

Системы типизации лямбда-исчисления

Лекция 2. λ -исчисление: программирование и λ -определимость

Денис Москвин

27.02.2011

CS Club при ПОМИ РАН

Программирование на языке λ -исчисления

λ -исчисление — полноценный язык программирования? Нет, но если его рассматривать в этом качестве, то:

- удобно работать с замкнутыми термами;
- в чистом λ -исчислении значения — это λ -абстракции;
- β -преобразования моделируют процесс вычисления.

Булевы значения (1)

Булевы значения можно определить так:

$$\text{TRU} \equiv \lambda t f. t$$
$$\text{FLS} \equiv \lambda t f. f$$

Стандартные булевы операции:

$$\text{IIF} \equiv \lambda b x y. b x y$$
$$\text{NOT} \equiv ??? \text{ (чуть позже)}$$
$$\text{AND} \equiv \lambda x y. x y \text{ FLS}$$
$$\text{OR} \equiv ??? \text{ (домашнее задание)}$$

Булевы значения (2)

Проверим, что ожидаемые свойства условного выражения $\text{IIF} \equiv \lambda b x y. b x y$ выполняются:

$$\begin{aligned} \text{IIF TRU } v w &\equiv (\lambda b x y. b x y) (\lambda t f. t) v w \\ &= (\lambda t f. t) v w \\ &= (\lambda f. v) w \\ &= v; \end{aligned}$$

$$\begin{aligned} \text{IIF FLS } v w &\equiv (\lambda b x y. b x y) (\lambda t f. f) v w \\ &= (\lambda t f. f) v w \\ &= (\lambda f. f) w \\ &= w. \end{aligned}$$

Булевы значения (3)

Проверим, что ожидаемые свойства логического оператора «И» $\text{AND} \equiv \lambda x y. x y \text{ FLS}$ выполняются:

$$\begin{aligned} \text{AND TRU } w &\equiv (\lambda x y. x y \text{ FLS}) \text{TRU } w \\ &= \text{TRU } w \text{ FLS} \\ &= w; \end{aligned}$$

$$\begin{aligned} \text{AND FLS } w &\equiv (\lambda x y. x y \text{ FLS}) \text{FLS } w \\ &= \text{FLS } w \text{ FLS} \\ &= \text{FLS}. \end{aligned}$$

Булевы значения (4)

Отрицание (версия I):

$$\text{NOT} \equiv \lambda b. \text{IIF } b \text{ FLS TRU} = \lambda b. b \text{ FLS TRU} \equiv \lambda b. b (\lambda t f. f) (\lambda t f. t)$$

Отрицание (версия II):

$$\text{NOT} \equiv \lambda b t f. b f t$$

Это разные термы! Но на булевых значениях одинаковые.

Типизация даст гарантию, что ничто, кроме булева значения, не пройдёт.

Пары (1)

Пару (двухэлементный кортеж) можно определить так:

$$\text{PAIR} \equiv \lambda x y f. f x y$$

При этом конкретная пара такова:

$$\text{PAIR } a \ b = \lambda f. f a b$$

Стандартные операции для пары (проекции):

$$\text{FST} \equiv \lambda p. p \ \text{TRU}$$

$$\text{SND} \equiv \lambda p. p \ \text{FLS}$$

Пары (2)

Проверим, что ожидаемые свойства проекций $FST \equiv \lambda p. p \text{ TRU}$ и $SND \equiv \lambda p. p \text{ FLS}$ выполняются:

$$\begin{aligned} FST (\text{PAIR } a \ b) &= \text{PAIR } a \ b \ \text{TRU} \\ &\equiv (\lambda x y f. f x y) \ a \ b \ \text{TRU} \\ &= \text{TRU } a \ b \\ &= a; \\ SND (\text{PAIR } a \ b) &= \text{PAIR } a \ b \ \text{FLS} \\ &\equiv (\lambda x y f. f x y) \ a \ b \ \text{FLS} \\ &= \text{FLS } a \ b \\ &= b. \end{aligned}$$

Числа (1)

Числа (нумералы) Чёрча:

$$\begin{aligned}\bar{0} &\equiv \lambda s z. z \\ \bar{1} &\equiv \lambda s z. s z \\ \bar{2} &\equiv \lambda s z. s (s z) \\ \bar{3} &\equiv \lambda s z. s (s (s z)) \\ \bar{4} &\equiv \lambda s z. s (s (s (s z))) \\ &\dots\end{aligned}$$

Чертю сверху будем опускать, если это не будет приводить к путанице.

