autor dokumentu: Rafał Stefański

Tłumaczenie λY-termów na schematy rekurencyjne

Pokażemy że dla każdego λY-termu można skonstruować schemat rekurencyjny który generuje to samo drzewo.

Weźmy dowolny zamknięty (czyli bez zmiennych wolnych) λY-term M_0 typu o i o złożoności n. Na początek skonstruujmy równoważny schemat rekurencyjny rzędu co najwyżej $n+1$. Zbiorem nieterminali tego schematu będzie $R = \{\langle N\rangle \mid N \text{ jest podtermem } M_0\}$, a nieterminal startowy to $\langle M_0\rangle$. Oznaczmy przez $\overrightarrow{\mathrm{FV}(N)}$ wektor zmiennych wolnych λY-termu N (zakładamy w tym celu jakiś porządek liniowy na nazwach zmiennych). Jeśli $\overrightarrow{\mathrm{FV}(N)} = (x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n})$ oraz N jest typu α, to nieterminal $\langle N\rangle$ jest typu $\alpha_1{\to}\alpha_2{\to}\ldots{\to}\alpha_n{\to}\alpha$.

Podajmy reguły wyprowadzeń, zaczynając od reguły dla aplikacji:

$$\langle N_1\, N_2\rangle\, \overrightarrow{\mathrm{FV}(N_1\, N_2)}\, \overrightarrow{t} \longrightarrow \langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}\, \left(\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}\right)\, \overrightarrow{t}\,.$$

Zatrzymajmy się na chwilę przy tej regule. Symbol $\langle N_1\, N_2\rangle$ jest typu $\alpha_1{\to}\alpha_2{\to}\ldots{\to}\alpha_n{\to}\alpha$, gdzie α to typ termu $N_1\, N_2$. Możemy założyć, że $\alpha = \beta_1{\to}\beta_2{\to}\ldots{\to}\beta_m{\to}o$. Przy takich oznaczeniach wektor $\overrightarrow{\mathrm{FV}(N_1\, N_2)}$ odpowiada za pierwsze n zmiennych typu $\alpha_1, \alpha_2, \ldots, \alpha_n$, a wektor $\overrightarrow{t}$ odpowiada za kolejne m zmiennych typu $\beta_1, \beta_2, \ldots, \beta_m$. Pierwszą część argumentów stanowi, w związku z tym, wartościowanie zmiennych wolnych termu $N_1\, N_2$. Rozdzielamy je pomiędzy termy N_1 i N_2 (każda ze zmiennych wolnych może występować tylko w N_1, tylko w N_2, bądź jednocześnie w N_1 i N_2) i otrzymujemy wyrażenia $\langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}$ oraz $\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}$, o typach równych odpowiednio typom termów N_1 i N_2. Skoro term $N_1\, N_2$ jest poprawnym termem, to N_1 musi być typu $\gamma{\to}\alpha$ a, N_2 typu γ, dla pewnego typu γ. Możemy, więc podać $\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}$ jako argument do $\langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}$, otrzymując $\langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}\, \left(\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}\right)$, typu α. Przypominając sobie, że $\alpha = \beta_1{\to}\beta_2{\to}\ldots{\to}\beta_m{\to}o$, możemy podać $\overrightarrow{t}$ jako argumenty do otrzymanego wyrażenia, tak by otrzymać $\langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}\, \left(\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}\right)\, \overrightarrow{t}$ typu o.

