Materiały do wykładu

Rachunek lambda

Paweł Urzyczyn
urzy@mimuw.edu.pl
3 września 2023, godzina 13:58

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 intensjonalnie 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 jezyków programowania. A wreszcie i w praktyce, gdy pojawiły sie języki programowania funkcyjnego. Dzisiaj znajomość podstaw rachunku lambda jest niezbędnym elementem wykształcenia nowoczesnego informatyka.

Poprzez związki z logiką intuicjonistyczną, rachunek lambda odzyskał też należne sobie miejsce w podstawach matematyki, zwłaszcza w teorii dowodu. Dlatego ta tematyka powinna być interesująca także dla studentów matematyki zainteresowanych logiką.

Te notatki obejmują najważniejsze wiadomości z rachunku lambda bez typów i podstawowe informacje o systemach z typami. Zrozumienie ich nie wymaga w zasadzie przygotowania wykraczającego poza materiał wykładany na I roku. Jednak wcześniejsze wysłuchanie wykładu z języków i automatów pozwoli na spojrzenie na pewne zagadnienia z szerszej perspektywy.

1 Składnia rachunku lambda

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

Przyjmijmy, że mamy pewien przeliczalny nieskończony zbiór *zmiennych przedmiotowych*.

– Zmienne przedmiotowe są lambda-wyrażeniami;
– Jeśli M i N są lambda-wyrażeniami, to (MN) też;
– Jeśli M jest lambda-wyrażeniem i x jest zmienną, to (λxM) jest lambda-wyrażeniem.

Wyrażenie postaci (MN) nazywamy *aplikacją*, a wyrażenie postaci (λxM) to λ-*abstrakcja*. Intuicyjny sens aplikacji (MN) to zastosowanie operacji M do argumentu N. Abstrakcję (λxM) interpretujemy natomiast jako definicję operacji (funkcji), która argumentowi x przypisuje M. Oczywiście x może występować w M, tj. M zależy od x. Narzuca się analogia z procedurą (funkcją) o parametrze formalnym x i treści M.

Stosujemy następujące konwencje notacyjne:

– opuszczamy zewnętrzne nawiasy;
– aplikacja wiąże w lewo, tj. MNP oznacza $(MN)P$;
– piszemy $\lambda x_1 \ldots x_n.M$ zamiast $\lambda x_1(\ldots(\lambda x_n M)\ldots)$.

Kropka w wyrażeniu $\lambda x_1 \ldots x_n.\, M$ oznacza to samo, co ujęcie w nawiasy całego wyrażenia M. Na przykład napis $(\lambda x.\, x(xx))y$ odczytamy jako $((\lambda x(x(xx)))y)$. Czasem piszemy niepotrzebne kropki, np. $\lambda x.\, x$ zamiast $\lambda x\, x$.

Operator lambda-abstrakcji λ wiąże zmienne, tj. wszystkie wystąpienia x w wyrażeniu λxM uważa się za *związane* (lokalne). Zmienne *wolne* (globalne) definiuje się tak:

– $\mathrm{FV}(x) = \{x\}$;
– $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$;
– $\mathrm{FV}(\lambda xM) = \mathrm{FV}(M) - \{x\}$.

Alfa-konwersja

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 lambda-wyrażenia różniące się tylko zmiennymi związanymi chcemy ze sobą utożsamiać. Utożsamienie to nazywa się *alfa-konwersją*.

Pojęcie alfa-konwersji można ściśle zdefiniować na wiele sposobów, my wybierzemy interpretację grafową.

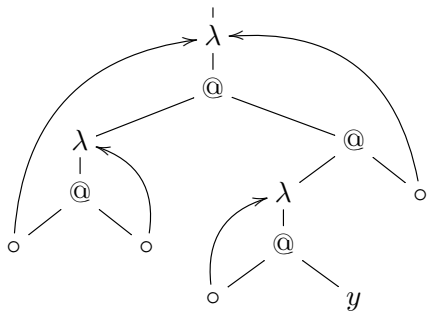
Lambda-grafy

Przez *drzewo* rozumiemy poniżej 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 przedmiotowymi;
- 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 *lambda-drzewem*. *Liśćmi* lambda-drzewa nazywamy jego wierzchołki końcowe (ich etykietami są zmienne). Na przykład taki graf jest lambda-drzewem (wierzchołki bez etykiet oznaczono przez $\circ$).



Teraz dowolnemu lambda-wyrażeniu M przypiszemy lambda-drzewo $G(M)$.

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

Na przykład $G(\lambda z.(\lambda u.zu)((\lambda u.uy)z))$ to lambda-drzewo na rysunku powyżej. Łatwo widzieć, że etykiety liści w $G(M)$ to dokładnie zmienne wolne M. Na dodatek mamy:

Fakt 1.1 *Jeśli T jest lambda-drzewem, to $T = G(M)$ dla pewnego lambda-wyrażenia M.*

Dowód: Indukcja ze względu na liczbę wierzchołków grafu 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 $T = G(M_1M_2)$. Pozostaje 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 lambda-drzewo T_1'. Z założenia indukcyjnego $T_1' = G(M_1)$ dla pewnego lambda-wyrażenia M_1 i możemy napisać $T = G(\lambda z.M_1)$. ∎

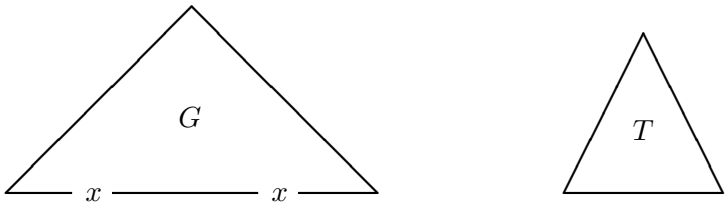
Relację alfa-konwersji (oznaczaną przez $=_\alpha$) definiujemy tak:

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

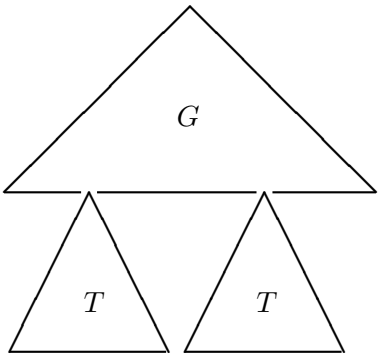
Od tej pory przez *lambda-termy* albo po prostu *termy* rozumiemy „lambda-wyrażenia z dokładnością do alfa-konwersji", czyli w istocie klasy abstrakcji relacji $=_\alpha$. Każda taka klasa jest wyznaczona przez pewne lambda-drzewo, możemy więc uważać, że lambda-termy to tak naprawdę lambda-drzewa. Lambda-wyrażenia są tylko ich syntaktyczną reprezentacją, przy czym dowolne dwie alfa-równoważne reprezentacje tego samego termu (lambda-drzewa) są tak samo dobre. Na 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 lambda-termów (a nie ich reprezentacji), napiszemy więc np. $\lambda x\, x = \lambda y\, y$.

Podstawienie

Dla danych lambda-grafow 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. Jeśli mamy grafy



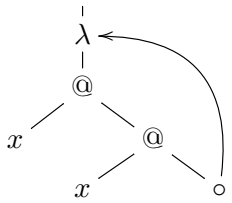
to graf $G[x := T]$ wygląda tak jak na Obrazku 1 (etykiety x już nie ma):



Obrazek 1: Podstawienie $G[x := T]$

Jeśli M i N są lambda-wyrażeniami to notacja $M[x := N]$ oznacza lambda-term o grafie $G(M)[x := G(N)]$. 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. Na przykład wynikiem podstawienia z na x w termie $\lambda z.x(xz)$ nie powinno być $\lambda z.z(zz)$. Chcemy bowiem podstawić z na miejsce x nie w napisie $\lambda z.x(xz)$ ale *w grafie*



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

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

przy czym wybór związanej zmiennej y jest nieistotny. Tekstowe podstawienie jest poprawne tylko wtedy, gdy nie występuje konflikt nazw wolnych i związanych (globalnych i lokalnych). Dlatego wcześniej należy dokonać odpowiedniej wymiany zmiennych związanych.

Następujący lemat stanowi syntaktyczną definicję podstawienia.

Lemat 1.2 *Równość $M[x := N] = R$ ma miejsce wtedy i tylko wtedy, gdy zachodzi jeden z następujących przypadków:*

1. *$M = x$, oraz $R = N$;*
2. *M jest zmienną różną od x, oraz $R = M$;*
3. *$M = PQ$, oraz $R = P[x := N]Q[x := N]$;*
4. *$M = \lambda y\, P$, gdzie $y \notin \{x\} \cup \mathrm{FV}(N)$, 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 lambda-drzewa termu M odrzucimy korzeń wraz z prowadzącymi do niego krawędziami, to otrzymamy graf 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 lambda-drzewie N, więc nie ma znaczenia czy najpierw w termie P zamienimy x na N, a potem dodamy lambdę wiążącą y, czy postąpimy w odwrotnej kolejności. ∎

Zapiszmy treść lematu 1.2 w nieco prostszej postaci:

- $x[x := N] = N$;
- $y[x := N] = y$, 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 $y \notin \mathrm{FV}(N)$.

Powyższe reguły można stosować w dowodach indukcyjnych.

Lemat 1.3

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

Łatwy dowód tego lematu pomijamy, udowodnimy za to następny.

Lemat 1.4 (o podstawieniu) *Jeśli* $x \neq y$ *oraz albo* $x \notin \mathrm{FV}(R)$ *albo* $y \notin \mathrm{FV}(M)$, *to*

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

Dowód: Dowód jest przez indukcję ze względu na rozmiar M. Rozważamy różne przypadki, w zależności od postaci tego termu.

Przypadek 1: Jeżeli $M = x$, to $M[x := N][y := R] = x[x := N][y := R] = N[y := R] = x[x := N[y := R]] = x[y := R][x := N[y := R]]$.

Przypadek 2: Jeżeli $M = y$, to $M[x := N][y := R] = y[x := N][y := R] = y[y := R] = R = R[x := N[y := R]] = y[y := R][x := N[y := R]]$, bo wtedy $x \notin \mathrm{FV}(R)$.

Przypadek 3: Jeżeli M jest jakąś inną zmienną z, to zarówno $M[x := N][y := R]$ jak też $M[y := R][x := N[y := R]]$ jest równe z.

Przypadek 4: Gdy $M = PQ$, to $M[x{:=}N][y{:=}R] = P[x{:=}N][y{:=}R]Q[x{:=}N][y{:=}R] = P[y{:=}R][x{:=}N[y{:=}R]]Q[y{:=}R][x{:=}N[y{:=}R]] = (PQ)[y{:=}R][x{:=}N[y{:=}R]]$, na mocy założenia indukcyjnego.

Przypadek 5: Jeżeli M jest abstrakcją to można zakładać, że $M = \lambda z\, Q$, gdzie $z \neq x, y$. Po lewej stronie mamy wtedy $\lambda z.\, Q[x := N][y := R]$ i z założenia indukcyjnego otrzymamy $M[x := N][y := R] = \lambda z.\, Q[y := R][x := N[y := R]] = (\lambda z.\, Q)[y := R][x := N[y := R]]$. ∎

Wniosek 1.5 *Jeśli* $y \notin \mathrm{FV}(M)$ *to* $M[x := y][y := R] = M[x := R]$.

Ćwiczenia

1. Jaki błąd popełniono w tej definicji lambda-termu:
$$M ::= x \mid MN \mid \lambda x.M \ ?$$
2. Udowodnić lemat 1.3.
3. Napis $M(a|b)$ oznacza wyrażenie powstałe z M przez jednoczesną zamianę wszystkich wystąpień symbolu a w M na wystąpienia b i odwrotnie. Przez $\equiv_\alpha$ oznaczmy najmniejszą relację równoważności w zbiorze lambda-wyrażeń, spełniającą następujące warunki:
   - $\lambda x.M \equiv_\alpha \lambda y.M(x|y)$, dla $y \notin \mathrm{FV}(M)$;
   - jeżeli $M \equiv_\alpha M'$, to $\lambda x.M \equiv_\alpha \lambda x.M'$;
   - jeżeli $M \equiv_\alpha M'$ i $N \equiv_\alpha N'$ to $MN \equiv_\alpha M'N'$.

   Udowodnić, że $M \equiv_\alpha N$ wtedy i tylko wtedy, gdy $M =_\alpha N$.

