

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 математике

- ▶ Раздел не претендует на полноту, предназначен для построения контекста.
- ▶ Парадокс Рассела: $\{ x \mid x \notin x \}$. Напрашивается вопрос: разве эта запись законна? $a \in b$ — ожидаем, что слева элемент, а справа множество.
- ▶ A. N. Whitehead, B. Russel, Principia Mathematica (Cambridge University Press, 1910-1913) — один из первых примеров формализации математики в рамках программы Гильберта, включает запрет таких записей с помощью введения типов. Всем пропозициональным функциям («предикатам») теории присваиваем тип — и определяем правила построения формул с учётом типов.
- ▶ По силе P.M. примерно соответствует аксиоматике Цермело, но менее удобна из-за явных типов. Аксиоматика ZFC явных типов не имеет, но позволяет приписать множествам схожую характеристику («ранг»). В ZFC также невозможны множества вида $\{ x \mid x \notin x \}$, хотя запрет не формулируется синтаксически.
- ▶ Не пытаюсь охватить всю тему про типы, сконцентрируемся только на типах в лямбда-исчислении.

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: формальная арифметика сложна (8 аксиом + схема аксиом индукции, исчисление предикатов...). В λ -исчислении арифметика выражается естественно. Попробуем λ -исчисление расширить до логики?
- ▶ С.Клини и Б.Россер, 1935, противоречие (модификация парадокса Ришара).

Краткое изложение формализма, напоминание

$$\Lambda ::= (\lambda x. \Lambda) | (\Lambda \ \Lambda) | x$$

Мета-язык:

- ▶ Мета-переменные:
 - ▶ $A \dots Z$ — мета-переменные для термов.
 - ▶ x, y, z — мета-переменные для переменных.
- ▶ Правила расстановки скобок аналогичны правилам для кванторов:
 - ▶ Лямбда-выражение ест всё до конца строки
 - ▶ Аппликация левоассоциативна

Пример

- ▶ $a \ b \ c \ (\lambda d. e \ f \ \lambda g. h) \ i \equiv \left(\left((a \ b) \ c \right) \left(\lambda d. ((e \ f) \ (\lambda g. h)) \right) \right) i$
- ▶ $0 := \lambda f. \lambda x. x; \quad (+1) := \lambda n. \lambda f. \lambda x. n \ f \ (f \ x); \quad (+2) := \lambda x. (+1) \ ((+1) \ x)$

Альфа-эквивалентность

1. $A \equiv x, B \equiv y, x \equiv y$;
2. $A \equiv P_a Q_a, B \equiv P_b Q_b$ и $P_a =_\alpha P_b, Q_a =_\alpha Q_b$;
3. $A \equiv (\lambda x.P), B \equiv (\lambda y.Q), P[x := t] =_\alpha Q[y := t]$, где t не входит в A и B .

Лемма

$$\lambda a. \lambda b. a \ b =_\alpha \lambda b. \lambda a. b \ a$$

Доказательство.

t	$=_\alpha$	t	Правило 1
s	$=_\alpha$	s	Правило 1
$t \ s$	$=_\alpha$	$t \ s$	Правило 2
$\lambda b. (t \ b)$	$=_\alpha$	$\lambda a. (t \ a)$	Правило 3
$\lambda a. \lambda b. (a \ b)$	$=_\alpha$	$\lambda b. \lambda a. (b \ a)$	Правило 3



Бета-редукция

Определение

Терм вида $(\lambda x.P) Q$ — бета-редекс.

Определение

$A \rightarrow_\beta B$, если:

1. $A \equiv (\lambda x.P) Q$, $B \equiv P [x := Q]$, при условии свободы для подстановки;
2. $A \equiv (P Q)$, $B \equiv (P' Q')$, при этом $P \rightarrow_\beta P'$ и $Q = Q'$, либо $P = P'$ и $Q \rightarrow_\beta Q'$;
3. $A \equiv (\lambda x.P)$, $B \equiv (\lambda x.P')$, и $P \rightarrow_\beta P'$.

