Системы типизации лямбда-исчисления

Лекция 8. Полиморфная система: связь с логикой и кодирование

Денис Москвин

27.03.2011

CS Club при ПОМИ РАН

Интуиционистская пропозициональная логика второго порядка PROP2

Существует изоморфизм между $\lambda 2$ и логической системой PROP2.

Типы в $\lambda 2$ — формулы в PROP2.
Термы в $\lambda 2$ — доказательства в PROP2.

PROP2 — конструктивная система; в ней, например, закон Пирса

$$\forall \alpha\, \beta.\, ((\alpha \to \beta) \to \alpha) \to \alpha$$

невыводим.

PROP2: $\to$

Импликация унаследована от PROP:

	Правило удаления $\to$	Правило введения $\to$
PROP2	$\dfrac{\sigma \to \tau \quad \sigma}{\tau}$	$\dfrac{\begin{array}{c}[\sigma] \\ \vdots \\ \tau\end{array}}{\sigma \to \tau}$
$\lambda 2$	$\dfrac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\,N : \tau}$	$\dfrac{\Gamma, x{:}\sigma \vdash M : \tau}{\Gamma \vdash \lambda x^{\sigma}.\, M : \sigma \to \tau}$

PROP2: $\forall$

Правила введения и удаления квантора $\forall$ соответствуют универсальным Λ-абстракции и применению:

	Правило удаления $\forall$	Правило введения $\forall$
PROP2	$\dfrac{\forall\alpha.\,\sigma}{\sigma[\alpha := \tau]}$	$\dfrac{\begin{array}{c}\tau_1 \ldots \tau_n \\ \vdots \\ \sigma\end{array}}{\forall\alpha.\,\sigma}, \quad \alpha \notin FV(\tau_1, \ldots, \tau_n)$
$\lambda 2$	$\dfrac{\Gamma \vdash M : \forall\alpha.\,\sigma \quad \Gamma \vdash \tau : *}{\Gamma \vdash M\,\tau : \sigma[\alpha := \tau]}$	$\dfrac{\Gamma, \alpha : * \vdash M : \sigma}{\Gamma \vdash \Lambda\alpha.\,M : \forall\alpha.\,\sigma}$

Стандартные логические связки

Система $\lambda 2$ позволяет определить стандартные логические связки:

$$
\begin{aligned}
\bot &\equiv \forall\alpha\,.\,\alpha \\
\neg\sigma &\equiv \sigma\to\bot \\
\sigma\wedge\tau &\equiv \forall\alpha.\,(\sigma\to\tau\to\alpha)\to\alpha \\
\sigma\vee\tau &\equiv \forall\alpha.\,(\sigma\to\alpha)\to(\tau\to\alpha)\to\alpha \\
\exists\alpha.\,\sigma &\equiv \forall\beta\,.\,(\forall\alpha.\,\sigma\to\beta)\to\beta
\end{aligned}
$$

При этом оказываются возможными все стандартные (конструктивные) выводы.

Свойства $\bot$

Напомним, что $\bot \equiv \forall\alpha.\,\alpha$.
Терм этого типа позволяет породить терм любого типа

$$\begin{array}{rcl} \sigma{:}*, x{:}\bot & \vdash & (x\,\sigma) : \sigma \\ \sigma{:}* & \vdash & \lambda x^{\bot}.\,(x\,\sigma) : \bot \to \sigma \end{array}$$

Однако тип $\bot$ не населён в $\lambda 2$ — не существует *замкнутого* терма с таким типом.

С логической точки зрения $\bot$ — это абсурд, заведомо ложное утверждение.

Связка $\neg$ (отрицание)

В $\lambda 2$ связка $\neg$ может быть определена так:

$$\neg\sigma \equiv \sigma \to \bot$$

Правило удаления $\neg$	Правило введения $\neg$
$\dfrac{\sigma \qquad \neg\sigma}{\tau}$	$\dfrac{\begin{array}{cc}[\sigma] & [\sigma] \\ \vdots & \vdots \\ \tau & \neg\tau\end{array}}{\neg\sigma}$

Покажем, что они выразимы в $\lambda 2$.

Связка $\neg$ (2)

Удаление $\neg$. Тип

$$\sigma \to \neg\sigma \to \tau \equiv \sigma \to (\sigma \to \bot) \to \tau \equiv \sigma \to (\sigma \to \forall\alpha.\, \alpha) \to \tau$$

Терм этого типа (в контексте $\Gamma = \sigma{:}*, \tau{:}*$)

$$\lambda x^{\sigma}.\, \lambda f^{\sigma \to \bot}.\, f\, x\, \tau$$

Из противоречия следует все что угодно.

Введение $\neg$. Тип

$$(\sigma \to \tau) \to (\sigma \to \neg\tau) \to \neg\sigma \equiv (\sigma \to \tau) \to (\sigma \to \tau \to \bot) \to \sigma \to \bot$$

Терм этого типа

$$\lambda\, f^{\sigma \to \tau}.\, \lambda\, g^{\sigma \to \tau \to \bot}.\, \lambda\, x^{\sigma}.\, g\, x\, (f\, x)$$

Доказательство приведением к нелепости — *reductio ad absurdum*.

