, volume 104 of *Studies in Logic and the Foundations of Mathematics*, pages 153–175. North-Holland, 1982.

[ML84] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory*. Bibliopolis, 1984.

[ML98] Per Martin-Löf. An intuitionistic theory of types. In Giovanni Sambin and Jan M. Smith, editors, *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 127–172. Oxford University Press, 1998.

[ML06] Per Martin-Löf. 100 years of Zermelo's axiom of choice: what was the problem with it? *The Computer Journal*, 49(3):345–350, 2006.

[Mog89] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1989.

[MP00] Ieke Moerdijk and Erik Palmgren. Wellfounded trees in categories. In *Proceedings of the Workshop on Proof Theory and Complexity, PTAC'98 (Aarhus)*, volume 104, pages 189–218, 2000.

[MP02] Ieke Moerdijk and Erik Palmgren. Type theories, toposes and constructive set theory: predicative aspects of AST. *Annals of Pure and Applied Logic*, 114(1–3):155–201, 2002.

[MRR88] Ray Mines, Fred Richman, and Wim Ruitenburg. *A course in constructive algebra*. SpringerVerlag, 1988.

[MS05] Maria Emilia Maietti and Giovanni Sambin. Toward a minimalist foundation for constructive mathematics. In Laura Crosilla and Peter Schuster, editors, *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics*, vol. 48 of *Oxford Logic Guides*, pp. 91–114. Clarendon Press, 2005.

[MvdB13] Ieke Moerdijk and Benno van den Berg. W-types in cartesian model categories. in preparation, 2013.

[Nor88] Bengt Nordström. Terminating general recursion. *BIT Numerical Mathematics*, 28(3):605–619, 1988.

[Nor07] Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers, Göteborg University, 2007.

[Pal07] Erik Palmgren. A constructive and functor{u,v:\mathds{R}_{\mathsf{c}}}ial embedding of locally compact metric spaces into locales. *Topology and its Applications*, 154(9):1854–1880, 2007.

[Pal09] Erik Palmgren. Constructivist and structuralist foundations: Bishop's and Lawvere's theories of sets. http://www.math.uu.se/~palmgren/cetcs.pdf, 2009.

[Pau86] Lawrence C. Paulson. Constructing recursion operators in intuitionistic type theory. *Journal of Symbolic Computation*, 2(4):325–355, 1986.

[Pie02] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[PM93] Christine Paulin-Mohring. Inductive Definitions in the System Coq — Rules and Properties. In Marc Bezem and Jan Friso Groote, editors, *Proceedings of the conference Typed Lambda Calculi and Applications*, number 664 in Lecture Notes in Computer Science, 1993.

[PPM90] Frank Pfenning and Christine Paulin-Mohring. Inductively defined types in the calculus of constructions. In Michael G. Main, Austin Melton, Michael W. Mislove, and David A. Schmidt, editors, *Mathematical Foundations of Programming Semantics, 5th International Conference, Tulane University, New Orleans, Louisiana, USA, March 29 – April 1, 1989, Proceedings*, number 442 in *Lecture Notes in Computer Science*, pages 209–228. Springer, 1990.

[PS89] Kent Petersson and Dan Synek. A set constructor for inductive sets in Martin-Löf's type theory. In David H. Pitt, David E. Rydeheard, Peter Dybjer, Andrew M. Pitts, and Axel Poigné, editors, *Category Theory and Computer Science, Manchester, UK, September 5–8, 1989, Proceedings*, volume 389 of *Lecture Notes in Computer Science*, pages 128–140. Springer, 1989.

[Rez01] Charles Rezk. A model for the homotopy theory of homotopy theory. *Transactions of the American Mathematical Society*, 353(3):973–1007, 2001. arXiv:math.AT/9811037.

[Rez05] Charles Rezk. Toposes and homotopy toposes. http://www.math.uiuc.edu/~rezk/homotopy-topos-sketch.pdf, 2005.

[Ric00] Fred Richman. The fundamental theorem of algebra: a constructive development without choice. *Pacific Journal of Mathematics*, 196(1):213–230, 2000.

[Ric08] Fred Richman. Real numbers and other completions. *Mathematical Logic Quarterly*, 54(1):98– 108, 2008.

[RS13] Egbert Rijke and Bas Spitters. Sets in homotopy type theory, 2013. arXiv:1305.3835.

[Rus08] Bertand Russell. Mathematical logic based on the theory of types. *American Journal of Mathematics*, 30:222–262, 1908.

[Sco70] Dana Scott. Constructive validity. In M. Laudet, D. Lacombe, L. Nolin, and M. Schützenberger, editors, *Symposium on Automatic Demonstration*, volume 125, pages 237–275. Springer-Verlag, 1970.

[Som10] Giovanni Sommaruga. *History and Philosophy of Constructive Type Theory*. Number 290 in Synthese Library. Kluwer, 2010.

[Spi11] Arnaud Spiwack. *A Journey Exploring the Power and Limits of Dependent Type Theory*. PhD thesis, École Polytechnique, Palaiseau, France, 2011.

[SS12] Urs Schreiber and Michael Shulman. Quantum gauge field theory in cohesive homotopy type theory. *Quantum Physics and Logic*, 2012.

[Str91] Thomas Streicher. *Semantics of type theory*. Progress in Theoretical Computer Science. Birkhäuser Boston Inc., 1991.

[Str93a] Thomas Streicher. *Investigations Into Intensional Type Theory*. PhD thesis, LMU München, 1993.

[Str93b] Thomas Streicher. Investigations into intensional type theory, 1993. Habilitationsschrift.

[Tai67] William W. Tait. Intensional interpretations of functionals of finite type. I. *The Journal of Symbolic Logic*, 32:198–212, 1967.

[Tai68] William W. Tait. Constructive reasoning. In *Logic, Methodology and Philos. Sci. III (Proc. Third Internat. Congr., Amsterdam, 1967)*, pages 185–199. North-Holland, Amsterdam, 1968.

[Tay96] Paul Taylor. Intuitionistic sets and ordinals. *The Journal of Symbolic Logic*, 61(3):705–744, 1996.

[Tay99] Paul Taylor. *Practical Foundations of Mathematics*. Cambridge University Press, 1999. (Cited on pages 342 and 344.)

[TV02] Bertrand Toën and Gabriele Vezzosi. Homotopical algebraic geometry I: Topos theory, 2002. arXiv:math/0207028.

[TvD88a] Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in mathematics. Vol. I*, volume 121 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1988. An introduction.

[TvD88b] Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in mathematics. Vol. II*, volume 123 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1988. An introduction.

[vdBG11] Benno van den Berg and Richard Garner. Types are weak w-groupoids. *Proceedings of the London Mathematical Society*, 102(2):370–394, 2011, http://plms.oxfordjournals.org/content/102/2/370.full.pdf+html

[Voe06] Vladimir Voevodsky. A very short note on the homotopy l-calculus. http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/Hlambda_short_current.pdf, 2006.

[Voe12] Vladimir Voevodsky. A universe polymorphic type system. http://uf-ias-2012.wikispaces.com/file/view/Universe+polymorphic+type+sytem.pdf, 2012.

[War08] Michael A. Warren. *Homotopy Theoretic Aspects of Constructive Type Theory*. PhD thesis, Carnegie Mellon University, 2008.

[Wik13] Wikipedia. Homotopy groups of spheres, April 2013.

[Wil10] Olov Wilander. Setoids and universes. *Mathematical Structures in Computer Science*, 20(4):563–576, 2010.

[WR27] Alfred North Whitehead and Bertrand Russell. *Principia mathematica, 3 vol.s*. Cambridge University Press, Cambridge, 1910–1913; Second edition, 1925–1927.

Список обозначений

$x :\equiv a$	определение, стр. 23
$a \equiv b$	дефинициальное равенство, стр. 23
$a =_A b$	тождественный тип, стр. 52
$a = b$	тождественный тип, стр. 52
$x \coloneqq b$	пропозициональное равенство по определению, стр. 190
$\mathsf{Id}_A(a, b)$	тождественный тип, стр. 52
$a =^P_p b$	зависимый тип пути, стр. 191
$a \neq b$	неравенство, стр. 59
refl_x	рефлексивный путь при x, стр. 52
p^{-1}	обращение пути, стр. 66
$p \centerdot q$	конкатенация путей, стр. 67
$p \centerdot_{\mathsf{l}} r$	левый вискеринг, стр. 73
$r \centerdot_{\mathsf{r}} q$	правый вискеринг, стр. 73
$r \star s$	горизонтальная конкатенация 2-путей, стр. 73
$g \circ f$	композиция функций, стр. 61
$g \circ f$	композиция морфизмов в предкатегории, стр. 322
f^{-1}	квазиобратная эквивалентность, стр. 84
f^{-1}	инверсия изоморфизма в предкатегории, стр. 323
$\mathbf{0}$	пустой тип, стр. 38
$\mathbf{1}$	единичный тип, стр. 30
$\star$	канонический обитатель в $\mathbf{1}$, стр. 30
$\mathbf{2}$	тип логических значений, стр. 39
$1_{\mathbf{2}}$, $0_{\mathbf{2}}$	конструкторы для $\mathbf{2}$, стр. 39
0_I, 1_I	точечные конструкторы интервала I, стр. 193
AC	аксиома выбора, стр. 126
AC_∞	«теоретико-типовая аксиома выбора», стр. 127
$\mathsf{acc}(a)$	предикат доступности, стр. 368
$P \wedge Q$	логическое соединение («и»), стр. 125
$\mathsf{ap}_f(p)$ или $f(p)$	применение $f : A \to B$ к $p : x =_A y$, стр. 75
$\mathsf{apd}_f(p)$	применение $f : \prod_{(a:A)} B(a)$ к $p : x =_A y$, стр. 78
$\mathsf{apd}^2_f(p)$	двумерно зависимое ap, стр. 196
$x \# y$	обособленность действительных чисел, стр. 392
base	базовая точка для $\mathbb{S}^1$, стр. 187
base	базовая точка для $\mathbb{S}^2$, стр. 189 и стр. 196

