

8. WYKŁAD 8: RACHUNEK λ . WPROWADZENIE.

Rachunek lambda i logika kombinatoryczna powstały w latach trzydziestych dwudziestego wieku. Początkowo miały stanowić alternatywne wobec teorii mnogości podejście do podstaw matematyki, w którym funkcja (rozumiana intuicyjnie jako definicja) była pojęciem pierwotnym. Ten zamiar się nie powiódł, ale szybko okazało się, że rachunek lambda jest niezwykle użytecznym aparatem w teorii obliczeń. Definicję funkcji obliczalnej sformułowano wcześniej za pomocą rachunku lambda, niż w języku maszyn Turinga. Później, w związku z rozwojem informatyki, język rachunku lambda okazał się niezastąpionym narzędziem w teorii języków programowania. A wreszcie i w praktyce, gdy pojawiły się języki programowania funkcyjnego. Dzisiaj znajomość podstaw rachunku lambda jest niezbędnym elementem wykształcenia nowoczesnego informatyka.

W rachunku λ rozważamy obiekty zwane **λ -termami**. Tradycyjnie λ -termy definiowane są jako formalne wyrażenia pewnego języka, podobnie jak zwykle termy w logice pierwszego rzędu. Istotna różnica polega na tym, że lambda-termi pozostające w pewnej relacji równoważności (α -konwersji) są ze sobą utożsamiane (uważane za identyczne). A więc λ -termy to w istocie nie wyrażenia, ale klasy abstrakcji **λ -wyrażeń** (nazywanych też **pre-termami**), które teraz zdefiniujemy.

Zbiór λ -wyrażeń rachunku λ konstruuje się indukcyjnie jako zbiór słów nad alfabetem $V \cup \{(\cdot), \lambda\}$, gdzie V jest przeliczalnym zbiorem zmiennych. Do zbioru Λ należą tylko λ -wyrażenia skonstruowane w poniższy sposób:

- Dowolna zmienna ze zbioru V jest λ -wyrażeniem rachunku λ , czyli jeżeli $x \in V$, to $x \in \Lambda$.
- Jeżeli weźmiemy dowolne λ -wyrażenie M ze zbioru Λ , oraz dowolną zmienną x ze zbioru V , to słowo postaci $(\lambda x.M)$ jest λ -wyrażeniem rachunku λ , czyli jeżeli $M \in \Lambda$ i $x \in V$, to $(\lambda x.M) \in \Lambda$. Powyższy sposób tworzenia λ -wyrażeń nazywamy **abstrakcją**.
- Jeżeli weźmiemy dowolne λ -wyrażenia M i N ze zbioru Λ , to słowo postaci (MN) jest λ -wyrażeniem rachunku λ . Powyższy sposób tworzenia λ -wyrażeń nazywamy **aplikacją**.

Intuicyjny sens aplikacji (MN) to zastosowanie operacji M do argumentu N . Abstrakcję $(\lambda x.M)$ interpretujemy natomiast jako definicję operacji (funkcji), która argumentowi x przypisuje M . Oczywiście x może występować w M , tj. M zależy od x .

Nawiasy są często pomijane według zasady łączności lewostronnej: na przykład $MNPQ \equiv (((MN)P)Q)$. Wieloargumentowe funkcje są reprezentowane przez funkcje jednoargumentowe, których argumentami są funkcje. Zatem to, co zwykle jest zapisywane w postaci $F(N, M)$, w rachunku λ zapisuje się $((FN)M) \equiv FNM$. Powtórzenia symbolu λ pomijane są według zasady łączności prawostronnej, tzn. $\lambda x y z.M \equiv (\lambda x.(\lambda y.(\lambda z.M)))$.

Zbiór **zmiennych wolnych** λ -wyrażeń M , oznaczany przez $F_V(M)$, zdefiniowany jest, w zależności od budowy termu M , w następujący sposób:

- Jeżeli M jest pojedynczą zmienną, tzn. $M \equiv x$, to zmienna ta jest wolna, a zatem $F_V(M) = \{x\}$.
- Jeżeli M jest postaci $\lambda x.M'$, to λx wiąże wszystkie wolne wystąpienia zmiennej x w λ -wyrażeniu M' , a zatem $F_V(M) = F_V(M') - \{x\}$.
- Jeżeli M jest postaci PQ , to zmiennymi wolnymi λ -wyrażenia M są wszystkie zmienne wolne występujące w λ -wyrażeniach P i Q , czyli $F_V(M) = F_V(P) \cup F_V(Q)$.