Связка $\wedge$ (конъюнкция)

В $\lambda 2$ связка $\wedge$ может быть определена так:

$$\sigma \wedge \tau \;\equiv\; \forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha$$

$\alpha \notin \mathsf{FV}(\sigma)$, $\alpha \notin \mathsf{FV}(\tau)$.

Правила удаления $\wedge$	Правило введения $\wedge$
$\dfrac{\sigma \wedge \tau}{\sigma} \qquad \dfrac{\sigma \wedge \tau}{\tau}$	$\dfrac{\sigma \qquad \tau}{\sigma \wedge \tau}$

Покажем, что они выразимы в $\lambda 2$.

Введение $\wedge$

Введение $\wedge$. Тип

$$\sigma \to \tau \to (\sigma \wedge \tau) \equiv \sigma \to \tau \to (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda\, x^{\sigma}\, y^{\tau}.\, \Lambda\, \alpha.\, \lambda\, f^{\sigma \to \tau \to \alpha}.\, f\, x\, y$$

В стандартной интерпретации $\lambda 2$, как языка программирования, этот терм описывает пару (двухэлементный кортеж):

$$\sigma{:}*, \tau{:}*, x{:}\sigma, y{:}\tau \vdash \langle x, y \rangle : \sigma \times \tau$$

Удаление $\wedge$

Удаление $\wedge$ (1). Тип $(\sigma \wedge \tau) \to \sigma \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \sigma$
Терм этого типа

$$\lambda f^{\sigma \wedge \tau}.\, f\, \sigma\, (\lambda x^{\sigma} y^{\tau}.\, x)$$

$$\sigma{:}*, \tau{:}*, f : \sigma \wedge \tau \;\vdash\; f\, \sigma : (\sigma \to \tau \to \sigma) \to \sigma$$
$$\sigma{:}*, \tau{:}*, f : \sigma \wedge \tau \;\vdash\; f\, \sigma\, (\lambda x^{\sigma} y^{\tau}.\, x) : \sigma$$

Удаление $\wedge$ (2). Тип $(\sigma \wedge \tau) \to \tau \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \tau$
Терм этого типа

$$\lambda f^{\sigma \wedge \tau}.\, f\, \tau\, (\lambda x^{\sigma} y^{\tau}.\, y)$$

$$\sigma{:}*, \tau{:}*, f : \sigma \wedge \tau \;\vdash\; f\, \tau : (\sigma \to \tau \to \tau) \to \tau$$
$$\sigma{:}*, \tau{:}*, f : \sigma \wedge \tau \;\vdash\; f\, \tau\, (\lambda x^{\sigma} y^{\tau}.\, y) : \tau$$

Связка $\vee$ (дизъюнкция)

В $\lambda 2$ связка $\vee$ может быть определена так:

$$\sigma \vee \tau \;\equiv\; \forall\alpha.\,(\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha$$

$\alpha \notin \mathsf{FV}(\sigma)$, $\alpha \notin \mathsf{FV}(\tau)$.

Правило удаления $\vee$	Правила введения $\vee$
$\dfrac{\sigma \vee \tau \qquad \begin{array}{c}[\sigma]\\ \vdots\\ \rho\end{array} \qquad \begin{array}{c}[\tau]\\ \vdots\\ \rho\end{array}}{\rho}$	$\dfrac{\sigma}{\sigma \vee \tau}$ $\qquad$ $\dfrac{\tau}{\sigma \vee \tau}$

Покажем, что они выразимы в $\lambda 2$.

Введение $\vee$

Введение $\vee$ (1). Тип

$$\sigma \to (\sigma \vee \tau) = \sigma \to (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda\, x^{\sigma}.\, \Lambda\, \alpha.\, \lambda\, f^{\sigma \to \alpha}\, g^{\tau \to \alpha}.\, f\, x$$

Введение $\vee$ (2). Тип

$$\tau \to (\sigma \vee \tau) = \tau \to (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha)$$

Терм этого типа

$$\lambda\, y^{\tau}.\, \Lambda\, \alpha.\, \lambda\, f^{\sigma \to \alpha}\, g^{\tau \to \alpha}.\, g\, y$$

Удаление $\vee$

Удаление $\vee$. Тип

$$\begin{aligned} & (\sigma \vee \tau) \to (\sigma \to \rho) \to (\tau \to \rho) \to \rho \\ = \quad & (\forall \alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha) \to (\sigma \to \rho) \to (\tau \to \rho) \to \rho \end{aligned}$$

Терм этого типа

$$\lambda\, h^{\sigma \vee \tau}\, f^{\sigma \to \rho}\, g^{\tau \to \rho}.\, h\, \rho\, f\, g$$

Построение терма

$$\sigma{:}*, \tau{:}*, \rho{:}*, h : \sigma \vee \tau \;\vdash\; h\, \rho : (\sigma \to \rho) \to (\tau \to \rho) \to \rho$$

Это правило носит название *доказательство разбором случаев*.

