

Лямбда-исчисление

Лямбда-исчисление, синтаксис

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

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

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

Примеры:

- ▶ $M := \lambda b. \lambda c. a \ c \ (b \ c); FV(M) = \{a\}$
- ▶ $N := x \ (\lambda x. (x \ (\lambda y. x))); FV(N) = \{x\}$

Определение

$A =_{\alpha} B$, если и только если выполнено одно из трёх:

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 .

Определение

$$L = \Lambda / =_{\alpha}$$

Альфа-эквивалентность, пример

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

Интуиция: вызов функции.

λ -выражение	Python
$\lambda f.\lambda x.f\ x$	<code>def one(f,x): return f(x)</code>
$(\lambda x.x\ x)\ (\lambda x.x\ x)$	<code>(lambda x: x x) (lambda x: x x)</code>
	<code>def omega(x): return x(x); omega(omega)</code>

Бета-редукция

Интуиция: вызов функции.

λ -выражение	Python
$\lambda f.\lambda x.f\ x$	<code>def one(f,x): return f(x)</code>
$(\lambda x.x\ x)\ (\lambda x.x\ x)$	<code>(lambda x: x x) (lambda x: x x)</code>
	<code>def omega(x): return x(x); omega(omega)</code>

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 $P \rightarrow_\beta P'$.

Бета-редукция, пример

Пример

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

Пример

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

Нормальная форма

Определение

Лямбда-терм N находится в нормальной форме, если нет $Q: N \rightarrow_{\beta} Q$.

Пример

В нормальной форме:

$\lambda f. \lambda x. x (f (f \lambda g. x))$

Нормальная форма

Определение

Лямбда-терм N находится в нормальной форме, если нет $Q: N \rightarrow_{\beta} Q$.

Пример

В нормальной форме:

$\lambda f. \lambda x. x (f (f \lambda g. x))$

Пример

Не в нормальной форме (редексы подчёркнуты):

$\lambda f. \lambda x. (\lambda g. x) (f (f x))$
 $(\underline{(\lambda x. x) (\lambda x. x)}) (\underline{(\lambda x. x) (\lambda x. x)})$

Определение

(\rightarrow_{β}) — транзитивное и рефлексивное замыкание (\rightarrow_{β}) .

Булевские значения

$T := \lambda x. \lambda y. x$ $F := \lambda x. \lambda y. y$

Тогда: $Or := \lambda a. \lambda b. a \ T \ b$:

$$\begin{aligned} Or \ F \ T &= ((\lambda a. \lambda b. a \ T \ b) \ F) \ T \rightarrow_{\beta} (\lambda b. F \ T \ b) \ T \rightarrow_{\beta} F \ T \ T = \\ &= (\lambda x. \lambda y. y) \ T \ T \rightarrow_{\beta} (\lambda y. y) \ T \rightarrow_{\beta} T \end{aligned}$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$(\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ \bar{0} = (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ (\lambda f'. \lambda x'. x') \rightarrow_{\beta}$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$\begin{aligned} (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ \bar{0} &= (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ (\lambda f'. \lambda x'. x') \rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda f'. \lambda x'. x') \ f \ (f \ x) &\rightarrow_{\beta} \end{aligned}$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$\begin{aligned} (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ \bar{0} &= (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ (\lambda f'. \lambda x'. x') \rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda f'. \lambda x'. x') \ f \ (f \ x) &\rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda x'. x') \ (f \ x) &\rightarrow_{\beta} \end{aligned}$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$\begin{aligned} (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ \bar{0} &= (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ (\lambda f'. \lambda x'. x') \rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda f'. \lambda x'. x') \ f \ (f \ x) &\rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda x'. x') \ (f \ x) &\rightarrow_{\beta} \\ \dots \lambda f. \lambda x. f \ x &= \bar{1} \end{aligned}$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$\begin{aligned} (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ \bar{0} &= (\lambda n. \lambda f. \lambda x. n \ f \ (f \ x)) \ (\lambda f'. \lambda x'. x') \rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda f'. \lambda x'. x') \ f \ (f \ x) &\rightarrow_{\beta} \\ \dots \lambda f. \lambda x. (\lambda x'. x') \ (f \ x) &\rightarrow_{\beta} \\ \dots \lambda f. \lambda x. f \ x &= \bar{1} \end{aligned}$$

Декремент: $Dec = \lambda n. \lambda f. \lambda x. n \ (\lambda g. \lambda h. h \ (g \ f)) \ (\lambda u. x) \ (\lambda u. u)$

