Wykład 2: Rachunek lambda

Systemy typów, II UWr, 2010

20 października 2010

λ-termy

- zmienne (Var)

  $\{x, y, z, ...\}$ – nieskończony, przeliczalny zbiór zmiennych

- termy (Term)

$$t \quad ::= \quad x \mid \lambda x.t \mid t\ t$$

- skróty notacyjne

$$\begin{aligned} \lambda x_1 x_2 \ldots x_n.t &\equiv \lambda x_1.\lambda x_2.\ldots.\lambda x_n.t \\ t_1\ t_2\ t_3 \ldots t_n &\equiv (\ldots((t_1\ t_2)\ t_3)\ \ldots)t_n \end{aligned}$$

α-równoważność i konwencja Barendregta

- λ-abstrakcja $\lambda x.t$ wiąże zmienną x w termie t
- zmienne wolne

$$\begin{aligned} \mathrm{FV}(x) &= \{x\} \\ \mathrm{FV}(\lambda x.t) &= \mathrm{FV}(t) - \{x\} \\ \mathrm{FV}(t_0\ t_1) &= \mathrm{FV}(t_0) \cup \mathrm{FV}(t_1) \end{aligned}$$

- termy t i u są α-równoważne, jeżeli różnią się wyłącznie nazwami zmiennych związanych
- α-równoważność jest relacją równoważności – utożsamiamy α-równoważne termy
- konwencja Barendregta
  zakładamy, że zmienne wolne są różne od zmiennych związanych w rozważanych termach (zmienne związane można zawsze odpowiednio przemianować)

Podstawienie

$$
\begin{array}{rcll}
x\{s/x\} & = & s & \\
y\{s/x\} & = & y & x \neq y \\
(\lambda y.t)\{s/x\} & = & \lambda y.t\{s/x\} & x \neq y, y \notin \mathsf{FV}(s) \\
(t_0\ t_1)\{s/x\} & = & t_0\{s/x\}\ t_1\{s/x\} &
\end{array}
$$

- implementując podstawienie trzeba przemianować zmienne związane w termie, w którym wykonuje się podstawienie na "świeże" zmienne

β-redukcja

- relacja β-redukcji

$$\frac{}{(\lambda x.t)\, s \to_\beta t\{s/x\}} \qquad \frac{t \to_\beta t'}{\lambda x.t \to_\beta \lambda x.t'}$$

$$\frac{t \to_\beta t'}{t\ s \to_\beta t'\ s} \qquad \frac{s \to_\beta s'}{t\ s \to_\beta t\ s'}$$

- $(\lambda x.t)\, s$ – β-redex
- t jest w postaci normalnej, jeżeli nie istnieje s taki, że $t \to_\beta s$

β-redukcja, c.d.

- relacja $\rightarrow_\beta^*$ wielokrokowej β-redukcji – przechodnio-zwrotne domknięcie $\rightarrow_\beta$
- relacja $=_\beta$ β-równości – przechodnio-zwrotno-symetryczne domknięcie $\rightarrow_\beta$

Twierdzenie (Churcha-Rossera)

Jeżeli $t \rightarrow_\beta^* r$ i $t \rightarrow_\beta^* s$, to istnieje u taki, że $r \rightarrow_\beta^* u$ i $s \rightarrow_\beta^* u$.
(Jeżeli $r =_\beta s$, to istnieje u taki, że $r \rightarrow_\beta^* u$ i $s \rightarrow_\beta^* u$.)

Wniosek

Każdy term t ma co najwyżej jedną postać normalną u (tj. taką, że $t \rightarrow_\beta^* u$).

Uwaga

Istnieją termy, które nie mają postaci normalnej, np. $\Omega = \omega\,\omega$, gdzie $\omega = \lambda x.x\,x$.

