

# Введение в Теорию Типов

## Конспект лекций

Штукенберг Д. Г.  
Университет ИТМО

11 декабря 2018 г.

## 1 Введение

Эти лекции были рассказаны студентам групп М3334–М3337, М3339 в 2018 году в Университете ИТМО, на Кафедре компьютерных технологий Факультета информационных технологий и программирования.

Конспект подготовили студенты Кафедры: Егор Галкин (лекции 1 и 2), Илья Кокорин (лекции 3 и 4), Никита Дугинец (лекции 5 и 6), Степан Прудников (лекции 7 и 8).  
(возможно, история сложнее)

## 2 Лекция 1

### 2.1 $\lambda$ -исчисление

**Определение 2.1** ( $\lambda$ -выражение).  $\lambda$ -выражение — выражение, удовлетворяющее грамматике:

$$\begin{aligned} \Phi ::= & x \\ & | (\Phi) \\ & | \lambda x. \Phi \quad (\textit{abstraction}) \\ & | \Phi \Phi \quad (\textit{application}) \end{aligned}$$

1. Аппликация левоассоциативна.
2. Абстракции жадные, едят все что могут.

**Пример.**  $(\lambda x. (\lambda f. ((fx)(fx) \lambda y. (yf))))$

Есть понятия связанного и свободного вхождения переменной (аналогично исчислению предикатов).  $\lambda x. A$  связывает все свободные вхождения  $x$  в  $A$ .

**Определение 2.2.** Функция  $FV(A)$  — множество свободных переменных, входящих в  $A$ :

$$FV(A) = \begin{cases} \{x\} & \text{если } A \equiv x \\ FV(P) \cup FV(Q) & \text{если } A \equiv PQ \\ FV(P) \setminus \{x\} & \text{если } A \equiv \lambda x. P \end{cases}$$

Договоримся, что:

- Переменные —  $x, a, b, c$ .
- Термы (части  $\lambda$ -выражения) —  $X, A, B, C$ .
- Фиксированные переменные обозначаются буквами из начала алфавита, метаварьируемые — из конца.

На самом деле, смысл в этом есть,  $\lambda$ -выражение можно понимать как функцию. Абстракция — это функция с аргументом, аппликация — это передача аргумента.

**Определение 2.3** ( $\alpha$ -эквивалентность).  $A =_\alpha B$ , если имеет место одно из следующих условий:

1.  $A \equiv x, B \equiv y$  ( $x, y$  — переменные) и  $x \equiv y$
2.  $A \equiv P_1 Q_1, B \equiv P_2 Q_2$  и  $P_1 =_\alpha P_2, Q_1 =_\alpha Q_2$
3.  $A \equiv \lambda x. P_1, B \equiv \lambda y. P_2$  и  $P_1[x := t] =_\alpha P_2[y := t]$ , где  $t$  — новая переменная.

**Пример.**  $\lambda x. \lambda y. xy =_\alpha \lambda y. \lambda x. yx$ .

*Доказательство.* Согласно второму правилу следующие утверждения верны:

$$\begin{aligned} \lambda y. ty =_\alpha \lambda x. tx &\implies \lambda x. \lambda y. xy =_\alpha \lambda y. \lambda x. yx \\ tz =_\alpha tz &\implies \lambda y. ty =_\alpha \lambda x. tx \end{aligned}$$

$tz =_\alpha tz$  верно по третьему условию. □

**Определение 2.4** ( $\beta$ -редекс).  $\beta$ -редекс — выражение вида:  $(\lambda x. A) B$

**Определение 2.5** ( $\beta$ -редукция).  $A \rightarrow_\beta B$ , если имеет место одно из следующих условий:

1.  $A \equiv P_1 Q_1, B \equiv P_2 Q_2$  и либо  $P_1 =_\alpha P_2, Q_1 \rightarrow_\beta Q_2$ , либо  $P_1 \rightarrow_\beta P_2, Q_1 =_\alpha Q_2$
2.  $A \equiv (\lambda x. P) Q, B \equiv P[x := Q]$  —  $Q$  свободна для подстановки вместо  $x$  в  $P$

**Пример.**  $X \rightarrow_\beta X, (\lambda x. x) y \rightarrow_\beta y$

**Пример.**  $a (\lambda x. x) y \rightarrow_\beta ay$

**Пример.**  $A \equiv \lambda x. P, B \equiv \lambda x. Q, P \rightarrow_\beta Q$

## 2.2 Представление некоторых функций в лямбда исчислении

Boolean значения легко представить в терминах  $\lambda$ -исчисления, к примеру

- $\text{True} = \lambda a \lambda b. a$
- $\text{False} = \lambda a \lambda b. b$

Также мы можем выражать и более сложные функции

**Определение 2.6.**  $\text{If} = \lambda c. \lambda t. \lambda e. (ct)e$

**Пример.**  $\text{If } T \ a \ b \rightarrow_\beta a$

Доказательство.