Упорядоченная пара и алгебраический тип

Определение

$Pair(a, b) := \lambda s. s \ a \ b$

$Fst := \lambda p. p \ T$

$Snd := \lambda p. p \ F$

Пример

$Fst(Pair(a, b)) = (\lambda p. p \ T) \ \lambda s. s \ a \ b \twoheadrightarrow_{\beta} (\lambda s. s \ a \ b) \ T \rightarrow_{\beta} a$

Определение

$InL \ L := \lambda p. \lambda q. p \ L$

$InR \ R := \lambda p. \lambda q. q \ R$

$Case \ t \ f \ g := t \ f \ g$

Теорема Чёрча-Россера

Теорема (Чёрча-Россера)

Для любых термов N, P, Q , если $N \twoheadrightarrow_{\beta} P$, $N \twoheadrightarrow_{\beta} Q$, и $P \neq Q$, то найдётся T : $P \twoheadrightarrow_{\beta} T$ и $Q \twoheadrightarrow_{\beta} T$.

Теорема

Если у терма N существует нормальная форма, то она единственна

Доказательство.

Пусть не так и $N \twoheadrightarrow_{\beta} P$ вместе с $N \twoheadrightarrow_{\beta} Q$, $P \neq Q$. Тогда по теореме Чёрча-Россера существует T : $P \twoheadrightarrow_{\beta} T$ и $Q \twoheadrightarrow_{\beta} T$, причём $T \neq P$ или $T \neq Q$ в силу транзитивности ($\twoheadrightarrow_{\beta}$) □

Бета-эквивалентность, неподвижная точка

Пример

$\Omega = (\lambda x. x \ x) (\lambda x. x)$ не имеет нормальной формы: $\Omega \rightarrow_{\beta} \Omega$

Определение

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

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 $R := Y \ N$:

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



Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

Просто-типизированное лямбда-исчисление.

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

Просто-типизированное лямбда-исчисление. Типы: $\tau ::= \alpha | (\tau \rightarrow \tau)$.

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

Просто-типизированное лямбда-исчисление. Типы: $\tau ::= \alpha | (\tau \rightarrow \tau)$. Язык:

$\Gamma \vdash A : \varphi$

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} \quad x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x. A : \varphi \rightarrow \psi} \quad x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Пример: тип чёреческих нумералов

Пусть $\Gamma = f : \alpha \rightarrow \alpha, x : \alpha$

$$\frac{\frac{\frac{\overline{\Gamma \vdash x : \alpha} \text{Ax}}{\Gamma \vdash f \ x : \alpha} \text{App} \quad \frac{\overline{\Gamma \vdash f : \alpha \rightarrow \alpha} \text{Ax}}{\Gamma \vdash f : \alpha \rightarrow \alpha} \text{App}}{\frac{\{f : \alpha \rightarrow \alpha, x : \alpha\} \Gamma \vdash f (f \ x) : \alpha}{f : \alpha \rightarrow \alpha \vdash \lambda x. f (f \ x) : (\alpha \rightarrow \alpha)} \lambda} \lambda$$

Изоморфизм Карри-Ховарда

λ -исчисление	исчисление высказываний
Выражение	доказательство
Тип выражения	высказывание
Тип функции	импликация
Упорядоченная пара	Конъюнкция
Алгебраический тип	Дизъюнкция
Необитаемый тип	Ложь

Изоморфизм Карри-Ховарда: отрицание

Определение

Ложь (\perp) — необитаемый тип; $failwith/raise/throw : \alpha \rightarrow \perp$; $\neg\varphi \equiv \varphi \rightarrow \perp$

Например, контрапозиция: $(\alpha \rightarrow \beta) \rightarrow (\neg\beta \rightarrow \neg\alpha)$

$$\frac{\frac{\frac{\overline{\Phi \vdash a : \alpha} \text{ Ax} \quad \overline{\Phi \vdash f : \alpha \rightarrow \beta} \text{ Ax}}{\Phi \vdash f a : \beta} \text{ App} \quad \overline{\Phi \vdash n : \beta \rightarrow \perp} \text{ Ax}}{\frac{f : \alpha \rightarrow \beta, n : \beta \rightarrow \perp, a : \alpha \vdash n (f a) : \perp}{f : \alpha \rightarrow \beta, n : \beta \rightarrow \perp \vdash \lambda a^\alpha. n (f a) : \neg\alpha} \lambda} \text{ App} \quad \lambda}{f : \alpha \rightarrow \beta \vdash \lambda n^{\beta \rightarrow \perp}. \lambda a^\alpha. n (f a) : \neg\beta \rightarrow \neg\alpha} \lambda \quad \lambda}{\lambda f^{\alpha \rightarrow \beta}. \lambda n^{\beta \rightarrow \perp}. \lambda a^\alpha. n (f a) : (\alpha \rightarrow \beta) \rightarrow (\neg\beta \rightarrow \neg\alpha)} \lambda$$

