

Теория типов

О курсе

Краткое содержание вступительного занятия

1. История вопроса: что вообще изучает теория типов, типы в математике, типы в лямбда-исчислении. Краткое повторение материала, знание которого ожидается от участников.
2. Содержание текущего курса: изоморфизм Карри-Ховарда и его применение в программировании и математике
3. Особенности преподавания.

Типы в теории множеств

- ▶ Парадокс Рассела: $\{ x \mid x \notin x \}$. Законна ли эта запись? $a \in b$ — ожидаем, что слева элемент, а справа множество.
- ▶ Давайте запретим. Например, введём тип множества: $\emptyset^0, \{x^n, y^n, z^n\}^{n+1}$ и т.п.
- ▶ В. Russel, A. Whitehead, Ramified type theory, 1908 (разветвлённая теория типов). Все объекты получают тип и порядок. Формулы $m + 1$ порядка работают с объектами, задаваемыми формулами m порядка:

$${}^{m+1}F^n : {}^mP^n \rightarrow {}^mQ^n$$

- ▶ По силе примерно Р.Т.Т. соответствует аксиоматике Цермело-Френкеля, но неудобна. В ZF можно приписать множествам схожую типу характеристику («ранг»), сложностью выражений можно управлять, например, средствами аксиомы конструктивности.

Лямбда-исчисление: история возникновения

- ▶ Готлоб Фреге, 1893 год, «карринг». Двуместную функцию $a + b$ можно представить как композицию двух одноместных функций:

$$f(a) = \lambda x. a + x \quad a + b = f(a)(b)$$

- ▶ Моисей Шейнфинкель, 1924, комбинаторы:

$$Kab = a \quad Sabc = ac(bc)$$

- ▶ Алонзо Чёрч, 1932, лямбда-исчисление:

$$(\lambda x. M) = M[x := N]$$

- ▶ Алонзо Чёрч, 1932, 1934: В λ -исчислении арифметика выражается естественно. Попробуем λ -исчисление расширить до логики?
- ▶ С.Клини и Б.Россер, 1935, противоречие (модификация парадокса Ришара).

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- ▶ В лямбда-исчисление введём логический символ \rightarrow . Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- ▶ Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
 1. $\vdash A \rightarrow A$
 2. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$
 3. Если $\vdash A$ и $\vdash A \rightarrow B$, то $\vdash B$.
- ▶ Менее очевидное: $\vdash A \rightarrow B$, если $A =_{\beta} B$. Мотивация: если $0 + 1 =_{\beta} 1$, то $X(0 + 1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу).

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- ▶ В лямбда-исчисление введём логический символ \rightarrow . Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- ▶ Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
 1. $\vdash A \rightarrow A$
 2. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$
 3. Если $\vdash A$ и $\vdash A \rightarrow B$, то $\vdash B$.
- ▶ Менее очевидное: $\vdash A \rightarrow B$, если $A =_{\beta} B$. Мотивация: если $0 + 1 =_{\beta} 1$, то $X(0 + 1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу). $1 = 0.5 \cdot 2$ влечёт $\sin 1 = \sin(0.5 \cdot 2)$, а как иначе?

Лямбда-исчисление как логический язык

«Анахроническое» изложение, пересказ современным языком: следуем J. Barkley Rosser: Highlights of the history of the Lambda-Calculus

- ▶ В лямбда-исчисление введём логический символ \rightarrow . Формулы исчисления будем считать логическими высказываниями. Добавим логические аксиомы. Ожидаем такое: $\vdash 0 + 1 = 1$
- ▶ Получение противоречия: определим минимальные требования к исчислению. Очевидно, хотя бы следующее мы должны уметь доказывать:
 1. $\vdash A \rightarrow A$
 2. $\vdash (A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$
 3. Если $\vdash A$ и $\vdash A \rightarrow B$, то $\vdash B$.
- ▶ Менее очевидное: $\vdash A \rightarrow B$, если $A =_{\beta} B$. Мотивация: если $0 + 1 =_{\beta} 1$, то $X(0 + 1)$ всегда можно заменить на $X(1)$ (равенство по Лейбницу). $1 = 0.5 \cdot 2$ влечёт $\sin 1 = \sin(0.5 \cdot 2)$, а как иначе?
- ▶ Заметим: $0 + 1 \twoheadrightarrow_{\beta} 1$, поэтому $\vdash (1 = 1) \rightarrow (0 + 1 = 1)$.