$$\begin{aligned} ((\lambda c. \lambda t. \lambda e. (ct)e) \lambda a \lambda b. a) a b \rightarrow_{\beta} (\lambda t. \lambda e. (\lambda a \lambda b. a) t e) a b \rightarrow_{\beta} \\ (\lambda t. \lambda e. (\lambda b. t) e) a b \rightarrow_{\beta} (\lambda t. \lambda e. t) a b \rightarrow_{\beta} (\lambda e. a) b \rightarrow_{\beta} a \end{aligned}$$

□

Как мы видим If true действительно возвращает результат первой ветки.  
Другие логические операции:

$$\text{Not} = \lambda a. a \text{ F T} \quad \text{Add} = \lambda a. \lambda b. a b \text{ F} \quad \text{Or} = \lambda a. \lambda b. a \text{ T } b$$

## 2.3 Черчевские нумералы

**Определение 2.7** (черчевский нумерал).

$$\bar{n} = \lambda f. \lambda x. f^n x, \quad \text{где} \quad f^n x = \begin{cases} f(f^{n-1}x) & \text{при } n > 0 \\ x & \text{при } n = 0 \end{cases}$$

**Пример.**

$$\bar{3} = \lambda f. \lambda x. f(f(fx))$$

Несложно определить прибавление единицы к такому нумералу:

$$(+1) = \lambda n. \lambda f. \lambda x. f(nfx)$$

Арифметические операции:

1. IsZero =  $\lambda n. n (\lambda x. \text{F}) \text{T}$
2. Add =  $\lambda a. \lambda b. \lambda f. \lambda x. a f (b f x)$
3. Pow =  $\lambda a. \lambda b. b (\text{Mul } a) \bar{1}$
4. IsEven =  $\lambda n. n \text{ Not } \text{T}$
5. Mul =  $\lambda a. \lambda b. a (\text{Add } b) \bar{0}$

Для того, чтобы определить  $(-1)$ , сначала определим «пару»:

$$\langle a, b \rangle = \lambda f. f a b \quad \text{First} = \lambda p. p \text{T} \quad \text{Second} = \lambda p. p \text{F}$$

Затем  $n$  раз применим функцию  $f(\langle a, b \rangle) = \langle b, b + 1 \rangle$  и возьмём первый элемент пары:

$$(-1) = \lambda n. \text{First}(n(\lambda p. \langle (\text{Second } p), (+1)(\text{Second } p) \rangle)) \langle \bar{0}, \bar{0} \rangle$$

## 3 Лекция 2

### 3.1 Формализация $\lambda$ -термов, классы $\alpha$ -эквивалентности термов

**Определение 3.1** ( $\lambda$ -терм). Рассмотрим классы эквивалентности  $[A] = \alpha$   
Будем говорить, что  $[A] \rightarrow_{\beta} [B]$ , если  $\exists A' \in [A], B' \in [B]$ , что  $A' \rightarrow_{\beta} B'$ .

**Лемма 3.1.**  $=_{\alpha}$  — отношение эквивалентности.

Пусть в  $A$  есть  $\beta$ -редекс  $\lambda x. Q$ , но  $P[x := Q]$  не может быть, тогда найдем  $y \notin V[P]$ ,  $y \notin V[Q]$ . Сделаем замену  $P[x := y]$ . Тогда замена  $P[x := y][y := Q]$  допустима.

**Лемма 3.2.**  $P[x := y] =_{\alpha} P[x := y][y := Q]$ , если замена допустима.

### 3.2 Нормальная форма, $\lambda$ -выражения без нормальной формы, комбинаторы $K$ , $I$ , $\Omega$

**Определение 3.2.** Нормальная форма — это  $\lambda$ -выражение без  $\beta$ -редексов.

**Лемма 3.3.**  $\lambda$ -выражение  $A$  в нормальной форме, т.е.т.т, когда  $\nexists B$ , что  $A \rightarrow_\beta B$ .

**Определение 3.3.**  $A$  — Н.Ф  $B$ , если  $\exists A_1 \dots A_n$ , что  $B =_\alpha A_1 \rightarrow_\beta A_2 \rightarrow_\beta \dots \rightarrow_\beta A_n =_\alpha A$ .

**Определение 3.4.** Комбинатор —  $\lambda$ -выражение без свободных переменных.

**Определение 3.5.**

- $I = \lambda x.x$  (Identitant)
- $K = \lambda a.\lambda b.a$  (Konstanz)
- $\Omega = (\lambda x.xx)(\lambda x.xx)$

**Лемма 3.4.**  $\Omega$  — не имеет нормальной формы.

*Доказательство.*  $\Omega \rightarrow_\beta \Omega$

□

### 3.3 $\beta$ -редуцируемость

**Определение 3.6.** Будем говорить, что  $A \twoheadrightarrow_\beta B$ , если  $\exists$  такие  $X_1 \dots X_n$ , что  $A =_\alpha X_1 \rightarrow_\beta X_2 \rightarrow_\beta \dots \rightarrow_\beta X_{n-1} \rightarrow_\beta X_n =_\alpha B$ .