Poniższe reguły wyprowadzeń opierają się na tej podobnej zasadzie:

- aplikacja: $\langle N_1 N_2\rangle\, \overrightarrow{\mathrm{FV}(N_1\, N_2)}\, \overrightarrow{t} \longrightarrow \langle N_1\rangle\, \overrightarrow{\mathrm{FV}(N_1)}\, \left(\langle N_2\rangle\, \overrightarrow{\mathrm{FV}(N_2)}\right)\, \overrightarrow{t}$,
- lambda-abstrakcja: $\langle \lambda y.P\rangle\, \overrightarrow{\mathrm{FV}(\lambda y.P)}\, y\, \overrightarrow{t} \longrightarrow \langle P\rangle\, \overrightarrow{\mathrm{FV}(P)}\, \overrightarrow{t}$,
- zmienna: $\langle y\rangle\, y\, \overrightarrow{t} \longrightarrow y\, \overrightarrow{t}$,
- konstruktor drzewa: $\langle a\, N_1\, N_2\, \ldots\, N_k\rangle\, \overrightarrow{\mathrm{FV}(a\, N_1\, N_2\, \ldots\, N_k)} \longrightarrow a\, \left(N_1\, \overrightarrow{\mathrm{FV}(N_1)}\right)\, \left(N_2\, \overrightarrow{\mathrm{FV}(N_2)}\right)\, \ldots\, \left(N_k\, \overrightarrow{\mathrm{FV}(N_k)}\right)$,
- Y(1): $\langle Yx.P\rangle\, \overrightarrow{\mathrm{FV}(Yx.P)}\, \overrightarrow{t} \longrightarrow \langle P\rangle\, \overrightarrow{z_1}\, \left(\langle Yx.P\rangle\, \overrightarrow{\mathrm{FV}(Yx.P)}\right)\, \overrightarrow{z_2}\, \overrightarrow{t}$ gdy $\overrightarrow{\mathrm{FV}(P)} = (\overrightarrow{z_1}, x, \overrightarrow{z_2})$,
- Y(2): $\langle Yx.P\rangle\, \overrightarrow{\mathrm{FV}(Yx.P)}\, \overrightarrow{t} \longrightarrow \langle P\rangle\, \overrightarrow{\mathrm{FV}(P)}\, \overrightarrow{t}$ gdy $\overrightarrow{x} \notin \mathrm{FV}(P)$.

Ten, raczej intuicyjnie utworzony, schemat rekurencyjny istotnie generuje to samo drzewo, co wyjściowy λY-term. Precyzyjny dowód zostawiamy, jako ćwiczenie, dla słuchacza.

Przykład:
Dla λY-termu $M_0 = Yx.a\, x\, x$, otrzymujemy następujący schemat:

$$\begin{aligned}
&\langle M_0\rangle \longrightarrow \langle a\, x\, x\rangle\, \langle M_0\rangle\\
&\langle a\, x\, x\rangle\, x \longrightarrow a\, (\langle x\rangle\, x)\, (\langle x\rangle\, x)\\
&\langle x\rangle\, x \longrightarrow x
\end{aligned}$$

Powyższy przykład pokazuje, że rząd wynikowego schematu może wzrosnąć o 1 względem złożoności λY-termu. W powyższym przykładzie złożoność M_0 wynosi 0—wszystkie jego podtermy ($Yx.a\, x\, x$, $a\, x\, x$ oraz x) mają typ o. Natomiast wynikowy schemat jest rzędu 1, ponieważ nieterminal $\langle axx\rangle$ jest typu $o{\to}o$. W dalszej części wykładu pokażemy, że można tak przetłumaczyć każdy λY-term na schemat rekurencyjny, by rząd wynikowego schematu był nie większy niż złożoność tego λY-termu. Zdefiniujmy w tym celu postać kanoniczną dla λY-termów.

Postać kanoniczna λY termów

Weźmy dowolny zamknięty λY-term M_0 typu o. Dla każdego N, będącego podtermem M_0 możemy zdefiniować rozłączne zbiory:

- $\mathrm{FV}_\lambda(N)$ – zmienne wolne wyrażenia N, które w M_0 są wiązane przez wyrażenie $\lambda x.(\dots)$,
- $\mathrm{FV}_Y(N)$ – zmienne wolne wyrażenia N, które w M_0 są wiązane przez wyrażenie $Yy.(\dots)$.