Reprezentacja struktur danych w rachunku λ

- liczby naturalne (numerały Churcha)

$$c_n \quad = \quad \lambda s.\lambda z.s^{(n)}(z)$$

gdzie $t^{(0)}(s) = s,\ t^{(n+1)}(s) = t\,(t^{(n)}(s))$

$$\begin{array}{rcl} \mathsf{suc} & = & \lambda m.\lambda s.\lambda z.m\ s\ (s\ z) \\ \mathsf{add} & = & \lambda m.\lambda n.\lambda s.\lambda z.m\ s\ (n\ s\ z) \\ \mathsf{mul} & = & \lambda m.\lambda n.\lambda s.\lambda z.m\ (n\ s)\ z \end{array}$$

$$\begin{array}{l} \mathsf{succ}\ c_n \to^*_\beta c_{n+1} \\ \mathsf{add}\ c_m\ c_n \to^*_\beta c_{m+n} \\ \mathsf{mul}\ c_m\ c_n \to^*_\beta c_{m*n} \end{array}$$

Reprezentacja struktur danych w rachunku λ, c.d.

- wartości logiczne

$$\begin{aligned} \text{true} &= \lambda x.\lambda y.x \\ \text{false} &= \lambda x.\lambda y.y \\ \text{cond} &= \lambda b.\lambda x.\lambda y.b\ x\ y \end{aligned}$$

$$\begin{aligned} \text{cond true } t\ s &\to_\beta^* t \\ \text{cond false } t\ s &\to_\beta^* s \end{aligned}$$

- pary, listy, drzewa, etc.
- rekursja

$$Y = \lambda f.(\lambda x.f\ (x\ x))\ (\lambda x.f\ (x\ x))$$

$$t\ (Y\ t) =_\beta Y\ t$$

Strategie redukcji – normalizacja

- postać normalna

$$v \quad ::= \quad \lambda x.v \mid x\ v_1 \ldots v_n \quad (n \geq 0)$$

- leftmost-outermost (porządek normalny) – w każdym kroku wybierany jest najbardziej na lewo położony redeks niezawarty w żadnym redeksie
- leftmost-innermost (porządek aplikatywny) – w każdym kroku wybierany jest najbardziej na lewo położony redeks niezawierający żadnego redeksu

Twierdzenie

Jeżeli term ma postać normalną, to redukcja w porządku normalnym ją znajdzie, a redukcja w porządku aplikatywnym niekoniecznie.

Ewaluacja w porządku normalnym (call by name)

- słaba czołowa postać normalna

$$v \quad ::= \quad \lambda x.t \mid x\ t_1 \ldots t_n \quad (n \geq 0)$$

- relacja redukcji w porządku normalnym $\rightarrow_n$

$$\frac{}{(\lambda x.t)\ s \rightarrow_n t\{s/x\}} \qquad \frac{t \rightarrow_n t'}{t\ s \rightarrow_n t'\ s}$$

- relacja ewaluacji $\Downarrow_n$ w stylu semantyki naturalnej

$$\frac{}{x \Downarrow_n x} \qquad \frac{}{\lambda x.t \Downarrow_n \lambda x.t}$$

$$\frac{t \Downarrow_n \lambda x.u \quad u\{s/x\} \Downarrow_n v}{t\ s \Downarrow_n v} \qquad \frac{t \Downarrow_n v}{t\ s \Downarrow_n v\ s} \quad (v \neq \lambda x.u)$$

Ewaluacja w porządku normalnym (call by name), c.d.

Twierdzenie

$t \to_n^* v$ *wtw* $t \Downarrow_n v$.

Dowód

- $\Rightarrow$ *Indukcja po długości ciągu redukcji* $t \to_n^* v$.
- $\Leftarrow$ *Indukcja po wyprowadzeniu* $t \Downarrow_n v$.

Maszyna Krivine'a (wariant z podstawieniem)

- reprezentacja termu
  $(t, t_1 : \cdots : t_n)$ reprezentuje term $t\ t_1\ \ldots t_n$
- przejścia maszyny

$$\begin{aligned} (\lambda x.t, s : \overline{s}) &\Rightarrow_n (t\{s/x\}, \overline{s}) \\ (t_0\ t_1, \overline{s}) &\Rightarrow_n (t_0, t_1 : \overline{s}) \end{aligned}$$

- konfiguracja początkowa

$$(t, \epsilon)$$

- konfiguracje końcowe

$$(\lambda x.t, \epsilon)$$
$$(x, \overline{t})$$

Twierdzenie

$t \Downarrow_n v$ *wtw* $(t, \epsilon) \Rightarrow_n^* (v_0, t_1 : \cdots : t_n)$ *gdzie* $v = v_0\ t_1 \ldots t_n$.