Пример

$(\lambda x.x x) (\lambda n.n) \rightarrow_\beta (\lambda n.n) (\lambda n.n) \rightarrow_\beta \lambda n.n$

Пример

$(\lambda x.x x) (\lambda x.x x) \rightarrow_\beta (\lambda x.x x) (\lambda x.x x)$

Бета-эквивалентность, неподвижная точка

Определение

$(=_{\beta})$ — транзитивное, рефлексивное и симметричное замыкание (\rightarrow_{β}) .

Теорема (следствие из теоремы Чёрча-Россера)

Если $A =_{\beta} B$, то найдётся C , что $A \rightarrow_{\beta} C$ и $B \rightarrow_{\beta} C$.

То есть, можно интуитивно воспринимать $A =_{\beta} B$ как вычислительное равенство, равенство результатов.

Теорема

Для любого терма N найдётся такой терм R , что $R =_{\beta} N R$.

Доказательство.

Пусть $Y = \lambda f. (\lambda x. f (x x)) (\lambda x. f (x x))$. Тогда $R := Y N$:

$$Y N =_{\beta} (\lambda x. N (\textcolor{red}{x} \textcolor{blue}{x})) (\lambda x. N (x x)) =_{\beta} N ((\lambda x. \textcolor{red}{N} (\textcolor{red}{x} \textcolor{red}{x})) (\lambda x. \textcolor{blue}{N} (\textcolor{blue}{x} \textcolor{blue}{x})))$$



Подробнее о лямбда-исчислении как логике

«Анахроническое» изложение, пересказ современным языком.

- ▶ В лямбда-исчисление введём логический символ \rightarrow . Все 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 $X(1)$, следует из определения равенства по Лейбницу.
- ▶ Заметим: $0 + 1 \twoheadrightarrow_{\beta} 1$, поэтому $\vdash (1 = 1) \rightarrow (0 + 1 = 1)$. Из чего уже видны возможные контуры исчисления.

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 |

Собственно 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.
- ▶ Применение типов в лямбда-исчислении позволяет достичь схожих результатов с Principia Mathematica: синтаксическое ограничение допустимых формул, исключение части формул из исчисления. Мы начнём с краткого повторения просто-типизированного лямбда-исчисления и покажем невыразимость в нём Y -комбинатора.

Импликационный фрагмент ИИВ

Рассмотрим подмножество ИИВ, со следующей грамматикой:

$\Phi ::= x \mid \Phi \rightarrow \Phi \mid (\Phi)$

Добавим в него схему аксиом

$$\overline{\Gamma, \varphi \vdash \varphi}$$

Правило введения импликации:

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi}$$

Правило удаления импликации:

$$\frac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

Theorem

Если Γ и φ состоят только из импликаций, то $\Gamma \vdash \varphi$ равносильна $\Gamma \vdash \rightarrow \varphi$.

Пример

Пример

Докажем $\vdash \varphi \rightarrow \psi \rightarrow \varphi$

$$\frac{\frac{\varphi, \psi \vdash \varphi}{\varphi \vdash \psi \rightarrow \varphi} \text{ (Введение импликации)}}{\vdash \varphi \rightarrow (\psi \rightarrow \varphi)} \text{ (Введение импликации)}$$

Пример

Докажем $\alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \gamma$

$$\frac{\frac{\alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \alpha \rightarrow \beta \rightarrow \gamma \quad \alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \alpha}{\alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \beta \rightarrow \gamma} \quad \alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \beta}{\alpha \rightarrow \beta \rightarrow \gamma, \alpha, \beta \vdash \gamma}$$

Просто типизированное по Карри λ -исчисление

Определение

Тип в просто типизированном λ -исчислении по Карри — это либо маленькая греческая буква ($\alpha, \phi, \theta, \dots$), либо импликация ($\theta_1 \rightarrow \theta_2$)

Таким образом, $\Theta ::= \theta_i | (\Theta \rightarrow \Theta)$

Импликация при этом считается правоассоциативной операцией.