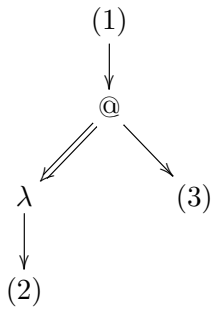
2 Redukcje

Relacja *beta-redukcji* to najmniejsza relacja w zbiorze lambda-termów, spełniająca warunki:

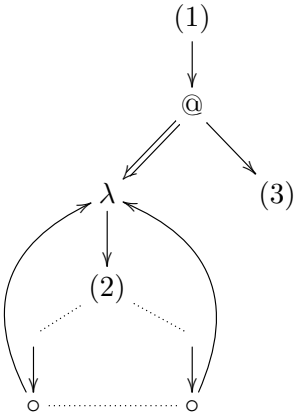
- $(\lambda x P)Q \to_\beta P[x := Q]$;
- jeśli $M \to_\beta M'$, to $MN \to_\beta M'N$, $NM \to_\beta NM'$ oraz $\lambda x M \to_\beta \lambda x M'$.

Inaczej mówiąc, $M \to_\beta M'$ zachodzi gdy podterm termu M postaci $(\lambda x P)Q$, czyli *β-redeks*, zostaje zastąpiony w termie M' przez $P[x := Q]$. Znakiem $\twoheadrightarrow_\beta$ lub $\to_\beta^*$ oznaczamy domknięcie przechodnio-zwrotne relacji $\to_\beta$, a znakiem $=_\beta$ najmniejszą relację równoważności zawierającą $\to_\beta$, którą nazywamy relacją *beta-równości*. Inne popularne oznaczenia to $\to_\beta^+$ na domknięcie przechodnie i $\to_\beta^=$ na domknięcie zwrotne relacji $\to_\beta$. Indeks $_\beta$ przy strzałkach bywa pomijany.

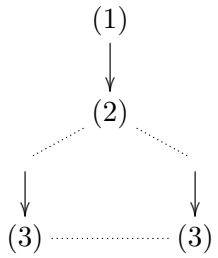
Z punktu widzenia grafowego, beta-redeks to krawędź od @ do λ:



Beta-redukcja polega na usunięciu tej krawędzi, tj. na „wyprostowaniu" drogi od (1) do (2). Każda krawędź wchodząca do λ (reprezentująca wystąpienie zmiennej związanej) zostaje przy tym skierowana do osobnej kopii termu zaczepionego w (3). Inaczej, podgraf postaci



zostaje zastąpiony podgrafem postaci:



Uwaga: może się zdarzyć, że lambda nie wiąże żądnego wystąpienia zmiennej (nie ma żadnej krawędzi prowadzącej do λ). Wtedy podgraf (3) znika. Odpowiada to redukcji postaci $(\lambda x.M)N \to_\beta M$, gdzie $x \notin \mathrm{FV}(M)$.

Następujący łatwy lemat stwierdza, że β-redukcja jest zgodna z operacją podstawienia.

Lemat 2.1 *Jeśli* $M \twoheadrightarrow_\beta M'$ *i* $N \twoheadrightarrow_\beta N'$ *to* $M[x := N] \twoheadrightarrow_\beta M'[x := N']$.

Dowód: Należy pokazać dwa prostsze stwierdzenia:

(1) Jeśli $M \to_\beta M'$, to $M[x := N] \to_\beta M'[x := N]$;

(2) Jeśli $N \to_\beta N'$, to $M[x := N] \twoheadrightarrow_\beta M[x := N']$,

a następnie zastosować indukcję ze względu na liczbę kroków redukcji. Zarówno (1) jak (2) można udowodnić przez indukcję ze względu na długość termu M. Istotny przypadek w dowodzie części (1) zachodzi wtedy, gdy $M = (\lambda y\, P)Q$ i $M' = P[y{:=}Q]$, dla pewnych P, Q. Przyjmując $y \notin \mathrm{FV}(N)$, mamy $M[x{:=}N] = (\lambda y.P[x{:=}N])Q[x{:=}N] \to_\beta P[x{:=}N][y{:=}Q[x{:=}N]] = P[y{:=}Q][x{:=}N] = M'[x{:=}N]$, na mocy lematu o podstawieniu 1.4. Szczegóły pozostawiamy jako ćwiczenie. ∎

Teoria λ

Rachunek lambda bez typów można przedstawić jako teorię równościową o aksjomatach i regułach jak na Obrazku 2. Jeśli w tym systemie można wyprowadzić równość $M = N$, to napiszemy $\lambda \vdash M = N$. Łatwo sprawdzić równoważność:

$$\lambda \vdash M = N \text{ wtedy i tylko wtedy, gdy } M =_\beta N.$$

Jednakże dla nas ważne są też własności beta-redukcji, a nie tylko beta-równości. Term postaci $(\lambda xP)Q$ nazywamy *beta-redeksem*. Jeśli w termie M nie występuje żaden beta-redeks, to dla żadnego N nie zachodzi $M \to_\beta N$. Mówimy wtedy, że M jest *w postaci β-normalnej,* lub po prostu *w postaci normalnej.* Nietrudno zauważyć, że postaci normalne to dokładnie termy kształtu $\lambda\vec{x}.\, y\vec{N}$, gdzie $\vec{N}$ są normalne.

$$(\beta)\ (\lambda x\, M)N = M[x := N] \qquad x = x$$

$$\frac{M = N}{MP = NP} \qquad \frac{M = N}{PM = PN} \qquad (\xi)\ \frac{M = N}{\lambda x\, M = \lambda x\, N}$$

$$\frac{M = N}{N = M} \qquad \frac{M = N, N = P}{M = P}$$

Obrazek 2: Rachunek lambda jako teoria równościowa

Przykład: Następujące termy są w postaci normalnej:

$$\begin{aligned} \mathbf{I} &= \lambda x.x \\ \mathbf{K} &= \lambda xy.x \\ \mathbf{S} &= \lambda xyz.xz(yz) \\ \omega &= \lambda x.xx \end{aligned}$$

Natomiast term $\Omega = \omega\omega$ nie jest w postaci normalnej, bo $\Omega \to_\beta \Omega$.

Jeśli $M \twoheadrightarrow_\beta N$ i N jest w postaci normalnej, to mówimy, że M *ma postać normalną*, i że N *jest postacią normalną termu* M. Mówimy, że term M ma własność *silnej normalizacji* (jest *silnie normalizowalny*) jeżeli nie istnieje nieskończony ciąg redukcji $M = M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta \cdots$ Oczywiście term silnie normalizowalny musi mieć postać normalną.

Jeszcze jedno przydatne czasem oznaczenie. Piszemy $M \downarrow_\beta N$, gdy istnieje taki term P, że $M \twoheadrightarrow_\beta P_\beta \twoheadleftarrow N$.

Głównym przedmiotem naszego zainteresowania jest beta-redukcja. Dlatego np. symbole $\to$ i $\twoheadrightarrow$ domyślnie[1] oznaczają $\to_\beta$ i $\twoheadrightarrow_\beta$. Ale czasami rozważamy także relację *eta-redukcji*. Jest to najmniejsza relacja $\to_\eta$, spełniającą warunki:

- $\lambda x.Mx \to_\eta M$, gdy $x \notin \mathrm{FV}(M)$;
- jeśli $M \to_\eta M'$, to $MN \to_\eta M'N$, $NM \to_\eta NM'$ oraz $\lambda xM \to_\eta \lambda xM'$.

Symbol $\to_{\beta\eta}$ oznacza sumę tych relacyj. Używamy odpowiednio notacji np. $\twoheadrightarrow_\eta$, $=_{\beta\eta}$, mówimy o „postaciach η-normalnych" itd.

Jeżeli do aksjomatów i reguł równościowych rozważanych powyżej dodamy nowy aksjomat

$$(\eta) \quad \lambda x.Mx = M,$$

dla $x \notin \mathrm{FV}(M)$, to oczywiście otrzymamy system wnioskowania, pozwalający na wyprowadzenie równości $M = N$ wtedy i tylko wtedy gdy $M =_{\beta\eta} N$. Mniej oczywiste jest to, że aksjomat (η) można równoważnie zastąpić przez następującą *regułę ekstensjonalności:*

$$(\mathrm{ext}) \quad \frac{Mx = Nx}{M = N}\ (x \notin \mathrm{FV}(M) \cup \mathrm{FV}(N))$$

[1] W odniesieniu do znaku równości nie stosujemy tej konwencji (ale spotyka się to często w literaturze).

Reguła ekstensjonalności wyraża taką myśl: funkcje, które dla tych samych argumentów dają te same wyniki, są równe.

Dygresja: Zauważmy, że beta- i eta-redukcja są w pewnym sensie dualne do siebie. Beta-redeks to term, w którym najpierw utworzyliśmy funkcję, tj. *wprowadziliśmy* abstrakcję (czyli zapytanie o argument, prompt), a potem ją *eliminujemy* przez dostarczenie argumentu. Eta-redeks odpowiada odwrotnej kolejności: najpierw eliminujemy prompt, dostarczając zmiennej jako argumentu, a potem wprowadzamy abstrakcję ze względu na tę zmienną.

Ćwiczenia

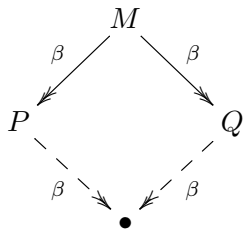
1. Napisać program, który drukuje własny tekst. (Nie wolno odwoływać się do nazwy pliku, miejsca w pamięci itp.) Zadanie łatwe w Lispie, trudne w Pascalu.
2. Udowodnić, że jeśli $(\lambda x P)Q = (\lambda y P')Q'$ to $P[x := Q] = P'[y := Q']$.
3. Udowodnić, że aksjomat (η) jest równoważny regule (ext), tj. te same równości można wyprowadzić w systemie z Obrazka 2 rozszerzonym o (η) i w systemie rozszerzonym o regułę (ext).
4. Sprawdzić, że $\lambda x.Mx =_\beta M$, gdy M jest abstrakcją.

3 Twierdzenie Churcha-Rossera

W jednym termie może występować więcej niż jeden redeks. Można więc go redukować na różne sposoby. Niedobrze jednak byłoby, gdyby te różne ciągi redukcji prowadziły do nieodwracalnie różnych rezultatów. Na szczęście mamy *twierdzenie Churcha-Rossera.*

Twierdzenie 3.1 *Jeśli* $P \;{}_\beta\!\twoheadleftarrow M \twoheadrightarrow_\beta Q$*, to* $P \downarrow_\beta Q$.

Treść twierdzenia Churcha-Rossera przedstawiamy za pomocą diagramu (Obrazek 3).



Obrazek 3: Twierdzenie Churcha-Rossera

Inaczej mówiąc, twierdzenie to mówi, że relacja $\twoheadrightarrow_\beta$ ma tzw. *własność rombu*, która niestety nie zachodzi dla relacji $\rightarrow_\beta$. Jako przykład można podać term $M = (\lambda x.xx)((\lambda x.x)y)$.

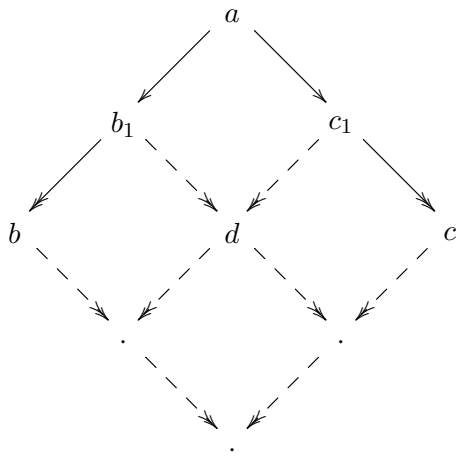
Zanim udowodnimy twierdzenie Churcha-Rossera, zauważmy, że pojęcie o którym mówimy, ma sens dla każdej relacji dwuargumentowej. Poniżej przyjmujemy taką konwencję: podwójna strzałka zawsze oznacza domknięcie przechodnio-zwrotne relacji oznaczonej strzałką pojedynczą.

