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

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

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.

$$\begin{array}{rcll}
t & =_\alpha & t & \text{Правило 1} \\
s & =_\alpha & s & \text{Правило 1} \\
t\ s & =_\alpha & t\ s & \text{Правило 2} \\
\lambda b.(t\ b) & =_\alpha & \lambda a.(t\ a) & \text{Правило 3} \\
\lambda a.\lambda b.(a\ b) & =_\alpha & \lambda b.\lambda a.(b\ a) & \text{Правило 3}
\end{array}$$

□

Бета-редукция

Интуиция: вызов функции.

λ-выражение	Python
$\lambda f.\lambda x.f\ x$	`def one(f,x): return f(x)`
$(\lambda x.x\ x)\ (\lambda x.x\ x)$	`(lambda x: x x) (lambda x: x x)`
	`def omega(x): return x(x); omega(omega)`

Определение

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

Определение

$A \to_\beta B$, *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* $P \to_\beta P'$ *и* $Q = Q'$, *либо* $P = P'$ *и* $Q \to_\beta Q'$;
3. $A \equiv (\lambda x.P)$, $B \equiv (\lambda x.P')$, *и* $P \to_\beta P'$.

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: $N \to_\beta Q$.

Пример

В нормальной форме:
$\lambda f.\lambda x.x\ (f\ (f\ \lambda g.x))$

Пример

Не в нормальной форме (редексы подчёркнуты):
$\lambda f.\lambda x.\underline{(\lambda g.x)\ (f\ (f\ x))}$
$(\underline{(\lambda x.x)\ (\lambda x.x)})\ (\underline{(\lambda x.x)\ (\lambda x.x)})$

Определение

$(\twoheadrightarrow_\beta)$ — *транзитивное и рефлексивное замыкание* $(\to_\beta)$.

Булевские значения

$T := \lambda x.\lambda y.x \; F := \lambda x.\lambda y.y$
Тогда: $Or := \lambda a.\lambda b.a \; T \; b$:
$Or \; F \; T = \underline{((\lambda a.\lambda b.a \; T \; b) \; F)} \; T \to_\beta (\lambda b.F \; T \; b) \; T \to_\beta F \; T \; T =$
$= (\lambda x.\lambda y.y) \; T \; T \to_\beta (\lambda y.y) \; T \to_\beta T$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

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

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

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

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:* $\Omega \to_\beta \Omega$

Определение

$(=_\beta)$ — *транзитивное, рефлексивное и симметричное замыкание* $(\to_\beta)$.

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 $R := Y\ N$:

$$Y\ N =_\beta (\lambda x.N\ (x\ x))\ (\lambda x.N\ (x\ x)) =_\beta N\ ((\lambda x.N\ (x\ x))\ (\lambda x.N\ (x\ x)))$$

$\square$

Интуиционистское И.В. (натуральный, естественный вывод)

- Формулы языка (секвенции) имеют вид: $\Gamma \vdash \alpha$. Правила вывода:
$$\frac{\text{посылка 1} \qquad \text{посылка 2} \qquad \dots}{\text{заключение}}\ (\text{аннотация})$$
- Аксиома:
$$\frac{}{\Gamma, \alpha \vdash \alpha}\ (\text{акс.})$$
- Правила введения связок:
$$\frac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta} \qquad \frac{\Gamma \vdash \alpha}{\Gamma \vdash \alpha \vee \beta}, \frac{\Gamma \vdash \beta}{\Gamma \vdash \alpha \vee \beta} \qquad \frac{\Gamma \vdash \alpha \qquad \Gamma \vdash \beta}{\Gamma \vdash \alpha \,\&\, \beta}$$
- Правила удаления связок:
$$\frac{\Gamma \vdash \alpha \quad \Gamma \vdash \alpha \to \beta}{\Gamma \vdash \beta} \qquad \frac{\Gamma \vdash \alpha \to \gamma \quad \Gamma \vdash \beta \to \gamma \quad \Gamma \vdash \alpha \vee \beta}{\Gamma \vdash \gamma}$$
$$\frac{\Gamma \vdash \alpha \,\&\, \beta}{\Gamma \vdash \alpha} \qquad \frac{\Gamma \vdash \alpha \,\&\, \beta}{\Gamma \vdash \beta} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \alpha}$$
- Пример доказательства:
$$\dfrac{\dfrac{\dfrac{}{A \,\&\, B \vdash A \,\&\, B}\ (\text{акс.})}{A \,\&\, B \vdash B}\ (\text{удал}\&) \qquad \dfrac{\dfrac{}{A \,\&\, B \vdash A \,\&\, B}\ (\text{акс.})}{A \,\&\, B \vdash A}\ (\text{удал}\&)}{A \,\&\, B \vdash B \,\&\, A}\ (\text{введ}\&)$$