Определение

Язык просто типизированного λ -исчисления — это язык бестипового λ -исчисления.

Определение

Контекст Γ — это список выражений вида $A : \theta$, где A — λ -терм, а θ — тип.

Исчисление по Карри

Аксиома

$$\overline{\Gamma, x : \theta \vdash x : \theta} \text{ если } x \text{ не входит в } \Gamma$$

Правило типизации абстракции

$$\frac{\Gamma, x : \varphi \vdash P : \psi}{\Gamma \vdash (\lambda x. P) : \varphi \rightarrow \psi} \text{ если } x \text{ не входит в } \Gamma$$

Правило типизации аппликации:

$$\frac{\Gamma \vdash P : \varphi \rightarrow \psi \quad \Gamma \vdash Q : \varphi}{\Gamma \vdash PQ : \psi}$$

Мы допускаем в исчислении только те лямбда-выражения, которые имеют тип.

Пример

Докажем $\vdash \lambda x. \lambda y. x : \alpha \rightarrow \beta \rightarrow \alpha$

$$\frac{\frac{x : \alpha, y : \beta \vdash x : \alpha}{x : \alpha \vdash \lambda y. x : \beta \rightarrow \alpha} \text{ (Правило типизации абстракции)}}{\vdash \lambda x. \lambda y. x : \alpha \rightarrow \beta \rightarrow \alpha} \text{ (Правило типизации абстракции)}$$

Пример

Докажем $\vdash \lambda x. \lambda y. x y : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$

$$\frac{\frac{\frac{x : \alpha \rightarrow \beta, y : \alpha \vdash x : \alpha \rightarrow \beta \quad x : \alpha \rightarrow \beta, y : \alpha \vdash y : \alpha}{x : \alpha \rightarrow \beta, y : \alpha \vdash x y : \beta}}{x : \alpha \rightarrow \beta \vdash \lambda y. x y : \alpha \rightarrow \beta}}{\vdash \lambda x. \lambda y. x y : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta}$$

Отсутствие типа у Y-комбинатора

Теорема

Y-комбинатор не типизируется в просто типизированном по Карри λ -исчислении.

Докажем от противного. Есть вывод типа выражения $x\ x$. Единственный вариант типизации:

$$\frac{\Gamma \vdash x : \varphi \rightarrow \psi \quad \Gamma \vdash x : \varphi}{\Gamma \vdash x x : \psi}$$

Рассмотрим типизацию $\Gamma \vdash x : \varphi \rightarrow \psi$ и $\Gamma \vdash x : \varphi$. x типизируется неизбежно следующим образом.

$$\frac{\Gamma', x : \varphi \rightarrow \psi, x : \varphi \vdash x : \varphi \rightarrow \psi \quad \Gamma', x : \varphi \rightarrow \psi, x : \varphi \vdash x : \varphi}{\Gamma', x : \varphi \rightarrow \psi, x : \varphi \vdash x x : \psi}$$

В Γ переменная x встречается два раза.

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

Часть 2. Анонс содержания курса

- ▶ Формальный материал до этого момента должен быть всем знаком — выше были соображения по поводу истории вопроса и напоминание общих определений. Если встретилось что-то неизвестное, задавайте вопросы сами.
- ▶ С этого момента начнётся нечто новое, сейчас будут анонсированы основные темы курса.
- ▶ Курс строится вокруг изоморфизма Карри-Ховарда — изоморфизма между матлогом и лямбда-исчислением. Он состоит из обсуждения двух частей: что мы можем получить из матлога для программирования, и что можем получить для матлога из программирования.

(\Rightarrow) : изучение языков программирования

- ▶ Малая выразительная сила просто-типизированного лямбда исчисления (полиномы).
- ▶ Добавим дополнительные конструкции (например, кванторы).
- ▶ 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>	$\forall x. x \rightarrow x$

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

(\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

- ▶ 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.

Часть 3. 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.