Mówimy, że relacja $\rightarrow$ w zbiorze A ma *własność Churcha-Rossera* (CR), jeżeli dla dowolnych elementów $a, b, c \in A$, takich że $b \twoheadleftarrow a \twoheadrightarrow c$, istnieje takie d, że $b \twoheadrightarrow d \twoheadleftarrow c$. Relacja $\rightarrow$ ma *słabą*

własność Churcha-Rossera (WCR), jeżeli $b \leftarrow a \rightarrow c$ implikuje $b \twoheadrightarrow d \twoheadleftarrow c$ dla pewnego d. Oczywiście własność rombu implikuje CR (ale nie na odwrót), a własność CR implikuje WCR. Mniej oczywiste jest to, że własność CR nie wynika z WCR. Najprostszy przykład jest taki:

$$\bullet \longleftarrow \bullet \longleftrightarrow \bullet \longrightarrow \bullet$$

Jeśli jednak relacja $\rightarrow$ ma *własność silnej normalizacji* (SN), tj. nie istnieje nieskończony ciąg postaci $a_0 \rightarrow a_1 \rightarrow a_2 \rightarrow \cdots$, to już jest dobrze.



Obrazek 4: Rysunek do lematu Newmana

Fakt 3.2 (Lemat Newmana) *Jednoczesne zachodzenie* WCR *i* SN *implikuje* CR*.*

Dowód: Jeśli relacja $\rightarrow$ ma własność SN to relacja $\twoheadleftarrow$ jest dobrym ufundowaniem zbioru A. Przez indukcję ze względu na porządek $\twoheadleftarrow$ dowodzimy, że każdy element $a \in A$ ma własność:

Dla dowolnych b, c*, jeśli* $b \twoheadleftarrow a \twoheadrightarrow c$*, to* $b \downarrow c$*.*

Jeśli $a = b$ lub $a = c$ to teza jest oczywista. Załóżmy więc, że $b \twoheadleftarrow b_1 \leftarrow a \rightarrow c_1 \twoheadrightarrow c$.

Na mocy WCR jest takie d, że $b_1 \twoheadrightarrow d \twoheadleftarrow c_1$. Z założenia indukcyjnego, zastosowanego do b_1 i c_1 mamy więc $b \downarrow d \downarrow c$ i możemy zastosować założenie indukcyjne dla d. ∎

Dowód twierdzenia Churcha-Rossera

Dla dowodu zdefiniujemy pewną pomocniczą relację. Będzie to relacja $\xrightarrow{1}$, określona jako najmniejsza relacja na termach, która spełnia następujące warunki:

– $x \xrightarrow{1} x$, gdy x jest zmienną;

– jeśli $M \xrightarrow{1} M'$, to $\lambda x M \xrightarrow{1} \lambda x M'$;

– jeśli $M \xrightarrow{1} M'$ i $N \xrightarrow{1} N'$, to $MN \xrightarrow{1} M'N'$, oraz $(\lambda x M)N \xrightarrow{1} M'[x := N']$.

Relacja $\xrightarrow{1}$ odpowiada jednoczesnej redukcji kilku beta-redeksów występujących w termie. Jeśli zredukujemy wszystkie redeksy w termie M, to otrzymamy jego *pełne rozwinięcie* $M^\bullet$:

– $x^\bullet = x$;

– $(\lambda x\, M)^\bullet = \lambda x\, M^\bullet$;

– $(MN)^\bullet = M^\bullet N^\bullet$, gdy M nie jest abstrakcją;

– $((\lambda x\, M)N)^\bullet = M^\bullet[x := N^\bullet]$.

Lemat 3.3

(1) Dla dowolnego M zachodzi $M \xrightarrow{1} M$ oraz $M \xrightarrow{1} M^\bullet$.

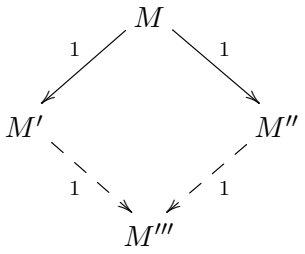
(2) Jeśli $M \xrightarrow{1} M'$ i $N \xrightarrow{1} N'$, to $M[x := N] \xrightarrow{1} M'[x := N']$.

(3) Jeśli $M \xrightarrow{1} M'$, to $M' \xrightarrow{1} M^\bullet$.

Dowód: Dowód części (1) jest przez indukcję ze względu na budowę M, a części (2) i (3) przez indukcję ze względu na wyprowadzenie $M \xrightarrow{1} M'$. W dowodzie punktu (2) nieoczywisty jest przypadek, gdy $M = (\lambda y P)Q \xrightarrow{1} P'[y := Q']$, gdzie $P \xrightarrow{1} P'$ oraz $Q \xrightarrow{1} Q'$. Zakładając, że zmienna związana y jest tak wybrana, że $y \notin FV(N)$, i korzystając z założenia indukcyjnego o $P \xrightarrow{1} P'$ i $Q \xrightarrow{1} Q'$, mamy wtedy na mocy lematu o podstawieniu:

$M[x{:=}N] = (\lambda y P[x{:=}N])Q[x{:=}N] \xrightarrow{1} P'[x{:=}N'][y{:=}Q'[x{:=}N']] = P'[y{:=}Q'][x{:=}N']$. ∎

Uwaga: Użyty w powyższym dowodzie zwrot „dowód przez indukcję ze względu na wyprowadzenie $M \xrightarrow{1} M'$” można ściśle rozumieć tak: Zbiór par termów $\{(M, M') \mid M \xrightarrow{1} M'\}$ jest sumą wstępującego ciągu zbiorów X_i, gdzie $X_0 = \{(x, x) \mid x \text{ jest zmienną}\}$, a każdy zbiór X_{i+1} powstaje z X_i przez dodanie wszystkich par $(MN, M'N')$, $((\lambda x M)N, M'[x := N'])$ oraz $(\lambda x M, \lambda x M')$, dla dowolnych (M, M') i (N, N'), które są już w X_i. Indukcję tak naprawdę prowadzimy ze względu na najmniejsze takie i, że $(M, M') \in X_i$.



Obrazek 5: Własność rombu

Wniosek 3.4 *Relacja* $\overset{1}{\to}$ *ma własność rombu, tj. jeśli* $M \overset{1}{\to} M'$ *i* $M \overset{1}{\to} M''$*, to dla pewnego termu* M''' *zachodzi* $M' \overset{1}{\to} M''' \overset{1}{\leftarrow} M''$*.*

Dowód: Z lematu 3.3(3) wynika, że jako M''' można wziąć $M^\bullet$. ∎

Dowód twierdzenia 3.1: Ponieważ relacja $\overset{1}{\to}$ ma własność rombu, więc tym bardziej jej domknięcie przechodnio-zwrotne $\overset{1}{\twoheadrightarrow}$ ma własność rombu.

Zauważmy teraz, że mamy następujące zawierania pomiędzy relacjami:

$$\to_\beta \subseteq \overset{1}{\to} \subseteq \twoheadrightarrow_\beta.$$

Stąd łatwo wywnioskować, że relacje $\overset{1}{\twoheadrightarrow}$ i $\twoheadrightarrow_\beta$ są równe, a więc $\twoheadrightarrow_\beta$ ma własność rombu. ∎

Skoro udowodniliśmy twierdzenie Churcha-Rossera, zobaczmy jakie ma ono konsekwencje.

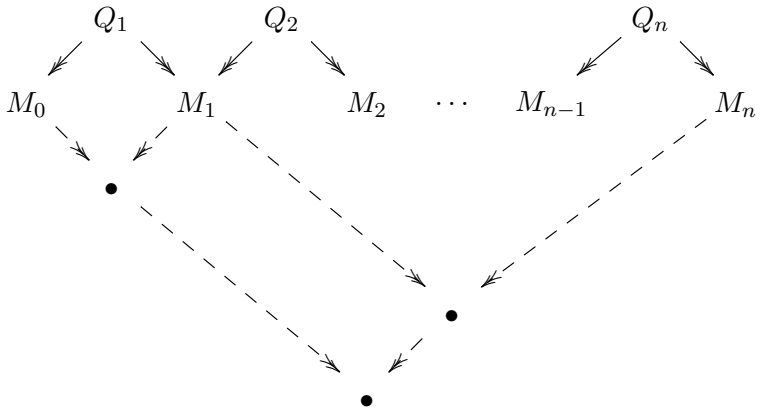
Wniosek 3.5

0) Jeśli $M \downarrow_\beta N \downarrow_\beta Q$*, to* $M \downarrow_\beta Q$*.*

1) Jeśli $M =_\beta N$*, to* $M \downarrow_\beta N$*.*

2) Każdy term ma co najwyżej jedną postać normalną.

3) Rachunek lambda jest niesprzeczny jako teoria równościowa: nie można np. wyprowadzić równości $x = y$*, ponieważ* $x \neq_\beta y$*.*



Obrazek 6: Dowód wniosku 3.5(1)

Dowód: Jedyna nieoczywista część to (1). Jeśli $M =_\beta N$, to mamy taki ciąg:

$$M = M_0 \twoheadleftarrow Q_1 \twoheadrightarrow M_1 \twoheadleftarrow Q_2 \twoheadrightarrow M_2 \twoheadleftarrow \cdots \twoheadrightarrow M_n = N$$

Dowodzimy $M \downarrow_\beta N$, przez indukcję ze względu na n. Jeśli $n = 0$, to sprawa jest oczywista, w przeciwnym razie z założenia indukcyjnego mamy $M_1 \downarrow_\beta N$. Ale $M_0 \downarrow_\beta M_1$ na mocy CR, więc $M \downarrow_\beta N$ wynika z części (0). Zob. Obrazek 6. ∎

A zatem sens twierdzenia Churcha-Rossera można wyrazić tak. Chociaż term może być redukowany na wiele sposobów, to wszystkie te sposoby są zgodne. Każde obliczenie zakończone sukcesem (uzyskaniem postaci normalnej) daje ten sam wynik.

Ćwiczenia

1. Jaki sens grafowy ma relacja $\stackrel{1}{\to}$ i pełne rozwinięcie termu?
2. Term $M^\bullet$ powstaje przez redukcję wszystkich redeksów w M. Czy $M^\bullet$ jest postacią normalną?

4 Eta-redukcja

Każdy krok η-redukcji $\lambda x.\, Mx \to_\eta M$ zmniejsza długość termu, zatem zachodzi

Fakt 4.1 *Relacja η-redukcji ma własność silnej normalizacji.*

Aby wywnioskować, że $\to_\eta$ ma własność Churcha-Rossera, wystarczy więc pokazać WCR. W istocie mamy prawie własność rombu, co można łatwo pokazać, badając możliwe przypadki wzajemnego położenia redeksów.

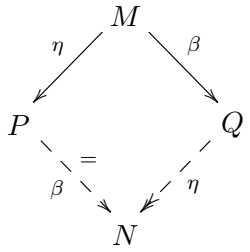
Lemat 4.2 *Domknięcie zwrotne $\to_\eta^=$ relacji $\to_\eta$ ma własność rombu.*

Własność rombu dla η-redukcji nie zachodzi bo $y \;{}_\eta\!\leftarrow \lambda x.\, yx \to_\eta y$, tymczasem nie ma termu, do którego y redukowałby się w jednym kroku.

Wniosek 4.3 *Relacja η-redukcji ma własność Churcha-Rossera.*

Pokażemy teraz, że własność Churcha-Rossera zachodzi także dla relacji $\to_{\beta\eta}$.