Aby zrobić to bardziej formalnie, powinniśmy podzielić zbiór wszystkich nazw zmiennych na rozłączne zbiory Var_λ i Var_Y oraz zakładać (bez straty ogólności), że wyrażenia $\lambda x.(\dots)$ używają wyłącznie nazw zmiennych $x \in \mathrm{Var}_\lambda$, natomiast wyrażenia $Yy.(\dots)$ wyłącznie nazw zmiennych $y \in \mathrm{Var}_Y$. Wówczas $\mathrm{FV}_\lambda(N)$ oraz $\mathrm{FV}_Y(N)$ nie zależą od nadtermu M_0.

Przykładowo dla $M_0 = (\lambda f.Yx.\underline{f\,x})\,(\lambda z.a\,z)$ oraz podkreślonego podtermu $N = f\,x$, $\mathrm{FV}_\lambda(N) = f$, oraz $\mathrm{FV}_Y(N) = x$.

Powiemy, że λY-term jest w postaci kanonicznej, gdy dla każdego jego podtermu N postaci $Yx.P$, $\mathrm{FV}_\lambda(N) = \varnothing$. Czyli w każdym jego podtermie postaci $Yx.P$ wszystkie zmienne wolne pochodzą z wyrażeń postaci $Yz.(\dots)$.

Na marginesie warto dodać, że opisana na poprzednim wykładzie konstrukcja λY-termów ze schematów rekurencyjnych produkuje termy w postaci kanonicznej.

λY-term M_0 z poprzedniego przykładu nie jest w postaci kanonicznej, gdyż jego podterm $K = Yx.f\,x$ zawiera zmienną wolną f, która pochodzi z wyrażenia λ (innymi słowy: $\mathrm{FV}_\lambda(K) = \{f\} \neq \varnothing$). Można jednak ten term przepisać tak by był w postaci kanonicznej i generował to samo drzewo:

$$M_0' = (\lambda\varphi.(Yx.\lambda f.f\,(x\,f))\,\varphi)\,(\lambda z.a\,z)$$

Nie jest to przypadek—każdy term można przepisać w taki sposób, o czym mówi lemat, który za chwilę udowodnimy.

Lemat. *Każdy zamknięty λY-term typu o można przekształcić do termu w postaci kanonicznej, który generuje to samo drzewo, co term wyjściowy, nie zwiększając przy tym jego złożoności.*

Dowód tego lematu wygląda następująco. Idąc od zewnątrz wyrażenia M_0 zastępujemy wszystkie podtermy postaci $Yx.P$ na

$$(Yz.\lambda\overrightarrow{y}.P[z\,\overrightarrow{y}/x])\,\overrightarrow{y}\,, \qquad \text{gdzie } \overrightarrow{y} = \overrightarrow{\mathrm{FV}_\lambda(P)}\,.$$

Tym sposobem wiążemy wszystkie zmienne z $\mathrm{FV}_\lambda(P)$ przez lambdę znajdującą się wewnątrz $Yz.(\dots)$, dzięki czemu żadna zmienna wewnątrz wyrażenia Y nie jest wiązana przez zewnętrzną lambdę.

Formalnie, możemy to zdefiniować za pomocą następującej operacji naprawiania, przypisującej dowolnemu termowi N term $(N)\!\downarrow$. Dla zmiennej nic nie zmieniamy, a dla aplikacji, lambdy i konstruktora drzewa naprawiamy podtermy:

$$(x)\!\downarrow = x\,, \qquad (K\,L)\!\downarrow = (K)\!\downarrow\,(L)\!\downarrow\,, \qquad (\lambda x.K)\!\downarrow = \lambda x.(K)\!\downarrow \qquad (a\,K_1\,\dots\,K_r)\!\downarrow = a\,(K_1)\!\downarrow\,\dots\,(K_r)\!\downarrow\,.$$

Pozostaje przypadek operatora Y, w którym postępujemy zgodnie z zapowiedzią:

$$(Yx.P)\!\downarrow = (Yz.\lambda\overrightarrow{y}.(P[z\,\overrightarrow{y}/x])\!\downarrow)\,\overrightarrow{y}\,, \qquad \text{gdzie } \overrightarrow{y} = \overrightarrow{\mathrm{FV}_\lambda(P)}\,. \tag{1}$$

Powyższa definicja jest przez indukcję po parze: liczba operatorów Y w termie, rozmiar termu (chociaż term $P[z\,\overrightarrow{y}/x]$ nie musi być mniejszy niż $Yx.P$, to jednak zawiera o jeden operator Y mniej).

