

Lambda rachunek dla kursu Teoria programowania

Marek Zaionc

Universytet Jagielloński

Rachunek lambda.

Termy. Opisy funkcji obliczalnych. Zakładamy istnienie przeliczalnie wielu zmiennych. Reguły formowania termów.

- Zmienne są termami
- jeżeli $M \in \Lambda$ oraz $N \in \Lambda$ to $MN \in \Lambda$ (aplikacja)
- jeżeli $M \in \Lambda$ oraz zmienna $x \in \Lambda$ to $\lambda x.M \in \Lambda$ (abstrakcja)

Notacja.

Term MN_1N_2 oznacza $(MN_1)N_2$

Term $\lambda xy.M$ oznacza $\lambda x.(\lambda y.M)$

Równości na termach.

Dla termu M niech $FV(M)$ oznacza zbiór zmiennych wolnych (nie będących w obszarze działania żadnej λ) występujących w M .

Przykład. $FV((\lambda x.(\lambda y.x))x) = \{x\}$

Podstawienia.

$M[x := N]$ oznacza term powstały z termu M przez wstawienia za każde wolne wystąpienia zmiennej x termu N .

Term domknięty to term bez zmiennych wolnych czyli taki że $FV(M) = \emptyset$

Konwersje.

α konwersja

$$\begin{aligned} M &=_{\alpha} M \\ \lambda x.M &=_{\alpha} \lambda y.(M[x := y]) \text{ gdy } y \notin FV(M) \end{aligned}$$

β konwersja

$$(\lambda x.M)N \rightarrow_{\beta} M[x := N] \text{ gdy } \dots$$

$$M \rightarrow_{\beta} N \text{ to } \lambda x.M \rightarrow_{\beta} \lambda x.N$$

$$M \rightarrow_{\beta} N \text{ to } MZ \rightarrow_{\beta} NZ$$

$$M \rightarrow_{\beta} N \text{ to } ZM \rightarrow_{\beta} ZN$$

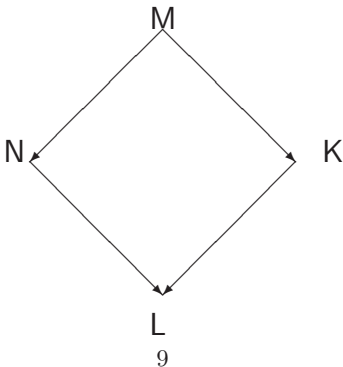
Relacje \rightarrow_{β} jest domknięciem przechodnim i zwrotnym relacji $\rightarrow_{\beta} \cup =_{\alpha}$.

Relacje $=_{\beta}$ jest domknięciem przechodnim symetrycznym i zwrotnym relacji $\rightarrow_{\beta} \cup =_{\alpha}$.

Term M jest w formie normalnej jeżeli nie ma takiego N że $M \rightarrow_{\beta} N$.

Twierdzenie Churcha Rossera (1936).

Dla dowolnych termów M, N, K takich że $M \rightarrow_{\beta} N$ oraz $M \rightarrow_{\beta} K$ istnieje term L taki że $N \rightarrow_{\beta} L$ oraz $K \rightarrow_{\beta} L$.



Twierdzenie (*) Dla dowolnych termów M, N takich że $M =_{\beta} N$ istnieje term L taki że $N \rightarrow_{\beta} L$ oraz $M \rightarrow_{\beta} L$.

oraz

Twierdzenie Dla dowolnych termów M, N, K takich że $M \rightarrow_{\beta} N$ oraz $M \rightarrow_{\beta} K$ oraz N i K są w formie normalnej to $K =_{\alpha} N$.

Przykłady β redukcji

$$I = \lambda x.x, \pi_1 = K = \lambda xy.x, \pi_2 = K^* = \lambda xy.y$$

Pojęcie pary uporządkowanej $\lambda yzx.xyz$ tak że $[P, Q] = \lambda x.xPQ$

Przykłady(*):

$$II \rightarrow_{\beta} I, K(II) \rightarrow_{\beta} \lambda x.(II), K(II) \rightarrow_{\beta} KI$$

zgodnie z **tw Churcha Rossera** istnieje taki wspólny term T że $\lambda x.(II) \twoheadrightarrow_{\beta} T$ oraz $KI \twoheadrightarrow_{\beta} T$. Tym termem jest w tym przypadku $\lambda x.I$.

$$[P, Q]\pi_1 \twoheadrightarrow_{\beta} (\lambda x.xPQ)\pi_1 \twoheadrightarrow_{\beta} \pi_1PQ \twoheadrightarrow_{\beta} P$$

Para nie jest surjektywna dlatego NIE zachodzi

$$[M\pi_1, M\pi_2] =_{\beta} M$$

$$\omega = \lambda x . x x$$

$$\Omega = \omega \omega$$

$$\Omega = \omega \omega = (\lambda x . x x)(\lambda x . x x) \rightarrow_{\beta} (\lambda x . x x)(\lambda x . x x)$$

Wyrażalność algorytmów w rachunku lambda.

Numerale Churcha.

$$\underline{n} = \lambda s x. s(\dots(sx)\dots)$$

Funkcja $f : \mathbb{N}^k \rightarrow \mathbb{N}$ jest λ definiowalna jeżeli istnieje taki term domknięty F że dla każdego liczb naturalnych $n_1 \dots n_k$ zachodzi

$$F \underline{n_1} \dots \underline{n_k} =_{\beta} \underline{f(n_1, \dots, n_k)}$$

Przykładowe termy opisujące funkcje obliczalne.

$\lambda n m s x. n s (m s x)$ opisuje dodawanie

$\lambda n m s x. n (\lambda z. m s z) x$ opisuje mnożenie

$\lambda n m k s x. n (\lambda y. k s y) (m s x)$ opisuje

$(n, m, k) \mapsto \text{if } n = 0 \text{ then } m \text{ else } k$

Twierdzenie (Kleene)

Wszystkie funkcje rekurencyjne są λ definiowalne.

Twierdzenie dowodzi się poprzez zastosowanie termu opisującego operator punktu stałego czyli takiego termu Y że dla dowolnego termu F

$$F(Y F) =_{\beta} Y F$$

Przykładowo takim Y może być term

$$Y = \lambda y.(\lambda x.y(xx))(\lambda x.y(xx))$$