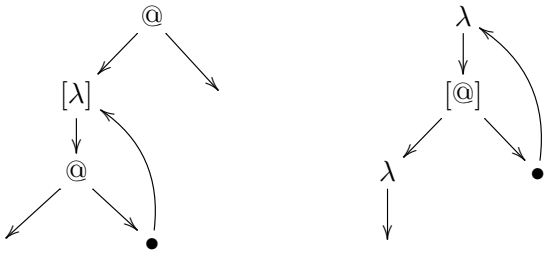
Lemat 4.4 *Jeśli $P \;{}_\eta\!\leftarrow M \to_\beta Q$ to istnieje takie N, że $P \to_\beta^= N \;{}_\eta\!\twoheadleftarrow Q$.*



Dowód: Badając wzajemne położenie beta- i eta-redeksów w grafie termu M (Obrazek 7), widzimy że kolejność, w której redukujemy te redeksy, na ogół nie ma znaczenia i rezultat jest taki sam. Pierwszy wyjątek, to sytuacja gdy eta-redeks znajduje się w argumencie beta-redeksu i wskutek redukcji może zostać usunięty lub powielony. Dlatego w treści lematu

Obrazek 7: Eta-redeks i beta-redeks

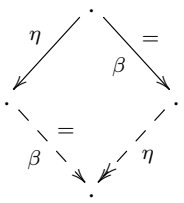
po prawej stronie mamy strzałkę podwójną $\twoheadleftarrow_\eta$ a nie pojedynczą. Pozostałe dwa wyjątki zachodzą wtedy, gdy nasze dwa redeksy wykorzystują ten sam wierzchołek grafu (lambdę lub aplikację). Ma to miejsce w podtermach postaci $(\lambda x.Mx)N$, gdzie $x \notin \mathrm{FV}(M)$ oraz postaci $\lambda x(\lambda y\, M)x$, gdzie $x \notin \mathrm{FV}(\lambda y\, M)$. Na obrazku 8 wspólne wierzchołki są w nawiasach kwadratowych. Szczęśliwie w obu przypadkach redukcja każdego z konkurujących redeksów daje ten sam rezultat. ∎



Obrazek 8: Pary krytyczne dla beta-eta-redukcji

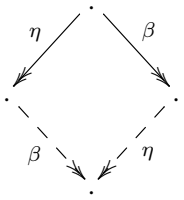
Sytuacja zilustrowana na Obrazku 8 może dotyczyć także innych systemów redukcyjnych. Jeśli redukcja każdego z dwóch redeksów wymaga zużycia tego samego „zasobu", to mówimy, że redeksy te tworzą *parę krytyczną*. Jeśli każdą parę krytyczną można „uzgodnić" to dana relacja redukcji ma słabą własność Churcha-Rossera.[2]

Z lematu 4.4 wynika łatwo, że mamy następujący diagram:



[2]Dla systemów o własności SN mamy więc ogólną metodę dowodzenia własności Churcha-Rossera: uzgodnić pary krytyczne i zastosować lemat Newmana 3.2.

Z tego diagramu otrzymamy następny. Mówi on, że relacje $\twoheadrightarrow_\beta$ i $\twoheadrightarrow_\eta$ są *przemienne*:



Ponieważ obie relacje mają własność Churcha-Rossera, nietrudno teraz pokazać

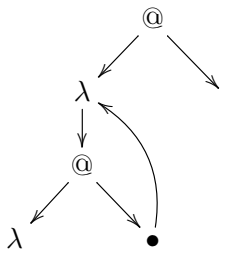
Twierdzenie 4.5 *Relacja $\rightarrow_{\beta\eta}$ ma własność Churcha-Rossera.*

Beta-eta-redukcja

Pokażemy teraz, że własność silnej β-normalizacji jest równoważna silnej $\beta\eta$-normalizacji. Wynika to łatwo z następującego prostego lematu.

Lemat 4.6 *Niech $M \rightarrow_\eta N \rightarrow_\beta P$ ale $M \not\rightarrow_\beta N$. Wtedy istnieje takie Q, że $M \rightarrow_\beta Q \twoheadrightarrow_\eta P$.*

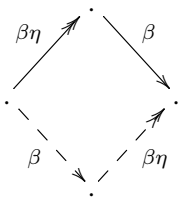
Dowód: Zauważmy, że jedyna sytuacja, w której eta-redukcja może wykreować nowy beta-redeks wygląda tak, jak na Obrazku 9. To jest akurat sytuacja zabroniona, bo zamiast eta-



Obrazek 9: Eta-redukcja tworzy beta-redeks

redeksu możemy z tym samym skutkiem zredukować beta-redeks. W pozostałych przypadkach beta-redeks redukowany w drugim kroku już istnieje w termie M i może zostać zredukowany zanim wykonana będzie eta-redukcja. Potem dopiero wykonamy tyle eta-redukcji ile kopii naszego eta-redeksu (być może zero) występuje w Q. ∎

Wniosek 4.7 *Jeśli $M \twoheadrightarrow_{\beta\eta} P \rightarrow_\beta N$ to $M \rightarrow_\beta Q \twoheadrightarrow_{\beta\eta} N$, dla pewnego Q. Graficznie:*



Wniosek 4.8 *Term ma nieskończoną β-redukcję wtedy i tylko wtedy gdy ma nieskończoną $\beta\eta$-redukcję.* (*Inaczej: term ma własność* β-SN *wtedy i tylko wtedy, gdy ma własność* $\beta\eta$-SN.)

Dowód: Załóżmy, że mamy nieskończoną $\beta\eta$-redukcję. Jeśli beta-kroki występują w tej redukcji nieskończenie wiele razy, to wniosek 4.7 pozwala z niej uzyskać nieskończoną β-redukcję (poprzez „przesuwanie" beta-kroków w lewo). A nie może być tak, że prawie wszystkie kroki są typu eta. Implikacja w przeciwną stronę jest oczywiście oczywista. ∎

Nieco trudniej jest udowodnić równoważność „zwykłej" β- i $\beta\eta$-normalizacji. Najpierw musimy pokazać, że każdy ciąg beta-eta-redukcji można „posortować" tak aby wszystkie eta-redukcje były na końcu (twierdzenie 4.10 o odkładaniu η-redukcji). Niech $\succ$ oznacza najmniejszą relację w zbiorze termów, spełniającą warunki:

- $x \succ x$, dla dowolnej zmiennej x;
- jeśli $M \succ M'$ oraz $x \notin \mathrm{FV}(M)$, to $\lambda x.Mx \succ M'$;
- jeśli $M \succ M'$, to $\lambda x.M \succ \lambda x.M'$;
- jeśli $M \succ M'$ oraz $N \succ N'$, to $MN \succ M'N'$.

Relacja $\succ$ tak się ma do η-redukcji, jak relacja $\xrightarrow{1}$ do β-redukcji. Na przykład $\lambda x(\lambda y.\, zy)x \succ z$, ale $\lambda xy.\, zxy \not\succ z$, chociaż $\lambda xy.\, zxy \twoheadrightarrow_\eta z$. Mamy więc znowu ostre inkluzje:

$$\to_\eta \subsetneq \succ \subsetneq \twoheadrightarrow_\eta.$$

Lemat 4.9

(1) Jeśli $M \succ M'$ oraz $N \succ N'$, to $M[x := N] \succ M'[x := N']$.

(2) Jeśli $M \succ M' \to_\beta P$ to istnieje taki term Q, że $M \twoheadrightarrow_\beta Q \succ P$.

Dowód: Część (1) można udowodnić przez łatwą indukcję ze względu na definicję $M \succ M'$. W części (2) najpierw udowodnimy szczególny przypadek:

$$\text{Jeśli } L \succ \lambda y\, R \text{ oraz } N \succ N', \text{ to istnieje takie } Q, \text{ że } LN \twoheadrightarrow_\beta Q \succ R[y := N'], \qquad (*)$$

przez indukcję ze względu na długość L. Są tu dwa przypadki. Pierwszy, gdy $L = \lambda y\, L_1$ i $L_1 \succ R$; wtedy $LN \to_\beta L_1[y := N] \succ R[y := N']$ z części (1). W drugim przypadku, gdy $L = \lambda z.\, L_1 z$ i $L_1 \succ \lambda y\, R$, stosujemy indukcję.

Mając (*) dowodzimy części (2) przez indukcję ze względu na definicję $M \succ M'$. ∎

Twierdzenie 4.10 (o odkładaniu η-redukcji)

Jeśli $M \twoheadrightarrow_{\beta\eta} N$ to $M \twoheadrightarrow_\beta P \twoheadrightarrow_\eta N$, dla pewnego P.

Dowód: Jeśli $M \twoheadrightarrow_{\beta\eta} N$, to każdy krok eta-redukcji można uważać za krok typu $\succ$, i „przesuwać" w prawo z pomocą lematu 4.9(2). ∎

Lemat 4.11 *Jeśli $M \succ N$ oraz N jest w postaci β-normalnej, to M ma postać β-normalną.*

Dowód: Przez indukcję ze względu na definicję $M \succ N$ dowodzimy, że postać β-normalna M' termu M istnieje i ma własność $M' \succ N$.

Nieoczywisty przypadek jest wtedy, gdy $M = PQ$, $N = P'Q'$ oraz $P \succ P'$ i $Q \succ Q'$. Z założenia indukcyjnego mamy postaci β-normalne P'' i Q'' termów P i Q i wszystko jest dobrze, jeżeli P'' nie jest abstrakcją.

Jeśli P'' jest abstrakcją, to musimy skorzystać z założenia indukcyjnego, że $P'' \succ P'$. Ponieważ term P' abstrakcją nie jest, więc musi być tak: $P'' = \lambda u.\, P'''u$ oraz $P''' \succ P'$. Przy tym P''' też nie jest abstrakcją, bo inaczej P'' nie jest β-normalne. I teraz mamy

$$PQ \twoheadrightarrow_\beta (\lambda u.\, P'''u)Q'' \to_\beta P'''Q'' \succ P'Q'.$$

Pozostałe przypadki są w zasadzie rutynowe. ∎

Lemat 4.12 *Jeśli $M \succ N$ oraz N ma postać β-normalną, to M też ma.*

Dowód: Tu jest indukcja ze względu na liczbę kroków redukcji N do postaci normalnej. Krok bazowy to lemat 4.11. W kroku indukcyjnym mamy $M \succ N \to_\beta N' \twoheadrightarrow_\beta N''$, gdzie N'' jest normalne. Korzystamy z lematu 4.9 i stosujemy indukcję. ∎

Twierdzenie 4.13 *Term ma postać beta-normalną wtedy i tylko wtedy, gdy ma postać beta-eta-normalną.*

Dowód: Aby udowodnić implikację z lewej do prawej, wystarczy zauważyć, że η-redukowanie postaci beta-normalnej nie tworzy nowych beta-redeksów (ćwiczenie 4), musi więc w końcu dać postać $\beta\eta$-normalną.

W dowodzie implikacji odwrotnej korzysta się z twierdzenia o odkładaniu eta-redukcji. Mamy więc redukcję $M \twoheadrightarrow_\beta N \twoheadrightarrow_\eta P$ do postaci $\beta\eta$-normalnej. Ponieważ $\to_\eta$ jest zawarte w $\succ$, więc z lematu 4.12 wynika, że N (a więc i M) ma postać β-normalną. ∎

Ćwiczenia

1. W jaki sposób beta-redukcja może spowodować powstanie nowego beta-redeksu? Eta-redeksu?
2. Uzupełnić szczegóły dowodu twierdzenia 4.10.
3. Dlaczego twierdzenie 4.10 nie wynika natychmiast z lematu 4.6?
4. Pokazać, że jeśli M jest w postaci β-normalnej oraz $M \to_\eta N$ to N jest w postaci β-normalnej.

5 Standaryzacja

W jednym termie może występować więcej niż jeden redeks. Możliwe są więc różne *strategie redukcji*. Jedna z nich polega na wybieraniu zawsze tego redeksu, który zaczyna się najbardziej na lewo. Mówimy wtedy o redukcji *lewostronnej*. Okazuje się, że strategia redukcji lewostronnej jest strategią *normalizujacą*, tj. tą metodą zawsze dojdziemy do postaci normalnej, jeśli ona istnieje. Popatrzmy na przykład na term $\mathbf{K}z\Omega$. W zależności od wyboru strategii redukcji możemy dojść do postaci normalnej lub redukować ten term w nieskończoność.