Łatwo zauważyć, że $\mathrm{FV}_\lambda((N)\!\downarrow) = \mathrm{FV}_\lambda(N)$ dla każdego termu N. Wynika z tego, że w przypadku operatora Y mamy $\mathrm{FV}_\lambda(\lambda\overrightarrow{y}.(P[z\,\overrightarrow{y}/x])\!\downarrow) = \emptyset$, gdyż $\mathrm{FV}_\lambda((P[z\,\overrightarrow{y}/x])\!\downarrow) = \mathrm{FV}_\lambda((P[z\,\overrightarrow{y}/x]))$ zawiera dokładnie zmienne z $\overrightarrow{y}$. Zatem $(M_0)\!\downarrow$ jest w postaci kanonicznej.[1]

Pozostaje uzasadnić, że operacja naprawiania nie zwiększa złożoności. Konkretnie: uzasadnimy, że jeśli $\mathrm{comp}(N) \leq n$ oraz wszystkie zmienne z $\mathrm{FV}_\lambda(N)$ są rzędu co najwyżej $n-1$, to $\mathrm{comp}((N)\!\downarrow) \leq n$. Przypadek zmiennej jest oczywisty. W przypadku aplikacji i konstruktora drzewa korzystamy bezpośrednio z założenia indukcyjnego oraz z tego, że termy $(N)\!\downarrow$ i N mają ten sam typ. Dla lambda-abstrakcji $\lambda x.K$ podobnie; ważne jest przy tym, że $\mathrm{ord}(x) < \mathrm{ord}(\lambda x.K) \leq n$.

[1] Zauważmy, że gdybyśmy w (1) zastąpili $(P[z\,\overrightarrow{y}/x])\!\downarrow$ przez $(P)\!\downarrow[z\,\overrightarrow{y}/x]$ (czyli wykonywali zamiany zaczynając od wewnątrz termu, a nie od zewnątrz), to wynikowy term nie musiałby być w postaci kanonicznej. Może się bowiem zdarzyć, że podstawienie $z\,\overrightarrow{y}$ za x popsuje kanoniczność termu $(P)\!\downarrow$—w $(P)\!\downarrow$ może istnieć podterm postaci $Yt.(\dots)$, w którym niektóre zmienne z $\overrightarrow{y}$ nie występują, natomiast po podstawieniu $z\,\overrightarrow{y}$ za x zaczną występować.

Pozostaje przypadek termu $Yx.P$. Przyjmijmy oznaczenia jak w (1); niech dodatkowo α będzie typem zmiennej x oraz niech $\overrightarrow{y} = (y_1^{\beta_1}, \ldots, y_k^{\beta_k})$. Wówczas podtermy $z\, y_1 \ldots y_i$ (dla $0 \leq i \leq k$) mają typ $\beta_{i+1} \to \ldots \to \beta_k \to \alpha$. Rząd tego typu jest nie większy niż n, gdyż x (czyli α) ma rząd nie większy niż n, a zmienne y_j (czyli typy β_j) mają rząd nie większy niż $n-1$. Wynika z tego, że $\mathrm{comp}(P[z\,\overrightarrow{y}/x]) \leq n$, czyli z założenia indukcyjnego także $\mathrm{comp}((P[z\,\overrightarrow{y}/x])\downarrow) \leq n$. Termy $\lambda y_{i+1}.\cdots.\lambda y_k.(P[z\,\overrightarrow{y}/x])\downarrow$ oraz $(Yz.\lambda\overrightarrow{y}.(P[z\,\overrightarrow{y}/x])\downarrow)\, y_1 \ldots y_i$ (dla $0 \leq i \leq k$) mają typ $\beta_{i+1} \to \ldots \to \beta_k \to \alpha$, który jest rzędu co najwyżej n, a zatem $\mathrm{comp}((Yx.P)\downarrow) \leq n$.