Zmienna jest zmienną wolną λ -wyrażenia M , jeżeli należy do zbioru $F_V(M)$, a **zmienną związaną** w przeciwnym wypadku. λ -wyrażenie M jest **domknięte** wtedy i tylko wtedy, gdy $F_V(M) = \emptyset$.

Przykłady:

- (1) W λ -wyrażeniu $M = y(\lambda x.x)$ zmienna y jest zmienną wolną, a x zmienną związaną. W termie $\lambda y.M$ nie ma zmiennych wolnych, jest to więc λ -wyrażenie domknięte.

- (2) W λ -wyrażeniu $M = x(\lambda x.x)$ tylko pierwsze wystąpienie zmiennej x jest wolne, chociaż $F_V(M) = \{x\}$. Oczywiście term M nie jest λ -wyrażeniem domkniętym.

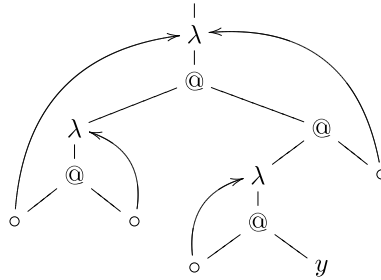
Wybór zmiennych używanych w wyrażeniu jako związane jest sprawą drugorzędną. Takie wyrażenia, jak na przykład $\lambda x.xy$ i $\lambda z.zy$, intuicyjnie definiują tę samą operację ("zaaplikuj dany argument do y ") więc z naszego punktu widzenia powinny być uważane za takie same. Dlatego λ -wyrażenia różniące się tylko zmiennymi związanymi chcemy ze sobą utożsamiać. Utożsamienie to nazywa się α -konwersją. Pojęcie α -konwersji można ściśle zdefiniować na wiele sposobów, my wybierzemy interpretację grafową.

Przez **drzewo de Bruijna** rozumiemy skończone drzewo etykietowane, w którym występować mogą następujące rodzaje wierzchołków:

- wierzchołki o etykiecie @, które mają zawsze dwa następniki: lewy i prawy;
- wierzchołki o etykiecie λ , które mają zawsze jeden następnik;
- liście etykietowane zmiennymi;
- liście bez etykiet.

Przy tym żądamy, aby każdy liść bez etykiety miał choć jednego przodka z etykietą λ . Jeśli z każdego wierzchołka bez etykiety poprowadzimy nową krawędź do któregoś z poprzedników tego wierzchołka o etykiecie λ , to otrzymany graf skierowany nazwiemy drzewem de Bruijna. Liśćmi drzewa de Bruijna nazywamy jego wierzchołki końcowe (ich etykietami są zmienne).

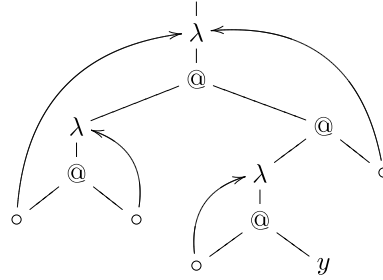
Przykład:



Dowolnemu λ -wyrażeniu M przypiszemy drzewo de Bruijna $G(M)$:

- Jeśli x jest zmienną to $G(x)$ składa się tylko z jednego wierzchołka o etykiecie x .
- Drzewo $G(PQ)$ ma korzeń o etykiecie @. Jego lewym następnikiem jest korzeń drzewa $G(P)$ a prawym korzeń drzewa $G(Q)$.
- Drzewo $G(\lambda xP)$ jest otrzymane z $G(P)$ poprzez
 - Dodanie nowego korzenia o etykiecie λ .
 - Połączenie go krawędzią z korzeniem drzewa $G(P)$.
 - Dodanie krawędzi od wszystkich wierzchołków z etykietą x do nowego korzenia.
 - Usunięcie etykiet x .