Экзистенциальные типы

Трактовка $\forall\alpha.\,\sigma$:
Функция из типов в термы, которая *любому* типу τ ставит в соответствие терм с типом $\sigma[\alpha := \tau]$.
Для $\sigma = \alpha \to \alpha$ имеем $\sigma[\alpha := \tau] = \tau \to \tau$:

$$\Lambda\alpha.\,\lambda x^{\alpha}.\,x : \forall\alpha.\,\alpha \to \alpha$$
$$(\Lambda\alpha.\,\lambda x^{\alpha}.\,x)\tau \to_{\beta} \lambda x^{\tau}.\,x : \tau \to \tau$$

Трактовка $\exists\alpha.\,\sigma$:
Пара из *некоторого* типа τ и терма, имеющего тип $\sigma[\alpha := \tau]$.
Для $\sigma = \alpha \to \gamma$ имеем:

$$\langle \gamma, \lambda x^{\gamma}.\,x \rangle : \exists\alpha.\,\alpha \to \gamma$$

поскольку $\sigma[\alpha := \gamma] = \gamma \to \gamma$ и $\lambda x^{\gamma}.\,x : \gamma \to \gamma$.

$\exists$ (квантор существования)

В $\lambda 2$ $\exists$ может быть определён так:

$$\exists\alpha\,.\,\sigma \;\equiv\; \forall\beta.\,(\forall\alpha.\,\sigma\to\beta)\to\beta$$

$\beta \notin \mathsf{FV}(\sigma)$.

Правило удаления $\exists$	Правило введения $\exists$
$\dfrac{\exists\alpha.\,\sigma \qquad \begin{array}{c}[\sigma]\\ \vdots\\ \rho\end{array}}{\rho}$	$\dfrac{\sigma[\alpha := \tau]}{\exists\alpha.\,\sigma}$

Покажем, что они выразимы в $\lambda 2$.

Введение $\exists$

Введение $\exists$. Тип

$$\sigma[\alpha := \tau] \to \exists\alpha.\,\sigma = \sigma[\alpha := \tau] \to \forall\beta.\,(\forall\alpha.\,\sigma \to \beta) \to \beta$$

Терм этого типа (в контексте $\Gamma = \tau{:}*$)

$$\lambda x^{\sigma[\alpha:=\tau]}.\,\Lambda\beta.\,\lambda f^{\forall\alpha.\,\sigma\to\beta}.\,f\,\tau\,x$$

Построение терма

$$\begin{array}{rcl} \tau{:}*, \sigma{:}*, \beta{:}*, f{:}\forall\alpha.\,\sigma\to\beta & \vdash & f\,\tau : \sigma[\alpha := \tau] \to \beta \\ \tau{:}*, \sigma{:}*, \beta{:}*, f{:}\forall\alpha.\,\sigma\to\beta, x{:}\sigma[\alpha := \tau] & \vdash & f\,\tau\,x : \beta \end{array}$$

Удаление $\exists$

Удаление $\exists$. Тип

$$\exists\alpha.\,\sigma\to(\sigma\to\rho)\to\rho = (\forall\beta.\,(\forall\alpha.\,\sigma\to\beta)\to\beta)\to(\sigma\to\rho)\to\rho$$

Терм этого типа

$$\lambda\, f^{\exists\alpha.\,\sigma}.\,\lambda\, g^{\sigma\to\rho}.\, f\,\rho\,(\Lambda\,\alpha.\, g)$$

Построение терма

$$\begin{aligned} f{:}\exists\alpha.\,\sigma &\vdash (f\,\rho) : (\forall\alpha.\,\sigma\to\rho)\to\rho \\ g{:}\sigma\to\rho &\vdash (\Lambda\alpha.\, g) : \forall\alpha.\,\sigma\to\rho \end{aligned}$$

Кодирование в $\lambda 2$: булев тип

$$\texttt{Bool} \;=\; \forall\alpha.\,\alpha\to\alpha\to\alpha$$

Констант этого типа в точности две: $\texttt{TRUE} : \texttt{Bool}$ и $\texttt{FALSE} : \texttt{Bool}$:

$$\begin{aligned}\texttt{TRUE} &= \Lambda\alpha.\,\lambda\, t^{\alpha}\, f^{\alpha}.\, t\\ \texttt{FALSE} &= \Lambda\alpha.\,\lambda\, t^{\alpha}\, f^{\alpha}.\, f\end{aligned}$$

Условный оператор задаёт способ «использования» для $\texttt{Bool}$:

$$\begin{aligned}\texttt{IIF} &: \texttt{Bool}\to\rho\to\rho\to\rho\\ \texttt{IIF} &= \lambda\, e^{\texttt{Bool}}.\lambda x^{\rho}\, y^{\rho}.\, e\,\rho\, x\, y\\ \texttt{IIF}\; e &= \lambda x^{\rho}\, y^{\rho}.\, e\,\rho\, x\, y\\ \texttt{IIF}\; e\, x\, y &= e\,\rho\, x\, y\end{aligned}$$

«Комбина́торный» синтаксис часто удобнее лямбд.