Снятие двойного отрицания: $((\alpha \rightarrow \perp) \rightarrow \perp) \rightarrow \alpha$, то есть $\lambda f^{(\alpha \rightarrow \perp) \rightarrow \perp}. ? : \alpha$.

f угадывает, что передать $x : \alpha \rightarrow \perp$. Тогда надо по f угадать, что передать x .

Исчисление по Чёрчу и по Карри

Определение

Просто-типизированное лямбда-исчисление по Карри.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Просто-типизированное лямбда-исчисление по Чёрчу.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x^{\varphi}. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Исчисление по Чёрчу и по Карри

Определение

Просто-типизированное лямбда-исчисление по Карри.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Просто-типизированное лямбда-исчисление по Чёрчу.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x^{\varphi}. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Пример

По Карри	По Чёрчу
$\lambda f. \lambda x. f (f x) : (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$	$\lambda f^{\alpha \rightarrow \alpha}. \lambda x^{\alpha}. f (f x) : (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$

Исчисление по Чёрчу и по Карри

Определение

Просто-типизированное лямбда-исчисление по Карри.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Просто-типизированное лямбда-исчисление по Чёрчу.

$$\frac{}{\Gamma, x : \varphi \vdash x : \varphi} x \notin \Gamma \quad \frac{\Gamma, x : \varphi \vdash A : \psi}{\Gamma \vdash \lambda x^\varphi. A : \varphi \rightarrow \psi} x \notin \Gamma \quad \frac{\Gamma \vdash A : \varphi \quad \Gamma \vdash B : \varphi \rightarrow \psi}{\Gamma \vdash BA : \psi}$$

Пример

По Карри	По Чёрчу
$\lambda f. \lambda x. f (f x) : (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$	$\lambda f^{\alpha \rightarrow \alpha}. \lambda x^\alpha. f (f x) : (\alpha \rightarrow \alpha) \rightarrow (\alpha \rightarrow \alpha)$
$\lambda f. \lambda x. f (f x) : (\beta \rightarrow \beta) \rightarrow (\beta \rightarrow \beta)$	$\lambda f^{\beta \rightarrow \beta}. \lambda x^\beta. f (f x) : (\beta \rightarrow \beta) \rightarrow (\beta \rightarrow \beta)$

Комбинаторы S, K

Определение

Комбинатор — лямбда-терм без свободных переменных

Определение

$S := \lambda x. \lambda y. \lambda z. x \ z \ (y \ z), \ K := \lambda x. \lambda y. x, \ I := \lambda x. x$

Теорема

Пусть N — некоторый замкнутый лямбда-терм. Тогда найдётся выражение C , состоящее из комбинаторов S, K , что $N =_{\beta} C$

Комбинаторы S,K

Определение

Комбинатор — лямбда-терм без свободных переменных

Определение

$S := \lambda x. \lambda y. \lambda z. x \ z \ (y \ z), \ K := \lambda x. \lambda y. x, \ I := \lambda x. x$

Теорема

Пусть N — некоторый замкнутый лямбда-терм. Тогда найдётся выражение C , состоящее из комбинаторов S, K , что $N =_{\beta} C$

Пример

$K := \lambda x^{\alpha}. \lambda y^{\beta}. x$

$\alpha \rightarrow \beta \rightarrow \alpha$

$S := \lambda x^{\alpha \rightarrow \beta \rightarrow \gamma}. \lambda y^{\alpha \rightarrow \beta}. \lambda z^{\alpha}. x \ z \ (y \ z) \quad (\alpha \rightarrow \beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma$

$I =_{\beta} S \ K \ K$

Дальнейшее развитие: изоморфизм Карри-Ховарда и вокруг него

Исчисление второго порядка

- ▶ Напомним о порядках:

Порядок	Объекты	Пример
0 (И.В.)	Атомарные	P
1 (И.П. 1)	Множества	$\{x P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \forall t.t > 0 \rightarrow P(t)\}$

Исчисление второго порядка

- ▶ Напомним о порядках:

