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
$\lambda f.\lambda x.f\ x$	`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)`

Лямбда-исчисление

Синтаксис:

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

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)`

Исчисление: изучаем преобразования формул — редукция.

$$(\lambda x.A)\ B \to_\beta A[x := B]$$

Пример

$(\lambda x.x\ x)\ (\lambda n.n) \to_\beta (\lambda n.n)\ (\lambda n.n) \to_\beta \lambda n.n$
$(\lambda x.x\ x)\ (\lambda x.x\ x) \to_\beta (\lambda x.x\ x)\ (\lambda x.x\ x)$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

$$(\lambda n.\lambda f.\lambda x.n\ f\ (f\ x))\ \overline{0} = (\lambda n.\lambda f.\lambda x.n\ f\ (f\ x))\ (\lambda f'.\lambda x'.x') \to_\beta$$

Чёрчевские нумералы

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

Определение

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

Пример

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

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

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

Чёрчевские нумералы

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

Определение

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

Пример

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

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

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

Чёрчевские нумералы

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

Определение

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

Пример

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

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

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

Чёрчевские нумералы

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

Определение

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

Пример

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

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

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

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

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

Просто-типизированное лямбда-исчисление.

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

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

Просто-типизированное лямбда-исчисление

Определение

Импликационный фрагмент интуиционистской логики:

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

Определение

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

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

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 ($\bot$) *— необитаемый тип;* $\mathtt{failwith/raise/throw} : \alpha \to \bot$; $\neg\varphi \equiv \varphi \to \bot$

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

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

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

Исчисление второго порядка

- Напомним о порядках:

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

Исчисление второго порядка

- Напомним о порядках:

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

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

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 $[\![\forall x.p(x)]\!]^{p:=W(x)}$? (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 $[\![\forall x.p(x)]\!]^{p:=W(x)}$? (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	$\{x \mid P(x)\}$
2 (И.П. 2)	Множества множеств	$\{P \mid \forall t.t > 0 \rightarrow P(t)\}$

- 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 $[\![\forall x.p(x)]\!]^{p:=W(x)}$? (импредикативность)
- Простой вариант: только пропозициональные переменные. $\forall p.p \rightarrow p$

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

Изоморфизм Карри-Ховарда для логики второго порядка

Типы и значения, зависящие от типов.

- Что такое $T : \forall x.x \to x$?

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); }
```

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 $T : \forall x.x \to x$?
  `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`?

Зависимые типы

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

Зависимые типы

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

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.

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`

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

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)))))`

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: $\texttt{Nat} ::= 0 | \texttt{suc Nat}$,

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

```
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.

Логика	λ-*исчисление*	*Топология*
Утверждение	*Тип*	*Пространство*
Доказательство	*Значение*	*Точка в пространстве*
Предикат $(=)$	*Зависимый тип* $(=)$	*Путь между точками*

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 $(a = b) = \{\langle a, b\rangle | a < 10 \,\&\, b < 10\}$? 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 $(a = b) = \{\langle a, b\rangle | a < 10 \,\&\, b < 10\}$? 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 $x^2 = y^2$. $\square$

Что ещё

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