Przykład:



jest drzewem λ -wyrażenia $\lambda z.(\lambda u.zu)((\lambda u.uy)z)$.

Łatwo zauważyć, że etykiety liści w $G(M)$ to dokładnie zmienne wolne M .

Stwierdzenie 1. *Jeśli T jest drzewem de Bruijna, to $T = G(M)$ dla pewnego λ -wyrażenia M .*

Dowód. Indukcja ze względu na liczbę wierzchołków drzewa T . Jeśli jest tylko jeden wierzchołek, to jego etykietą jest pewna zmienna x i mamy $T = G(x)$. Jeśli etykietą korzenia jest $@$, to T składa się z korzenia i dwóch rozłącznych podgrafów T_1 i T_2 , do których stosujemy założenie indukcyjne. Mamy więc $T_1 = G(M_1)$ oraz $T_2 = G(M_2)$. Oczywiście wtedy $T = G(M_1 M_2)$. Pozosta je przypadek gdy etykietą korzenia jest λ . Wtedy T składa się z korzenia, podgrafu T_1 zaczepionego w następniku korzenia i pewnej liczby krawędzi prowadzących do korzenia z wierzchołków bez etykiet. Przypisując tym wierzchołkom etykietę z , która nie występuje w grafie T_1 , otrzymujemy z T_1 drzewo T'_1 . Z założenia indukcyjnego $T'_1 = G(M_1)$ dla pewnego λ -wyrażenia M_1 i możemy napisać $T = G(\lambda z.M_1)$. $\square$

Relację α -konwersji (oznaczaną przez $=_\alpha$) definiujemy tak:

$$M =_\alpha N \text{ wtedy i tylko wtedy, gdy } G(M) = G(N).$$

Od tej pory przez λ -termy albo po prostu termy rozumiemy "lambda-wyrażenia z dokładnością do α -konwersji", czyli w istocie klasy abstrakcji relacji $=_\alpha$. Każda taka klasa jest wyznaczona przez pewne drzewo de Bruijna, możemy więc uważać, że λ -termy to tak naprawdę drzewa de Bruijna. λ -wyrażenia są tylko ich syntaktyczną reprezentacją, przy czym dowolne dwie α -równoważne reprezentacje tego samego termu (drzewa de Bruijna) są tak samo dobre.

Przykład:

$$(\lambda x.x(\lambda x.xy))(\lambda y.yx)$$

to to samo co

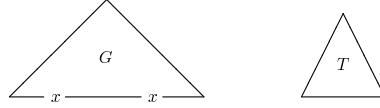
$$(\lambda u.u(\lambda x.xy))(\lambda v.vx),$$

ale nie to samo co

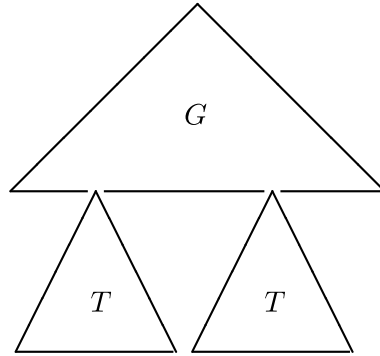
$$(\lambda y.y(\lambda y.yx))(\lambda x.xy).$$

Od tej pory znak równości oznacza równość λ -termów (a nie ich reprezentacji), napiszemy więc np. $\lambda xx = \lambda yy$.

Dla danych drzew de Bruijna G i T przez **podstawienie** $G[x := T]$ rozumiemy graf otrzymany z G przez zamianę każdego liścia z etykietą x na korzeń odrębnej kopii grafu T . Ilustracja jest chyba prostsza i bardziej zrozumiała niż jakakolwiek formalna definicja. Dla grafów:

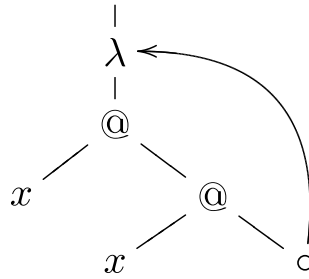


graf $G[x := T]$ wygląda tak:



Jeśli M i N są λ -wyrażeniami to notacja $M[x := N]$ oznacza λ -term o drzewie $G(M)[x := G(N)]$. Zwykle mówimy, że $M[x := N]$ to wynik podstawienia wyrażenia N na miejsce wszystkich wolnych wystąpień zmiennej x w M . Ale ta interpretacja nie zawsze jest poprawna. Problem powstaje wtedy, gdy jakaś zmienna z wolna w N przy "naiwnym" tekstowym podstawieniu znalazłaby się w zasięgu wiązania λz .