Наблюдение: $\bar{0} \equiv \text{FLS}$.

Числа (2)

Более формально:

Выражение $F^n(X)$, где $n \in \mathbb{N}$, а $F, X \in \Lambda$, определим индуктивно:

$$\begin{aligned}F^0(X) &\equiv X; \\F^{n+1}(X) &\equiv F(F^n(X)).\end{aligned}$$

Тогда n -ое число Чёрча определяется так:

$$\bar{n} \equiv \lambda s z. s^n(z).$$

Числа (3)

Проверка числа на ноль ($\bar{0} \equiv \lambda s z. z$):

$$\text{ISZRO} \equiv \lambda n. n \text{ ?? TRU}$$

Вместо ?? нужна функция, которая всегда возвращает FLS:

$$\text{ISZRO} \equiv \lambda n. n (\lambda x. \text{FLS}) \text{ TRU}$$

Альтернативное определение, конструирующее булево значение «на лету»:

$$\text{ISZRO2} \equiv \lambda n t f. n (\lambda x. f) t$$

Числа (4)

Функция следования для чисел

$$\text{SUCC} \equiv \lambda n s z. s (n s z)$$

Действительно,

$$\text{SUCC } \bar{n} \equiv \lambda s z. s (\bar{n} s z) \equiv \lambda s z. s(s^n(z)) \equiv \lambda s z. s^{n+1}(z) \equiv \overline{n+1}.$$

Альтернативное определение

$$\text{SUCC2} \equiv \lambda n s z. n s (s z)$$

«надстраивает» n -кратное применение s не над нулём (z), а над единицей ($s z$).

Числа: сложение (1)

Сложение чисел Чёрча основывается на аналогичной идее

$$\text{PLUS} \equiv \lambda m n s z. m s (n s z)$$

Здесь m -кратное применение s «надстраивается» над его n -кратным применением. Например,

$$\begin{aligned} \text{PLUS } 3 \ 2 &= \lambda s z. 3 \ s \ (2 \ s \ z) = \lambda s z. 3 \ s \ (s \ (s \ z)) \\ &= \lambda s z. s \ (s \ (s \ (s \ z))) \equiv 5 \end{aligned}$$

Числа: сложение (2)

Лемма. $\overline{m} s (\overline{n} s z) = \overline{m+n} s z$.

Индукция по m .

База ($m = 0$): $\overline{0} s (\overline{n} s z) = \overline{n} s z = \overline{0+n} s z$.

ИН: Пусть верно для m , то есть $s^m (s^n (z)) = s^{n+m}(z)$.

Покажем, что верно и для $m+1$:

$$\begin{aligned} \overline{m+1} s (\overline{n} s z) &= s^{m+1} (s^n (z)) = s (s^m (s^n (z))) \\ &\stackrel{\text{ИН}}{=} s (s^{m+n} (z)) = s^{(m+n)+1} (z) \\ &= s^{(m+1)+n} (z) = \overline{(m+1)+n} s z \quad \blacksquare \end{aligned}$$

То есть PLUS $\equiv \lambda m n s z. m s (n s z)$ действительно складывает.