$\twoheadrightarrow_\beta$  — рефлексивное и транзитивное замыкание  $\rightarrow_\beta$ .  $\twoheadrightarrow_\beta$  не обязательно приводит к нормальной форме

**Пример.**  $\Omega \twoheadrightarrow_\beta \Omega$

### 3.4 Ромбовидное свойство

**Определение 3.7** (Ромбовидное свойство). Отношение  $R$  обладает ромбовидным свойством, если  $\forall a, b, c$ , таких, что  $aRb$ ,  $aRc$ ,  $b \neq c$ ,  $\exists d$ , что  $bRd$  и  $cRd$ . Далее будем обозначать ромбовидное свойство как  $<>$ .

**Пример.**  $(\leq)$  на множестве натуральных чисел обладает  $<>$   $(>)$  не обладает  $<>$  на множестве натуральных чисел

### 3.5 Теорема Чёрча-Россера, следствие о единственности нормальной формы

**Теорема 3.1** (Черча-Россера).  $(\twoheadrightarrow_\beta)$  обладает ромбовидным свойством.

**Следствие 3.1.** Если у  $A$  есть Н.Ф, то она единственная с точностью до  $(=_\alpha)$  (переименования переменных).

*Доказательство.* Пусть  $A \twoheadrightarrow_\beta B$  и  $A \twoheadrightarrow_\beta C$ .  $B, C$  — нормальные формы и  $B \neq_\alpha C$ . Тогда по теореме Черча-Россера  $\exists D$ :  $B \twoheadrightarrow_\beta D$  и  $C \twoheadrightarrow_\beta D$ . Тогда  $B =_\alpha D$  и  $C =_\alpha D \Rightarrow B =_\alpha C$ . Противоречие. □

**Лемма 3.5.** Если  $B$  — Н.Ф, то  $\nexists Q$ :  $B \rightarrow_\beta Q$ . Значит если  $B \twoheadrightarrow_\beta Q$ , то количество шагов редукции равно 0.

**Лемма 3.6.** Если  $R$  — обладает  $<>$ , то и  $R^*$  (транзитивное, рефлексивное замыкание  $R$ ) обладает  $R^*$ .

*Доказательство.* content...

□

**Лемма 3.7** (Грустная лемма).  $(\rightarrow_\beta)$  не обладает  $<>$

**Определение 3.8** (Параллельная  $\beta$ -редукция).  $A \rightrightarrows_\beta B$ , если

1.  $A =_\alpha B$
2.  $A \equiv P_1 Q_1, B \equiv P_2 Q_2$  и  $P_1 \rightrightarrows_\beta P_2, Q_1 \rightrightarrows_\beta Q_2$
3.  $A \equiv \lambda x. P_1, B \equiv \lambda x. P_2$  и  $P_1 \rightrightarrows_\beta P_2$
4.  $A =_\alpha (\lambda x. P) Q, B =_\alpha P[x := Q]$

**Лемма 3.8.**  $(\rightrightarrows_\beta)$  обладает  $<>$

$$P_1 \rightrightarrows_\beta P_2 \text{ и } Q_1 \rightrightarrows_\beta Q_2, \text{ то } P_1[x := Q_1] \rightrightarrows_\beta P_2[x := Q_2]$$

### 3.6 Нормальный и аппликативный порядок вычислений

**Пример.** Выражение  $KI\Omega$  можно редуцировать двумя способами:

1.  $KI\Omega =_\alpha ((\lambda a. \lambda b. a)I)\Omega \rightarrow_\beta (\lambda b. I)\Omega \rightarrow_\beta I$
2.  $KI\Omega =_\alpha ((\lambda a. \lambda b. a)I)((\lambda x. x x)(\lambda x. x x)) \rightarrow_\beta ((\lambda a. \lambda b. a)I)((\lambda x. x x)(\lambda x. x x)) \rightarrow_\beta KI\Omega$

Как мы видим, в первом случае мы достигли нормальной формы, в то время как во втором мы получаем бесконечную редукцию. Разница двух этих способов в порядке редукции. Первый называется нормальный порядок, а второй аппликативный.

**Определение 3.9** (нормальный порядок редукции). Редукция самого левого  $\beta$ -редекса.

**Определение 3.10** (аппликативный порядок редукции). Редукция самого левого  $\beta$ -редекса из самых вложенных.

**Утверждение 3.1.** Если нормальная форма существует, она может быть достигнута нормальным порядком редукции.