# Теория типов

Человек, который поспорил на 2 торта ♡

## 1 $\lambda$-исчисление

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

$$\begin{aligned} \Lambda ::= & \lambda x.\Lambda && \text{(абстракция)} \\ & |\, \Lambda\Lambda && \text{(аппликация)} \\ & |\, x \\ & |\, (\Lambda) \end{aligned}$$

(a) аппликация левоассоциативна

(b) абстракция распространяется как можно дальше вправо

(c) смысла в этом нет

*Пример.* $((\lambda z.(z(yz)))(zx)z) = (\lambda z.z(yz))(zx)z$

Есть понятия связанного и свободного вхождения переменной (аналогично ИП). $\lambda x.A$ связывает все свободные вхождения $x$ в $A$. Договоримся, что:

(a) Переменные — $x$, $a$, $b$, $c$.

(b) Термы (части $\lambda$-выражения) — $X$, $A$, $B$, $C$.

(c) Фиксированные переменные обозначаются буквами из начала алфавита, метапеременные — из конца.

**Определение** ($\alpha$-эквивалентность)**.** $A$ и $B$ называются $\alpha$-эквивалентными ($A =_\alpha B$), если выполнено одно из следующих условий:

1. $A \equiv x$ и $B \equiv x$ .
2. $A \equiv \lambda x.P$ и $B \equiv \lambda x.Q$. Пусть $t$ — новая переменная, тогда $P_{[x:=t]} =_\alpha Q_{[y:=t]}$ .
3. $A \equiv PQ$, $B \equiv RS$, $P =_\alpha R$, $Q =_\alpha S$ .

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

*Доказательство.*

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

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

**Определение** ($\beta$-редекс)**.** Терм вида $(\lambda a.A)\,B$ называется $\beta$-редексом.

*Пример.* В выражении $(\lambda f.\underbrace{(\lambda x.\overbrace{f(xx)}^{A_1})\overbrace{(\lambda x.f(xx))}^{B_1}}_{A_2})\ \underbrace{g}_{B_2}$ два $\beta$-редекса.

**Определение.** Множество $\lambda$-термов $\mathbf{\Lambda}$ назовём множеством классов эквивалентности $\Lambda$ по $(=_\alpha)$.

**Определение** ($\beta$-редукция)**.** $A \to_\beta B$ (состоят в отношении $\beta$-редукции), если выполняется одно из условий:

1. $A \equiv PQ$, $B \equiv RS$ и
$$\begin{gathered}\text{либо } P \to_\beta R \text{ и } Q =_\alpha S\\ \text{либо } P =_\alpha R\ \text{ и } Q \to_\beta S\end{gathered}$$
2. $A \equiv \lambda x.P$, $B \equiv \lambda x.Q$, $P \to_\beta Q$ ($x$ из какого-то класса из $\mathbf{\Lambda}$).
3. $A \equiv (\lambda x.P)Q$, $B \equiv P_{[x:=Q]}$, $Q$ свободно для подстановки в $P$ вместо $x$.

**Итак, лулзы.** Хотите знать, что такое истина?

$$\begin{aligned}\mathrm{T} &= \lambda x\lambda y.x\\ \mathrm{F} &= \lambda x\lambda y.y\\ \mathrm{Not} &= \lambda a.a\mathrm{FT}\end{aligned}$$

Похоже на тип boolean, не правда ли?

*Пример.*
$$\mathrm{Not\ T} = (\lambda a.a\mathrm{FT})\mathrm{T} \to_\beta \mathrm{TFT} = (\lambda x.\lambda y.x)\mathrm{FT} \to_\beta (\lambda y.\mathrm{F})\mathrm{T} \to_\beta \mathrm{F}$$

Можно продолжить:

$$\begin{aligned}\mathrm{And} &= \lambda a.\lambda b.ab\mathrm{F}\\ \mathrm{Or} &= \lambda a.\lambda b.a\mathrm{T}b\end{aligned}$$

Попробуем определить числа:

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

Это называют числами Чёрча.

$$\begin{aligned}(+1) &= \lambda n.\lambda f.\lambda x.f(nfx)\\ \mathrm{IsZero} &= \lambda n.n(\lambda x.\mathrm{T})\mathrm{F}\\ \mathrm{Add} &= \lambda a.\lambda b.\lambda f.\lambda x.af(bfx)\\ \mathrm{Mul} &= \lambda a.\lambda b.a(\mathrm{Add}\ b)\overline{0}\\ \mathrm{Pow} &= \lambda a.\lambda b.a(\mathrm{Mul}\ b)\overline{1}\\ \mathrm{Pow}^* &= \lambda a.\lambda b.ba\end{aligned}$$

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

$$\begin{aligned}\langle a, b\rangle &= \lambda f.fab\\ \text{First} &= \lambda p.\text{T}p\\ \text{Second} &= \lambda p.\text{F}p\end{aligned}$$

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

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

Сокращение записи:

$$\lambda xy.A = \lambda x.\lambda y.A$$

**Определение** (Нормальная форма)**.**
Терм $A$ — нормальная форма (н.ф.), если в нём нет $\beta$-редексов.
Нормальной формой $A$ называется такой $B$, что $A \twoheadrightarrow_\beta B$, $B$ — н.ф.
$\twoheadrightarrow_\beta$ — транзитивно-рефлексивное замыкание $\rightarrow_\beta$.

**Утверждение 1.** *Существует $\lambda$-выражение, не имеющее н.ф.*

$$\begin{aligned}\Omega &= \omega\omega\\ \omega &= \lambda x.xx\end{aligned}$$

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

Комбинатор неподвижной точки:

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

**Определение** ($\beta$-эквивалентность)**.** $A =_\beta B$, если $\exists C : C \twoheadrightarrow_\beta A, C \twoheadrightarrow_\beta B$

**Утверждение 2.**

$$Yf =_\beta f(Yf)$$

*Доказательство.* (на лекции не давалось)

$$\begin{aligned}Yf &=_\beta (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))f\\ &=_\beta (\lambda x.f(xx))(\lambda x.f(xx))\\ &=_\beta f((\lambda x.f(xx))(\lambda x.f(xx)))\\ &=_\beta f(Yf)\end{aligned}$$

□

Таким образом, с помощью $Y$-комбинатора можно определять рекурсивные функции.

*Пример.*

$$\text{Fact} = Y(\lambda fn.\text{IsZero}\ n\ \overline{1}\ (\text{Mul}\ n\ (f\ (-1)\ n)))$$