Кодирование в $\lambda 2$: булев тип (2)

Проверка (в контексте $\rho{:}*, a{:}\rho, b{:}\rho$)

$$\begin{array}{rcl}
\texttt{IIF TRUE}\ a\,b & \to_\beta & \texttt{TRUE}\ \rho\,a\,b \\
& = & (\Lambda\alpha.\,\lambda t^\alpha\, f^\alpha.\, t)\,\rho\,a\,b \\
& \to_\beta & (\lambda\, t^\rho\, f^\rho.\, t)\,a\,b \\
& \to_\beta & a
\end{array}$$

$$\begin{array}{rcl}
\texttt{IIF FALSE}\ a\,b & \to_\beta & \texttt{FALSE}\ \rho\,a\,b \\
& = & (\Lambda\alpha.\,\lambda t^\alpha\, f^\alpha.\, f)\,\rho\,a\,b \\
& \to_\beta & (\lambda\, t^\rho\, f^\rho.\, f)\,a\,b \\
& \to_\beta & b
\end{array}$$

Кодирование в $\lambda 2$: булев тип (3)

Операторы AND, OR и NOT.

$$\begin{aligned} \texttt{AND} &: \texttt{Bool} \to \texttt{Bool} \to \texttt{Bool} \\ \texttt{AND}\ x\,y &= \Lambda\alpha.\,\lambda t^{\alpha}\, f^{\alpha}.\, x\,\alpha\,(y\,\alpha\,t\,f)\,f \end{aligned}$$

Синим выделено отличие от бестиповой лямбды.

$$\begin{aligned} \texttt{OR} &: \texttt{Bool} \to \texttt{Bool} \to \texttt{Bool} \\ \texttt{OR}\ x\,y &= \text{??? самостоятельно} \end{aligned}$$

$$\begin{aligned} \texttt{NOT} &: \texttt{Bool} \to \texttt{Bool} \\ \texttt{NOT}\ x &= \text{??? самостоятельно} \end{aligned}$$

Кодирование в $\lambda 2$: двухэлементный кортеж

Тип пары значений с типами σ и τ (произведение типов $\sigma \times \tau$) можно задать так:

$$\sigma \times \tau \;\equiv\; \forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha$$

Отметим, что $\sigma \times \tau \;\equiv\; \sigma \wedge \tau$.

Любые два значения $a : \sigma$ и $b : \tau$ порождают значение $\langle a, b \rangle$ типа их произведения $\sigma \times \tau$:

$$\mathtt{PAIR}\ a\, b \;\equiv\; \langle a, b \rangle \;\equiv\; \Lambda\, \alpha.\, \lambda\, f^{\sigma \to \tau \to \alpha}.\, f\, a\, b$$

Тип этого конструктора пары

$$\sigma \to \tau \to \sigma \times \tau \quad \equiv \quad \sigma \to \tau \to \forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha$$

Кодирование в $\lambda 2$: двухэлементный кортеж (2)

Две проекции пары $p : \sigma \times \tau$

$$\begin{aligned} \texttt{FST} &: \sigma \times \tau \to \sigma \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \sigma \\ \texttt{FST}\ p &\equiv p\, \sigma\, (\lambda x^{\sigma} y^{\tau}.\, x) \end{aligned}$$

$$\begin{aligned} \texttt{SND} &: \sigma \times \tau \to \tau \equiv (\forall \alpha.\, (\sigma \to \tau \to \alpha) \to \alpha) \to \tau \\ \texttt{SND}\ p &\equiv p\, \tau\, (\lambda x^{\sigma} y^{\tau}.\, y) \end{aligned}$$

Итератор для пары $p : \sigma \times \tau$

$$\begin{aligned} \texttt{UNCURRY} &: (\sigma \to \tau \to \rho) \to \sigma \times \tau \to \rho \\ \texttt{UNCURRY} &\equiv \lambda f^{\sigma \to \tau \to \rho}\, p^{\sigma \times \tau}.\, p\, \rho\, f \\ \texttt{UNCURRY}\ f\, p &\equiv p\, \rho\, f \end{aligned}$$

Выразите `FST` и `SND` в терминах `UNCURRY`.

Кодирование в $\lambda 2$: размеченное объединение

Тип размеченного объединения значений типов σ и τ (суммы типов $\sigma + \tau$) можно задать так:

$$\sigma + \tau \;\equiv\; \forall\alpha.\,(\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha$$

Отметим, что $\sigma + \tau \;\equiv\; \sigma \vee \tau$.