Парадокс Карри

$$\Phi_\alpha := Y (\lambda x. x \rightarrow \alpha)$$

Редуцируя Φ_α , получаем:

$$\Phi_\alpha \rightarrow_\beta (\lambda x. x \rightarrow \alpha) (Y (\lambda x. x \rightarrow \alpha)) \rightarrow_\beta \Phi_\alpha \rightarrow \alpha$$

И доказательство:

- | | |
|--|--|
| 1) $\Phi_\alpha \rightarrow (\Phi_\alpha \rightarrow \alpha)$ | $(A \rightarrow A)$ и $\Phi_\alpha =_\beta \Phi_\alpha \rightarrow \alpha$ |
| 2) $(\Phi_\alpha \rightarrow \Phi_\alpha \rightarrow \alpha) \rightarrow (\Phi_\alpha \rightarrow \alpha)$ | Так как $(A \rightarrow (A \rightarrow B)) \rightarrow (A \rightarrow B)$ |
| 3) $\Phi_\alpha \rightarrow \alpha$ | MP 1, 2 |
| 4) $(\Phi_\alpha \rightarrow \alpha) \rightarrow \Phi_\alpha$ | $(A \rightarrow A)$ и $\Phi_\alpha \rightarrow \alpha =_\beta \Phi_\alpha$ |
| 5) Φ_α | MP 3, 4 |
| 6) α | MP 5, 3 |

Парадокс Карри: «Если данное высказывание верно, то луна сделана из зелёного сыра». То есть,

$$\Phi_\alpha \leftrightarrow (\Phi_\alpha \rightarrow \alpha)$$

Лямбда-исчисление как вычислительная модель

- ▶ Из исчисления А. Чёрч выделил некоторую часть и доказал её непротиворечивость: Church, A. (1935). "A Proof of Freedom from Contradiction." Proceedings of the National Academy of Sciences of the United States of America, 21(5):275–281.
- ▶ Но затем предложил смотреть на исчисление как на вычислительную модель: Church, A. (1936). "An Unsolvable Problem of Elementary Number Theory." American Journal of Mathematics, 58(2):345–363, 1936.
- ▶ Начала современного понимания теории типов были заложены в этой работе: Church, A. (1940). A formulation of the simple theory of types, Journal of Symbolic Logic 5, pp. 56–68.

Примеры вычислений

$$f^{(n)}(x) = \begin{cases} x, & n = 0 \\ f(f^{(n-1)}(x)), & n > 0 \end{cases}$$

Определение

Чёрчевский нумерал $\bar{n} := \lambda f. \lambda x. f^{(n)}(x)$. Например, $\bar{3} = \lambda f. \lambda x. f(f(f(x)))$

Пример

Инкремент: $Inc := \lambda n. \lambda f. \lambda x. n \ f \ (f \ x)$

$IsZero := \lambda n. n \ (\lambda x. F) \ T$

$Pair \ a \ b := \lambda s. s \ a \ b$, $Fst := \lambda p. p \ T$, $Snd := \lambda p. p \ F$

$Dec := \lambda n. Snd \ (n \ (\lambda p. Pair \ (Snd \ p) \ (Inc \ (Snd \ p)))) \ (Pair \ \bar{0} \ \bar{0})$

$Y := \lambda f. (\lambda x. f \ (x \ x)) \ (\lambda x. f \ (x \ x))$

Пример

$Fact \ n := Y \ (\lambda f. \lambda x. (IsZero \ x) \ \bar{1} \ (Mul \ x \ (f \ (Dec \ x)))) \ n$

Порядок вычислений

$Body := \lambda f. \lambda x. (IsZero\ x) \ \bar{1} \ (Mul\ x \ (f \ (Dec\ x)))$

$Fact\ n := Y\ Body\ n$

Пример

Рассмотрим $Fact\ \bar{3} = Y\ Body\ \bar{3} \rightarrow_{\beta} Body\ (Y\ Body)\ \bar{3}$

Вычислять двумя способами:

1. $Body\ (Y\ Body)\ \bar{3} \rightarrow_{\beta} Body\ (Body\ (Y\ Body))\ \bar{3}$
2. $Body\ (Y\ Body)\ \bar{3} = [\lambda f. \lambda x. (IsZero\ x) \ \bar{1} \ (Mul\ x \ (f \ (Dec\ x)))]\ (Y\ Body)\ \bar{3} \rightarrow_{\beta}$
 $(IsZero\ \bar{3}) \ \bar{1} \ (Mul\ \bar{3} \ (Y\ Body\ (Dec\ \bar{3})))$

Ну и дальше (при втором способе):

$(Mul\ \bar{3} \ (Y\ Body\ (Dec\ \bar{3}))) \rightarrow_{\beta}$

$Mul\ \bar{3} \ (Y\ Body\ \bar{2}) \rightarrow_{\beta} Mul\ \bar{3} \ (Body\ (Y\ Body\ \bar{2})) \rightarrow_{\beta} \dots$

Теорема Чёрча-Россера гарантирует, что результат будет одинаков, если будет.