$\mathsf{biinv}(f)$	высказывание, что f би-обратима, стр. 144
$x \sim y$	би-имитация, стр. 379
$-$	символ, используемый для неявной λ-абстакции, стр. 26
$\mathcal{C}$	тип аппроксимаций Коши, стр. 401
Card	тип кардинальных чисел, стр. 364
$\bigcirc A$	рефлектор или модальность, примененная к A, стр. 259 и стр. 261
$\mathsf{cocone}_X(Y)$	тип коконусов, стр. 205
code	семейство кодов для путей, стр. 97, стр. 277, стр. 315
$A \setminus B$	дополнение к подмножеству, стр. 125
$\mathsf{cons}(x, \ell)$	конструктор конкатенации для списков, стр. 158 и стр. 216
contr_x	путь к центру сжатия, стр. 131
$\mathcal{F} \triangleleft (\mathcal{J}, \mathcal{G})$	индуктивное покрытие, стр. 421
$\mathsf{isCut}(L, U)$	свойство быть Дедекиндовым сечением, стр. 390
$\{L \mid R\}$	сечение, определяющее сюрреалистическое число, стр. 425
$X^\dagger$	обращение морфизма в †-категории, стр. 340
decode	функция декодирования путей, стр. 97, стр. 277, стр. 315
encode	функция кодирования путей, стр. 97, стр. 277, стр. 315
$\eta_A^\bigcirc$ или η_A	функция $A \to \bigcirc A$, стр. 259 и стр. 261
$A \twoheadrightarrow B$	эпиморфизм или сюръекция
$\mathsf{eq}_{\mathsf{No}}(x, y)$	конструктор пути сюрреалистических чисел, стр. 424
$\mathsf{eq}_{\mathbb{R}_c}(u, v)$	конструктор пути вещественных чисел Коши, стр. 397
$a \sim b$	отношение эквивалентности, стр. 211
$X \simeq Y$	тип эквивалентностей, стр. 83
$\mathsf{Equiv}(X, Y)$	тип эквивалентностей (такой же, как $X \simeq Y$)
$A \simeq B$	тип эквивалентностей категорий, стр. 331
$P \Leftrightarrow Q$	логическая эквивалентность, стр. 125
$\exists (x : A).\, B(x)$	логическое обозначение простого существования, стр. 125
$\mathsf{ext}(f)$	продолжение $f : A \to B$ вдоль η_A, стр. 240
$\bot$	логическое противоречие, стр. 125
$\mathsf{fib}_f(b)$	слой $f : A \to B$ в $b : B$, стр. 142
$\mathsf{Fin}(n)$	стандартный конечный тип, стр. 28
$\forall (x : A).\, B(x)$	логическое обозначение для типа зависимой функции, стр. 125
funext	функциональная экзистенсиональность, стр. 91
$A \to B$	функциональный тип, стр. 25
B^A	предкатегория функторов, стр. 326
glue	конструктор пути $A \sqcup^C B$, стр. 203
happly	функция, превращающая путь функций в гомотопию, стр. **??**
$\mathsf{hom}_A(a, b)$	hom-множество в предкатегории, стр. 322
$f \sim g$	гомотопия между функциями, стр. 81
I	тип интервала, стр. 193
id_A	тождественная функция для A, стр. 30
1_a	тождественный морфизм в предкатегории, стр. 322

idtoeqv	функция $(A = B) \to (A \simeq B)$, инвертирующая унивалентность, стр. 94
idtoiso	функция $(a = b) \to (a \cong b)$ в предкатегории, стр. 323
$\mathsf{im}(f)$	образ отображения f, стр. 254
$\mathsf{im}_n(f)$	n-образ отображения f, стр. 254
$P \Rightarrow Q$	логическая импликация («влечет за собой»), стр. 125
$a \in P$	принадлежность к подмножеству или подтипу, стр. 122
$x \in v$	принадлежность к кумулятивной иерархии, стр. 378
$x \mathrel{\tilde{\in}} v$	изменяемая степень принадлежности, стр. 382
$\mathsf{ind}_{\mathbf{0}}$	индукция для $\mathbf{0}$, стр. 39,
$\mathsf{ind}_{\mathbf{1}}$	индукция для $\mathbf{1}$, стр. 34,
$\mathsf{ind}_{\mathbf{2}}$	индукция для $\mathbf{2}$, стр. 40,
$\mathsf{ind}_{\mathbb{N}}$	индукция для $\mathbb{N}$, стр. 43, и
$\mathsf{ind}_{=_A}$	индукция пути для $=_A$, стр. 54,
$\mathsf{ind}'_{=_A}$	индукция базированного пути для $=_A$, стр. 55,
$\mathsf{ind}_{A\times B}$	индукция для $A \times B$, стр. 34,
$\mathsf{ind}_{\sum_{(x:A)} B(x)}$	индукция для $\sum_{(x:A)} B$, стр. 36,
ind_{A+B}	индукция для $A + B$, стр. 39,
$\mathsf{ind}_{W_{(x:A)}B(x)}$	индукция для $W_{(x:A)}B$, стр. 176
$A_{/a}$	начальный сегмент ординала, стр. 372
$\mathsf{inj}(A, B)$	тип инъекций, стр. 366
inl	первая инъекция в копроизведение, стр. 38
inr	вторая инъекция в копроизведение, стр. 38
$A \cap B$	пересечение подмножеств, стр. 125, классов, стр. 381, или интервалов, стр. 420
$\mathsf{isContr}(A)$	высказывание, что A является стягиваемым, стр. 131
$\mathsf{isequiv}(f)$	высказывание, что f является эквивалентностью, стр. 83, стр. 137, и стр. 146
$\mathsf{ishae}(f)$	высказывание, что f является полусопряженной эквивалентностью, стр. 140
$a \cong b$	тип изоморфизмов в (пред)категории, стр. 323
$A \cong B$	тип изоморфизмов между предкатегориями, стр. 333
$A \cong B$	тип изоморфизмов между множествами, стр. 83
$a \cong^{\dagger} b$	тип унитарных изоморфизмов, стр. 341
isotoid	инверсия idtoiso в категории, стр. 324
$\mathsf{is}\text{-}\mathsf{n}\text{-}\mathsf{type}(X)$	высказывание, что X является n-типом, стр. 231
$\mathsf{isProp}(A)$	высказывание, что A является простым высказыванием, стр. 118
$\mathsf{isSet}(A)$	высказывание, что A является множеством, стр. 113
$A * B$	соединение A и B, стр. 207
$\mathsf{ker}(f)$	ядро отображения точечных множеств, стр. 287
$\lambda x. b(x)$	λ-абстракция, стр. 29
$\mathsf{lcoh}_f(g, \eta)$	тип данных левой сопряженной когерентности, стр. 143
LEM	закон исключения третьего, стр. 119
LEM_∞	противоречивые высказывания-как-типы LEM, стр. 117 и стр. 120

$x < y$	строгое неравенство на натуральных числах, стр. 50, ординалах, стр. 368, действительных числах Коши, стр. 411, сюрреалистических числах, стр. 424, и т.д.
$x \leqslant y$	не-строгое неравенство на натуральных числах, стр. 50, действительных числах Коши, стр. 411, сюрреалистических числах, стр. 424, и т.д.
$\preccurlyeq$, $\prec$	рекурсивные версии $\leqslant$ и $<$ для сюрреалистических чисел, стр. 430
$\trianglelefteq, \triangleleft, \sqsubseteq, \sqsubset$	упорядочения в кодомене NO-рекурсии, стр. 427
$\mathsf{lim}(x)$	предел аппроксимации Коши, стр. 397
$\mathsf{linv}(f)$	тип левых обратных к f, стр. 143
$\mathsf{List}(X)$	тип списков элементов X, стр. 158 и стр. 216
loop	конструктор пути для $\mathbb{S}^1$, стр. 187
$\mathsf{Map}_*(A, B)$	тип базовых отображений, стр. 199
$x \mapsto b$	альтернативное обозначение для λ-абстракции, стр. 26
$\mathsf{max}(x, y)$	максимум в каком-то порядке, например, стр. 392 и стр. 411
$\mathsf{merid}(a)$	меридиан ΣA в $a : A$, стр. 197
$\mathsf{min}(x, y)$	минимум в каком-то порядке, например, стр. 392 и стр. 411
$A \rightarrowtail B$	мономорфизм или вложение
$\mathbb{N}$	тип натуральных чисел, стр. 41
N	северный полюс ΣA, стр. 197
$\mathbf{N^w}$, $0^{\mathbf{w}}$, $\mathsf{sucw}^{\mathbf{w}}$	натуральные числа, закодированные как W-тип, стр. 163
$\mathbb{N}\mathsf{alg}$	тип $\mathbb{N}$-алгебр, стр. 166
$\mathbb{N}\mathsf{Hom}(C, D)$	тип $\mathbb{N}$-гомоморфизмов, стр. 166
nil	пустой список, стр. 158 и стр. 216
No	тип сюрреалистических чисел, стр. 424
$\neg P$	логическое отрицание («не»), стр. 125
n-Type, n-$\mathsf{Type}_{\mathcal{U}}$	универсум n-типов, стр. 234
$\Omega(A, a)$, ΩA	пространство петель точечного типа, стр. 75
$\Omega^k(A, a)$, $\Omega^k A$	итеративное пространство петель, стр. 75
A^{op}	обратная предкатегория, стр. 336
$P \vee Q$	логическая дизъюнкция («или»), стр. 125
Ord	тип ординальных чисел, стр. 372
(a, b)	(зависимая) пара, стр. 30 и стр. 35
$\mathsf{pair}^{=}$	конструктор для $=_{A \times B}$, стр. 86
$\pi_n(A)$	n-я гомотопическая группа A, стр. 216 и стр. 274
$\mathcal{P}(A)$	степенное множество, стр. 123
$\mathcal{P}_+(A)$	просто обитаемое степенное множество, стр. 375
pred	функция-предшественник $\mathbb{Z} \to \mathbb{Z}$, стр. 278
$A \times B$	тип декартова произведения, стр. 30
$\prod_{(x:A)} B(x)$	зависимый функциональный тип, стр. 29
$\mathsf{pr}_1(t)$	первая проекция из пары, стр. 32 и стр. 35
$\mathsf{pr}_2(t)$	вторая проекция из пары, стр. 32 и стр. 35
Prop, $\mathsf{Prop}_{\mathcal{U}}$	универсум простых высказываний, стр. 122