Эквивалентность натурального и гильбертовского выводов

Определение

$$|\alpha|_\perp = \begin{cases} X, & \alpha \equiv X \\ |\sigma|_\perp \star |\tau|_\perp, & \alpha \equiv \sigma \star \tau \\ |\sigma|_\perp \to \perp, & \alpha \equiv \neg\sigma \end{cases} \qquad |\alpha|_\neg = \begin{cases} X, & \alpha \equiv X \\ |\sigma|_\neg \star |\tau|_\neg, & \alpha \equiv \sigma \star \tau \\ A \,\&\, \neg A, & \alpha \equiv \perp \end{cases}$$

Теорема

1. $\Gamma \vdash_n \alpha$ *тогда и только тогда, когда* $|\Gamma|_\neg \vdash_h |\alpha|_\neg$.
2. $\Gamma \vdash_h \alpha$ *тогда и только тогда, когда* $|\Gamma|_\perp \vdash_n |\alpha|_\perp$.

Доказательство.

Индукция по структуре $\square$

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

$$\frac{}{\Gamma, \varphi \vdash_{\to} \varphi} \qquad \frac{\Gamma, \varphi_{\to} \vdash \psi}{\Gamma \vdash_{\to} \varphi \to \psi} \qquad \frac{\Gamma \vdash_{\to} \varphi \qquad \Gamma \vdash_{\to} \varphi \to \psi}{\Gamma \vdash_{\to} \psi}$$

Теорема

Если $\Gamma \vdash \alpha$, *то* $\Gamma \vdash_{\to} \alpha$.

Доказательство.

Определим модель Крипке:

- миры — замкнутые множества формул: $\alpha \in \Gamma$ т.и.т.т. $\Gamma \vdash_{\to} \alpha$,
- порядок — $(\subseteq)$,
- $\Gamma \Vdash X$ т.и.т.т. $X \in \Gamma$.

Из корректности моделей Крипке следует, что что если $\Gamma \vdash \alpha$, то $\Gamma \Vdash \alpha$. Требуемое следует из того, что $\Gamma \Vdash \alpha$ влечёт $\Gamma \vdash_{\to} \alpha$. $\square$

$\Gamma \Vdash \alpha$ т.и.т.т. $\Gamma \vdash_{\to} \alpha$

Индукция по структуре α.

- $\alpha \equiv X$. Утверждение следует из определения;
- $\alpha \equiv \varphi \to \psi$.
  - Пусть $\Gamma \Vdash \varphi \to \psi$. То есть, $\Gamma \subseteq \Delta$ и $\Delta \Vdash \varphi$ влечёт $\Delta \Vdash \psi$. Возьмём Δ как замыкание $\Gamma \cup \{\varphi\}$. Значит, $\Gamma \vdash_{\to} \varphi$ и, по индукционному предположению, $\Delta \Vdash \varphi$. Тогда $\Delta \Vdash \psi$. По индукционному предположению, $\Delta \vdash_{\to} \psi$. То есть, $\Gamma, \varphi \vdash_{\to} \psi$, откуда

$$\frac{\Gamma, \alpha \vdash \beta}{\Gamma \vdash \alpha \to \beta}$$

  - Пусть $\Gamma \vdash_{\to} \varphi \to \psi$. Проверим $\Gamma \Vdash \varphi \to \psi$. Пусть $\Gamma \subseteq \Delta$ и пусть $\Delta \Vdash \varphi$. По индукционному предположению, $\varphi \in \Delta$. То есть, $\Delta \vdash_{\to} \varphi$ и $\Delta \vdash_{\to} \varphi \to \psi$. Тогда

$$\frac{\Delta \vdash_{\to} \varphi \quad \Delta \vdash_{\to} \varphi \to \psi}{\Delta \vdash_{\to} \psi}$$

    По индукционному предположению, $\Delta \Vdash \psi$, отчего $\Gamma \Vdash \varphi \to \psi$.