Нормальный и аппликативный порядок вычислений

Пример

$\mathcal{K} := \lambda x. \lambda y. x$, $\mathcal{I} := \lambda x. x$, $\Omega := (\lambda x. x \ x) (\lambda x. x \ x)$

Выражение $\mathcal{K}\mathcal{I}\Omega$ можно редуцировать двумя способами:

1. $\mathcal{K}\mathcal{I}\Omega =_{\alpha} ((\lambda a. \lambda b. a) \mathcal{I})\Omega \rightarrow_{\beta} (\lambda b. \mathcal{I})\Omega \rightarrow_{\beta} \mathcal{I}$
2. $\mathcal{K}\mathcal{I}\Omega =_{\alpha} ((\lambda a. \lambda b. a) \mathcal{I})((\lambda x. x \ x)(\lambda x. x \ x)) \twoheadrightarrow_{\beta} ((\lambda a. \lambda b. a) \mathcal{I})((\lambda x. x \ x)(\lambda x. x \ x)) \rightarrow_{\beta} \mathcal{K}\mathcal{I}\Omega$

Определение (нормальный порядок редукции)

Редукция самого левого β -редекса.

Определение (аппликативный порядок редукции)

Редукция самого левого β -редекса из самых вложенных.

Теорема (Приводится без доказательства)

Если нормальная форма существует, она может быть достигнута нормальным порядком редукции.

Нормальный порядок — медленный

Пример

Рассмотрим λ -выражение $(\lambda x.x\ x\ x\ x)(II)$. Попробуем редуцировать его нормальным порядком:

$$(\lambda x.x\ x\ x\ x)(II) \rightarrow_{\beta} (II)(II)(II)(II) \rightarrow_{\beta} I(II)(II)(II) \rightarrow_{\beta} (II)(II)(II) \rightarrow_{\beta} \dots$$

Как мы увидим, в данной ситуации аппликативный порядок редукции оказывается значительно эффективней:

$$(\lambda x.x\ x\ x\ x)(II) \rightarrow_{\beta} (\lambda x.x\ x\ x\ x)I \rightarrow_{\beta} IIII \rightarrow_{\beta} III \rightarrow_{\beta} II \rightarrow_{\beta} I$$

Как программировать? Любое значение – замыкание

```
let x = sqrt 256  let x = fun () -> sqrt 256
```

Плюс мемоизация:

```
let x = fun () -> sqrt 256;;  
let y = x;;  
y () + x () (* вычисляется два раза *)
```

Давайте запоминать результаты!

```
type int-value = Compute of unit -> int | Result of int;;  
let compute v = match !v with  
    Compute f -> let res = f () in v := Result res; res  
    | Result r -> r;;
```

```
let x = ref (Compute (fun () -> sqrt 256));;  
let y = x;;  
compute y + compute x
```

Ленивые и энергичные вычисления

Энергичные вычисления: аппликативный порядок. Ленивые вычисления: нормальный порядок + мемоизация.

If всегда ленив

```
let fact n = if n > 1 then n * fact (n-1) else 1
```

Ленивое общение с внешним миром бессмысленно.

Изоморфизм Карри-Ховарда

Просто типизированное λ -исчисление	Импликативный фрагмент ИИВ
$\Gamma, x : \theta \vdash x : \theta$	$\Gamma, \varphi \vdash \varphi$
$\frac{\Gamma, x : \varphi \vdash P : \psi}{\Gamma \vdash (\lambda x. P) : \varphi \rightarrow \psi}$	$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi}$
$\frac{\Gamma \vdash P : \varphi \rightarrow \psi \quad \Gamma \vdash Q : \varphi}{\Gamma \vdash PQ : \psi}$	$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$

Просто типизированное λ -исчисление	Импликативный фрагмент ИИВ
Тип	Высказывание
Терм	Доказательство высказывания
Проверка типа	Проверка доказательства на корректность
Обитаемый тип	Доказуемое высказывание

(\Rightarrow) : изучение языков программирования

- ▶ Просто типизированное исчисление соответствует исчислению высказываний. Малая выразительная сила просто-типизированного лямбда исчисления (полиномы).

- ▶ Метод: усложняем исчисление — смотрим получающийся язык.

- ▶ Логика первого порядка: зависимые типы. Какой тип у `sprintf`?