W istocie prawdziwe jest nieco silniejsze twierdzenie, zwane *twierdzeniem o standaryzacji.* Powiemy, że wyprowadzenie

$$M_0 \to M_1 \to \cdots \to M_n$$

jest *standardowe*, jeżeli dla dowolnego $i = 0, \ldots, n-2$, w kroku $M_i \to M_{i+1}$ redukujemy redeks położony[3] nie dalej od początku termu niż redeks, który redukujemy w następnym kroku $M_{i+1} \to M_{i+2}$. Na przykład ta redukcja jest standardowa:

$$\begin{aligned}(\lambda x.xx)((\lambda zy.z(zy))(\lambda u.u)(\lambda w.w)) &\to (\lambda x.xx)((\lambda y.(\lambda u.u)((\lambda u.u)y))(\lambda w.w)) \to \\ \to (\lambda x.xx)((\lambda u.u)((\lambda u.u)(\lambda w.w))) &\to (\lambda x.xx)((\lambda u.u)(\lambda w.w)) \to (\lambda x.xx)(\lambda w.w).\end{aligned}$$

a ta nie jest:

$$\begin{aligned}(\lambda x.xx)((\lambda zy.z(zy))(\lambda u.u)(\lambda w.w)) &\to (\lambda x.xx)((\lambda y.(\lambda u.u)((\lambda u.u)y))(\lambda w.w)) \to \\ (\lambda x.xx)((\lambda y.(\lambda u.u)y)(\lambda w.w)) &\to (\lambda x.xx)((\lambda y.y)(\lambda w.w)) \to (\lambda x.xx)(\lambda w.w).\end{aligned}$$

Twierdzenie 5.1 *Jeśli $M \twoheadrightarrow_\beta N$, to istnieje standardowa redukcja z M do N.*

Zajmiemy się teraz dowodem twierdzenia 5.1. Zaczniemy od tego, że każdy term M jest jednej z dwóch możliwych postaci:

1) $\lambda\vec{x}.z\vec{R}$;

2) $\lambda\vec{x}.(\lambda y.P)Q\vec{R}$,

gdzie długość wektora $\vec{x}$ może być zerem, a z może występować w $\vec{x}$ lub nie. W przypadku (1) mówimy, że term jest w *czołowej postaci normalnej*[4]. Natomiast w przypadku (2) mówimy, że term ma *redeks czołowy* $(\lambda y.P)Q$. Krok redukcji

$$M = \lambda\vec{x}.(\lambda y.P)Q\vec{R} \to_\beta \lambda\vec{x}.P[y := Q]\vec{R} = N$$

nazywamy wtedy *redukcją czołową* i zapisujemy $M \xrightarrow{\mathrm{h}} N$. Inne kroki redukcji nazywamy *wewnętrznymi*, a w takim przypadku używamy symbolu $\xrightarrow{\mathrm{i}}$. Kluczowy lemat 5.3 mówi, że redukcje czołowe można wykonać przed wewnętrznymi. Dowód tego lematu nie jest jednak tak oczywisty, jak mogłoby się wydawać (ćwiczenie 3) i wymaga pewnych przygotowań, w tym dwóch pomocniczych relacji. Idea dowodu polega na użyciu redukcji postaci $\xrightarrow{1}$, które łatwiej rozbić na część czołową i wewnętrzną.

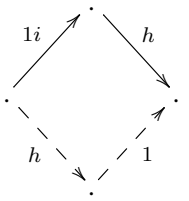
[3] Liczy się początek redeksu.

[4] Uwaga: czołowa postać normalna nie jest na ogół postacią normalną.

- Napiszemy $M \overset{1i}{\to} N$, gdy istnieje ciąg redukcji wewnętrznych z M do N, które jednocześnie stanowią redukcję postaci $\overset{1}{\to}$ (ćwiczenie 4).
- Natomiast $M \Rrightarrow N$ zachodzi wtedy gdy $M \overset{h}{\twoheadrightarrow} P \overset{1i}{\to} N$, oraz każdy term Q występujący w redukcji $M \overset{h}{\twoheadrightarrow} P$ spełnia warunek $Q \overset{1}{\to} N$ (ćwiczenie 5).

Lemat 5.2

(1) Jeśli $M \overset{1i}{\to} N \overset{h}{\to} P$, *to* $M \overset{h}{\to} Q \overset{1}{\to} P$, *dla pewnego* Q.



(2) Jeśli $M \Rrightarrow M'$ *oraz* $N \Rrightarrow N'$ *to* $M[x := N] \Rrightarrow M'[x := N']$ (∗∗)

(3) Jeśli $M \overset{1}{\to} N$, *to* $M \Rrightarrow N$.

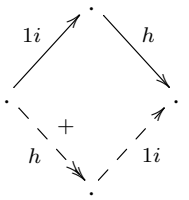
Dowód: Ćwiczenie. ∎

Lemat 5.3 *Jeżeli* $M \twoheadrightarrow_\beta N$, *to* $M \overset{h}{\twoheadrightarrow} P \overset{i}{\twoheadrightarrow} N$, *dla pewnego* P.

Dowód: Relacja $\Rrightarrow$ rozkłada się z definicji na $\overset{h}{\twoheadrightarrow}$ i $\overset{1i}{\to}$, i na dodatek zawiera relację $\overset{1}{\to}$ (lemat 5.2(3)). Zatem,

$$\text{jeśli } M \overset{1}{\to} N, \text{ to } M \overset{h}{\twoheadrightarrow} L \overset{1i}{\to} N, \text{ dla pewnego } L,$$

czyli $\overset{1}{\to}$ rozkłada się na fazę czołową i fazę postaci $\overset{1i}{\to}$. Z lematu 5.2(1) wynika taki obrazek:



A więc relacje $\overset{1i}{\to}$ i $\overset{h}{\twoheadrightarrow}$ można permutować, a to już wystarczy, bo przecież $\overset{i}{\to} \subseteq \overset{1i}{\to} \subseteq \overset{i}{\twoheadrightarrow}$. ∎

Dowód twierdzenia: Na mocy lematu 5.3 mamy $M \overset{h}{\twoheadrightarrow} M' \overset{i}{\twoheadrightarrow} N$. Po wykonaniu wszystkich redukcji czołowych, kształt termu się częściowo „ustala”, jeśli więc $M' = \lambda\vec{x}.z\vec{R}$ albo $M' = \vec{x}.(\lambda y.P)Q\vec{R}$, to dalsze redukcje odbywają się wewnątrz P, Q i $\vec{R}$. Stosując indukcję do termów P, Q i $\vec{R}$, otrzymujemy standardowe redukcje, które wykonujemy „od lewej” tj. najpierw w P, potem w Q i potem w kolejnych wyrazach ciągu $\vec{R}$. ∎

Udowodniliśmy więc twierdzenie o standaryzacji. A oto najważniejszy wniosek z niego.

Wniosek 5.4 *Jeśli M ma postać normalną N, to istnieje lewostronna redukcja z M do N.*

Powyższy wniosek[5] można odczytać tak: Jeśli istnieje nieskończona redukcja lewostronna zaczynająca się od M, to term M nie ma postaci normalnej. Można to stwierdzenie wzmocnić w taki sposób. Powiemy, że nieskończony ciąg redukcji jest *quasi-lewostronny*, jeśli nieskończenie wiele razy następuje w tym ciągu redukcja redeksu położonego najbardziej na lewo.

Fakt 5.5 *Jeśli istnieje nieskończony quasi-lewostronny ciąg redukcji zaczynający się od M, to term M nie ma postaci normalnej.*

Dowód: Na potrzeby tego dowodu powiedzmy, że term jest *wytrzymały*, gdy ma redukcję quasi-lewostronną, w której wystepuje nieskończenie wiele kroków postaci $\xrightarrow{\mathrm{h}}$. Zauważmy, że jeśli N jest wytrzymały, to $N \xrightarrow{\mathrm{h}} N'$ dla pewnego wytrzymałego termu N' (ćwiczenie 6).

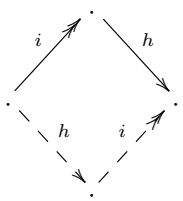
Załóżmy teraz, że M ma postać normalną. Przez indukcję ze względu na jej długość udowodnimy, że M nie może mieć nieskończonej redukcji quasi-lewostronnej. Przypuśćmy najpierw, że M jest termem wytrzymałym. Używając ćwiczenia 6 można wtedy zdefiniować ciąg wytrzymałych termów N_i, takich że $N_0 = M$ oraz $N_i \xrightarrow{\mathrm{h}}{}^{+} N_{i+1}$ dla dowolnego $i \in \mathbb{N}$. Inaczej mówiąc, „wyciągając na początek" redukcje czołowe otrzymamy nieskończoną redukcję czołową (w szczególności lewostronną) termu M. Nie może więc istnieć postać normalna.

Zatem jeśli $M = M_0 \to M_1 \to M_2 \to \cdots$ jest nieskończoną redukcją quasi-lewostronną, to od pewnego miejsca mamy tylko redukcje wewnętrzne postaci $\lambda\vec{z}.y\vec{P} \xrightarrow{\mathrm{i}} \lambda\vec{z}.y\vec{P'} \xrightarrow{\mathrm{i}} \lambda\vec{z}.y\vec{P''} \xrightarrow{\mathrm{i}} \cdots$ Spośród redukcji lewostronnych występujących w tym ciągu, nieskończenie wiele dotyczy tej samej składowej wektora $\vec{P}$, więc wystarczy użyć założenia indukcyjnego. ∎

Ćwiczenia

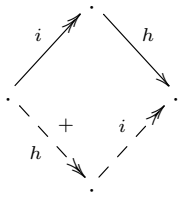
1. Jaki jest parametr indukcji w dowodzie twierdzenia 5.1?
2. Pokazać, że niepusta redukcja jest standardowa wtedy i tylko wtedy, gdy jest jednej z postaci:
   - $(\lambda x\, P)Q \to P[x := Q] = M_1 \to \cdots \to M_n$;
   - $\lambda x\, M_1 \to \lambda x\, M_2 \to \cdots \to \lambda x\, M_n$;
   - $M_1 N_1 \to \cdots \to M_n N_1 \to \cdots \to M_n N_m$,

   gdzie ciągi $M_1 \to \cdots \to M_n$ i $N_1 \to \cdots \to N_m$ są standardowymi redukcjami.
3. Diagram poniżej przedstawia naiwną próbę dowodu lematu 5.3. Na czym polega błąd?



[5] Ten fakt bywa nazywany *twierdzeniem o normalizacji*. Obecnie rzadziej używa się tej nazwy, bo „normalizacja" zwykle oznacza istnienie postaci normalnych.

4. Zdefiniować poprawnie relację $\xrightarrow{\text{1i}}$ i udowodnić lemat 5.2(1). Czy ten dowód pozostanie poprawny, jeśli zdefiniujemy $\xrightarrow{\text{1i}}$ jako iloczyn relacji $\xrightarrow{\text{1}}$ i $\xrightarrow{\text{i}}{\!\!\!\!\rightarrow}$?
5. Zdefiniować poprawnie relację $\Rrightarrow$ i udowodnić lemat 5.2(2–3).
6. Term jest *wytrzymały*, gdy ma redukcję quasi-lewostronną, w której jest nieskończenie wiele kroków postaci $\xrightarrow{\text{h}}$. Udowodnić, że jeśli N jest wytrzymały, to $N \xrightarrow{\text{h}} N'$ dla pewnego wytrzymałego termu N'. *Wskazówka: Z twierdzenia o standaryzacji wynika poprawność diagramu:*



6 Rachunek $\lambda\mathbf{I}$

Term M jest nazywany $\lambda\mathbf{I}$-termem, jeżeli nie występuje w nim jako podterm żadna abstrakcja postaci $\lambda x\, N$, gdzie $x \notin \mathrm{FV}(N)$. To znaczy, że każda abstrakcja wiąże przynajmniej jedno wystąpienie zmiennej. Nietrudno zauważyć, że jeśli M jest $\lambda\mathbf{I}$-termem, oraz $M \twoheadrightarrow_\beta M'$, to M' też jest $\lambda\mathbf{I}$-termem (chociaż nie na odwrót). Dlatego można mówić o *rachunku* $\lambda\mathbf{I}$, w którym rozważa się tylko takie termy. Pełny rachunek lambda nazywany też bywa rachunkiem $\lambda\mathbf{K}$. Ta terminologia bierze się z tradycyjnych oznaczeń:

$$\mathbf{I} = \lambda x.x$$
$$\mathbf{K} = \lambda xy.x$$

Najciekawsza własność $\lambda\mathbf{I}$-termów jest taka:

Twierdzenie 6.1 *Jeśli $\lambda\mathbf{I}$-term ma postać normalną, to ma własność silnej normalizacji.*

Dowód: Załóżmy, że M jest $\lambda\mathbf{I}$-termem o postaci normalnej M'. Na mocy standaryzacji mamy lewostronną redukcję $M \twoheadrightarrow^\ell M'$. Przez indukcję ze względu na długość tej redukcji pokażemy, że $M \in \mathrm{SN}$. W tym celu wystarczy wiedzieć, że jeśli $M \to^\ell N \in \mathrm{SN}$, to $M \in \mathrm{SN}$. To zaś udowodnimy przez indukcję ze względu na długość postaci normalnej termu N.