Tłumaczenie λY-termów w postaci kanonicznej na schematy rekurencyjne, z zachowaniem rzędu

Zachowująca rząd konstrukcja schematu rekurencyjnego generującego to samo drzewo co dany λY-term M_0 jest bardzo podobna do poprzedniej konstrukcji. Zbiór nieterminali pozostaje ten sam. Tym razem podczas określania typu nieterminala patrzymy tylko na zmienne wolne wprowadzone przez lambda abstrakcję. Tak więc jeśli $\overrightarrow{\mathrm{FV}_\lambda(N)} = (x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n})$ oraz N jest typu α, to nieterminal $\langle N \rangle$ jest typu $\alpha_1 \to \alpha_2 \to \ldots \to \alpha_n \to \alpha$. Większość reguł wyprowadzenia pozostaje praktycznie bez zmian. Różnią się tylko reguły wyprowadzenia dla zmiennych i Y-termów:

- Y: $\langle Yx.P \rangle\, \overrightarrow{\mathrm{FV}_\lambda(P)}\, \overrightarrow{t} \longrightarrow \langle P \rangle\, \overrightarrow{\mathrm{FV}(P)}\, \overrightarrow{t}$,
- zmienna(1): $\langle x \rangle\, x\, \overrightarrow{t} \longrightarrow x\, \overrightarrow{t}$, gdy zmienna y została wprowadzona przez term postaci $\lambda x.(\ldots)$,
- zmienna(2): $\langle y \rangle\, \overrightarrow{t} \longrightarrow \langle P \rangle\, \overrightarrow{t}$, gdy zmienna y została wprowadzona przez term $Yy.P$.

Chcąc, aby term P w regule „zmienna(2)" był jednoznacznie wyznaczony, musimy założyć, że każde wyrażenie postaci $Yy.P$ używa innej nazwy zmiennej. Założenie to możemy poczynić nie tracąc przy tym ogólności. Zauważmy też, że reguła ta ma sens tylko dlatego, że $\mathrm{FV}_\lambda(P) = \emptyset$: gdyby tak nie było, to powinniśmy podać do $\langle P \rangle$, jako początkowe argumenty, wartości podstawione za zmienne wolne, a wartości tych w tym miejscu nie mielibyśmy skąd wziąć.

Przykład:
Dla λY-termu $M_0 = Yx.a\,x\,x$ (będącego w postaci kanonicznej) otrzymujemy następujący schemat rzędu 0:

$$\begin{aligned} \langle M_0 \rangle &\longrightarrow \langle a\,x\,x \rangle \\ \langle a\,x\,x \rangle &\longrightarrow a\,\langle x \rangle\,\langle x \rangle \\ \langle x \rangle &\longrightarrow \langle M_0 \rangle \end{aligned}$$

Rozmiar powstałego schematu

Przypomnijmy, że nieterminale powstałego schematu odpowiadają podtermom naszego λY-termu. Typy tych nieterminali są nieco bardziej złożone niż typy odpowiadających im podtermów, ale różnią się tylko tym, że dodajemy (jednorazowo) jako argumenty listę zmiennych wolnych danego podtermu; wiemy jednak, że każda z tych zmiennych jest także podtermem całego termu. Ponadto prawe strony reguł są niewielkie, ich rozmiary są proporcjonalne do rozmiarów lewych stron. Wynika z tego, że rozmiar powstałego schematu jest wielomianowy względem rozmiaru grafowego λY-termu, od którego zaczynamy (czyli sumy, po wszystkich jego różnych podtermach, rozmiarów typów tych podtermów).

Uzyskana zależność jest odwrotna do tej z poprzedniego wykładu, gdzie dla danego schematu rekurencyjnego stworzyliśmy równoważny λY-term mający rozmiar grafowy (ale niekoniecznie klasyczny rozmiar) proporcjonalny do rozmiaru schematu rekurencyjnego.