Числа: умножение (1)

Благодаря каррированию PLUS n прибавляет n к любому переданному аргументу. Поэтому умножение может быть задано так

$$\text{MULT1} \equiv \lambda m n. m (\text{PLUS } n) 0$$

Например,

$$\begin{aligned} \text{MULT1 } 3 \ 2 &= 3 (\text{PLUS } 2) 0 = (\lambda s z. (s (s (s z)))) (\text{PLUS } 2) 0 \\ &= (\text{PLUS } 2) ((\text{PLUS } 2) ((\text{PLUS } 2) 0)) = 6 \end{aligned}$$

Альтернативные определения

$$\text{MULT2} \equiv \lambda m n s z. m (n s) z$$

$$\text{MULT} \equiv \lambda m n s. m (n s)$$

Числа: умножение (2)

Проверим $MULT2 \equiv \lambda m n s z. m (n s) z$:

$$\begin{aligned}MULT2\ 3\ 2 &= \lambda s z. 3 (2 s) z = \lambda s z. (2 s) ((2 s) (2 s z)) \\ &= \lambda s z. (2 s) ((2 s) (s s z)) = 6\end{aligned}$$

Версия $MULT \equiv \lambda m n s. m (n s)$ получается из $MULT2$ с помощью η -преобразования.

Числа: умножение (3)

Лемма. $\overline{m} (\overline{n} s) z = \overline{m \cdot n} s z$.

Индукция по m .

База ($m = 0$): $\overline{0} (\overline{n} s) z = z = \overline{0 \cdot n} s z$.

ИН: Пусть верно для m , то есть $(\overline{n} s)^m (z) = s^{m \cdot n} (z)$.

Покажем, что верно и для $m + 1$:

$$\begin{aligned} \overline{m+1} (\overline{n} s) z &= (\overline{n} s)^{m+1} (z) = \overline{n} s ((\overline{n} s)^m (z)) \\ &\stackrel{IH}{=} \overline{n} s (s^{m \cdot n} (z)) = s^n (s^{m \cdot n} (z)) \\ &= s^{n+m \cdot n} (z) = s^{(m+1)n} (z) = \overline{(m+1)n} s z \quad \blacksquare \end{aligned}$$

То есть $MULT2 \equiv \lambda m n s z. m (n s) z$ действительно умножает.

Числа: предшествование

Вспомогательные функции

$$ZP \equiv \text{PAIR } \bar{0} \bar{0}$$

$$SP \equiv \lambda p. \text{PAIR } (\text{SND } p) (\text{SUCC } (\text{SND } p))$$

Вторая работает так

$$SP (\text{PAIR } \bar{i} \bar{j}) = \text{PAIR } \bar{j} \overline{\bar{j} + 1}$$

$$SP^0 (ZP) = \text{PAIR } \bar{0} \bar{0}$$

$$SP^m (ZP) = \text{PAIR } \overline{\bar{m} - 1} \bar{m}$$

(здесь $m > 0$). Тогда **функция предшествования**:

$$\text{PRED} = \lambda m. \text{FST } (m \text{ SP } ZP)$$

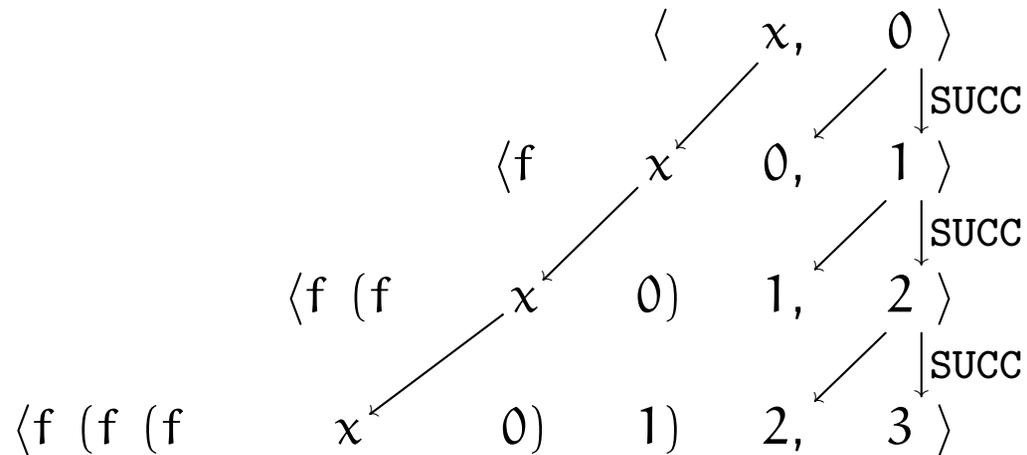
А какая, кстати, у неё временная сложность?