Przypadek 1: Term M jest w czołowej postaci normalnej $\lambda\vec{x}.\, yR_1R_2\ldots R_k$. Mamy wtedy $N = \lambda\vec{x}.\, yR'_1\ldots R'_k$, gdzie $R_i \to R'_i$, dla pewnego i, a w pozostałych przypadkach $R_i = R'_i$. Ponieważ $N \in \mathrm{SN}$, więc $R'_i \in \mathrm{SN}$. Ponadto, postać normalna termu R'_i jest krótsza niż postać normalna N, więc teza wynika z założenia indukcyjnego.

Przypadek 2: Term M ma redeks czołowy, tj. $M = \lambda\vec{x}.\,(\lambda y\, P)QR_1\ldots R_k$. Wtedy N ma postać $N = \lambda\vec{x}.\, P[y := Q]R_1\ldots R_k$. Ponieważ $N \in \mathrm{SN}$, to wszystkie termy $P, Q, R_1, \ldots, R_k$ są w SN (ćwiczenie 1).

Z tego powodu, każdy ciąg redukcji wewnętrznych $M \xrightarrow{\text{i}} M_1 \xrightarrow{\text{i}} M_2 \xrightarrow{\text{i}} \cdots$ musi być skończony. A każdy ciąg redukcji postaci $M \xrightarrow{\text{i}}{\!\!\!\!\rightarrow} \lambda\vec{x}.\,(\lambda y\, P')Q'R'_1\ldots R'_k \xrightarrow{\text{h}} \lambda\vec{x}.\, P'[y := Q']R'_1\ldots R'_k \twoheadrightarrow \cdots$ też musi się skończyć, bo $N \twoheadrightarrow \lambda\vec{x}.\, P'[y := Q']R'_1\ldots R'_k$, oraz $N \in \mathrm{SN}$ ∎

Ćwiczenia

1. Pokazać, że jeśli $N[y := Q] \in \mathrm{SN}$, to $N \in \mathrm{SN}$. Jeśli dodatkowo $y \in \mathrm{FV}(N)$, to także $Q \in \mathrm{SN}$.
2. Czy zawsze jeśli $M \to N \in \mathrm{SN}$ to $M \in \mathrm{SN}$?
3. Czy nastepujący dowód wniosku 5.4 jest poprawny?
   Załóżmy, że term M ma postać normalną ale ma też nieskończoną redukcję. Z Twierdzenia 6.1 wynika, że przyczyną tego musi być jakiś podterm właściwy B termu M, który nie ma postaci normalnej i który jest usuwany przez pewną redukcję postaci $(\lambda x.\, P)Q \to P$, gdzie $x \notin FV(P)$ (bo jest podtermem Q). Wybierzmy takie B możliwie najbardziej na lewo. Skoro ten term jest usuwany przez jakąś redukcję, to tym bardziej przez redukcję lewostronną.
4. Czy jeśli M ma postać normalną i każdy jego podterm właściwy jest SN, to $M \in \mathrm{SN}$?
5. Czy istnieje zbiór $\mathcal{A} \subseteq \mathrm{SN}$ o tej własności, że $M \in \mathcal{A}$ wtedy i tylko wtedy, gdy $MN_1 \ldots N_k \in \mathrm{SN}$, dla dowolnych $N_1, \ldots, N_k \in \mathcal{A}$?

7 Siła wyrazu

Rachunek lambda jest systemem pozornie bardzo prostym. Abstrakcja i aplikacja wydają się trywialnymi operacjami, i może się zdawać, że niczego ciekawego nie da się z nich uzyskać. Okazuje się jednak, że siła wyrazu tego rachunku jest tak duża jak moc obliczeniowa maszyn Turinga.

Kombinator(y) punktu stałego

Zaczniemy od dość paradoksalnej własności rachunku lambda. Każde równanie stałopunktowe ma w nim rozwiązanie. Odpowiedzialny za to jest na przykład taki term:

$$\mathbf{Y} := \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)).$$

Główna własność termu $\mathbf{Y}$ jest taka: Dla dowolnego termu F zachodzi beta-równość:

$$(*) \qquad F(\mathbf{Y}(F)) =_\beta \mathbf{Y}(F).$$

Rzeczywiście, $\mathbf{Y}(F) =_\beta (\lambda x.F(xx))(\lambda x.F(xx)) =_\beta F((\lambda x.F(xx))(\lambda x.F(xx)) =_\beta F(\mathbf{Y}(F))$. Termy o własności $(*)$ nazywamy *kombinatorami punktu stałego*.

Fakt 7.1 *Dla dowolnego termu M istnieje taki term N, że $N =_\beta M[x := N]$.*

Dowód: Wystarczy przyjąć $N = \mathbf{Y}(\lambda x.M)$. ∎

Przykład 7.2

- Istnieje taki term M, że $Mxy =_\beta MyxM$. Można przyjąć $M = \mathbf{Y}(\lambda mxy.myxm)$. Wtedy $M =_\beta \lambda xy.MyxM$, skąd także $Mxy =_\beta MyxM$.
- Istnieje taki term M, że dla dowolnego N zachodzi $MN =_\beta MNN$, na przykład $M = \mathbf{Y}(\lambda mx.mxx)$.

Proste rzeczy

Teraz pokażemy jak w rachunku lambda można zinterpretować pewne proste konstrukcje. Zaczniemy od wartości logicznych

$$\textbf{true} = \lambda xy.x \qquad\qquad \textbf{false} = \lambda xy.y$$

i instrukcji warunkowej

$$\textbf{if } P \textbf{ then } Q \textbf{ else } R = PQR.$$

Łatwo sprawdzić, że **if true then** Q **else** $R \twoheadrightarrow_\beta Q$ oraz **if false then** Q **else** $R \twoheadrightarrow_\beta R$.

Następna konstrukcja to para uporządkowana. Przyjmiemy

$$\begin{array}{rcl} \langle M, N\rangle & = & \lambda x.xMN; \\ \pi_i & = & \lambda x_1x_2.x_i, \text{ dla } i = 1, 2; \\ \Pi_i & = & \lambda p.\, p\pi_i. \end{array}$$

Jak należy się spodziewać, mamy $\Pi_i\langle M_1, M_2\rangle \twoheadrightarrow_\beta M_i$. Zauważmy jednak, że nie zachodzi równość $\langle \Pi_1 M, \Pi_2 M\rangle =_\beta M$, tj. nasza para uporządkowana nie jest *surjektywna*.

Uwaga: Dodanie do rachunku lambda surjektywnej pary uporządkowanej powoduje utratę własności Churcha-Rossera. Oznacza to w szczególności, że w (beztypowym) rachunku lambda nie można takiej operacji zdefiniować.

Liczby naturalne reprezentujemy w rachunku lambda jako tzw. *liczebniki Churcha*:

$$c_n = \lambda fx.f^n(x),$$

gdzie notacja $f^n(x)$ oznacza oczywiście term $f(f(\cdots(x)\cdots))$, w którym f występuje n razy. Zwykle zamiast c_n będziemy pisać $\mathbf{n}$, co jest mniej precyzyjne ale wygodne. Więc na przykład:

$$\begin{array}{rcl} \mathbf{0} & = & \lambda fx.x; \\ \mathbf{1} & = & \lambda fx.fx; \\ \mathbf{2} & = & \lambda fx.f(fx), \end{array}$$

i tak dalej. Zauważmy, że $\mathbf{1} =_{\beta\eta} \mathbf{I}$, ale $\mathbf{1} \neq_\beta \mathbf{I}$.

Powiemy, że funkcja częściowa $f : \mathbb{N}^k \rightharpoonup \mathbb{N}$ jest *definiowalna* w beztypowym rachunku lambda (λ-*definiowalna*) jeżeli istnieje term zamknięty F, spełniający dla dowolnych liczb $n_1, \ldots, n_k \in \mathbb{N}$ następujące warunki:

- Jeżeli $f(n_1, \ldots, n_k) = m$, to $F\mathbf{n}_1 \ldots \mathbf{n}_k =_\beta \mathbf{m}$;
- Jeżeli $f(n_1, \ldots, n_k)$ jest nieokreślone, to $F\mathbf{n}_1 \ldots \mathbf{n}_k$ nie ma postaci normalnej.

Mówimy oczywiście, że F *definiuje* lub *reprezentuje* funkcję f w rachunku lambda. W przypadku, gdy dodatkowo dla wszystkich $n_1, \ldots, n_k \in \mathbb{N}$ spełniony jest warunek

- Jeżeli $f(n_1, \ldots, n_k)$ jest określone, to $F\mathbf{n}_1 \ldots \mathbf{n}_k$ ma własność silnej normalizacji,

to powiemy, że F *dobrze* definiuje funkcję f, którą nazwiemy *dobrze* definiowalną. Kilka przykładów termów dobrze definiujących pewne funkcje mamy poniżej.

Przykład 7.3

- Następnik: $\mathbf{succ} = \lambda nfx.f(nfx)$;
- Dodawanie: $\mathbf{add} = \lambda mnfx.mf(nfx)$;
- Mnożenie: $\mathbf{mult} = \lambda mnfx.m(nf)x$;
- Potęgowanie: $\mathbf{exp} = \lambda mnfx.mnfx$;
- Test na zero: $\mathbf{zero} = \lambda m.m(\lambda y.\mathbf{false})\mathbf{true}$;
- Funkcja k-argumentowa stale równa zeru: $\mathbf{Z}_k = \lambda m_1 \ldots m_k.\mathbf{0}$;
- Rzut k-argumentowy na i-tą współrzędną: $\mathbf{\Pi}^i_k = \lambda m_1 \ldots m_k.m_i$.

Reprezentowanie funkcji częściowo rekurencyjnych

Przypomnijmy, że funkcje częściowo rekurencyjne[6] tworzą najmniejszą klasę funkcji częściowych nad $\mathbb{N}$ zawierającą *funkcje bazowe*:

- następnik, $succ(n) = n + 1$;
- rzuty, $\Pi^i_k(n_1, \ldots, n_k) = n_i$;
- funkcje stale równe zeru, $Z_k(n_1, \ldots, n_k) = 0$,

i zamkniętą ze względu na składanie, rekursję prostą i operację minimum. Na wszelki wypadek, przypomnijmy też znaczenie tych terminów.

- Funkcja $f : \mathbb{N}^k \multimap \mathbb{N}$ powstaje przez *składanie* funkcji $h : \mathbb{N}^\ell \multimap \mathbb{N}$ z funkcjami $g_1, \ldots, g_\ell : \mathbb{N}^k \multimap \mathbb{N}$, gdy
  - $\mathrm{Dom}(f) = \{\vec{n} \mid \vec{n} \in \mathrm{Dom}(g_i)$ dla wszystkich i, oraz $(g_1(\vec{n}), \ldots, g_\ell(\vec{n})) \in \mathrm{Dom}(h)\}$;
  - $f(\vec{n}) = h(g_1(\vec{n}), \ldots, g_\ell(\vec{n}))$, dla dowolnego $\vec{n} \in \mathrm{Dom}(f)$.
- Funkcja $f : \mathbb{N}^{k+1} \multimap \mathbb{N}$ powstaje przez *rekursję prostą* z funkcji $h : \mathbb{N}^{k+2} \multimap \mathbb{N}$ i funkcji $g : \mathbb{N}^k \multimap \mathbb{N}$, jeżeli dla dowolnego $n \in \mathbb{N}$ i dowolnego $\vec{n} \in \mathbb{N}^k$ spełnione są równania
  - $f(0, \vec{n}) = g(\vec{n})$;
  - $f(n + 1, \vec{n}) = h(f(n, \vec{n}), n, \vec{n})$,

  przy czym równanie uważamy za spełnione, gdy obie strony są określone i równe, lub obie strony są nieokreślone. Wartość wyrażenia $h(f(n, \vec{n}), n, \vec{n})$ jest określona wtedy i tylko wtedy gdy $(n, \vec{n}) \in \mathrm{Dom}(f)$ oraz $(f(n, \vec{n}), n, \vec{n}) \in \mathrm{Dom}(h)$.

[6] To to samo co klasa funkcji częściowych obliczalnych przez maszyny Turinga. Dowód twierdzenia 7.9 wykorzystujący maszyny Turinga jest w Dodatku A.