В $\sigma + \tau$ хранится либо значение типа σ, либо типа τ, инъектированные в него одним из двух конструкторов

$$\begin{aligned} \texttt{INJL} \;&:\; \sigma \to \sigma + \tau \\ \texttt{INJL}\ a \;&\equiv\; \Lambda\alpha.\,\lambda\, f^{\sigma\to\alpha}\, g^{\tau\to\alpha}.\, f\, a \end{aligned}$$

$$\begin{aligned} \texttt{INJR} \;&:\; \tau \to \sigma + \tau \\ \texttt{INJR}\ b \;&\equiv\; \Lambda\alpha.\,\lambda\, f^{\sigma\to\alpha}\, g^{\tau\to\alpha}.\, g\, b \end{aligned}$$

Кодирование в $\lambda 2$: размеченное объединение (2)

$$\sigma + \tau \;\equiv\; \forall\alpha.\,(\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha$$

«Итератор» для размеченного объединения $\mathfrak{h}^{\sigma+\tau}$
(разбор случаев):

$$\begin{aligned} \texttt{CASES} &: (\sigma \to \rho) \to (\tau \to \rho) \to (\sigma + \tau) \to \rho \\ \texttt{CASES} &\equiv \lambda \mathfrak{f}^{\sigma\to\rho}\, \mathfrak{g}^{\tau\to\rho}\, \mathfrak{h}^{\sigma+\tau}.\, \mathfrak{h}\, \rho\, \mathfrak{f}\, \mathfrak{g} \\ \texttt{CASES}\ \mathfrak{f}\, \mathfrak{g}\, \mathfrak{h} &\equiv \mathfrak{h}\, \rho\, \mathfrak{f}\, \mathfrak{g} \end{aligned}$$

ρ задает тип результата разбора, а $\mathfrak{f}$ и $\mathfrak{g}$ являются обработчиками допустимых вариантов хранения.

Кодирование в $\lambda 2$: размеченное объединение (3)

$$\texttt{TST1} \;:\; \texttt{Bool} + \texttt{Nat}$$

$$\texttt{TST1} \equiv \texttt{INJL FALSE} \equiv \Lambda\alpha.\,\lambda\, \mathrm{f}^{\texttt{Bool}\to\alpha}\, \mathrm{g}^{\texttt{Nat}\to\alpha}.\, \mathrm{f}\ \texttt{FALSE}$$

$$\texttt{TST2} \;:\; \texttt{Bool} + \texttt{Nat}$$

$$\texttt{TST2} \equiv \texttt{INJR}\ \overline{42} \equiv \Lambda\alpha.\,\lambda\, \mathrm{f}^{\texttt{Bool}\to\alpha}\, \mathrm{g}^{\texttt{Nat}\to\alpha}.\, \mathrm{g}\ \overline{42}$$

$\texttt{CASES}\ \mathrm{f}\,\mathrm{g}\,\mathrm{h} \equiv \mathrm{h}\,\rho\,\mathrm{f}\,\mathrm{g}$, где $\mathrm{f} : \sigma \to \rho$, $\mathrm{g} : \tau \to \rho$, $\mathrm{h} : \sigma + \tau$

$$\texttt{NOT} : \texttt{Bool} \to \texttt{Bool} \qquad \texttt{ISZRO} : \texttt{Nat} \to \texttt{Bool}$$

$$\texttt{CASES NOT ISZRO TST1} \;:\; \texttt{Bool}$$

$$\texttt{CASES NOT ISZRO TST1} \to_\beta \texttt{NOT FALSE} \to_\beta \texttt{TRUE}$$

$$\texttt{CASES NOT ISZRO TST2} \;:\; \texttt{Bool}$$

$$\texttt{CASES NOT ISZRO TST2} \to_\beta \texttt{ISZRO}\ \overline{42} \to_\beta \texttt{FALSE}$$

Кодирование в $\lambda 2$: натуральные числа

Каков тип, например, $\overline{3} = \lambda z\, s.\, s\,(s\,(s\,z))$?
Если выбрать $z : \alpha$, то из аппликаций ясно, что $s : \alpha \to \alpha$.
Поэтому естественно определить

$$\mathtt{Nat} = \forall \alpha.\, \alpha \to (\alpha \to \alpha) \to \alpha$$

(или, переставив в абстракции s и z, $\forall \alpha.\, (\alpha \to \alpha) \to \alpha \to \alpha$)

Конструкторы для `Nat`

$$\begin{aligned} \mathtt{ZERO} &: \mathtt{Nat} \\ \mathtt{ZERO} &= \Lambda \alpha.\, \lambda z^{\alpha}\, s^{\alpha \to \alpha}.\, z \end{aligned}$$

$$\begin{aligned} \mathtt{SUCC} &: \mathtt{Nat} \to \mathtt{Nat} \\ \mathtt{SUCC}\ n &= \Lambda \alpha.\, \lambda z^{\alpha}\, s^{\alpha \to \alpha}.\, s\,(n\,\alpha\, z\, s) \end{aligned}$$

Индуктивный параметр «перевычисляется».