Порядок	Объекты	Пример
0 (И.В.)	Атомарные	P
1 (И.П. 1)	Множества	$\{x P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \forall t.t > 0 \rightarrow P(t)\}$

- ▶ Можно заменить схемы аксиом на аксиомы: $\forall a.\forall b.a \rightarrow b \rightarrow a$

Исчисление второго порядка

- ▶ Напомним о порядках:

Порядок	Объекты	Пример
0 (И.В.)	Атомарные	P
1 (И.П. 1)	Множества	$\{x P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \forall t.t > 0 \rightarrow P(t)\}$

- ▶ Можно заменить схемы аксиом на аксиомы: $\forall a.\forall b.a \rightarrow b \rightarrow a$
- ▶ Острый угол: импредикативность (формулы могут говорить о себе). Что такое «предикат»? Произвольное выражение, а подстановка — буквальная замена текста? Тогда каково $\llbracket p(p) \rrbracket$ при $p(x) = x(x) \rightarrow \perp$?

Исчисление второго порядка

- ▶ Напомним о порядках:

Порядок	Объекты	Пример
0 (И.В.)	Атомарные	P
1 (И.П. 1)	Множества	$\{x P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \forall t.t > 0 \rightarrow P(t)\}$

- ▶ Можно заменить схемы аксиом на аксиомы: $\forall a.\forall b.a \rightarrow b \rightarrow a$
- ▶ Острый угол: импредикативность (формулы могут говорить о себе). Что такое «предикат»? Произвольное выражение, а подстановка — буквальная замена текста? Тогда каково $\llbracket p(p) \rrbracket$ при $p(x) = x(x) \rightarrow \perp$?
Нужна точная формализация.
- ▶ Самый простой вариант: переменные второго порядка — только булевские пропозициональные переменные.

Исчисление второго порядка

- ▶ Напомним о порядках:

Порядок	Объекты	Пример
0 (И.В.)	Атомарные	P
1 (И.П. 1)	Множества	$\{x P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \forall t.t > 0 \rightarrow P(t)\}$

- ▶ Можно заменить схемы аксиом на аксиомы: $\forall a.\forall b.a \rightarrow b \rightarrow a$
- ▶ Острый угол: импредикативность (формулы могут говорить о себе). Что такое «предикат»? Произвольное выражение, а подстановка — буквальная замена текста? Тогда каково $\llbracket p(p) \rrbracket$ при $p(x) = x(x) \rightarrow \perp$?
Нужна точная формализация.
- ▶ Самый простой вариант: переменные второго порядка — только булевские пропозициональные переменные.

$$\llbracket \forall p.Q \rrbracket = \begin{cases} \text{И,} & \llbracket Q \rrbracket^{p:=\text{И}} = \llbracket Q \rrbracket^{p:=\text{Л}} = \text{И} \\ \text{Л,} & \text{иначе} \end{cases}$$

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- ▶ Что такое $T : \forall x. x \rightarrow x$?

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- ▶ Что такое $T : \forall x. x \rightarrow x$?

```
template <class x> class T { x f (x); }
```

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- ▶ Что такое $T : \forall x. x \rightarrow x$?

```
template <class x> class T { x f (x); }
```

- ▶ Что такое $T : \exists x. \tau(x)$?

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- ▶ Что такое $T : \forall x. x \rightarrow x$?

```
template <class x> class T { x f (x); }
```

- ▶ Что такое $T : \exists x. \tau(x)$?

Абстрактный тип данных: `interface T { τ }; f(T x)`

Зависимые типы

- ▶ Рассмотрим код
`int n; cin >> n; int arr[n];`
Каков тип `arr`?

Зависимые типы

- ▶ Рассмотрим код
`int n; cin >> n; int arr[n];`
Каков тип `arr`?
- ▶ $\text{sizeof}(\text{arr}) = n \cdot \text{sizeof}(\text{int})$

Зависимые типы

- ▶ Рассмотрим код
`int n; cin >> n; int arr[n];`
Каков тип `arr`?
- ▶ `sizeof(arr) = n · sizeof(int)`
- ▶ `arr = Πintn.int[n]`

Зависимые типы

- ▶ Рассмотрим код
`int n; cin >> n; int arr[n];`
Каков тип `arr`?
- ▶ `sizeof(arr) = n · sizeof(int)`
- ▶ `arr = Πintn.int[n]`
- ▶ Аналогично, `printf(const char*, ...)` — капитуляция.