`sprintf "%d" : int -> string`

`sprintf "%d + %d" : int*int -> string`

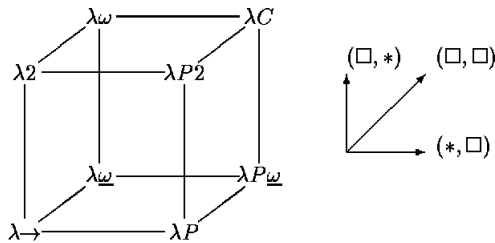
Например, Идрис позволяет тип выписать.

- ▶ Логика второго порядка: генерики.

Программа	Доказывает
<code>let id x = x</code>	$\forall \tau. \tau \rightarrow \tau$

- ▶ Классические функциональные языки — типовая система Хиндли-Милнера (разрешимый вариант системы F, соответствующей логике второго порядка).
- ▶ Алгоритмы вывода типов, анализ и верификация программ — используют матлог.

Лямбда-куб Барендрегта



Типовые системы и языки программирования:

Классические и функциональные языки:

$\lambda \rightarrow$ Классический Паскаль

λ_{ω} Система F

λ_{ω} Haskell, Ocaml

Языки с зависимыми типами данных (обычно около λC):

Idris, Coq, Agda, Arend, C++ :).

(\Leftarrow) : изучение логики и доказательств через написание программ

- ▶ Пер Мартин-Лёф, Intuitionistic Type Theory: версии 1972 и 1979.
- ▶ Множество расширений и вариантов.
- ▶ Такие инструменты как Coq, Agda, Lean используют варианты этой теории.
- ▶ Мы будем рассматривать некоторую родственную теорию, гомотопическую теорию типов.

Гомотопическая теория типов

Владимир Александрович Воеводский, 1966-2017.

... Математика находится на пороге кризиса, а точнее двух кризисов. Первый связан с отрывом математики «чистой» от математики прикладной. Понятно, что рано или поздно встанет вопрос о том, а почему общество должно платить деньги людям, которые занимаются вещами, не имеющими никаких практических приложений. Второй, менее очевидный, связан с усложнением чистой математики, которое ведет к тому, что, опять же рано или поздно, статьи станут слишком сложными для детальной проверки и начнется процесс накопления незамеченных ошибок ...

Гомотопическая теория типов

- ▶ Центральный вопрос — что такое равенство.
- ▶ Классический матлог: это предикат, удовлетворяющий свойствам.
- ▶ Однако, свойства обычно слишком общие (класс эквивалентности?).
Интуитивно хочется большего, равенство не всегда просто эквивалентность.

- ▶ Изоморфизм Карри-Ховарда-Воеводского:

Логика	λ -исчисление	Топология
Утверждение	Тип	Пространство
Доказательство	Значение	Точка в пространстве
Предикат ($=$)	Зависимый тип ($=$)	Путь между точками

- ▶ Реализация: кубическая теория типов, Аренд.

Чем естественно такое определение равенства?

Целые числа — класс эквивалентности пар $\langle a, b \rangle$ при $a, b \in \mathbb{N}_0$, понимаемых как $a - b$.

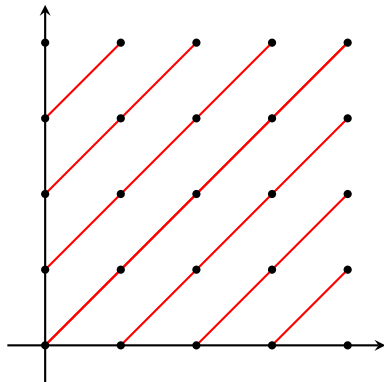
Точнее: $\langle a, b \rangle \approx \langle c, d \rangle := a + d = b + c$

$\mathbb{Z} := \{\langle a, b \rangle \mid a, b \in \mathbb{N}_0\} / \approx$

Рассмотрим топологическое пространство с носителем

$Z = \{\langle a, b \rangle \mid a, b \in \mathbb{N}_0\}$,

открытыми множествами назовём все диагональные прямые $x = y + c$ и $y = x + c$.



$\langle a, b \rangle = \langle c, d \rangle$ эквивалентно существованию пути $\pi : [0, 1] \rightarrow Z$, что $\pi(0) = \langle a, b \rangle$ и $\pi(1) = \langle c, d \rangle$.

Построение курса

1. Аналогично с матлогом, будет разделение на теорию и практику.
2. Теория: знание определений, идей, теорем.
3. Практика: лабы на доказательства теорем с использованием языка Аренд, возможны дополнительные околокомпиляторные лабы.
4. Для закрытия предмета надо набрать баллы практическими заданиями и сдать зачёт/экзамен.