$A \times_C B$	обратный образ A и B над C, стр. 107
$A \sqcup^C B$	амальгама A и B под C, стр. 203
$\mathbb{Q}$	тип рациональных чисел, стр. 388
$\mathbb{Q}_+$	тип положительных рациональных чисел, стр. 388
$\mathsf{qinv}(f)$	тип квази-обратных к f, стр. 82
A/R	частное множества по отношению эквивалентности, стр. 210
$A \mathbin{/\!\!/} R$	альтернативное определение частного, стр. 211
$\mathbb{R}$	тип действительных чисел (любой), стр. 416
$\mathbb{R}_c$	тип действительных чисел Коши, стр. 397
$\mathbb{R}_d$	тип действительных чисел Дедекинда, стр. 390
$\mathsf{rat}(q)$	рациональное число, рассматриваемое как действительное число Коши, стр. 397
$\mathsf{rcoh}_f(g, \epsilon)$	тип данных правосопряженной когерентности, стр. 143
$\mathsf{rec}_{\mathbf{0}}$	рекурсор для $\mathbf{0}$, стр. 38
$\mathsf{rec}_{\mathbf{1}}$	рекурсор для $\mathbf{1}$, стр. 33
$\mathsf{rec}_{\mathbf{2}}$	рекурсор для $\mathbf{2}$, стр. 39
$\mathsf{rec}\mathbb{N}$	рекурсор для $\mathbb{N}$, стр. 42
$\mathsf{rec}_{A\times B}$	рекурсор для $A \times B$, стр. 32
$\mathsf{rec}\sum_{(x:A)} B(x)$	рекурсор для $\sum_{(x:A)} B$, стр. 36
rec_{A+B}	рекурсор для $A + B$, стр. 38
$\mathsf{rec}_{W_{(x:A)}B(x)}$	рекурсор для $W_{(x:A)}B$, стр. 164
rinv	тип правых обратных к f, стр. 143
S	южный полюс ΣA, стр. 197
$\mathbb{S}^n$	n-мерная сфера, стр. 195
seg	конструктор пути интервала I, стр. 193
Set, $\mathsf{set}_{\mathcal{U}}$	универсум множеств, стр. 122
$\mathcal{S}et$	категория множеств, стр. 323
$\mathsf{set}(A, f)$	конструктор кумулятивной иерархии, стр. 377
$x \sim_\epsilon y$	отношение ϵ-близости для $\mathbb{R}_c$, стр. 397
$x \approx_\epsilon y$	рекурсивная версия $\sim_\epsilon$, стр. 405
$\frown_\epsilon$ or $\smile_\epsilon$	отношения близости на кодоменах $\mathbb{R}_c$-рекурсии, стр. 399
$A \wedge B$	скрещенное (стянутое) произведение A и B, стр. 207
$\{x : A \mid P(x)\}$	тип подмножества, стр. 121
$\{f(x) \mid P(x)\}$	образ подмножества, стр. 359
$B \subseteq C$	включение подмножества типов, стр. 122
$(q, r) \subseteq (s, t)$	включение интервалов, стр. 420
succ	функция преемника $\mathbb{N} \to \mathbb{N}$, стр. 41
succ	функция преемника $\mathbb{Z} \to \mathbb{Z}$, стр. 275
$A + B$	тип копроизведения, стр. 38
$\sum_{(x:A)} B(x)$	тип зависимой пары, стр. 34
$\mathsf{sup}(a, f)$	конструктор для W-типа, стр. 163
surf	конструктор 2-пути $\mathbb{S}^2$, стр. 189 и стр. 196
ΣA	надстройка A, стр. 197

$\mathsf{total}(f)$	индуцированное отображение на пространствах расслоений, стр. 149
$p_*(u)$	транспортирование $u : P(x)$ по $p : x = y$, стр. 76
$\mathsf{transport}^p(p, u)$	транспортирование $u : P(x)$ по $p : x = y$, стр. 76
$\mathsf{transport}^2(X, Y)$	двумерное транспортирование, стр. 196
$\mathsf{transportconst}^X_Y(Z)$	транспортирование в постоянное семейство, стр. 79
$\lVert A\rVert_n$	n-усечение A, стр. 238
$\lvert a\rvert^A_n$, $\lvert a\rvert_n$	образ $a : A$ в $\lVert A\rVert_n$, стр. 238
$\lVert A\rVert$	пропозициональное усечение A, стр. 124 и стр. 207
$\lvert a\rvert$	образ $a : A$ в $\lVert A\rVert$, стр. 124 и стр. 207
$\top$	логическая истина, стр. 125
_	неименованный объект или переменная
$A \cup B$	объединение подмножеств, стр. 125
$\mathsf{uniq}_{A\times B}$	принцип уникальности произведения $A \times B$, стр. 33
$\mathsf{uniq}_{\mathbf{1}}$	принцип уникальности для $\mathbf{1}$, стр. 34
$\mathcal{U}$	тип универсума, стр. 28
$\mathcal{U}_{\bigcirc}$	универсум модальных типов, стр. 261
$\mathcal{U}_\bullet$	универсум точечных типов, стр. 75
ua	обратное к $\mathsf{idtoeqv}$ из унивалентности, стр. 94
V	кумулятивная иерархия, стр. 377
$\mathsf{WAlg}(A, B)$	тип w-алгебр, стр. 167
$\mathsf{WHom}_{A,B}(C, D)$	тип W-гомоморфизмов, стр. 167
$W_{(x:A)}B(x)$	W-тип (индуктивный тип), стр. 162
$A \vee B$	клин из A и B, стр. 207
$\mathbf{y}$	вложение Йонеды, стр. 337
$\mathbb{Z}$	тип целых чисел, стр. 212

Предметный указатель

$\bigcirc$-связная функция, **262**
$\bigcirc$-усеченная функция, **262**
$\dagger$-категория, **341**
$\dagger$-предкатегория, **340**
  унитарный морфизм, 341
∞-связная функция, **314**
∞-функтор, 66
∞-группа, 215
∞-группоид, 5, 64–66, 108, 166, 180, 188, 271, 273, 274, 311
  структура типа, 66–75
  фундаментальный, 64
∞-усеченный тип, **314**
$(\infty, 1)$-категория, 109, 155, 166, 240, 321, 352
$(\infty, 1)$-топос, 15, 152, 222, 262, 263, 271, 308, 311, 314, 351, 352
  не-гиперполный, 311
1-тип, **115**
2-категория, **353**
2-мерный путь, *см.* путь, 2-
2-путь, *см.* путь, 2-
3-мерный путь, *см.* путь, 3-
3-путь, *см.* путь, 3-

h-высказывание, *см.* простое высказывание
h-инициальная, *см.* гомотопически-инициальная
h-уровень, *см.* n-тип
hom-множество, **322**
hom-функтор, **337**

$\mathcal{J}$, *см.* принцип индукции для тождественного типа

n-путь, *см.* путь, n-
n-мерная петля, *см.* петля, n-
n-мерный путь, *см.* путь, n-
n-петля, *см.* петля, n-
n-тип, 115
  определяемый в теории типов, 133

Π-тип, *см.* тип, зависимой функции

UIP, *см.* уникальность доказательств тождественности

Аккермана функция, **62**
Больцано-Вейерштрасс, *см.* компактность
Бурбаки, 109
Ван Кампена теорема, 353
Галуа
  группа, 340
  расширение, 340
Гейне-Борель, *см.* компактность
Дедекинд
  сечение, *см.* сечение, Дедекинда
Дедекинда
  действительные числа, *см.* действительные числа, Дедекинда
  полнота, **395**
  пополнение, *см.* пополнение, Дедекинда
Диаконеску, теорема, 364
Евклид Александрийский, 65
Йонеды
  вложение, **337**
  лемма, 180, **337**, 336–339
Кантора, теорема, 367
Коши
  аппроксимация, **393**, **398**, 417
    зависимая, **399**
    тип, 401
  действительные числа, *см.* действительные числа, Коши
  полнота, **414**
  пополнение, *см.* пополнение, Коши
  последовательность, 387, **393**, 396, 398, 415, 419, 438
Лифшица, функция, постоянная, **404**
Ловер, 8, 11, 59, 183, 184, 262, 355, 364, 383, 438
Мартин-Лёф, 183
Пеано, 183
Рассел, Бертран, 4, 134
Сигала, категория, пространство, 352
Скотт, 184
Тирней, 262
Фейта-Томпсона теорема, 8

