

Содержание

1	Лямбда-исчисление	2
2	Типизированное лямбда-исчисление	3

Литература:

- ???

1 Лямбда-исчисление

Безтиповое лямбда-исчисление: термы, α -конверсия, β , η -редукция.

Определение 1. $V = \{v_0, \dots, v_1\} \sim \mathbb{N}$ — алфавит. Определим λ -терм:

- $x \in V \Rightarrow x \in \Lambda$
- $M, N \in \Lambda \rightarrow (MN) \in \Lambda$
- $x \in V, M \in \Lambda \Rightarrow (\lambda x.M) \in \Lambda$

Замечание. Примечания: аппликация (применение) левоассоциативно, абстракция — наоборот.

Конверсии бывают некорректные. Можно сформулировать правило, определяющее корректную подстановку. Но мы будем далее считать, что в любом контексте ни одна переменная не имеет одновременно свободных и связанных вхождений. Для этого примерним автоматические переименование в не встречающиеся: $\lambda x.M \rightarrow_\alpha \lambda y.(M[x := y]), y \notin V(M)$.

Определение 2. Определим абстрактные редукции. $R \subset \Lambda \times \Lambda$ — редукция. Условия совместимости:

1. $MRN \Rightarrow M \rightarrow_R N$
2. $M \rightarrow_R N \Rightarrow LM \rightarrow_R LN, ML \rightarrow_R NL, \lambda x.M \Rightarrow_R \lambda x.N$

Также замкнём нашу редукцию: \rightarrow_R и построим соответствующее отношение эквивалентности $=_R$.

Определение 3. Терм R -нормален, если $\nexists NM \rightarrow_R N$. Терм N является нормальной формой терма M , если он R -нормальный и $M =_R N$.

Терм нормализуемый, если у него есть нормальная форма и сильно нормализуемый, если любая цепочка редукций обрывается (на R -нормальной форме).

Лемма 1 (О ромбе). Пусть $R \in \{\beta, \beta\eta\}$. Тогда если $M \twoheadrightarrow N_1, M \twoheadrightarrow N_2$, то $\exists L : N_1 \twoheadrightarrow L, N_2 \twoheadrightarrow L$.

Теорема 1 (Чёрча-Россера). Если $M =_R N \Rightarrow \exists L : M \twoheadrightarrow L, N \twoheadrightarrow L$.

Следствие. Нормальная форма (если существует) единственна.

Нормальная форма существует не всегда. Не каждая цепочка преобразований ведёт к нормальной форме.

Утверждение 1. На базе лямбда-термов можно построить логику высказываний и примитивно-рекурсивную арифметику, нужным образом закодировав нужные функции.

Комбинатор неподвижной точки: $Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ обладает интересным свойством $F(YF) = YF$.

Пример 1. Хотим найти комбинатор M , заданный условием $MXY = XM(MM)Y(MMM)$. Рассмотрим $F = \lambda txy.xt(mt)y(mmt)$ и положим $M = YF$. Тогда все получится.

Можно находить и совместные неподвижные точки: $X = FXY, Y = GXY$. Возьмём $\Phi = \lambda p.\langle F(\pi_1 p)(\pi_2 p), G(\pi_1 p)(\pi_2 p) \rangle$. Тогда $Z = Y\Phi$ неподвижная точка, значит $Z = \Phi Z = \langle F(\pi_1 Z)(\pi_2 Z), G(\pi_1 Z)(\pi_2 Z) \rangle$. Тогда остаётся взять $X = \pi_1 Z, Y = \pi_2 Z$.

Используя эту машинерию, можно построить все примитивно-рекурсивные функции, а также другие виды рекурсии, а также и минимизацию.

Теорема 2. Следующие условия равносильны:

- f представимо лямбда-термом
- f вычислимо на машинах Тьюринга
- f частично-рекурсивна (примитивная рекурсия + минимизация, возможны не тотальные функции)

2 Типизированное лямбда-исчисление

Типы: пропозициональные формулы с импликацией.

Определение 4. Счетный алфавит типов

$$A, B \in Tp, (A \rightarrow B) \in Tp$$

$$A_1 \rightarrow (A_2 \rightarrow \dots \rightarrow (A_n \rightarrow B)) = A_1 \rightarrow \dots \rightarrow A_n \rightarrow B$$

Определение 5. $PdTm$ — псевдотермы.

$$X \in Var \Rightarrow X \in PdTm$$

$$M, N \in PdTm \Rightarrow (MN) \in PdTm$$

$$x \in Var, A \in Tp, M \in PdTm \Rightarrow (\lambda x^A.M) \in PdTm.$$

Псевдотермы, потому что не все из них будут считаться корректными.

Контекст $\Gamma = \{x_1 : A_1, \dots, x_n : A_n\}$.