- Funkcja $f : \mathbb{N}^k \multimap \mathbb{N}$ powstaje z funkcji $h : \mathbb{N}^{k+1} \multimap \mathbb{N}$ przez zastosowanie *operacji minimum*, co zapisujemy $f(\vec{n}) = \mu y[h(\vec{n}, y) = 0]$, gdy dla dowolnego $\vec{n} \in \mathbb{N}^k$,
  - Jeśli istnieje takie m, że $h(\vec{n}, m) = 0$ oraz wszystkie wartości $h(\vec{n}, i)$ dla $i < m$ są określone i różne od zera, to $f(\vec{n}) = m$.
  - Jeśli takiego m nie ma, to $f(\vec{n})$ jest nieokreślone.

Klasa funkcji *pierwotnie rekurencyjnych* to najmniejsza klasa funkcji całkowitych nad $\mathbb{N}$ zawierająca funkcje bazowe i zamknięta ze względu na składanie i rekursję prostą. Szczególne funkcje pierwotnie rekurencyjne to *funkcja pary*

$$p(m, n) = \frac{(m + n)(m + n + 1)}{2} + m,$$

i dwie funkcje *left* i *right*, takie że dla dowolnego n,

$$n = p(\mathit{left}(n), \mathit{right}(n)),$$

czyli funkcje odwrotne do funkcji pary. Mamy następujące

Twierdzenie 7.4 (o postaci normalnej Kleene'go) *Każdą funkcję częściowo rekurencyjną można przedstawić w postaci*

$$f(\vec{n}) = \mathit{left}(\mu y[g(\vec{n}, y) = 0]),$$

gdzie g jest funkcją pierwotnie rekurencyjną.[7]

Udowodnimy teraz, że każda funkcja częściowo rekurencyjna jest definiowalna w rachunku lambda. Zaczynamy od najłatwiejszego.

Lemat 7.5 *Funkcje bazowe są lambda-definiowalne.*

Dowód: Przykład 7.3. ∎

Lemat 7.6 *Jeśli funkcja całkowita f powstaje przez składanie lambda-definiowalnych funkcji całkowitych, to też jest lambda-definiowalna.*

Dowód: Oczywiste. Ale tylko dlatego, że mowa o funkcjach całkowitych. ∎

Lemat 7.7 *Jeśli funkcja całkowita f powstaje przez rekursję prostą z lambda-definiowalnych funkcji całkowitych, to też jest lambda-definiowalna.*

Dowód: To już nie jest oczywiste. Załóżmy, że f jest zdefiniowana równaniami

$$\begin{array}{lcl} f(0, \vec{n}) & = & g(\vec{n}); \\ f(n + 1, \vec{n}) & = & h(f(n, \vec{n}), n, \vec{n}), \end{array}$$

[7] Tak naprawdę to jest nawet funkcja elementarnie rekurencyjna.

i że funkcje g i h są definiowalne odpowiednio za pomocą termów G i H. Zdefiniujmy pomocnicze termy

$$\begin{array}{lcl} \mathbf{Step} & = & \lambda p.\langle \mathbf{succ}(\Pi_1 p), H(\Pi_2 p)(\Pi_1 p)x_1 \ldots x_m\rangle; \\ \mathbf{Init} & = & \langle \mathbf{0}, Gx_1 \ldots x_m\rangle. \end{array}$$

Funkcja f jest wtedy definiowalna termem

$$F = \lambda x x_1 \ldots x_m.\, \Pi_2(x\,\mathbf{Step}\,\mathbf{Init}).$$

Ta definicja wyraża następujący algorytm obliczania wartości funkcji f: generujemy ciąg par

$$(0, a_0), (1, a_1), \ldots, (n, a_n),$$

gdzie $a_0 = g(n_1, \ldots, n_m)$, każde a_{i+1} to $h(a_i, i, n_1, \ldots, n_m)$ oraz $a_n = f(n, n_1, \ldots, n_m)$. ∎

Wniosek 7.8 *Funkcje pierwotnie rekurencyjne są lambda-definiowalne.*

Ponieważ mamy twierdzenie Kleene'go, więc pozostaje już (prawie) tylko pokazać lambda-definiowalność funkcji określonych przez minimum. Można do tego wykorzystać kombinator punktu stałego $\mathbf{Y}$, rozwiązując odpowiednie równanie stałopunktowe. My to zrobimy trochę delikatniej, „opóźniając" działanie kombinatora, dzieki czemu uzyskamy silniejszy wynik.

Twierdzenie 7.9 *Wszystkie funkcje częściowo rekurencyjne są lambda-definiowalne.*

Dowód: Niech $f(\vec{n}) = \mathit{left}(\mu y[g(\vec{n}, y) = 0])$, gdzie g jest funkcją pierwotnie rekurencyjną. Na mocy wniosku 7.8, zarówno g jak *left* są lambda-definiowalne pewnymi termami G i L. Określimy pomocniczy term

$$W = \lambda y.\mathbf{if}\ \mathbf{zero}(G\vec{x}y)\ \mathbf{then}\ \lambda w.Ly\ \mathbf{else}\ \lambda w.w(\mathbf{succ}\, y)w.$$

Funkcja f jest definiowana termem

$$F = \lambda \vec{x}.W\mathbf{0}W.$$

Rzeczywiście, przypuśćmy najpierw, że $\mu y[g(\vec{n}, y) = 0] = n$, oraz $\mathit{left}(n) = r$. Wtedy mamy taki ciąg redukcji:

$$F\vec{\mathbf{n}} \twoheadrightarrow_\beta \overline{W}\mathbf{0}\overline{W} \twoheadrightarrow_\beta \overline{W}\mathbf{1}\overline{W} \twoheadrightarrow_\beta \cdots \twoheadrightarrow_\beta \overline{W}\mathbf{n}\overline{W} \twoheadrightarrow_\beta L\mathbf{n} \twoheadrightarrow_\beta \mathbf{r}, \tag{1}$$

gdzie $\overline{W}$ oznacza wynik podstawienia $\vec{\mathbf{n}}$ na $\vec{x}$ w termie W.

Jeśli minimum jest nieokreślone, to znaczy, że wszystkie wartości $g(\vec{n}, m)$ są określone i różne od zera, bo funkcja g jest całkowita. Wtedy mamy nieskończony ciąg redukcji

$$F\vec{\mathbf{n}} \twoheadrightarrow_\beta \overline{W}\mathbf{0}\overline{W} \twoheadrightarrow_\beta \overline{W}\mathbf{1}\overline{W} \twoheadrightarrow_\beta \cdots \twoheadrightarrow_\beta \overline{W}\mathbf{n}\overline{W} \twoheadrightarrow_\beta \cdots$$

Ten ciąg jest quasi-lewostronny, a zatem na mocy Faktu 5.5 postaci normalnej nie ma. ∎

W istocie prawdziwy jest silniejszy fakt:

Fakt 7.10 *Wszystkie funkcje częściowo rekurencyjne są dobrze lambda-definiowalne.*

Dowód Faktu 7.10 opiera się na tej samej konstrukcji co dowód twierdzenia 7.9, ale wymaga dodatkowo dowodu silnej normalizacji, który opuszczamy.

Wniosek 7.11 *Następujące problemy są nierozstrzygalne:*

- *Dane termy M i N. Czy $M \twoheadrightarrow_\beta N$?*
- *Dane termy M i N. Czy $M =_\beta N$?*
- *Dany term M. Czy M ma postać normalną?*
- *Dany term M. Czy M ma własność silnej normalizacji?*

Druga i trzecia część wniosku 7.11 są szczególnymi przypadkami twierdzenia 7.13, które jest w rachunku lambda odpowiednikiem twierdzenia Rice'a. Aby udowodnić to twierdzenie wykorzystamy numerację Gödla: niech $\mathrm{nr}(M)$ oznacza numer termu M w pewnej ustalonej numeracji. Jeśli $\mathrm{nr}(M) = n$, to napis $\underline{M}$ oznaczać będzie liczebnik Churcha $\mathbf{n}$. Mamy rekurencyjne funkcje *app* i *num*, takie że:

$$app(\mathrm{nr}(M))(\mathrm{nr}(N)) = \mathrm{nr}(MN)$$
$$num(n) = \mathrm{nr}(\mathbf{n}).$$

Z twierdzenia 7.9 wynika, że są to funkcje λ-definiowalne: są takie termy *App* i *Num*, że:

$$App\,\underline{M}\,\underline{N} = \underline{MN} \qquad \text{and} \qquad Num\,\mathbf{n} = \underline{\mathbf{n}}.$$

Zauważmy, że $Num\,\underline{M} = \underline{\underline{M}}$. Jest to liczebnik Churcha odpowiadający numerowi liczebnika odpowiadającego numerowi termu M, czyli niejako „numer numeru termu M". Zachodzi następujące twierdzenie o punkcie stałym:

Twierdzenie 7.12 *Dla dowolnego termu F istnieje term X o własności $F(\underline{X}) =_\beta X$.*

Dowód: Niech $X = Z\underline{Z}$, gdzie $Z = \lambda x.\, F(App\,x(Num\,x))$. Wtedy

$$X = Z\underline{Z} =_\beta F(App\,\underline{Z}\,(Num\,\underline{Z})) =_\beta F(App\,\underline{Z}\,\underline{\underline{Z}}) =_\beta F(\underline{Z\underline{Z}}) =_\beta F(\underline{X}).$$ ∎

Twierdzenie 7.13 (D. Scott) *Niech $\mathcal{A}$ będzie zbiorem termów, który jest:*

- *nietrywialny, tj. niepusty i nie pełny;*
- *zamknięty ze względu na beta-konwersję, tj. $M =_\beta N \in \mathcal{A}$ implikuje $M \in \mathcal{A}$.*

Wówczas $\mathcal{A}$ jest nierozstrzygalny.

Dowód: Załóżmy, że $\mathcal{A}$ jest nietrywialny i rozstrzygalny. Rozstrzygalność oznacza, że funkcja charakterystyczna ch_A zbioru $A = \{\mathrm{nr}(M) \mid M \in \mathcal{A}\}$ jest rekurencyjna, a zatem definiowalna. Istnieje więc taki term F, że

$$F\underline{M} =_\beta \begin{cases} \mathbf{0}, & \text{jeśli } M \in A; \\ \mathbf{1}, & \text{jeśli } M \notin A. \end{cases}$$

Ustalmy $M_1 \in A$, $M_2 \notin A$ i niech:

$$G = \lambda x.\, \mathbf{if}\ \mathbf{zero}(Fx)\ \mathbf{then}\ M_2\ \mathbf{else}\ M_1.$$

Z twierdzenia 7.12 wynika, że $G(\underline{N}) =_\beta N$, dla pewnego N. Wtedy jednak:

- Jeśli $N \in A$, to $N = G(\underline{N}) =_\beta M_2 \notin A$.
- Jeśli $N \notin A$, to $N = G(\underline{N}) =_\beta M_1 \in A$.

A zatem zbiór $\mathcal{A}$ nie może być zamknięty ze względu na beta-konwersję. ∎

Uwaga: Warunek zamkniętości ze względu na beta-konwersję nie jest spełniony przez wiele istotnych zbiorów termów (problemów decyzyjnych). Dotyczy to na przykład pierwszej i ostatniej części wniosku 7.11, a także zbiorów termów typowalnych w rachunkach z typami.

Ćwiczenia

1. Niech $G = \mathbf{SI} =_\beta \lambda yf.f(yf)$. Term jest kombinatorem punktu stałego wtedy i tylko wtedy, gdy jest punktem stałym G.
2. Zdefiniować w rachunku lambda poprzednik dla liczb naturalnych.
3. Pokazać, że w definicji funkcji definiowalnej można żądać $F\mathbf{n}_1 \dots \mathbf{n}_k \twoheadrightarrow_\beta \mathbf{f}(\mathbf{n_1}, \dots, \mathbf{n_k})$, i że nic się nie zmieni, jeśli dopuścić $\mathrm{FV}(F) \neq \varnothing$.
4. Nic się też nie zmieni, jeśli w definicji funkcji λ-definiowalnej zamiast $=_\beta$ użyjemy $=_{\beta\eta}$.