Фреге, 183
Хедберга теорема, 237
Цермело-Френкеля теория множеств, *см.* теория множеств
Шрёдера-Бернштейна, теорема, 367
абсолютное значение, 411
абстрактная двойственность Стоуна, 435
абстракция, 443
  λ-, *см.* λ-абстракция
автоморфизм
  $\mathbb{Z}$, функция следования, 274
  без неподвижной точки, 12, 116
  для **2**, нетождественный, 227, 229
  из $\mathbb{S}^1$, 196
  экстенсиональных вполне обоснованных отношений, 371
аксиома
  Аксиома K Стрейчера, **235**, 263
  Аксиома K Стрейчера
    обобщение на n-типы, 237
  Стрейчера, Аксиома K, 59, 60
  бесконечности, 380
  выбора, 11, 12, **125–127**, 135, 321, 354, 364, 366, 375
    $\mathsf{AC}_{n,m}$, **265**
    единственного, *см.* единственность выбора
    зависимого, **436**
    n-связная, **265**
    счетного, 387, **396**, 415, 436
    теоретико-типовая, 37, 106, 109, 116
  двойного отрицания, **119**
  Δ_0-отделения, 382
  замены, 380
  исключения третьего, *см.* исключение третьего
  нестабильная октаэдрическая, 156
  ограниченный принцип всеведения, *см.* ограниченный принцип всеведения
  отделимости, 380
  приводимости, 134
  принцип Маркова, **438**
  принцип Уайтхеда, **314**, 311–314
  пропозиционального изменения размера, *см.* пропозициональное изменение размера
  против правил, 24, 85, 190
  сильной совокупности, 382, **385**
  совокупности подмножеств, 382
  теории множеств, для кумулятивной иерархии, 380
  унивалентности, 3, 6, 11, **94**
    конструктивность, 14
  функциональной экстенсиональности, *см.* функциональная экстенсиональность
  экстенсиональности, 371
аксиома унивалентности, 85, 102, 108, 116, 153, 161, 182, 274, 324, 453
аксиоматическая свобода, 48
акцентирование, 28, 77, 120, 179, 218, 271–320, 332, 346, 421–423, 434–435
алгебра
  2-клетки, **168**
  W-, **167**
  для полиномиального функтора, **167**
  для эндофунктора, 167, 183
  инициальная, *см.* гомотопически-инициальная, 175
  $\mathbb{N}$-, **166**
  свободная, 216
алгебраическая теория множеств, 355, 383
алгоритм, 7, 8, 10, 11, 23, 48, 66, 217
α-конверсия, 443
амальгама, **204–207**, 291, 317
  в n-типах, **245**
  множеств, 209
амальгированное свободное произведение, **219**, 310
анализ
  классический, 416
  конструктивный, 416
анализ случаев, **38**, 158
аналитическая математика, 65
аппроксимация, Коши, *см.* Коши аппроксимация
аргумент Экманна-Хилтона, 73, 216, 274, 289
арность, 163, 344
архимедово свойство, *см.* упорядоченное поле, архимедово
ассоциативность, 217
  Σ-типов, 110
  в группе, 215
  в моноиде, 215
  композиции функторов, 329
    согласованная, 329
  конкатенации пути, 70
    согласованность, 72
  конкатенации списков, 216
  полугрупповой операции, 102
  сложения
    действительных чисел Коши, 411
    натуральных чисел, 43
  соединения, 294

  функциональной композиции, 61
  функциональных типов, 27
базируемое отображение, 199
башня Постникова, 11, 238
бесконечная алгебраическая теория, 220
бесточечная топология, 420
β-конверсия, *см.* β-редукция
β-редукция, 26, 31
би-имитация, **379**
би-обратимая функция, **144**
биекция, 83, **146**
бимодуль, 430
биполное отношение, *см.* отношение, биполное
бит, 118
Блейкерса-Месси теорема, *см.* теорема, Блейкерса-Месси
булевы значения
  тип, *см.* тип булевых значений
бутылка Клейна, 201
W-алгебра, **167**
W-гомоморфизм, **167**
W-тип, 163
  как гомотопически-инициальная алгебра, 167
  непредикативное кодирование, 185
Ван Кампена теорема, 302–311
введения, правило, 450
вектор, 177
  принцип индукции, 177
векторное пространство, 413
векторное пространство конечномерное, 341
верхние действительные числа Дедекинда, **437**
вершина ко-конуса, 205
винтовое отображение, 276
вискеринг, **74**, 238
включение
  интервалов, **420**
  подмножеств, **122**
вложение, *см.* функция, вложения
  Йонеды, **337**
возведение в степень, кардинальных чисел, **365**
вполне обоснованная индукция, **369**
вполне обоснованное отношение, **369**
вполне ограниченное метрическое пространство, **417**
вполне точный функтор, **331**
выведение, 448
выделитель
  индуктивного типа
    зависимый, *см.* принцип индукции
    независимый, *см.* принцип рекурсии
  типа, **31**
высказывание
  как тип, 10, 116–117
  простое, *см.* простое высказывание
высказывания как типы, **46–52**
высший индуктивный тип, *см.* тип, высший индуктивный
вычислительное правило, 450
вычислительный эффект, 263
гейм Конвея, 423, 425
геметрическая реализация, 64
генерация типа, индуктивная, 53–57
геометрическая реализация, 271, 308
геометрия, синтетическая, 65
гетерогенное равенство, 191
гиперполный тип, **314**
гипотеза, **24**, 47, 49
  гомотопическая, 64
  индуктивная, 44
гладкая функция следования, 386
гладкий ординал, 386, 439
глобулярная операда, 72
гомология, 272
гомоморфизм
  алгебры для функтора, **167**
  W-, **167**
  групповой, **217**
  моноидный, **216**
  $\mathbb{N}$-, **166**
  Ω-структур, **344**
  полугрупп, 104
  поля, 340, 395
  структур, **342**
гомотопическая
  группа, 216, 273, **274**
    сферы, 290, 301, 316
  (пред)категория типов, 325, 351
  эквивалентность, *см.* эквивалентность
    топологическая, 4, 272
гомотопическая группа сферы, 273
гомотопическаяягипотеза, 64
гомотопически-индуктивный тип, 169
гомотопически-инициальная
  алгебра для функтора, **168**
  W-алгебра, **168**
  $\mathbb{N}$-алгебра, **166**
гомотопический тип, 5
гомотопия, **80–82**, 91–93
  индукция, 183
  топологическая, 4, 63

горизонтальная композиция
естественных преобразований, 329
путей, **74**
граф, 109, **264**
с композицией, **266**
группа, **215**
абелева, 74, 140, 216, 274, 285, 310, 317, 391
точная последовательность, 289
гомотопическая, *см.* гомотопическая группа
свободная, 217–219
фундаментальная, *см.* фундаментальная группа
группа циклическая, 273
групповой гомоморфизм, 217
группоид, **325**
высший, 72
фундаментальный, *см.* фундаментальный группоид
двойное отрицание, закон, 116, **119**
де Моргана законы, 47–48
дедуктивная система, 21
как игра, 21
действие
зависимой функции на пути, **78**
функции на пути, **75**
действительные числа, 387–439
Дедекинда, **390**, 389–395, 415–416
верхние, **437**
нижние, **437**
Коши, **397**, 396–416
Эскардо-Симпсона, 436
гомотопические, **281**
расширенные, **436**
согласованные, 416
декартово произведение, *см.* тип, произведения
декодирование, *см.* метод кодирования-декодирования
дерево, вполне фундированное, 162
дефинициальное равенство, *см.* равенство, дефинициальное, 85, 189
диаграмма, 81, 109, 264, 266
диадические рациональные числа, *см.* рациональные числа, диадические
дизъюнктивная отмеченная точка, 199
дизъюнкция, 46, **125**
диск, 200, 202
дискретная категория, **325**
дискретное пространство, 8, 10, 14, 113
доказательство, 22, 46–52
истинности предложения, 22
компьютерное, 4, 9, 10, 59, 227, 228, 272, 349, 441
Agda, 60
Coq, 60, 108, 134
NuPRL, 134
от противного, 47, 48, 120
доказательство единственности
тождественности, **235**
дополнение, подмножества, **125**
допустимое
правило, *см.* правило, допустимое
допустимость
упорядоченного поля, *см.* упорядоченное поле, допустимость
доступность, **368**, 368, 383
доступный, *см.* доступность
единица
группы, **215**
закон для конкатенации пути, 70
кольца, 391, 393
моноида, **215**
сопряжения, **330**
единицы тип, *см.* тип, единицы
единичный интервал, 6
единственности
принцип, пропозициональный
для функций на $\mathbb{N}$, 159
единственность
выбора, **127–128**
принцип, пропозициональный
для функций на усечении, 240
тождественных типов, 160
ϵ-сеть, **417**
естественное преобразование, 138, **326**, 343
«естественность» гомотопий, 81
естественный изоморфизм, 327
зависимая
n-петля, 227, 229
аппроксимация Коши, **399**
функция, *см.* функция, зависимая
зависимое сечение, **427**
зависимый
путь, *см.* путь, зависимый
тип, *см.* семейство, типов
зависимый выделитель, *см.* принцип индукции
закон
двойного отрицания, **119**
де Моргана, 47–48
исключения третьего, 48, 117, *см.* исключение третьего

закон замены, 328
закон исключения третьего, 11, 12
закрытый интервал, **413**
замена, 24, 27
замена слоя, 352
замкнутая модальность, 265
замкнутый терм, 455
захват, переменной, **27**
ZF, *см.* теория множеств
ZF-алгебра, 386
ZFC, *см.* теория множеств
зигзагообразное тождество, **330**
злоупотребление
обозначением, 7
обозначениями, 102
языком, 129, 130
значение
истинностное, 41
функции, **25**
идемпотент
модальность, 262
функция, **212**
иерархия
кумулятивная, теоретико-множественная, **377**
универсумов, *см.* тип, универсума
извлечение алгоритмов, 10, 11
изменение размера, 134, 361, 379
пропозициональное, *см.* пропозициональное изменение размера
изменяющийся вдоль конструктора пути, 191
изометрия, **341**
изоморфизм
(пред)категорий, **333**
в (пред)категории, **323**
естественный, 81, 327
множеств, 83, **146**
перенос через, 161
полугрупп, 104
унитарный, **341**
имитация, **371**
ограниченная, **373**
импликация, 46, **125**
инвариантность относительно изоморфизма, 352
инверсия
в (пред)категории, 323
в группе, 215
индекс индуктивного семейства, **177**
индкутивная гипотеза, 44
индуктивно-индуктивный тип, 178
высший, 397
индуктивно-рекурсивный тип, 179
индуктивное
определение, 157
покрытие, **421**
семейство типов, 177
индуктивный
предикат, 177
тип, *см.* тип, индуктивный
высший, *см.* тип, высший индуктивный
инициальная
алгебраическая характеризация индуктивных типов, *см.* гомотопически-инициальная
инициальное множество, 362
инициальный объект категории полей, 220
инициальный тип, *см.* тип, пустой
интенсиональная теория типов, 59, 107
интервал
арифметический, 437
открытый и закрытый, **413**, 417, 422
поточечное покрытие, **420**
семейство, базовых, **419**
тип, *см.* тип, интервала
топологическая единица, 6
интервальная
арифметика, 391
область, **437**
инфиксная нотация, 443
инъективная функция, *см.* функция, инъективная
инъекция, *см.* функция, инъективная
иррефлективность
вполне обоснованного отношения, 375
исключение третьего, **119**, 237, 364, 366, 374, 375, 389, 415, 420
$\mathsf{LEM}_{n,m}$, **264**
истина, 46, **125**
источник
конструктора пути, 201
построенного пути, 226
пути конструктора, **187**
исходный
σ-каркас, 390
σ-фрейм, 415
итератор
для натуральных чисел, 61
k-морфизм, **64**
каноничность, 14, 25, **455**
кардинальное число, **364**
возведение в сепень, 365
неравенство, 366

