Automaty a Rekurencja - notatki z wykładu 3

Marek Sommer
prowadzący wykład: Paweł Parys

2016-10-18

1 Rachunek λY z typami

1.1 Opis

- Typy proste: o, $o \to o$, $(o \to o) \to o \to o$, ... (takie same jak dla schematów rekurencyjnych).
- Nie ma nieterminali.
- Termy:

Nazwa termu	Przykład	Otypowanie
Zmienna	x	x^α, gdzie α jest typem prostym (np. $o \to o$)
Konstruktor drzewa	$a\ K_1\ \dots\ K_{\mathrm{rank}(a)}$	$\left(a\ K_1^o\ \dots\ K_{\mathrm{rank}(a)}^o\right)^o$
Aplikacja	$K\ L$	$\left(K^{\alpha\to\beta}\ L^\alpha\right)^\beta$
λ-abstrakcja	$\lambda x.\, K$	$\left(\lambda x^\alpha.\, K^\beta\right)^{\alpha\to\beta}$
Operator rekurencji	$Yx.\, K$	$(Yx^\alpha.\, K^\alpha)^\alpha$

1.2 Rodzaje zmiennych

W termach mogą występować zmienne dwóch rodzajów:

- *Zmienne związane* – są to zmienne wprowadzone wewnątrz termu przez pewną λ-abstrakcję lub przez pewien operator rekurencji Y.

$\lambda x.\,(\ \dots\ x\ \dots\ )$

Zmienna związana przez tę λ.

$Yx.\,(\ \dots\ x\ \dots\ )$

Zmienna związana przez ten operator rekurencji Y.

- *Zmienne wolne* – są to wszystkie zmienne, które nie są związane. Zbiór zmiennych wolnych termu K oznacza się przez $\mathrm{FV}(K)$.

Termy różniące się tylko nazwami zmiennych związanych uznajemy za identyczne.

Przykład: $\lambda x.\, \lambda x.\, a\ x = \lambda x.\, \lambda y.\, a\ y$

1.3 Podstawienia

Jeśli K, L są termami, $x \in \mathrm{FV}(K)$, oraz L jest tego samego typu co x, to można zastosować podstawienie:

$$K\,[L/x]\,.$$

Ten napis oznacza wynik podstawienia termu L pod wszystkie wystąpienia zmiennej wolnej x w termie K.

Przykład:

$$(\lambda y.\, a\ x\ y)\,[L/x] = \lambda y.\, a\ L\ y$$

Jeśli y jest zmienną związaną, to pozostaje bez zmian:

$$(\lambda y.\, a\ x\ y)\,[L/y] = \lambda y.\, a\ x\ y$$

Poniższe podstawienie jest niepoprawne ze względu na kolizję nazw zmiennych:

$$\cancel{(\lambda y.\, a\ x\ y)\,[(b\ y)/x] = \lambda y.\, a\ (b\ y)\ y}$$

Aby uniknąć problemów z powtarzającymi się nazwami zmiennych, warto używać następującej procedury w celu wykonania prawidłowego podstawienia $K\,[L/x]$:

1. Zmień nazwy zmiennych związanych w termie K, aby nie pokrywały się z nazwami zmiennych wolnych termu L.
2. Podstaw L za wszystkie wystąpienia zmiennej wolnej x w termie K.

Poprawne podstawienie z poprzedniego przykładu:

$$(\lambda y.\, a\ x\ y)\,[(b\ y)/x] = (\lambda z.\, a\ x\ z)\,[(b\ y)/x] = \lambda z.\, a\ (b\ y)\ z$$

1.4 Redukcje

1.4.1 β-redukcja

$$\left(\left(\lambda x^{\alpha}.\, K^{\beta}\right)^{\alpha\to\beta}\ L^{\alpha}\right)^{\beta} \longrightarrow_{\beta}\ K^{\beta}\,[L^{\alpha}/x^{\alpha}]$$

β-redukcję można wykonać na dowolnym podtermie całego termu, niekoniecznie tylko na najbardziej zewnętrznej aplikacji. Aby dokładnie zdefiniować taką rozszerzoną redukcję, należy ją rekurencyjnie zdefiniować dla każdego rodzaju termu. Zatem, jeśli $K \longrightarrow_{\beta} K'$, $L \longrightarrow_{\beta} L'$, oraz $K_i \longrightarrow_{\beta} K_i'$, to definiujemy:

- $K\ L \longrightarrow_{\beta} K'\ L$,
- $K\ L \longrightarrow_{\beta} K\ L'$,
- $\lambda x.\, K \longrightarrow_{\beta} \lambda x.\, K'$,

- $Yx.\,K \longrightarrow_\beta Yx.\,K'$,
- $a\ K_1\ \ldots\ K_i\ \ldots\ K_n \longrightarrow_\beta a\ K_1\ \ldots\ K_i'\ \ldots\ K_n$, dla każdego $i \in \{1, \ldots, n\}$.