Кодирование в $\lambda 2$: натуральные числа (2)

Любое натуральное число выразимо через конструкторы

$$\overline{n} \;=\; \Lambda\alpha.\,\lambda z^{\alpha}\, s^{\alpha\to\alpha}.\, s^{n}(z) \;=\; \mathtt{SUCC}^{n}(\mathtt{ZERO})$$

В ЯП конструкторы часто вводят как примитивы:

```
data Nat = ZERO | SUCC Nat
```

Итератор для $n : \mathtt{Nat}$

$$\begin{aligned} \mathtt{IT} &: \rho \to (\rho \to \rho) \to \mathtt{Nat} \to \rho \\ \mathtt{IT} &\equiv \lambda x^{\rho}\, f^{\rho\to\rho}\, n^{\mathtt{Nat}}.\, n\, \rho\, x\, f \\ \mathtt{IT}\; x\, f\, n &\equiv n\, \rho\, x\, f \end{aligned}$$

Кодирование в $\lambda 2$: натуральные числа (3)

Итерирование происходит ожидаемым образом (контекст $\rho{:}*, x{:}\rho, f{:}\rho\to\rho$; $\mathtt{IT}\ x\,f\,n \equiv n\,\rho\,x\,f$):

$$
\begin{array}{rcl}
\mathtt{IT}\ x\,f\ \mathtt{ZERO} & \equiv & (\Lambda\alpha.\,\lambda z^{\alpha}\,s^{\alpha\to\alpha}.\,z)\,\rho\,x\,f \\
& \to_\beta & (\lambda z^{\rho}\,s^{\rho\to\rho}.\,z)\ x\,f \\
& \to_\beta & (\lambda s^{\rho\to\rho}.\,x)\,f \\
& \to_\beta & x \\
\mathtt{IT}\ x\,f\ (\mathtt{SUCC}\ n) & \equiv & (\Lambda\alpha.\,\lambda z^{\alpha}\,s^{\alpha\to\alpha}.\,s\,(n\,\alpha\,z\,s))\,\rho\,x\,f \\
& \to_\beta & (\lambda z^{\rho}\,s^{\rho\to\rho}.\,s\,(n\,\rho\,z\,s))\ x\,f \\
& \to_\beta & (\lambda s^{\rho\to\rho}.\,s\,(n\,\rho\,x\,s))\ f \\
& \to_\beta & f\,(n\,\rho\,x\,f) \\
& = & f\,(\mathtt{IT}\ x\,f\,n)
\end{array}
$$

Кодирование в $\lambda 2$: натуральные числа (4)

Рекурсия в терминах итерации

$$\begin{array}{rcl}
\mathtt{XZ} & : & \sigma \to \sigma \times \mathtt{Nat} \\
\mathtt{XZ}\ x & \equiv & \langle x,\ \mathtt{ZERO} \rangle \\
\mathtt{FS} & : & (\sigma \to \mathtt{Nat} \to \sigma) \to \sigma \times \mathtt{Nat} \to \sigma \times \mathtt{Nat} \\
\mathtt{FS}\ f\, p & \equiv & \langle f\,(\mathtt{FST}\ p)\,(\mathtt{SND}\ p),\ \mathtt{SUCC}\ (\mathtt{SND}\ p) \rangle \\
\mathtt{REC} & : & \mathtt{Nat} \to \sigma \to (\sigma \to \mathtt{Nat} \to \sigma) \to \sigma \\
\mathtt{REC}\ n\, x\, f & \equiv & \mathtt{FST}\ (\mathtt{IT}\ (\mathtt{XZ}\ x)\ (\mathtt{FS}\ f)\ n)
\end{array}$$

Итерирование $\mathtt{IT}\ x\, f\, n \equiv n\, \rho\, x\, f$ идёт с типом $\rho \equiv \sigma \times \mathtt{Nat}$.
В частности,

$$\begin{array}{rcl}
\vdash \mathtt{PRED} & : & \mathtt{Nat} \to \mathtt{Nat} \\
\mathtt{PRED}\ n & \equiv & \mathtt{REC}\ n\ \mathtt{ZERO}\ (\lambda x^{\mathtt{Nat}}\, y^{\mathtt{Nat}}.\, y)
\end{array}$$

Кодирование в $\lambda 2$: списки

Каков тип $[\overline{3},\overline{7},\overline{42}] = \lambda n\, c.\, c\,\overline{3}\,(c\,\overline{7}\,(c\,\overline{42}\,n))$?
Если выбрать $n : \alpha$, то $c : \mathtt{Nat} \to \alpha \to \alpha$. Определим

$$\begin{aligned} \mathtt{ListNat} &\equiv \forall\alpha.\, \alpha \to (\mathtt{Nat} \to \alpha \to \alpha) \to \alpha \\ \mathtt{List}\,\sigma &\equiv \forall\alpha.\, \alpha \to (\sigma \to \alpha \to \alpha) \to \alpha \end{aligned}$$