сложение, 365
умножение, 365
кардинальность, **365**
каррирование, **27**
категорий
изоморфизм, **333**
произведение, 336
эквивалентность, **331**
категория, **323**
$(\infty,1)$-, *см.* $(\infty,1)$-категория
дискретная, **325**
кополная, 356
локально декартово замкнутая, 362
обратная, **336**
полная, 356
регулярная, **356**
скелетальная, 324
срезов, **352**, 384
строгая, **340**
типов, 351
тощая, **324**, 340
функторов, **327**
центр, 138
четко акцентированная, 364, **384**
квазиобратная, **82**
квазиобратный, 138–140
квантор, 49, 125
всеобщности, 49, 123, **125**
ограниченный, 381
существования, 49, 124, 125, 390
класс, **378**
малый, **379**
отделимый, **381**
эквивалентности, **211**
классификатор
объектов, **152**, 155
подобъектов, 362
классическая
теория гомотопий, 63–64, 272–273, 275–276
теория категорий, 321, 322, 329, 341, 345, 346
классический анализ, 387, 416
CW-комплекс, **200–203**
клеточный комплекс, **200–203**
клин, **207**, 296
ко-конус, **205**, 244
ко-предел типов, 264, 266
ко-расслоение, 310
ко-слой функции, **207**
ко-транзитивность отделенности, 393
ковариантный функтор, 173
когерентность, 140, 143
когомология, 272
кодирование, *см.* метод кодирования-декодирования
коединица сопряжения, **330**
кольцо, 388, 391–393
коммутативный квадрат, **110**
комонада, 263
компактность, **416**
Больцано-Вейерштрасса, 416, 419
Гейне-Бореля, 416, **420**, 422
метрическая, **387**, **416**, 417
комплекс
CW, 7
Кана, 6, 13
CW-комплекс, 310
комплекс Кана, 271, 272, 456
композиция
морфизмов в (пред)категории, **322**
путей, **67**
горизонтальная, **74**
функций, **61**
компонент пары, *см.* проекция
компьютерная система AUTOMATH, 456
конверсия
α-, *см.* α-конверсия
β-, *см.* β-редукция
η-, *см.* η-расширение
конвртируемость термов, 444
конечная точка пути, 65
конечные
множества, семейства, 30
списки, типа, 216
конкатенация путей, **67**
константа
определенная, 443
примитивная, 443
явная, 443
конструктивная теория множеств, 382
конструктивность, 13
конструктивный анализ, 416
конструктор, 174
пути, **187**
типа, **31**
точки, 187
конструкция Гротендика, 221
конструкция Хопфа, **293**
контекст, 24, 442, 448
правильно построенный, 442
контравариантный функтор, 173

конус
сферы, 202
функции, **207**, 358
конфлюэнтность, 455
концентратор и спица, **202–203**, 239, 310
конъюнкция, 46, **125**
кообласть функции, 25
кополная категория, 356
копредел
алгебраических объектов, 219
множеств, 209, 356
типов, 106, 203
копроизведение, *см.* тип, копроизведения
коуравнитель, **221**
множеств, *см.* set-coequalizer
кумулятивная иерархия, теоретико-множественная, **377**
кумулятивные универсумы, **28**
левая обратимая функция, 143
левое сопряжение, **330**
левый обратный, **143**
лемма, 21
сглаживания, **221**, 291
линейное отображение, *см.* функция, линейная
линейный порядок, 391
логика
высказывания как типы, 46–52
интуиционистская, 12
конструктивная, 12
конструктивная против классической, 12, 47–49, 116–117, 119–121
первого порядка, 21, 128
пропозициональная, 46
простых высказываний, 117–119, 123–125, 128–131
рондикатная, 49
усеченная, 128
логическая нотация, традиционная, **125**
логическая эквивалентность, **51**
логический топос, 364
ложь, 46, 47, **125**
из лжи следует все, 38
локализация индуктивного покрытия, 421
локализованность, 389, 390
локаль, 420
локально декартово замкнутая категория, 362
локально равномерно непрерывное отображение, **438**
λ-абстракция, **25**, 27, **29**, 44, 50, 444
λ-исчисление, 4
магма, **37**, 51
малое множество, 323
малый
класс, **379**
тип, 6, **28**
Мартин-Лёф, 456
математика
классическая, 10, 11, 41, 48, 59, 106, 116–118, 120, 121, 123, 126, 321, 324, 326, 341, 345, 346, 362, 363, 370, 374, 375, 387, 389, 390, 416–423
конструктивная, 10–13, 43, 215, 272, 317, 355, 362, 387, 389, 392, 396, 416–424
предикативная, **122**, 174, 355, 362, 383, 421
релевантная для доказательства, 24, 36, 51, 69, 82, 83, 222
синтетическая, 271
формализованная, 4, 8, **9–10**, 15, 23, 69, 146, 178, 272, 317, 434
меридиан, **197**, 202
местоположение, **438**
метатеория, 454–456
метод кодирования-декодирования, **99**, 97–101, 278–280, 298–306, 308–309, 315, 350, 360, 409
метрическая компактность, **416**
метрически компактный, **387**
метрическое пространство, **416**, 416–438
вполне органиченное, **417**
полное, **417**
множество, 8–9, **113–115**, 146, 208–210, 232, 235–237, 323, 355–386
в кумулятивной иерархии, **378**
отмеченных точек, 307
множество-амальгама, **209**
множество-коуравнитель, 356, **357**
множество-частное, **210–215**, 228, 360–362
модальная логика, 262
модальность, **261**, 258–263
замкнутая, **265**
открытая, **265**
тождественная, 262
модальный
оператор, 262, 263, 265
тип, **261**
модельная категория, 6
модельная категория Квиллена, 6, 271
модуль
равномерной непрерывности, **417**
сходимости, **393**

монада, 228, 263
монарный, *см.* мономорфизм
моно, *см.* мономорфизм
моноид, **215**, 215–219
свободный, 216–217, 229
моноидный гомоморфизм, **216**
мономорфизм, 340, **357**, 362
монотонность, 405
индуктивного покрытия, 421
морфизм
в ∞-группоиде, **64**
в (пред)категории, **322**
унитарный, **341**
$\mathbb{N}$-алгебра гомотопически-инициальная, **166**
$\mathbb{N}$-алгебра, **166**
$\mathbb{N}$-гомоморфизм, **166**
n-образ, **254**
функции, **254**
n-связная
аксиома выбора, **265**
n-связная функция, *см.* функция, n-связная
n-связный тип, *см.* тип, n-связный
n-сфера, *см.* тип, n-сферы
n-тип, 11, **232**, 231–258
n-усечение, *см.* усечение
n-усеченная функция, **254**
n-усеченный тип, *см.* n-тип
надстройка, **197–200**, 207
наименьшая верхняя граница, *см.* супремум
наречие, 130, 262
конструктивно, **130**
просто, **130**
чисто, **130**
натуральные числа, **41–44**, 100–101, 158, 213
закодированные как $\mathsf{List}(\mathbf{1})$, 161
закодированные как W-тип, 172
изоморфное определение, 160
как гомотопически-инициальная алгебра, 165
кодированные как W-тип, 163
начальная точка пути, **65**
начальное упорядоченное поле, 388
начальный сегмент, **372**, 372
негативный тип, 97
недискретная предкатегория, **335**
недостижимые кардинальные числа, 13
независимый выделитель, *см.* принцип рекурсии
непересекающаяся
сумма, *см.* тип, копроизведения
непредикативное
кодирование W-типа, 185
усечение, 134
частное, 211, 361
непредикативность, *см.* математика, предикативная
для простых высказываний, *см.* пропозициональное изменение размера
«непрерывность» функций в теории типов, 5, 75, 77, 81, 91, 113, 116, 131, 236
непротиворечивость, 441, 455, 456
арифметики, 383
непустое подмножество, **370**, 375
неравенство треугольника, *см.* треугольника неравенство для $\mathbb{R}_\mathsf{c}$
неразличимость тождественности, 53, 76
нерасширяющаяся функция, 410
нерефлексивность
$<$ в поле, 393
of $<$ для действительных чисел, 391
отделенности, 393
несвязное объединение, *см.* тип, копроизведения
нестрогий порядок, 411, 425
несущая, 37
неформальная теория типов, **9–10**
нечеткого порядка теорема, 8
неэквивалентность, **59**
неявный аргумент, 453
нижние действительные числа Дедекинда, **437**
нисходящие данные, **354**
нормализация, 455
сильная, 455
нормализуемый терм, **455**
нормальная форма терма, 455
носитель, 102
нулевое отображение, 286
нуль, 41, 159, 163, 164
нульарное
копроизведение, *см.* тип, пустой
произведение, *см.* тип, единицы
обитаемый тип, **52**, 118, 120
просто, **130**
область, **26**, 30, 443
конструктора, 173
переменной, **443**
область функции, 25
образ, **254**, 287, 356
n-образ, **254**
подмножества, **359**
сохранение под обратным образом, 257
образующая
группы, 310, 311