Ewaluacja w porządku aplikatywnym (call by value)

- słaba postać normalna

$$v \quad ::= \quad \lambda x.t \mid x\ v_1 \ldots v_n \quad (n \geq 0)$$

- relacja redukcji w porządku aplikatywnym $\rightarrow_v$

$$\frac{}{(\lambda x.t)\ v \rightarrow_v t\{v/x\}}$$

$$\frac{t \rightarrow_v t'}{t\ s \rightarrow_v t'\ s} \qquad \frac{s \rightarrow_v s'}{v\ s \rightarrow_v v\ s'}$$

- relacja ewaluacji $\Downarrow_v$ w stylu semantyki naturalnej

$$\frac{}{x \Downarrow_v x} \qquad \frac{}{\lambda x.t \Downarrow_v \lambda x.t} \qquad \frac{t \Downarrow_v \lambda x.u \quad s \Downarrow_v w \quad u\{w/x\} \Downarrow_v v}{t\ s \Downarrow_v v}$$

$$\frac{t \Downarrow_v v \quad s \Downarrow_v w}{t\ s \Downarrow_v v\ w} \quad (v \neq \lambda x.u)$$

Maszyna CK

- stos

$$\overline{s} \quad ::= \quad \epsilon \mid \mathsf{arg}(t) : \overline{s} \mid \mathsf{fun}(v) : \overline{s}$$

- przejścia maszyny

$$\begin{array}{rcl}
(x, \overline{s}) & \Rightarrow_v & (\overline{s}, x) \\
(\lambda x.t, \overline{s}) & \Rightarrow_v & (\overline{s}, \lambda x.t) \\
(t_0\ t_1, \overline{s}) & \Rightarrow_v & (t_0, \mathsf{arg}(t_1) : \overline{s}) \\
(\mathsf{arg}(t_1) : \overline{s}, v_0) & \Rightarrow_v & (t_1, \mathsf{fun}(v_0) : \overline{s}) \\
(\mathsf{fun}(\lambda x.t) : \overline{s}, v_1) & \Rightarrow_v & (t\{v_1/x\}, \overline{s}) \\
(\mathsf{fun}(x\ v_1\ \ldots\ v_n) : \overline{s}, v) & \Rightarrow_v & (\overline{s}, x\ v_1\ \ldots\ v_n\ v)
\end{array}$$

- konfiguracja początkowa

$$(t, \epsilon)$$

- konfiguracja końcowe

$$(\epsilon, v)$$

Indeksy de Bruijna

Cel

- reprezentacja termów, która nie wymaga przemianowania zmiennych przy podstawieniu
- α-równoważne termy są identyczne

Rozwiązanie

- zmienne reprezentowane za pomocą liczb naturalnych (indeksy de Bruijna)
- indeks de Bruijna określa odległość (liczba dzielących je λ) zmiennej od wiążącej ją lambdy
- przykłady

$$\begin{aligned} \lambda x.x &\mapsto \lambda.0 \\ \lambda x.\lambda y.x\,(y\,x) &\mapsto \lambda.\lambda.1\,(0\,1) \end{aligned}$$

Indeksy de Bruijna, c.d.

- termy

$$t \quad ::= \quad n \mid \lambda.t \mid t\ t$$

- termy otwarte mają sens w pewnym kontekście

$$\Gamma = x_n, x_{n-1}, \ldots, x_1, x_0$$

  wiążącym nazwę x_i z indeksem i

- przykład ($\Gamma = c, b, a$)

$$\begin{array}{rcl} a\,(b\,c) & \mapsto & 0\,(1\,2) \\ \lambda x.a\,x & \mapsto & \lambda.1\,0 \\ \lambda x.\lambda y.c & \mapsto & \lambda.\lambda.4 \end{array}$$

Indeksy de Bruijna, c.d.