Конструкторы для $\mathtt{List}\,\sigma$

$$\begin{aligned} \mathtt{NIL} &: \mathtt{List}\,\sigma \\ \mathtt{NIL} &\equiv \Lambda\alpha.\, \lambda n^{\alpha}\, c^{\sigma\to\alpha\to\alpha}.\, n \\ \mathtt{CONS} &: \sigma \to \mathtt{List}\,\sigma \to \mathtt{List}\,\sigma \\ \mathtt{CONS}\ x\ l &\equiv \Lambda\alpha.\, \lambda n^{\alpha}\, c^{\sigma\to\alpha\to\alpha}.\, c\,x\,(l\,\alpha\,n\,c) \end{aligned}$$

$$[\overline{3},\overline{7},\overline{42}] = \mathtt{CONS}\ \overline{3}\ (\mathtt{CONS}\ \overline{7}\ (\mathtt{CONS}\ \overline{42}\ \mathtt{NIL}))$$

Кодирование в $\lambda 2$: списки (2)

Итератор для $\mathtt{l} : \mathtt{List}\,\sigma$ (свёртка)

$$\begin{aligned} \mathtt{FOLD} &: \rho \to (\sigma \to \rho \to \rho) \to \mathtt{List}\,\sigma \to \rho \\ \mathtt{FOLD} &\equiv \lambda x^{\rho}\, f^{\sigma\to\rho\to\rho}\, l^{\mathtt{List}\,\sigma}.\, l\,\rho\,x\,f \\ \mathtt{FOLD}\ x\,f\,l &\equiv l\,\rho\,x\,f \end{aligned}$$

Проверьте, что свёртка ведёт себя ожидаемо, то есть

$$\begin{aligned} \mathtt{FOLD}\ x\,f\ \mathtt{NIL} &\twoheadrightarrow_\beta x \\ \mathtt{FOLD}\ x\,f\ (\mathtt{CONS}\ e\,l) &\twoheadrightarrow_\beta f\,e\,(l\,\rho\,x\,f) \\ &= f\,e\,(\mathtt{FOLD}\ x\,f\,l) \end{aligned}$$

Пример свёртки:

$$\mathtt{FOLD}\ [\overline{3}, \overline{7}, \overline{42}]\ x\,f = f\ \overline{3}\ (f\ \overline{7}\ (f\ \overline{42}\ x))$$

Общая теория (индуктивных типов Мартин-Лёфа)

Рассмотрим

$$\texttt{SomeType} \;= \forall\alpha.\, \tau_1 \to \tau_2 \to \ldots \to \tau_n \to \alpha$$

где τ_i зависит от α, имея α по крайней мере в самой правой позиции: $\tau_i = \ldots \to \alpha$

▶ Количество конструкторов равно n («арности» `SomeType`):

$$\begin{aligned}
\top &\equiv \forall\alpha.\, \alpha \to \alpha \quad \text{1 штука} \\
\sigma \times \tau &\equiv \forall\alpha.\, (\sigma \to \tau \to \alpha) \to \alpha \quad \text{1 штука} \\
\texttt{Bool} &\equiv \forall\alpha.\, \alpha \to \alpha \to \alpha \quad \text{2 штуки} \\
\sigma + \tau &\equiv \forall\alpha.\, (\sigma \to \alpha) \to (\tau \to \alpha) \to \alpha \quad \text{2 штуки} \\
\texttt{List}\,\sigma &\equiv \forall\alpha.\, \alpha \to (\sigma \to \alpha \to \alpha) \to \alpha \quad \text{2 штуки}
\end{aligned}$$

Теория типов Мартин-Лёфа: тип конструктора

$$\texttt{SomeType} = \forall\alpha.\, \tau_1 \to \tau_2 \to \ldots \to \tau_n \to \alpha$$

- Арность i-го конструктора равна арности τ_i.
- Тип i-го конструктора $f_i : \tau_i[\alpha := \texttt{SomeType}]$.

Например, для списка $f_i : \tau_i[\alpha := \texttt{List}\,\sigma]$:

$$\texttt{List}\,\sigma \equiv \forall\alpha.\, \alpha \to (\sigma \to \alpha \to \alpha) \to \alpha$$

$$\begin{aligned} \texttt{NIL} &: \texttt{List}\,\sigma \\ \texttt{CONS} &: \sigma \to \texttt{List}\,\sigma \to \texttt{List}\,\sigma \end{aligned}$$