для группы, 216
обратная к предкатегории, **336**
обратный
аппроксимирующий, 388
левый, **143**
правый, **143**
путь, **66**
обратный образ, **106**, **110**, 152, 203, 233, 356
объединение
несвязное, *см.* тип, копроизведения
подмножеств, **125**
объект
в (пред)категории, **322**
классификатор, **152**, 155
подтерминальный, *см.* простое высказывание
ограниченная имитация, **373**
ограниченный квантор, 381
ограниченный принцип всеведения, **419**, 437
ограниченный принцип всезнания, 385
односвязный тип, **249**
одноэлементный тип, *см.* тип, синглетон
округление
отношение, **409**
округленное
сечение Дедекинда, 390, 437
округленное сечение Дедекинда, 389
октаэдрическая аксиома, нестабильная, 156
Ω-структура, *см.* структура
операда, 72
оператор, *см.* функция
выбора, 117
операция следования, **41**
определение, 453
индуктивное, 157
наречий, 130
общерекурсивное, 444
по сопоставлению с образцом, 44–45
посредством сопоставления с образцом, 176
путем сопоставления с образцом, 457
структурной рекурсией, 444
функции, прямое, **25**
функции, явное, **29**
определяющее уравнение, 443
опция сюрреалистических чисел, **424**
ординал, **372**, 426
гладкий, 386, 439
псевдо-, 439
трихотомический, 374
ординальное число, 368
ординальное числоl, 377
ортогональная система факторизации, 231, 254–257, 262, 359
основная теорема теории Галуа, 340
основы, 3
основы, унивалентность, 3
отделение
Δ_0, **381**
полное, **383**
отделенность, **392**, 414, 438
отделимый класс, **381**
открытая
модальность, **265**
проблема, 14–15, 109, 265, 266, 319, 320, 436, 454, 456
открытое
отношение, 405
сечение, 389
открытый интервал, **413**
отмеченная точка, **75**, 251
примыкание, 199
отношение
антисимметричное, 324, 429
биполное, **385**, 385
вполне обоснованности, **369**
иррефлексивное, **375**
ко-транзитивности, **393**
монотонности, 405
нерефлексивное, 391, 393
округления, *см.* отношение с округлением
открытости, 405
простое, **210**
разделяемое семейство, **403**
рефлексивное, 235
рефлексивности, **211**
симметричности, **211**
транзитивности, **211**
эквивалентности, **211**
эквивалентности, эффективность, 360
экстенсиональное, **371**
эффективное
эквивалентность, **360**
отношение с округлением, **405**
отображающий конус, *см.* конус функции
отображение, *см.* функция
послойное, *см.* послойное преобразование
пролетов, **246**
отрицание, 47, 119–121, 125–127, 374–377
оценивание, *см.* применение, функции
пара
зависимая, **35**

неупорядоченная, 381
упорядоченная, **31**
парадокс, 28, 122, 173
параллельные пути, **65**
параметр индуктивного определения, **176**
первопорядковая сигнатура, **344**
перезаписи, правило, 454
переменная, **24**, 25, 29, 44, 47, 50, 164, 176, 442, 443, 448
в контексте, 442
захваченная, **27**
и подстановка, 442
область, **26**
связанная, **27**, **443**, 451
связанная в своей области, 443
тип, 173
типа, 176
фиктивная, **27**
пересечение
интервалов, **420**
подмножеств, **125**
петля, 63, 73, **75**
n-, **75**
n-мерная, *см.* петля, n-
зависимая n-, **197**
n-, 197, 231, 274, 314
постоянная, *см.* путь, постоянный
ΠW-предтопос, 362
плотность, 392
плотный порядок, *см.* плотность
подмножеств, совокупность, 382
подмножество, **122**
отношение на кумулятивной иерархии, **378**
поднятие
пути, **77**
эквивалентностей, 102
подобъектов классификатор, **362**
подсемейство, конечное, интервалов, **420**
подсинглетон, *см.* простое высказывание
подстановка, 442
подтерминальный объект, *см.* простое высказывание
подтип, 50, **122**
подуниверсум, рефлективный, **259**
позитивность, строгая, *см.* строгая позитивность
позитивный тип, 97
покрывающее пространство
универсальное, 276–278
покрытие
индуктивное, **421**
поточечное, **420**
универсальное, 276–278
покрытие пространства, 317
поле
аппроксимирующее, **388**
упорядоченное, *см.* упорядоченное поле
полиморфная функция, **30**
полиномиальный функтор, *см.* эндофунктор, полиномиальный
полная категория, 356
полное
метрическое пространство, **417**
отношение, 436
пространство, **77**
упорядоченное поле, по Дедекинду, **395**
упорядоченное поле, по Коши, **414**
полный функтор, **331**
положительные рациональные числа, **388**
полугруппа, 51, 102
структура, **102**
полукольцо, 62, 365
полусопряженная эквивалентность, **140–144**
типов, **140–144**
полюс, **197**
полярность, 97
помощник по компьютерному доказательству, *см.* доказательство помощник
пополнение
Дедекинда, 394, 395
Коши, 396–397
Резка, 322, **348–351**, 362, 384
метрического пространства, 438
стека, 352
точное, 362
пополнение Резка, *см.* пополнение, Резка
порождение
∞-группоида, 188
индуктивного типа, 157–160, 180
типа, индуктивного, 187–189
порядок
линейный, 391
нестрогий, 366, 411, 425
слабо линейный, 391, 393
строгий, 393, 411, 425
последовательность, 66, 175, 415, 416, 419
слоев, **286**, 285–290
точная, **288**, 290, 314
послойная эквивалентность, **150**
послойно, **77**
n-связное семейство функций, 252

послойное
  отображение, *см.* послойное преобразование
  преобразование, **149–150**, 252
постоянная
  Лифшица, **404**
  функция, **26**
поточечная функциональность, 61
поточечное
  покрытие, **420**
  равенство функций, 91
поточечные операции на функциях, 92
правая обратимая функция, **143**
правила теории типов, 441–454
правило, **21**, **447**
  введения, **31**, 450
  вывода, *см.* правило
  выделения, *см.* выделитель, типа
  вычисления, **31**, *см.* вычислительное правило
    для типов произведений, 32
  допустимое, **449**
  замещения, **449**
  исключения, 450
  ослабления, **449**
  отражения, 107
  перезаписи, 454
  против аксиом, 24, 190
  структурное, 448–449
  формирования, **31**, 450
правило вычислений
  для $\mathbb{S}^1$, 189, 192
  для W-типов, 164
  для высших индуктивных типов, 189–190
  для индуктивных типов, 175
  для натуральных чисел, 41, 43, 158
  для типа булевых значений, 158
  для типа копроизведения, 39
  для типов зависимых функций, 29
  для тождественных типов, 54
  для функциональных типов, **26**, 444, 454
  пропозициональное, **31**, 169, 170, 189–190, 203
    для тождественностей между парами, 86
    для тождественостей между функциями, 92
    для унивалентности, 94
правое сопряжение, **330**
правый обратный, **143**
пред-2-категория, **353**
пред-бикатегория, **353**
предгруппоид, фундаментальный, *см.* фундаментальный предгруппоид
предел
  аппроксимации Коши, 393, 398, 414, 417
  множеств, 209, 356
  типов, 106, 109, 203
предикат
  индуктивный, 177
  точечный, **180**
предикативная математика, *см.* математика, предикативная
предикатная логика, 49
предкатегорий
  изоморфизм, **333**
  произведение, 336
  эквивалентность, **331**
предкатегория, **322**
  †-, **340**
  обратная, **336**
  срезов, *см.* категория, срезов
  (P, H)-структур, **342**
  типов, **325**
  функторов, **326**
предположение, **24**
предпорядок, **324**
  на кардинальных числах, 366
представимый функтор, **338**, 338
представление
  ∞-группоида, 274
  ∞-группоида, 188
  группы, 220, 311
  позитивного типа конструкторами, 97
  пространства как CW-комплекса, 8
предстек, **354**
предтопос, *см.* ΠW-предтопос
преобразование
  естественное, *см.* естественное преобразование
  послойное, *см.* послойное преобразование
признаки, истинности высказывания, 46
прикрепляющее отображение, 201, 202, 310
прилагательное, 130
применение
  гипотезы или теоремы, 48
  зависимой функции, **29**
  зависимой функции к пути, **78**
  функции, **25**
  функции к пути, **75**
примитивная
  константа, 443

рекурсия, **41**
примыкание к дизъюнктивной отмеченной точке, 199
принадлежность, для кумулятивной иерархии, **378**
принуждение, поднятие универсума, 60
принцип
отвлекающего маневра, 340
структурной тождественности, **343**, 342–344
уникальности, *см.* уникальность, принцип
принцип Маркова, **438**
принцип базированной индукции
для тождественного типа, 55
принцип индукции, 34, 158, 450
для W-типов, 163
для $\mathbb{S}^1$, 190, 195
для $\mathbb{S}^2$, 197
для гомотопий, 183
для действительных чисел Коши, 399
для доступности, 368
для индуктивного типа, 175
для копроизведения, 38
для кумулятивной иерархии, 378
для модальности, 261
для надстройки, 197
для натуральных чисел, 43
для произведения, 34
для пустого типа, 39
для связных отображений, 249
для сюрреалистических чисел, 427
для типа булевых значений, 40
для типа векторов, 177
для типа зависимой пары, 35
для типа интервала, 194
для тождественного типа, 53–57, 66
для тора, 201
для усечения, 136, 208, 239
для целых чисел, 213
для эквивалентностей, 183
принцип рекурсии, 159
для $\mathbb{S}^1$, 189, 192
для $\mathbb{S}^2$, 196
для действительных чисел Коши, 402
для декартова произведения, 33
для индуктивного типа, 175
для копроизведения, 38
для модальности, 261
для надстройки, 197
для натуральных чисел, 41
для пустого типа, 38
для типа булевых значений, 39
для типа зависимой пары, 35
для типа интервала, 194
для усечения, 124, 208, 240
программирование, 4, 12, 27, 160, 263
проективная плоскость, 201
проекция
из типа декартова произведения, **32**
из типа зависимой пары, **35**
произведение
(пред)категорий, 336
типов, *см.* тип, произведения
пролет, 204, 209, **244**, 291
пропозициональная логика, 46
пропозициональное
изменение размера, **122**, 134, 135, 174, 355, 361, 363, 379, 389, 421, 424
равенство, 23, **52**
пропозициональное равенство, 160
пропозициональный
принцип уникальности, *см.* уникальность, принцип, пропозициональный
просто, 262
обитаем, **130**
разрешимое равенство, **237**
простое высказывание, **117–119**, 121–125, 128–131
простое отношение, **210**
простое число, 129
простоты теорема, 427
пространство
векторное, 341
метрическое, *см.* метрическое пространство
параметров, 24
расслоений, 88, 275, 281, 286, 295
топологическое, *см.* топологическое пространство
пространство Эйленберга-МакЛейна, **311**, 317
пространство петель, 73, **75**, 199, 216, 237, 275, 276, 286, 312, 315
n-кратных, *см.* пространство петель, кратных
кратных, 73, **75**, 197, 200, 216, 238, 274, 286
функториальность, 286
пространство расслоений Сигала, 352
пространство расслоений, 149
противоречие, **47**
псевдо-ординал, 439
пустой тип, *см.* тип, пустой
путь, **52**, 65–75
2-, 63, **65**, 72

