

Изоморфизм Карри-Ховарда

Лямбда-исчисление

Синтаксис:

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

Лямбда-исчисление

Синтаксис:

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

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>

Лямбда-исчисление

Синтаксис:

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

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>

Исчисление: изучаем преобразования формул — редукция.

$$(\lambda x. A) \ B \rightarrow_{\beta} A[x := B]$$

Пример

$$\begin{aligned} (\lambda x. x \ x) \ (\lambda n. n) &\rightarrow_{\beta} (\lambda n. n) \ (\lambda n. n) \rightarrow_{\beta} \lambda n. n \\ (\lambda x. x \ x) \ (\lambda x. x \ x) &\rightarrow_{\beta} (\lambda x. x \ x) \ (\lambda x. x \ x) \end{aligned}$$

Чёрчевские нумералы

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

Определение

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

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 $\bar{n} = f^{(n)}(x)$

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 $\bar{n} = f^{(n)}(x)$

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 $\bar{n} = f^{(n)}(x)$

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 $\bar{n} = f^{(n)}(x)$

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

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:

$\Gamma \vdash A : \varphi$

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

Изоморфизм Карри-Ховарда

λ -исчисление	исчисление высказываний
Выражение	доказательство
Тип выражения	высказывание
Тип функции	импликация
Упорядоченная пара	конъюнкция
Алгебраический тип	дизъюнкция

Изоморфизм Карри-Ховарда: отрицание

Определение

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

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

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

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

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: $\forall a. \forall b. a \rightarrow b \rightarrow a$
- ▶ Не так просто! $W(x) := \forall p. \forall x. p(x)$, рассматриваем ли $\llbracket \forall x. p(x) \rrbracket^{p:=W(x)}$? (импредикативность)

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: $\forall a. \forall b. a \rightarrow b \rightarrow a$
- ▶ Не так просто! $W(x) := \forall p. \forall x. p(x)$, рассматриваем ли $\llbracket \forall x. p(x) \rrbracket^{p:=W(x)}$? (импредикативность)
- ▶ Простой вариант: только пропозициональные переменные. $\forall p. p \rightarrow p$

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: $\forall a. \forall b. a \rightarrow b \rightarrow a$
- ▶ Не так просто! $W(x) := \forall p. \forall x. p(x)$, рассматриваем ли $\llbracket \forall x. p(x) \rrbracket^{p:=W(x)}$? (импредикативность)
- ▶ Простой вариант: только пропозициональные переменные. $\forall p. p \rightarrow p$

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

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`?
- ▶ `sizeof(arr) = n · sizeof(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 = $\prod n^{\text{int}}$.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: $\text{Nat} ::= 0 \mid \text{suc } \text{Nat}$,

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

```
func pmap A B :
```

```
Type (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^* , ...)