Вывод типов: $\vdash \subset Context \times Statement$. Правила

- $\Gamma, x : A \vdash x : A$
- $\frac{\Gamma \vdash M : A \rightarrow B, \Gamma \vdash N : A}{\Gamma \vdash MN : B}$
- $\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash (\lambda x^A.M) : A \rightarrow B}$

Каждое выведение может быть получено единственным образом (лемма об обращении).

Лемма 2 (О подстановке).

- $\Gamma \vdash M : A \Rightarrow \Gamma[\alpha := B] \vdash M : A[\alpha := B]$
- $\frac{\Gamma, x:A \vdash M:B, \Gamma \vdash N:A}{\Gamma \vdash M[x:=N]:B}$

Лемма 3 (subject reduction). $\Gamma \vdash M : A, M \rightarrow_{\beta\eta} N \Rightarrow \Gamma \vdash N : A$

Утверждение 2. $\Gamma \vdash M : A, \Gamma \vdash M : B \Rightarrow A = B$

Доказательство. Разберём один из случаев: $\Gamma \vdash \lambda x^C.N : A \Rightarrow A = C \rightarrow D, \Gamma, x : C \vdash N : D$. Если $B = C \rightarrow E$, то аналогично $\Gamma, x : C \vdash N : E$. По индукции $D = E$, значит $A = B$. \square

Лемма 4 (о ромбе). Если $\Gamma \vdash M, N_1, N_2 : A$, причём $M \rightarrow N_1, M \rightarrow N_2$, тогда $\exists L : \Gamma \vdash L : A, N_1 \rightarrow L, N_2 \rightarrow L$.

Теорема 3 (Чёрча-Россера). Пусть $\Gamma \vdash M, N : A, M =_{\beta\eta} N$, тогда $\exists L : \Gamma \vdash L : A, M \rightarrow L, N \rightarrow L$.

Более того, верно, что у любого терма нормальная форма существует, единственна и любая последовательность редукций конечна. В частности, терму $\omega_2 = \lambda x.xx$ нельзя приписать никакой тип.

Пример 2. $\lambda x^A y^B.x : A \rightarrow (B \rightarrow A)$.

Берём контекст $\Gamma = \{x : A, y : B\}$. Выводим:

$$\begin{aligned} \Gamma &\vdash x : A \\ x : A &\vdash \lambda y^B.x : B \rightarrow A \\ &\vdash \lambda x^A y^B.x : A \rightarrow B \rightarrow A \end{aligned}$$

Теперь можно определить «корректные» псевдотермы: $\Lambda_{\rightarrow}^{\Gamma}(A) = \{M \in PdTm \mid \Gamma \vdash M : A\}$.

$$\begin{aligned} \Lambda_{\rightarrow}(A) &= \bigcup_{\Gamma} \Lambda_{\rightarrow}^{\Gamma}(A) \\ \Lambda_{\rightarrow} &= \bigcup_{A \in Tb} \Lambda_{\rightarrow}(A) \end{aligned}$$

Теорема 4 (сильная нормализуемость). Если $M \in \Lambda_{\rightarrow}$, то любая цепочка $\beta\eta$ -редукций конечна и может быть продолжена до нормальной формы.

Что можно вычислить в типизированном лямбда-исчислении?

$Int = (\alpha_0 \rightarrow \alpha_0) \rightarrow (\alpha_0 \rightarrow \alpha_0)$ — целочисленный тип, взятый из привычных нумералов Чёрча: $\underline{n} = \lambda f^{\alpha_0 \rightarrow \alpha_0} x_0^{\alpha_0}. f^n x$.

Утверждение 3. $\forall n$ выполнено $\underline{n} \in \Lambda_{\rightarrow}(Int). \vdash \lambda f^{\alpha_0 \rightarrow \alpha_0} x_0^{\alpha_0}. f^n x$.

Аналогично перенесём понятие представимости функций $f : \mathbb{N}^k \rightarrow \mathbb{N}$.

Заметим, что $Sc = \lambda k f x. f(k f x)$ типизуем типом $Int \rightarrow Int$. Точно также типизируются сложением, умножением. Но с возведением в степень проблемы: $E = \lambda k l. l k$ не типизируется.

Можно показать, что при таком кодировании чисел получится представить только эти операции, проекторы, проверку на 0 и все их композиции.

Определение 6.

- $PVar = \{p_0, \dots, p_1\} \sim \mathbb{N}$
- $p_i \in PVar \Rightarrow p_i \in IFm$
- $A, B \in IFm \Rightarrow (A \rightarrow B) \in IFm$

Натуральный вывод: помеченное дерево с правилами элиминации импликации (modus ponens) и добавлением импликации. Такой вывод двойственен выводу типа.

Интуиционистская импликация: f «доказательство» импликации $A \rightarrow B$, если f вычислима и преобразует любое доказательство A в доказательство B . В этом смысле типизированные функции интерпретируются как описания доказательств (изоморфизм Карри-Ховарда).