2-мерный, *см.* путь, 2-
3-, 64, **65**, 73
3-мерный, *см.* path, 3-
n-, **73**, 109
n-мерный, *см.* путь, n-
базированная индукция, 55
зависимый, 79, 191
в зависимых функциональных типах, 93
в тождественных типах, 97
в функциональных типах, 93
индукция, 53–57
композиция, **67**
конечная точка, **65**
конкатенация, **67**
n-кратная, **215**
конструктор, **187**
n-, 197, 228
начальная точка, **65**
обратный, **66**
параллельный, **65**
поднятие, **77**
постоянный, **53**, 56
применение зависимой функции, **78**
применение функции, **75**
расслоение, 275
топологический, 6, 63
пятиугольник, МакЛейна, 64, 329

равенство
гетерогенное, 191
дефинициальное, **23**, 189
пропозициональное, 23, **52**
просто разрешимое, **237**
разрешимое, *см.* разрешимое равенство
рефлексивность, 69
симметричность, 69
симметрия, 66
субъективное, **23**
суждений, 442
типов, *см.* тип, тождественности
транзитивность, 67, 69
равное можно заменить равным, 53
равномерно непрерывная функция, **417**
разделяемое семейство отношений, **403**
размерность
конструкторов пути, 188
путей, 65
разрешенное подмножество, 41
разрешимое
равенство, 50, **121**, 219, 236–237, 319, 388
семейство типов, **121**
разрешимый тип, **121**
расслоение, 77, 88, 149, 275
Хопфа, **290**, *см.* Хопфа расслоение
путей, 275
расслоённая последовательность, **286**, 285–290
расстояние, 397, 412, 438
расширение, η-, *см.* η-расширение
расширенные действительные числа, **436**
расщепленная
сюръекция, *см.* функция, расщепленная сюръективная
расщепляемый эссенциально сюръективный функтор, **331**
рациональные числа, **388**
двоичные, 388
диадические, **426**
как действительные числа Коши, 398
положительные, **388**
регулярная категория, **356**
регулярный эпиморфизм, **356**
редукция
β-, *см.* β-редукция
слова в свободной группе, 219
редуцированное слово в свободной группе, **219**
рекурентность, 42, **159**, 184
рекурсивный вызов, 42
рекурсия
примитивная, **41**
структурная, 444
рекурсор, *см.* принцип рекурсии
ретракт
типа, **132**, 154
функции, **148–149**
ретракция, **132**, 147
рефлексивность
индуктивного покрытия, 421
отношения, **211**, 235
равенства, **53**
рефлективная подкатегория, 240
рефлективный подуниверсум, **259**
решетка, 390
свидетельство
истинности высказывания, 46
истинности предложения, 22
свободная
алгебраическая структура, 216
группа, *см.* группа, свободная
свободное
генерирование индуктивного типа, 158

полное метрическое пространство, 396
порождение индуктивного типа, 180, 188
произведение, **220**
амальгированное, **219**, 310
свободный моноид, *см.* моноид, свободный
свойство 2-из-3, 148
свойство 2-из-6, 156
свойство неподвижной точки, 184
связанная переменная, *см.* переменная, связанная
связная
функция, *см.* функция, n-связная
связность, 170
связный
категорно, тип, **267**
тип, **249**
связующая структура, **26**
сглаживания лемма, **221**
сегмент, начальный, *см.* начальный сегмент
семейство
базовых интервалов, **419**
конечных множеств, 28, 62
постоянных типов, **29**
типов, **28**
сетоид, 134, 228, 362
сечение, **132**
Дедекинда, 387, 389, **390**, 394, 395, 415, 437
семейства типов, **77**
сюрреалистических чисел, **424**
зависимое, **427**
σ-каркас
исходный, **390**
σ-фрейм исходный, 415
Σ-тип, *см.* тип, зависимой пары
сигнатура
алгебраической теории, 220
первого порядка, **344**
сильная
индукция, 369
совокупность, 382, **385**
сильно нормализуемый терм, 455
симметричность отношения, **211**
симметрия равенства, 66
симплициальное
множество, 352
ядро, 308
симплициальные множества, 6, 271
синтетическая математика, 65
система, тождественности, *см.* тождественная система
скелет CW-комплекса, 308, 310
скелетальная категория, 324
скелетон CW-комплекса, 200
скобки, 25–27
скобочный тип, *см.* усечение, пропозициональное
скрещенное произведение, **207**
слабая эквивалентность
предкатегорий, **332**, 346–351, 376
типов, **314**
слабо линейный порядок, 391, 393
слабый копредел, 221, 222
следствие, 21
сложение
действительных чисел Дедекинда, **391**
действительных чисел Коши, **411**
кардинальных чисел, **365**
натуральных чисел, **42**
ординальных чисел, 384
сюрреалистических чисел, 433
слой, **142**, 285
совокупность
подмножеств, 382
сильная, 382, **385**
совпадение, приближений Коши, 396, 402
согласованность, 13, 52, 70, 72, 174, 329
соединение
в решетке, 390
типов, **207**, 294
сопоставление, *см.* сопоставление с образцом
сопоставление с образцом, **44–45**, 60, 176, 457
сопряжение, 64
линейное отображение, 341
функторное, 106
сопряженная
эквивалентность, 82, 155, 353
(пред)категорий, **330**
сопряженный
функтор, **330**, 353, 362
функтор, теорема, 218
сохранение образов под обратным образом, 258
спираль, 276
список, *см.* тип списков
списочный тип, *см.* тип, списков
спица, *см.* концентратор и спица
срезов (пред)категория, *см.* категория, срезов
стабильность гомотопических групп сфер, 301
стек, 351, 352, **354**
стека пополнение, 352
степенное множество, 41, **123**, 174, 211, 361, 362, 370, 383, 421

стрелка, *см.* морфизм
строгая
  категория, 321, **340**, 353, 354
  позитивность, **174**, 226
  положительность, 424
строгий порядок, 393, 411, 425
структура
  (P, H)-, **342**
  гомоморфизма, **342**
  гомоморфизма Ω-, **344**
  нотация, **342**
  Ω-, **344**
  полугруппы, 102
  предкатегории (P, H)-, **342**
  стандартная нотация, **342**
(P, H)-структура, *см.* структура
структурная
  рекурсия, 444
  теория множеств, 362–363
структурные правила, 448–449
стягиваемая функция, **145–146**
стягиваемый тип, **131–133**
стягивание, 232
  типа, 232
  функции, 249
субъективное равенство, **23**
суждение, **21**, 23, 442
суждений равенство, 442
сумма
  зависимая, *см.* тип, зависимой пары
  непересекающаяся, *см.* тип, копроизведения
супремум
  конструктор W-типа, 163
  равномерно непрерывной функции, **418**
существительное, 130
сферы тип, *см.* тип, сферы
сюрреалистические числа, **424**, 423–435
сюръективная
  функция, *см.* функция, сюръективная
    расщепленная, *см.* функция, расщепленная сюръективная
сюръекция, *см.* функция, сюръективная
  расщепленная, *см.* функция, расщепленная сюръективная
теорема, 21
  Блейкерса-Месси, **317**
  Ван Кампена, 302–311, 353
  Диаконеску, 364
  Кантора, 367
  Конвея, 0, 428
  Конвея, простоты, 427
  Уайтхеда, 311
  Фейта-Томпсона, 8
  Фрейденталя о надстройке, 296–301
  Фубини для копределов, 295
  Хедберга, 133, 237
  Шрёдера-Бернштейна, 367
  нечетного порядка, 8
  о четырех красках, 10
теория
  алгебраическая, 220
  эссенциально алгебраическая, 220
теория высших категорий, 63–64
теория множеств
  Цермело-Френкеля, 8, 21, 220, 355, 383
  алгебраическая, 355, 383
теория типов, 4, 21
  интенсиональная, 59, 107
  неформальная, **9–10**, 24
  формальная, 9–10, 441–457
  экстенсиональная, 59, 107
терм, **22**
  замкнутый, 455
  конвртируемость, 444
  нормализуемый, **455**
  сильно нормализуемый, **455**
терминальный тип, *см.* тип, единицы
тип
  ∞-усеченный, **314**
  2-сферы, 189, 196–197
  Π-, *см.* тип, зависимой функции
  W-, *см.* W-тип
  амальгама, *см.* амальгама
  булевых значений, **39–41**
  векторов, 177
  взаимной индукции, 178
  высший индуктивный, 7, 8, 187–229
  гиперполный, 314
  гомотопически-индуктивный, 169
  декартова произведения, *см.* тип, произведения
  единицы, **31–34**, 39, 90–91, 106, 131, 161, 163, 452
  зависимой пары, **34–38**, 88–90, 105
  зависимой суммы, *см.* тип, зависимой пары
  зависимой функции, **29–30**, 91–93, 450
  зависимый, *см.* семейство, типов
  зависимый, пары, 450
  индуктивно-индуктивный, 178
  индуктивно-рекурсивный, 179