Просто-типизированное лямбда-исчисление

Определение

Просто-типизированное лямбда-исчисление (по Карри). Типы: $\tau ::= \alpha | (\tau \to \tau)$*.*
Язык: $\Gamma \vdash A : \varphi$

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

Пример: тип чёрчевских нумералов

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

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash x : \alpha}\,Ax \qquad \dfrac{}{\Gamma \vdash f : \alpha \to \alpha}\,Ax}{\Gamma \vdash f\ x : \alpha}\,App \qquad \dfrac{}{\Gamma \vdash f : \alpha \to \alpha}\,Ax}{\{f : \alpha \to \alpha, x : \alpha\}\ \Gamma \vdash f\ (f\ x) : \alpha}\,App}{\dfrac{f : \alpha \to \alpha \vdash \lambda x.f\ (f\ x) : (\alpha \to \alpha)}{\vdash \lambda f.\lambda x.f\ (f\ x) : (\alpha \to \alpha) \to (\alpha \to \alpha)}\,\lambda}\,\lambda$$

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;* $\mathtt{failwith/raise/throw} : \alpha \to \perp$; $\neg\varphi \equiv \varphi \to \perp$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Phi \vdash a : \alpha}\,Ax \quad \dfrac{}{\Phi \vdash f : \alpha \to \beta}\,Ax}{\Phi \vdash f\ a : \beta}\,App \quad \dfrac{}{\Phi \vdash n : \beta \to \perp}\,Ax}{f : \alpha \to \beta, n : \beta \to \perp, a : \alpha \vdash n\ (f\ a) : \perp}\,App}{f : \alpha \to \beta, n : \beta \to \perp \vdash \lambda a^{\alpha}.n\ (f\ a) : \neg\alpha}\,\lambda}{f : \alpha \to \beta \vdash \lambda n^{\beta\to\perp}.\lambda a^{\alpha}.n\ (f\ a) : \neg\beta \to \neg\alpha}\,\lambda}{\lambda f^{\alpha\to\beta}.\lambda n^{\beta\to\perp}.\lambda a^{\alpha}.n\ (f\ a) : (\alpha \to \beta) \to (\neg\beta \to \neg\alpha)}\,\lambda$$

Снятие двойного отрицания: $((\alpha \to \perp) \to \perp) \to \alpha$, то есть $\lambda f^{(\alpha\to\perp)\to\perp}.? : \alpha$.
f угадывает, что передать $x : \alpha \to \perp$. Тогда надо по f угадать, что передать x.

Исчисление по Чёрчу и по Карри

Определение

Просто-типизированное лямбда-исчисление по Карри.

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

Просто-типизированное лямбда-исчисление по Чёрчу.

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

Пример

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

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 $N =_\beta C$

Пример

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

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

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	$\{x \vert P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \vert \forall t.t > 0 \rightarrow P(t)\}$

- 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 $[\![p(p)]\!]$ при $p(x) = x(x) \rightarrow \perp$?
  Нужна точная формализация.
- Самый простой вариант: переменные второго порядка — только булевские пропозициональные переменные.

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

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- Что такое $T : \forall x.x \to x$?
  `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`?
- $\texttt{sizeof(arr)} = n \cdot \texttt{sizeof(int)}$
- $arr = \Pi n^{\texttt{int}}.\texttt{int}[n]$
- Аналогично, `printf(const char*, ...)` — капитуляция.
- Есть языки, где тип выписывается (например, Идрис).

Прямолинейное: доказательства в коде

- `Div2: (l: int) -> (even l) -> int`
- `even l` — что это?
- $$even(x) ::= \begin{cases} EZ, & x = 0 \\ EP(even(y)), & x = y'' \end{cases}$$
- `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: $\mathtt{Nat} ::= 0 | \mathtt{suc\ Nat}$,

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

```
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))
```

Что ещё

- Гомотопическая теория типов...
- Метод резолюций и рядом — Prolog, SMT-солверы,...
- Можно пытаться совмещать (F*, ...)