1.4.2 δ-redukcja

$$Yx.\,M \longrightarrow_\delta M\,[Yx.\,M/x]$$

δ-redukcję można wykonać na dowolnym podtermie całego termu, niekoniecznie tylko na najbardziej zewnętrznym operatorze rekurencyjnym. Aby dokładnie zdefiniować taką rozszerzoną redukcję, to podobnie jak z β-redukcją, należy ją rekurencyjnie zdefiniować dla każdego rodzaju termu. Zatem, jeśli $K \longrightarrow_\delta K'$, $L \longrightarrow_\delta L'$, oraz $K_i \longrightarrow_\delta K_i'$, to definiujemy:

- $K\ L \longrightarrow_\delta K'\ L$,
- $K\ L \longrightarrow_\delta K\ L'$,
- $\lambda x.\,K \longrightarrow_\delta \lambda x.\,K'$,
- $Yx.\,K \longrightarrow_\delta Yx.\,K'$,
- $a\ K_1\ \ldots\ K_i\ \ldots\ K_n \longrightarrow_\delta a\ K_1\ \ldots\ K_i'\ \ldots\ K_n$, dla każdego $i \in \{1, \ldots, n\}$.

1.4.3 Domknięcie zwrotno-przechodnie redukcji

Definiuje się również następujące relacje: dla każdej redukcji, następująco nazywamy jej domknięcie zwrotno-przechodnie:

- $\longrightarrow_\beta^*$ – domknięcie zwrotno-przechodnie β-redukcji,
- $\longrightarrow_\delta^*$ – domknięcie zwrotno-przechodnie δ-redukcji,
- $\longrightarrow_{\beta\delta}^*$ – domknięcie zwrotno-przechodnie ($\longrightarrow_\beta \cup \longrightarrow_\delta$).

1.5 Generowanie drzew

Niech K będzie termem zamkniętym (bez zmiennych wolnych), typu o.

- Jeśli $K \longrightarrow_{\beta\delta}^* a\ K_1\ \ldots\ K_n$, to drzewo generowane przez term K ma a w korzeniu, a pod korzeniem są poddrzewa generowane przez K_1, …, K_n.
- Natomiast jeśli $K \not\longrightarrow_{\beta\delta}^* a\ K_1\ \ldots\ K_n$, to drzewo generowane przez term K ma ω w korzeniu.

1.5.1 Uogólnienie gdy term zawiera zmienne wolne (Drzewo Böhma)

Niech K będzie dowolnym termem.

- Jeśli $K \longrightarrow_{\beta\delta}^* \lambda x_1. \ldots \lambda x_k.\, a\ K_1\ \ldots\ K_n$, to w korzeniu jest $\lambda x_1. \ldots \lambda x_k.\, a$, a poddrzewa to drzewa generowane przez K_1, …, K_n.
- Jeśli $K \longrightarrow_{\beta\delta}^* \lambda x_1. \ldots \lambda x_k.\, y\ K_1\ \ldots\ K_n$, gdzie y jest zmienną wolną, to w korzeniu jest $\lambda x_1. \ldots \lambda x_k.\, y$, a poddrzewa to drzewa generowane przez K_1, …, K_n.

1.5.2 Jednoznaczność drzewa

Lemat (Własność Churcha-Rossera)**.** *Drzewo generowane przez term jest jednoznacznie wyznaczone.*

Dowód. Być może pojawi się kiedyś na zajęciach. □

2 Rachunek λ bez typów

Rozważa się rachunek λ bez typów. Jest on Turing-zupełny, tzn. może symulować maszynę Turinga.