индуктивный, 157–160, 172–179
обощения, 176
интервала, **193–195**
кардинальных чисел, **364**
копредела, 106, 203
копроизведения, **38–39**, 39, 40, 62, 97–100, 451
коуравнителя, **221**
малый, 6, **28**
модальный, **261**
n-связный, **248**
n-сферы, 197, **199**, 200, 301
n-тип, *см.* n-тип
n-усеченный, *см.* n-тип
надстройка, *см.* надстройка
натуральных чисел, *см.* натуральные числа, 452
негативный, 97
обитаемый, *см.* обитаемый тип
односвязный, **249**
одноэлементного множества, 56
окружности, 187, 189–193, 195–196, 198, 454
отождествлений, 52
подмножества, **122**
позитивный, 97
предела, 106, 203
произведения, **30–34**, 62, 85–88, 104, 450
пустой, **38–39**, 106, 163, 451
равенства, *см.* тип, тождественности
разрешимый, **121**
связный, **249**
семейства, 40
индуктивного, 177
семейство, 76–80
постоянных, **29**
разрешимое, **121**
Σ-, *см.* тип, зависимой пары
синглетон, **131**
сквоша, *см.* усечение, пропозициональное
скобочный, *см.* усечение, пропозициональное
списков, 158, 160, 163, 177, 184, **216**, 417
стягиваемый, **131–133**
тождественности, **52–59**, 66–75, 95–97, 106, 452
как индуктивный, 179
точечный, **75**, 197
универсума, **28–29**, 93–95, 115, 444, 449
в стиле Расселла, 60
в стиле Тарского, 60
кумулятивный, **28**
унивалентного, **94**
уровень, *см.* уровень универсума
f-локальный, **266**
функции, 28, 91–93, 450
функциональный, **25**
частного, *см.* множество-частное
членов, **380**
тип сквоша, *см.* усечение, пропозициональное
типовая двусмысленность, 389, 444
типовая неоднозначность, **28**, 373, 453
тождественная
модальность, 262
система, **182**
при точке, **180–182**
функция, **30**, 82, 94
тождественность, 7
типов, *см.* тип, тождественности
тождественный морфизм в (пред)категории, **322**
тождество
зигзагообразное, **330**
треугольное, **330**
топологический путь, 6, 63
топологическое пространство, 4, 5, 63, 271
топология
Ловера-Тирней, 262
бесточечная, 420
формальная, 422
топос, 12, 134, 262, 352, 355, 362–364, 383
логический, 364
тор, 201, 202, 229, 310
принцип индукции, 201
точечное
отображение, **285**
ядро, **287**
точечный
предикат, **180**
тип, *см.* тип, точечный
точка
конструктор, **187**
типа, **22**
точная последовательность, **288**, 288
точный функтор, 331
тощая категория, **324**, 340
традиционная логическая нотация, **125**
транзитивность
индуктивного покрытия, 421
$<$ для сюрреалистических чисел, 433
$\leqslant$ для сюрреалистических чисел, 433
отношения, **211**
равенства, 67

$<$ в поле, 393
$<$ для действительных чисел, 391
of $\leqslant$ для действительных чисел, 391
транспортирование, **76–80**, 88, 94
в единичном типе, 91
в зависимых парных типах, 90
в зависимых функциональных типах, 92
в типах копроизведений, 100
в типах произведений, 87
в тождественных типах, 96
в функциональных типах, 92
треугольника неравенство для $\mathbb{R}_\mathsf{c}$, 409
треугольное тождество, **330**
трихотомия ординалов, 374
Уайтхеда
принцип, 311–314
теорема, 311
умножение
в группе, **215**
в моноиде, **215**
действительных чисел Дедекинда, **391**
действительных чисел Коши, **413**
кардинальных чисел, **365**
натуральных чисел, 62
ординальных чисел, 384
унивалентный универсум, **94**
универсальное
покрытие, 276–278
свойство, 104–107
амальгамы, 205, 209, 247
декартова произведения, 105
для W-типа, 167
для $\mathbb{S}^1$, 193
зависимых типов пар, 222
копроизведения, 110
метрического пополнения, 438
множества-коуравнителя, 357
модальности, 261
$\mathbb{S}^n$, 200
надстроек, 199
натуральных чисел, 166
пополнения Резка, 348
свободной группы, 217
типа зависимой пары, 105
типа тождественности, 106
усечений, 240
усечения, 209
универсальное свойство, 162
универсум, *см.* тип, универсума
уникальная
система факторизации, 254–257, 262
уникальности
принцип, пропозициональной
для гомотопических W-типов, 170
принцип, пропозициональный
для функций на окружности, 193
уникальность
принцип, **31**, 60, 450
для типов зависимых функций, 29
для типов произведений, 60
для тождественостей между функциями, 92
для функциональных типов, 26, 60
принцип, пропозициональная
для зависимых парных типов, 90
для типов произведений, 86
для тождественностей между парами, 87
для функций на амальгаме, 204
принцип, пропозициональной
для унивалентности, 94
для функций на W-типах, 165
принцип, пропозициональный, **31**
для модальности, 261
для типов произведений, 33
уникальность
доказательств тождественности (UIP), 59
унитарный морфизм, 341
упорядоченное поле, 387, **392**, 414
архимедово, 392, **393**, 411, 437
допустимость, **394**, 415
уравнение, определяющее, 443
уровень, *см.* уровень универсум или n-тип
уровень универсума, **28**, 322, 373, 376, 389
усечение
множества, 208–210
n-усечение, 210, **238–244**
пропозициональное, **124–125**, 207–208
устойчивость
свойство спуска копределов, 222
уточнение, в теории типов, 453
f-локальный тип, **266**
фактор-множество, 360–362
факторизация
система, ортогональная, *см.* ортогональная система факторизации
сохранение под обратным образом, 257
фибрантная замена, 362
фиктивная переменная, *см.* переменная, связанная

формализация математики, *см.* математика, формализованная
формальная
  теория типов, 441–457
  топология, 421
формирующее правило, 450
Фрейденталя теорема о надстройке, 296–301
фундаментальная
  группа, 64, **216**, 272, 302, 306–307, 309–311
    окружности, 275–282
фундаментальный
  ∞-группоид, 64
  группоид, 302, **351**, 353
  предгруппоид, **325**, 351, 353
функтор, **326**
  вполне точный, **331**
  ковариантный, 173
  контравариантный, 173
  полиномиальный, *см.* эндофунктор, полиномиальный
  полный, **331**
  представимый, **338**, 338
  пространства петель, 286
  расщепляемый эссенциально сюръективный, **331**
  слабая эквивалентность, *см.* слабая эквивалентность
  сопряженный, 330
  точный, **331**
  эссенциально сюръективный, **332**
«функториальность» функций в теории типов, 75, 81, 91, 113, 116, 131, 236
функторная эквивалентность, 331
функторов категория, **327**
функциональная композиция, **61**
функциональная экстенсиональность, 60, 85, **91**, 108, 132, 154, 453
  доказательство от типа интервала, 194
  доказательство от унивалентности, 152
  независимая, **156**
  слабая, **153**
функциональное отношение, 25
функциональный тип, *см.* тип, функциональный
функция, **25–28**, 75–76
  $\bigcirc$-связная, **262**
  $\bigcirc$-усеченная, **262**
  ∞-связная, **314**
  Аккермана, **62**
  Лифшица, **404**
  би-обратимая, 137, **144**
  биективная, *см.* биекция
  вложения, **147**, 232, 254
  зависимая, **29–30**, 78–80
    применение, **29**
    применение к пути, **78**
  идемпотент, **212**
  имитации, *см.* имитация
  инъективная, **147**, 254, 356, 366
  каррированная, **27**
  квазиобратная, *см.* квазиобратная
  левая обратимая, **143**
  линейная, 341
  локально равномерно непрерывная, **438**
  λ-абстракция, *см.* λ-абстракция
  n-связная, **248**, 249
  n-усеченная, **254**
  непрерывная, 404, 409
    в классической теории гомотопий, 4
  нерасширяющаяся, 410
  нулевая, **286**
  полиморфная, **30**
  послойная, *см.* послойное преобразование
  постоянная, **26**
  правая обратимая, **143**
  применение, **25**
  применение к пути, **75**
  проекции, *см.* проекция
  равномерно непрерывная, **417**
  расщепленная сюръективная, **147**
  ретракции, **132**, 147
  с кообластью, 25
  с областью, 25
  сечения, **132**
  слоя, *см.* слой
  стягиваемая, **145–146**
  сюръективная, **147**, 249, 356, 366
  тождественная, **30**, 82, 94
  точечная, *см.* точечное отображение
функция возведения в квадрат, 413
функция предшествования, 159, 164
  изоморфизм на $\mathbb{Z}$, 277
  ординала, **373**
  усеченная, 190
функция следования, 101, 159, 163, 164
  гладкая, **386**
  изоморфизм на $\mathbb{Z}$, 275, 277
  ординала, 386
H-пространство, **292**
хаотическая предкатегория, **335**
Хопфа

расслоение, 293–295
младшее, **320**
целые числа, **212**, 276
принцип индукции, 213
целые числа, 273
цель
конструктора пути, 201
построенного пути, 226
пути конструктора, **187**
центр
категории, 138
стягивания, **131**
циклическая группа, 273
частично упорядоченное множество, **324**
частичный порядок, **324**, 391
частное множеств, *см.* set-quotient
четыре краски, теорема, 10
числа целые, **212**
число
кардинальное, *см.* кардинальное число
натуральное, *см.* натуральные числа
ординальное, *см.* ординал
рациональное, *см.* рациональные числа
сюрреалистическое, *см.* сюрреалистические числа
число Лебега, 423, **438**
число витков пути, 279
чисто, 262, 263
членство, 22
эквивалентность, 82–84, 93–95, 146
(пред)категорий, **331**
слабая, *см.* weak equivalence
индукция, 183
как би-обратимая функция, **144**
как стягиваемая функция, **145–146**
логическая, **51**
отношение, *см.* отношение, эквивалентности
полусопряженная, **140–144**
послойная, **150**
свойства, 146, 148
слабая, 314
экспоненциальный идеал, **259**
экстенсиональная теория типов, 59, 107
экстенсиональное отношение, **371**
экстенсиональность, функций, *см.* функциональная экстенсиональность
элемент, **22**
элементарная теория категории множеств, 8, 355, 364, 383
эндофунктор
алгебра, *см.* алгебра для эндофунктора
полиномиальный, **167**, 175
эпи, *см.* эпиморфизм
эпиморфизм, **357**
регулярный, **356**
эссенциально сюръективный функтор, **332**
η-конверсия, *см.* η-расширение
η-расширение, 26, 31
эффективная процедура, 48
эффективное отношение, **360**
эффективность
отношения эквивалентности, **360**, 360–362
ядерная пара, 308, **356**, 360, 402
ядро, **287**
симплициальное, 308
язык программирования Haskell, 263