Przykład: Wynikiem podstawienia z na x w termie $\lambda z.x(xz)$ nie powinno być $\lambda z.z(zz)$. Podstawiamy z na miejsce x nie w napisie $\lambda z.x(xz)$ ale w drzewie



w którym w ogóle nie ma żadnego z . A więc zgodnie z definicją:

$$(\lambda z.x(xz))[x := z] = \lambda y.z(zy),$$

przy czym wybór związanej zmiennej y jest nieistotny.

Następujące stwierdzenie stanowi syntaktyczną definicję podstawienia.

Stwierdzenie 2. *Równość $M[x := N] = R$ ma miejsce wtedy i tylko wtedy, gdy zachodzi jeden z następujących przypadków:*

- $M = x$, oraz $R = N$;
- M jest zmienną różną od x , oraz $R = M$;
- $M = PQ$, oraz $R = P[x := N]Q[x := N]$;

- $M = \lambda y P$, gdzie $y \notin \{x\} \cup F_V(N)$, oraz $R = \lambda y. P[x := N]$.

Dowód. Każdy term M jest albo zmienną, albo aplikacją, albo abstrakcją. Ta ostatnia możliwość to jedyny nieoczywisty przypadek, niech więc M będzie abstrakcją. Jeśli z drzewa de Bruijna termu M odrzucimy korzeń wraz z prowadzącymi do niego krawędziami, to otrzymamy drzewo M' , w którym niektóre wierzchołki końcowe nie mają etykiet. Nadając tym wierzchołkom "nową" etykietę y otrzymamy taki term P , że $M = \lambda y P$. Ponieważ y nie występuje jako etykieta w drzewie N , więc nie ma znaczenia czy najpierw w termie P zamienimy x na N , a potem dodamy λ wiążącą y , czy postąpimy w odwrotnej kolejności. $\square$

Treść stwierdzenia możemy prościej zapisać jako:

- $x[x := N] = N$;
- $y[x := N] = y$, gdy y jest zmienną różną od x ;
- $(PQ)[x := N] = P[x := N]Q[x := N]$;
- $(\lambda y P)[x := N] = \lambda y. P[x := N]$, gdy $y \neq x$ oraz $y \notin F_V(N)$.

Stwierdzenie 3. *Zachodzą następujące własności podstawiania:*

- (1) $F_V(M[x := N]) = \begin{cases} (F_V(M) \setminus \{x\}) \cup F_V(N), & \text{jeśli } x \in F_V(M); \\ F_V(M), & \text{w przeciwnym przypadku.} \end{cases}$
- (2) *Jeśli $x \notin F_V(M)$ to $M[x := N] = M$.*
- (3) $M[x := x] = M$.
- (4) *Jeśli $\lambda x. P = \lambda y. Q$ to $P[x := y] = Q$ i $Q[y := x] = P$.*

Dowód jest oczywisty i go pomijamy.

Stwierdzenie 4 (lemat o podstawianiu). *Jeśli $x \neq y$ oraz albo $x \notin F_V(R)$ albo $y \notin F_V(M)$, to*

$$M[x := N][y := R] = M[y := R][x := N[y := R]].$$

Dowód. Dowód prowadzimy przez indukcję względem budowy M . Udowodnimy dla przykładu przypadek, gdy $M = x$. Wówczas

$$M[x := N][y := R] = x[x := N][y := R] = N[y := R] = x[x := N[y := R]] = x[y := R][x := N[y := R]].$$

$\square$

Wniosek 1. *Jeśli $y \notin F_V(M)$ to $M[x := y][y := R] = M[x := R]$.*