2.0.3 Rekurencja

Aby uzyskać rekurencję w rachunku λ bez typów, nie potrzeba operatora rekurencji Y.

$$Y = Z\ Z, \text{ dla } Z = \lambda z.\, \lambda x.\, x\ (z\ z\ x)$$

$$\underset{\substack{\uparrow \\ \text{W rachunku } \lambda \text{ bez typów.}}}{Y\ \ (\lambda x.\, M)} \qquad \approx \qquad \underset{\substack{\uparrow \\ \text{W rachunku } \lambda Y \text{ z typami.}}}{Y x.\, M}$$

Sprawdzenie:

$$Y\ \ (\lambda x.\, M) = Z\ Z\ \ (\lambda x.\, M)\ \longrightarrow_\beta^*\ (\lambda x.\, M)\ \ (Z\ Z\ \ (\lambda x.\, M))$$

3 Sprowadzenie deterministycznego HORS do rachunku λY z typami

Niech A_1, ..., A_k będą nieterminalami, Niech A_k będzie nieterminalem startowym. Niech reguły będą następujące: $A_i\ x_1\ \ldots\ x_{n_i}\ \longrightarrow\ M_i$.

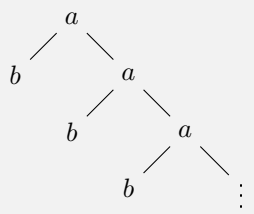
Oznaczmy $\mathcal{R}\,(A_i) = \lambda x_1.\, \ldots\, \lambda n_i.\, M_i$. Zdefiniujmy termy:

Term	Zmienne wolne
$T_1 = Y A_1.\, \mathcal{R}\,(A_1)$	$A_2, A_3, \ldots, A_k$
$T_2 = Y A_2.\, (\mathcal{R}\,(A_2)\,[T_1/A_1])$	$A_3, A_4, \ldots, A_k$
$\vdots$	$\vdots$
$T_k = Y A_k.\, ((\ldots((\mathcal{R}\,(A_k)\,[T_1/A_1])\,[T_2/A_2])\ldots)\,[T_{k-1}/A_{k-1}])$	$\emptyset$

Term T_k generuje to samo drzewo w rachunku λY z typami co term A_k w poprzednim modelu.

Przykład.

$$\begin{array}{rcl|l} A\ x & \longrightarrow & a\ x\ (A\ x) & \text{rank}(a) = 2 \\ S & \longrightarrow & A\ b & \text{rank}(b) = 0 \end{array}$$



Transformacja:

$Y A.\, \lambda x.\, a\ x\ \ (A\ x)$ term dla A

$Y S.\, (Y A.\, \lambda x.\, a\ x\ \ (A\ x))\ \ b$ term dla S

$\downarrow \delta$-redukcja

$(Y A.\, \lambda x.\, a\ x\ \ (A\ x))\ \ b$

$\downarrow \delta$-redukcja

$\lambda x.\, a\ x\ \ ((Y A.\, \lambda x.\, a\ x\ \ (A\ x))\ \ x)\ \ b$

$\downarrow \beta$-redukcja

$a\ b\ \ ((Y A.\, \lambda x.\, a\ x\ \ (A\ x))\ \ b)$

Na poniższym przykładzie można przekonać się, że otrzymany term może być wykładnoczo większy od schematu rekurencyjnego z którego zaczynaliśmy:

$$A_0 \to b \qquad\qquad A_k \to a\, A_{k-1}\, A_{k-1} \qquad \text{dla } 1 \leq k \leq n\,.$$

Term odpowiadający nieterminalowi A_n będzie po prostu drzewem mającym 2^n liści. Jednocześnie nietrudno przekonać się, że wzrost nie będzie nigdy większy niż wykładniczy.

Możemy jednak zdefiniować „rozmiar grafowy" termu. *Rozmiarem grafowym* termu M nazwiemy sumę po wszystkich różnych jego podtermach rozmiarów tych podtermów.[1] Widzimy, że liczba podtermów termu M powstałego ze schematu rekurencyjnego $\mathcal{S}$ będzie wielomianowa (a nawet liniowa) względem rozmiaru $\mathcal{S}$. Będzie tak dlatego, że każdy podterm termu M odpowiada pewnemu podtermowi prawej strony jednej z reguł $\mathcal{S}$. Również typy podtermów M będą typami podtermów prawych stron reguł $\mathcal{S}$, nie będą więc bardziej skomplikowane niż typy nieterminali $\mathcal{S}$. Zatem rozmiar grafowy termu M będzie wielomianowy względem rozmiaru schematu $\mathcal{S}$.

[1] Jest to więc z grubsza liczba różnych podtermów. Dodajemy jednak rozmiary typów, ze względu na termy takie jak $(\lambda x^{\alpha\to o}.a)\,(\lambda y^{\alpha}.a)$; w tym termie, chociaż jest zamkniętym termem typu o, typ α może być dowolnie skomplikowany. Dość łatwo można podać reprezentację termu (za pomocą grafu podtermów) mieszczącą się w pamięci proporcjonalnej do jego rozmiaru grafowego.