Зависимые типы

- ▶ Рассмотрим код
`int n; cin >> n; int arr[n];`
Каков тип `arr`?
- ▶ `sizeof(arr) = n · sizeof(int)`
- ▶ `arr = Πintn.int[n]`
- ▶ Аналогично, `printf(const char*, ...)` — капитуляция.
- ▶ Есть языки, где тип выписывается (например, Идрис).

Прямолинейное: доказательства в коде

► `Div2: (l: int) -> (even l) -> int`

Прямолинейное: доказательства в коде

- ▶ `Div2: (l: int) -> (even l) -> int`
- ▶ `even l` — что это?

Прямолинейное: доказательства в коде

► `Div2: (l: int) -> (even l) -> int`

► `even l` — что это?

►

$$\text{even}(x) ::= \begin{cases} EZ, & x = 0 \\ EP(\text{even}(y)), & x = y'' \end{cases}$$

Прямолинейное: доказательства в коде

► `Div2: (l: int) -> (even l) -> int`

► `even l` — что это?



$$\text{even}(x) ::= \begin{cases} EZ, & x = 0 \\ EP(\text{even}(y)), & x = y'' \end{cases}$$

► `Div2 10 (EP (EP (EP (EP (EP EZ))))))`

Прямолинейное: доказательства в коде

► `Div2: (l: int) -> (even l) -> int`

► `even l` — что это?



$$\text{even}(x) ::= \begin{cases} EZ, & x = 0 \\ EP(\text{even}(y)), & x = y'' \end{cases}$$

► `Div2 10 (EP (EP (EP (EP (EP EZ))))))`

► А если `Div2 p`? В общем случае сложно.

`Plus2: (l: int) -> (p: even l) -> (l+2, even (l+2)) = (l+2, EP p)`

Интереснее: доказательства утверждений

Натуральные числа: $\text{Nat} ::= 0 \mid \text{suc Nat}$,

$$a + b = \begin{cases} a, & b = 0 \\ \text{suc } (a + c), & b = \text{suc } c \end{cases}$$

```
func pmap A B :  
(f : A -> B) {a a' : A} (p : a = a') : f a = f a' => ...
```

```
func +-comm (n m : Nat) : n + m = m + n  
| 0, 0 => idp  
| suc n, 0 => pmap suc (+-comm n 0)  
| 0, suc m => pmap suc (+-comm 0 m)  
| suc n, suc m => pmap suc (+-comm (suc n) m *>  
pmap suc (inv (+-comm n m)) *> +-comm n (suc m))
```

Гомотопическая теория типов

Определение

Изоморфизм Карри-Ховарда-Воеводского.

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

1. Точный смысл равенства.
2. Позволяет легко формулировать утверждения про топологию, гомологическую алгебру и т.п.
3. Можно реализовать (кубическая теория типов). Реализации для Агды, Кока, ..., отдельные языки (Аренд)

Какие вопросы пытаемся решать?

Пример

Самое простое: $x = y$. Почему $x^2 = y^2$?

Какие вопросы пытаемся решать?

Пример

Самое простое: $x = y$. Почему $x^2 = y^2$?

А что если так $(a = b) = \{\langle a, b \rangle \mid a < 10 \ \& \ b < 10\}$? Тогда $5 = 7$, но $25 \neq 49$.

Какие вопросы пытаемся решать?

Пример

Самое простое: $x = y$. Почему $x^2 = y^2$?

А что если так $(a = b) = \{\langle a, b \rangle | a < 10 \ \& \ b < 10\}$? Тогда $5 = 7$, но $25 \neq 49$.

Постулируется в формальной арифметике: (A2) $a = b \rightarrow a' = b'$

Какие вопросы пытаемся решать?

Пример

Самое простое: $x = y$. Почему $x^2 = y^2$?

А что если так $(a = b) = \{\langle a, b \rangle \mid a < 10 \ \& \ b < 10\}$? Тогда $5 = 7$, но $25 \neq 49$.

Постулируется в формальной арифметике: (A2) $a = b \rightarrow a' = b'$

Доказательство.

Путь x в y — функция $f : [0, 1] \rightarrow S$, $f(0) = x$, $f(1) = y$. $f(x) = x^2$ — непрерывная функция. Тогда $f(x^2)$ — тоже непрерывная, то есть $x^2 = y^2$. □

Что ещё

- ▶ Метод резолюций и рядом — Prolog, SMT-солверы,...
- ▶ Можно пытаться совмещать (F^* , ...)