- podstawienie

$$\begin{aligned} k\{s/k\} &= s \\ k\{s/j\} &= k, \quad k \neq j \\ (\lambda.t)\{s/j\} &= \lambda.t\{\uparrow_0^1(s)/j+1\} \\ (t_0\ t_1)\{s/j\} &= t_0\{s/j\}\ t_1\{s/j\} \end{aligned}$$

- przesunięcie indeksów

$$\begin{aligned} \uparrow_c^d(k) &= k, \quad k < c \\ \uparrow_c^d(k) &= k+d, \quad k \geq c \\ \uparrow_c^d(\lambda.t) &= \lambda.\uparrow_{c+1}^d(t) \\ \uparrow_c^d(t_0\ t_1) &= \uparrow_c^d(t_0)\ \uparrow_c^d(t_1) \end{aligned}$$

- β-redukcja

$$(\lambda.t)\ s \rightarrow_\beta \uparrow_0^{-1}(t\{\uparrow_0^1(s)/0\})$$

Modelowy język funkcyjny (CBV)

- składnia

$$\begin{array}{rcl} t & ::= & x \mid \lambda x.t \mid t\ t \\ & \mid & 0 \mid \mathsf{suc}\ t \mid \mathsf{case}\ t\ \mathsf{of}\ 0 \Rightarrow t \parallel \mathsf{suc}\ x \Rightarrow t \\ & \mid & \mathsf{fix}\ x.t \end{array}$$

- syntaktyczny cukier

$$\begin{array}{rcl} \mathsf{let}\ x = t_1\ \mathsf{in}\ t_2 & \equiv & (\lambda x.t_2)\ t_1 \\ \mathsf{letrec}\ x = t_1\ \mathsf{in}\ t_2 & \equiv & \mathsf{let}\ x = \mathsf{fix}\ x.t_1\ \mathsf{in}\ t_2 \end{array}$$

- program = zamknięty term
- przykład programu

$$\begin{array}{l} \mathsf{letrec}\ \mathsf{add} = \lambda m.\lambda n.\mathsf{case}\ m\ \mathsf{of}\ 0 \Rightarrow n \parallel \mathsf{suc}\ k \Rightarrow \mathsf{add}\ k\ (\mathsf{suc}\ n) \\ \mathsf{in}\ \mathsf{add}\ 2\ 3 \end{array}$$

Modelowy język funkcyjny (CBV), c.d.

- wartości

$$\begin{array}{rcl} v & ::= & \mathrm{nv} \mid \lambda x.t \\ \mathrm{nv} & ::= & 0 \mid \mathrm{suc}\ \mathrm{nv} \end{array}$$

- semantyka programów

$$\frac{}{(\lambda x.t)\ v \to_v t\{v/x\}} \qquad \frac{t \to_v t'}{t\ s \to_v t'\ s} \qquad \frac{s \to_v s'}{v\ s \to_v v\ s'}$$

$$\frac{t \to_v t'}{\mathrm{suc}\ t \to_v \mathrm{suc}\ t'}$$

$$\frac{}{\mathrm{case}\ 0\ \mathrm{of}\ 0 \Rightarrow t \parallel \mathrm{suc}\ x \Rightarrow s \to_v t}$$

$$\frac{}{\mathrm{case}\ (\mathrm{suc}\ \mathrm{nv})\ \mathrm{of}\ 0 \Rightarrow t \parallel \mathrm{suc}\ x \Rightarrow s \to_v s\{\mathrm{nv}/x\}}$$

$$\frac{t \to_v t'}{\mathrm{case}\ t\ \mathrm{of}\ 0 \Rightarrow r \parallel \mathrm{suc}\ x \Rightarrow s \to_v \mathrm{case}\ t'\ \mathrm{of}\ 0 \Rightarrow r \parallel \mathrm{suc}\ x \Rightarrow s}$$

$$\frac{}{\mathrm{fix}\ x.t \to_v t\{\mathrm{fix}\ x.t/x\}}$$