Теория типов Мартин-Лёфа: код конструктора

Пусть $\tau_i = \sigma \to \tau \to \alpha \to \alpha$. Тип i-го конструктора

$$f_i : \sigma \to \tau \to \mathtt{SomeType} \to \mathtt{SomeType}$$

- Построение i-ого конструктора — однозначная процедура:

$$\begin{aligned}
f_i &= \lambda x_1^{\sigma}\, x_2^{\tau}\, a_1^{\mathtt{SomeType}}.\, \{\ldots\} : \sigma \to \tau \to \mathtt{SomeType} \to \mathtt{SomeType} \\
f_i\, x_1\, x_2\, a_1 &= \{\ldots\} : \mathtt{SomeType} \\
f_i\, x_1\, x_2\, a_1 &= \Lambda\alpha.\, \lambda t_1^{\tau_1}\, t_2^{\tau_2}\, \ldots\, t_n^{\tau_n}.\, t_i\, \{\ldots\} \\
f_i\, x_1\, x_2\, a_1 &= \Lambda\alpha.\, \lambda t_1^{\tau_1}\, t_2^{\tau_2}\, \ldots\, t_n^{\tau_n}.\, t_i\, x_1\, x_2\, (a_1\, \alpha\, t_1\, t_2\, \ldots\, t_n)
\end{aligned}$$

Индуктивные параметры «реконструируются».

$$\begin{aligned}
\mathtt{NIL} &\equiv \Lambda\alpha.\, \lambda n^{\alpha}\, c^{\sigma \to \alpha \to \alpha}.\, n \\
\mathtt{CONS}\ x\ l &\equiv \Lambda\alpha.\, \lambda n^{\alpha}\, c^{\sigma \to \alpha \to \alpha}.\, c\, x\, (l\, \alpha\, n\, c)
\end{aligned}$$

Теория типов Мартин-Лёфа: итераторы

Если $\mathtt{SomeType} = \forall\alpha.\,\tau_1 \to \tau_2 \to \ldots \to \tau_n \to \alpha$, то *тип итератора*

$$\mathtt{IT} : \tau_1[\alpha := \rho] \to \tau_2[\alpha := \rho] \to \ldots \to \tau_n[\alpha := \rho] \to \mathtt{SomeType} \to \rho$$

Код итератора тривиален; для $s : \mathtt{SomeType}$ имеем

$$\mathtt{IT} \;=\; \lambda x_1^{\tau_1[\alpha:=\rho]}\, x_2^{\tau_2[\alpha:=\rho]} \ldots\, x_n^{\tau_n[\alpha:=\rho]}\, s^{\mathtt{SomeType}}.\, s\,\rho\,x_1\,x_2\,\ldots\,x_n$$

$$\mathtt{IT}\;x_1\,x_2\,\ldots\,x_n\,s \;=\; s\,\rho\,x_1\,x_2\,\ldots\,x_n$$

Вспомним итераторы

$$\begin{array}{rl} \sigma + \tau & \mathtt{CASES}\;f\,g\,h \;\equiv\; h\,\rho\,f\,g \\ \sigma \times \tau & \mathtt{UNCURRY}\;f\,p \;\equiv\; p\,\rho\,f \\ \mathtt{List}\,\sigma & \mathtt{FOLD}\;x\,f\,l \;\equiv\; l\,\rho\,x\,f \end{array}$$

Домашнее задание

Реализуйте функции $\texttt{OR} : \texttt{Bool} \to \texttt{Bool} \to \texttt{Bool}$ и $\texttt{NOT} : \texttt{Bool} \to \texttt{Bool}$.

Задайте в $\lambda 2$ перечислительный тип `Three`, населённый ровно тремя константами $\texttt{ONE} : \texttt{Three}$, $\texttt{TWO} : \texttt{Three}$, $\texttt{THREE} : \texttt{Three}$. Выпишите его конструкторы и «итератор» для него. Реализуйте функцию $\texttt{SHIFT} : \texttt{Three} \to \texttt{Three}$ со следующим поведением:

$$\begin{aligned} \texttt{SHIFT ONE} &= \texttt{TWO} \\ \texttt{SHIFT TWO} &= \texttt{THREE} \\ \texttt{SHIFT THREE} &= \texttt{ONE} \end{aligned}$$

Домашнее задание (2)

Выразите `FST` и `SND` в терминах `UNCURRY`.

Выразите `PLUS` и `MULT` для `Nat` в терминах `IT`.

Выразите $\texttt{MAP} : (\sigma \to \tau) \to \texttt{List}\,\sigma \to \texttt{List}\,\tau$ в терминах `FOLD`.

Определите в $\lambda 2$ индуктивный тип $\texttt{Tree}\,\sigma\,\tau$, задающий бинарное дерево, хранящее в узлах значения типа σ, а в листьях — типа τ.
Выпишите его конструкторы и «итератор» для него.

Литература (1)

TAPL гл. 23, 24
Benjamin C. Pierce, Types and Programming Languages, MIT Press, 2002

`http://www.cis.upenn.edu/~bcpierce/tapl`

ITT гл. 5
Herman Geuvers, Introduction to Type Theory
Alfa Lernet Summer school 2008, Uruguay

`http://www.cs.ru.nl/H.Geuvers/Uruguay2008SummerSchool.html`

Литература (2)

ОЯП гл. 9

Дж.Митчелл, Основания языков программирования,
М.-Ижевск, НИЦ РХД, 2010

PAT гл. 11

Jean-Yves Girard, Paul Taylor, Yves Lafont, Proofs and Types,
Cambridge University Press, 1989