А что нужно поменять, чтобы вышел факториал?

Числа: примитивная рекурсия

Обобщим предыдущую схему

$$\begin{aligned} \text{XZ} &\equiv \lambda x. \text{PAIR } x \ \bar{0} \\ \text{FS} &\equiv \lambda f p. \text{PAIR } (f \ (\text{FST } p)) \ (\text{SND } p)) \ (\text{SUCC } (\text{SND } p)) \\ \text{REC} &\equiv \lambda m f x. \text{FST } (m \ (\text{FS } f) \ (\text{XZ } x)) \end{aligned}$$



В частности, $\text{PRED} = \lambda m. \text{REC } m \ (\lambda x y. y) \ \bar{0}$. А факториал?

Списки

Конструкторы списков:

$$\begin{aligned} \text{NIL} &\equiv \lambda c n. n \\ \text{CONS} &\equiv \lambda e l c n. c e (l c n) \end{aligned}$$

Например,

$$[] = \text{NIL} = \lambda c n. n$$
$$[5, 3, 2] = \text{CONS } 5 (\text{CONS } 3 (\text{CONS } 2 \text{NIL})) = \lambda c n. c 5 (c 3 (c 2 n))$$

Стандартные функции:

$$\text{EMPTY} \equiv \lambda l. l (\lambda h t. \text{FLS}) \text{TRU}$$
$$\text{HEAD} \equiv ?? \text{ домашнее задание}$$
$$\text{TAIL} \equiv ?? \text{ домашнее задание **}$$

λ -определимость (1)

- Числовая функция — это отображение $f : \mathbb{N}^p \rightarrow \mathbb{N}$.
- Числовая функция f называется λ -определимой, если для некоторого комбинатора F равенство

$$F \overline{n_1} \dots \overline{n_p} = \overline{f(n_1, \dots, n_p)}$$

выполняется для всех $n_1, \dots, n_p \in \mathbb{N}$. При этом f называют λ -определенной посредством F .

λ -определимость (2)

Клини (Kleene) 1936: все рекурсивные функции λ -определимы и наоборот.

Замечание. Под рекурсивными здесь, конечно, понимаются частичные рекурсивные функции ($\bar{0}$, succ , проекции и оператор минимизации) [LCWT 2.2, ВФ 11]

Понятия рекурсивности, λ -определенности и вычислимости по Тьюрингу эквивалентны.

Бестиповое λ -исчисление полно по Тьюрингу.

Назрешимость λ -исчисления

Пусть $\mathcal{A} \subseteq \Lambda$.

\mathcal{A} называется *замкнутым относительно $=$* , если

$$M \in \mathcal{A}, \lambda \vdash M = N \Rightarrow N \in \mathcal{A}.$$

\mathcal{A} называется *нетривиальным*, если $\mathcal{A} \neq \emptyset$ и $\mathcal{A} \neq \Lambda$.

Теорема. Пусть $\mathcal{A} \subseteq \Lambda$ является нетривиальным и замкнутым относительно $=$. Тогда \mathcal{A} не является рекурсивным.
[LCWT 2.2]

Следствие. Множество $\{M \mid M \in \Lambda, M = \text{TRU}\}$ не является рекурсивным.

Бестиповое λ -исчисление не является разрешимой теорией.

Домашнее задание

Закодируйте в бестиповом λ -исчислении:

- логическую операцию OR;
- пятиэлементный кортеж TUPLE5 и функции, осуществляющие проекции PRJ1, ..., PRJ5;
- стандартные функции для списков;
- функции вычитания, проверки на равенство и возведения в степень для чисел Чёрча.

В последнем случае приведите индуктивное доказательство.

Литература (1)

TAPL гл. 5

Benjamin C. Pierce, Types and Programming Languages, MIT Press, 2002

LCWT гл. 2.2

Henk Barendregt, Lambda calculi with types, Handbook of logic in computer science (vol. 2), Oxford University Press, 1993

Литература (2)

ВФ гл. 11

Н.К. Верещагин, А. Шень, Вычислимые функции
М:МЦНМО 2008

I2FP гл. 3

John Harrison, Introduction to Functional Programming