8 Lambda-teorie i drzewa Böhma

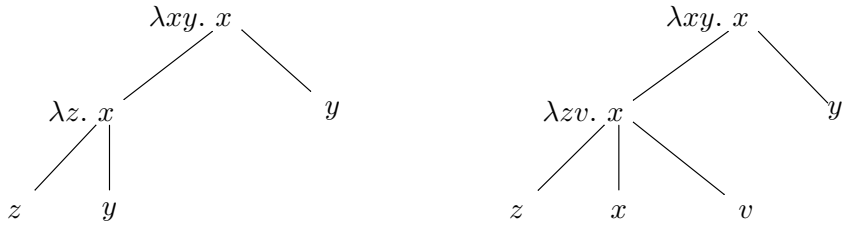
Jak już mówiliśmy, zwykły rachunek lambda można uważać (w uproszczeniu) za pewną teorię równościową. Inną teorię otrzymamy, jeśli do aksjomatów beta-konwersji dołożymy ekstensjonalość w postaci reguły (ext) lub aksjomatu (η). A co jeśli dodać jako nowy aksjomat na przykład równość $\mathbf{K} = \mathbf{S}$? Wtedy dla dowolnego termu M wywnioskujemy

$$M = \mathbf{SI}(\mathbf{K}M)\mathbf{I} = \mathbf{KI}(\mathbf{K}M)\mathbf{I} = \mathbf{I},$$

a wiec takie rozszerzenie rachunku lambda byłoby sprzeczne. Termy $\mathbf{K}$ i $\mathbf{S}$ to tylko przykład. Okazuje się, że sprzeczna musi być każda teoria, która skleja ze sobą dwie różne postaci $\beta\eta$-normalne. Wynika to z następującego twierdzenia Böhma:

Twierdzenie 8.1 (C. Böhm, 1968) *Niech M i N będą kombinatorami w postaci β-normalnej i niech $M \neq_{\beta\eta} N$. Wtedy istnieje taki ciąg zamkniętych termów $\vec{P}$, że $M\vec{P} =_\beta \mathbf{true}$ oraz $N\vec{P} =_\beta \mathbf{false}$.*

Zobaczmy na przykładzie jak można rozróżnić dwa termy. Niech $M = \lambda xy.x(\lambda z.xzy)y$ i $N = \lambda xy.x(\lambda zv.xzxv)y$. Wygodnie jest przedstawić te termy za pomocą drzew (Obrazek 10).



Obrazek 10: Drzewa dla $M = \lambda xy.x(\lambda z.xzy)y$ i $N = \lambda xy.x(\lambda zv.xzxv)y$

Istotna różnica pomiędzy tymi drzewami to liście odpowiadające zmiennej y w termie M i zmiennej x w termie N. (Wystąpienie v w termie N jest nieistotne, bo $\lambda zv.xzxv =_\eta \lambda z.xzx$.) Aby wykorzystać tę różnicę do „odseparowania" termów M i N należy zaaplikować je do takich argumentów, które wyciągną z nich „na wierzch" właśnie te liście. Posłużymy się trickiem, zwanym „*Böhm-out technique*". Niech $P = \lambda uw.\langle u, w\rangle$. Wtedy dla dowolnego Q zachodzi $MPQ =_\beta \langle\lambda z.\langle z, Q\rangle, Q\rangle$. Dalej $MPQ\,\mathbf{true}\,R\,\mathbf{false} =_\beta Q$ dla dowolnego R. Tymczasem $NPQ\,\mathbf{true}\,R\,\mathbf{false} =_\beta P$. Wybierzmy więc jakiekolwiek R oraz $Q = \lambda uvw.\,\mathbf{true}$ i niech $\vec{P}$ oznacza ciąg $(P, Q, \mathbf{true}, R, \mathbf{false}, \mathbf{false}, \mathbf{I}, \mathbf{true})$. Wtedy $M\vec{P} =_\beta \mathbf{true}$ oraz $N\vec{P} =_\beta \mathbf{false}$.

Dowód twierdzenia Böhma

Przypomnijmy, że $\mathbf{true} = \lambda xy.\,x$, $\mathbf{false} = \lambda xy.\,y$ oraz $\pi_i^k = \lambda x_1 \ldots x_k.\,x_i$ dla $i \leq k$. Uogólnieniem pary uporządkowanej $\langle M, N\rangle = \lambda f.\,fMN$ jest *p-krotka*:

$$\langle M_1, \ldots, M_p\rangle = \lambda f.\,fM_1 \ldots M_p.$$

Przez T_p oznaczymy operator tworzenia krotki, tj. $T_p = \lambda x_1 \ldots x_p \lambda f.\,fx_1 \ldots x_p$.

Podstawienie postaci $S = [x_i := T_{p_i}]_{i=1,\ldots,n}$ nazwiemy m-podstawieniem, gdy liczby $p_1, \ldots p_n$ są parami różne i większe od m. Termy M i N są *m-rozróżnialne* gdy dla dowolnego m-podstawienia S o dziedzinie $\mathrm{FV}(M) \cup \mathrm{FV}(N)$ istnieje taki ciąg kombinatorów $\vec{L}$, że:

$$M[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true} \qquad \text{oraz} \qquad N[S]\vec{L} \twoheadrightarrow_\beta \mathbf{false}.$$

Termy *rozróżnialne*, to takie, które są m-rozróżnialne dla pewnego m. Zauważmy, że rozróżnialność jest relacją symetryczną. Jeśli bowiem M i N są m-rozróżnialne, to także N i M są m-rozróżnialne, bo $\mathbf{true}\,\mathbf{false}\,\mathbf{true} \twoheadrightarrow_\beta \mathbf{false}$ oraz $\mathbf{false}\,\mathbf{false}\,\mathbf{true} \twoheadrightarrow_\beta \mathbf{true}$.

Twierdzenie Böhma udowodnimy pokazując, że eta-różne postaci normalne muszą być rozróżnialne (lemat 8.6).

Lemat 8.2 *Jeśli termy M i N są rozróżnialne, to $\lambda x\,M$ i $\lambda x\,N$ też są rozróżnialne.*

Dowód: Załóżmy, że M i N są m-rozróżnialne. Weźmy m-podstawienie S i niech $p > m$ będzie nowe (takie, że T_p nie występuje w S). Wtedy $S' = S[x := T_p]$ jest m-podstawieniem, więc $M[S']\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ oraz $N[S']\vec{L} \twoheadrightarrow_\beta \mathbf{false}$ dla odpowiedniego ciągu kombinatorów $\vec{L}$.

Zatem $(\lambda x\, M)[S]T_p\vec{L} \to_\beta M[S][x := T_p]\vec{L} = M[S']\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ i podobnie term $(\lambda x\, N)[S]T_p\vec{L}$ redukuje się do **false**. ∎

Lemat 8.3 *Jeśli $x \notin \mathrm{FV}(N)$ i termy M i Nx są rozróżnialne, to termy $\lambda x\, M$ i N też.*

Dowód: Załóżmy, że M i Nx są m-rozróżnialne. Niech S będzie m-podstawieniem i niech $S(x) = T_p$. Istnieje taki ciąg $\vec{L}$, że $M[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ oraz $N[S]T_p\vec{L} \twoheadrightarrow_\beta \mathbf{false}$. Ponieważ x jest związane w termie $\lambda x\, M$, więc $(\lambda x\, M)[S] = \lambda x\, M[S']$ gdzie S' to S ograniczone do zmiennych różnych od x. Zatem $(\lambda x\, M)[S]T_p\vec{L} = (\lambda x\, M[S'])T_p\vec{L} \to_\beta M[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ i już dobrze. ∎

Lemat 8.4 *Jeśli $x \neq y$ i termy $M = xP_1 \dots P_k$ i $N = yQ_1 \dots Q_\ell$ są w postaci normalnej, to M i N są rozróżnialne.*

Dowód: Wybierzmy $m \geq k, \ell$ i rozpatrzmy dowolne m-podstawienie S. Niech $S(x) = T_p$ i $S(y) = T_q$. Wtedy $M[S] = T_pP_1[S] \dots P_k[S] \twoheadrightarrow_\beta \lambda\vec{u}f.\, fP_1[S] \dots P_k[S]\vec{u}$, gdzie wektor zmiennych $\vec{u}$ jest długości $p - k$. Podobnie $N[S] = T_qQ_1[S] \dots Q_\ell[S] \twoheadrightarrow_\beta \lambda\vec{v}g.\, gQ_1[S] \dots Q_\ell[S]\vec{v}$, gdzie wektor $\vec{v}$ ma długość $q - \ell$.

Załóżmy najpierw, że wektory $\vec{u}$ i $\vec{v}$ są różnej długości, np. $p - k < q - \ell$. Wtedy możemy bez straty ogólności założyć, że $\vec{v} = \vec{u}f\vec{w}$, czyli że $N[S] \twoheadrightarrow_\beta \lambda\vec{u}f\vec{w}\, g.\, gQ_1[S] \dots Q_\ell[S]\vec{u}\, f\vec{w}$. Jeśli teraz $\vec{L} = L_1 \dots L_{q-\ell+1}$ jest dowolnym ciągiem kombinatorów długości $q - \ell + 1$, to:

$$M[S]\vec{L} \twoheadrightarrow_\beta L_{p-k+1}P_1[S] \dots P_k[S]L_1 \dots L_{p-k}L_{p-k+2} \dots L_{q-\ell+1};$$

$$N[S]\vec{L} \twoheadrightarrow_\beta L_{q-\ell+1}Q_1[S] \dots Q_\ell[S]L_1 \dots L_{q-\ell}.$$

Przy tym $p - k + 1 \neq q - \ell + 1$, więc wystarczy wziąć $L_{p-k+1} = \lambda y_1 \dots y_{q-l+k}.\, \mathbf{true}$ oraz $L_{q-\ell+1} = \lambda z_1 \dots z_q.\, \mathbf{false}$ i dostaniemy $M[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ i $N[S]\vec{L} \twoheadrightarrow_\beta \mathbf{false}$. Pozostałe termy L_i mogą być jakiekolwiek, np. $L_i = \mathbf{I}$ dla $i \neq p - k + 1, q - \ell + 1$.

Załóżmy teraz, że $p - k = q - \ell$. Możemy wtedy przyjąć, że $N[S] \twoheadrightarrow_\beta \lambda\vec{u}f.\, fQ_1[S] \dots Q_\ell[S]\vec{u}$. Ponieważ $p \neq q$, więc $k \neq \ell$; przypuśćmy, że $k > \ell$ i $p > q$. Niech $\vec{L} = L_1 \dots L_{p-k}$, gdzie $L_1 = \lambda z_1 \dots z_{k-\ell}.\, \mathbf{true}$, oraz $L_{k-\ell+1} = \mathbf{false}$. (Uwaga: $k - \ell + 1 \leq p - k = q - \ell$, bo $k < q$.) Pozostałe termy L_i są nieistotne. Aplikując $M[S]$ i $N[S]$ do argumentów $\vec{L}\pi^p_{k+1}$, dostajemy:

$M[S]\vec{L}\pi^p_{k+1} \twoheadrightarrow_\beta \pi^p_{k+1}P_1[S] \dots P_k[S]\vec{L} \twoheadrightarrow_\beta L_1$, ponieważ ciąg $P_1[S] \dots P_k[S]\vec{L}$ ma długość p.

$N[S]\vec{L}\pi^p_{k+1} \twoheadrightarrow_\beta \pi^p_{k+1}Q_1[S] \dots Q_\ell[S]\vec{L} \twoheadrightarrow_\beta \lambda\vec{w}.\, L_{k-\ell+1}$, bo teraz ciąg argumentów jest krótszy.

Wektor zmiennych $\vec{w}$ jest długości $k - \ell$. Dlatego nasze termy zaaplikujemy jeszcze do $k - \ell$ egzemplarzy kombinatora $\mathbf{I}$ i dostaniemy:

$$M[S]\vec{L}\pi^p_{\ell+1}\mathbf{I} \dots \mathbf{I} \twoheadrightarrow_\beta L_1\mathbf{I} \dots \mathbf{I} \twoheadrightarrow_\beta \mathbf{true} \quad \text{oraz} \quad N[S]\vec{L}\pi^p_{\ell+1}\mathbf{I} \dots \mathbf{I} \twoheadrightarrow_\beta L_{k-\ell+1} = \mathbf{false}.$$ ∎

Lemat 8.5 *Jeśli $k \neq \ell$ i termy $M = xP_1 \dots P_k$ i $N = xQ_1 \dots Q_\ell$ są w postaci normalnej, to M i N są rozróżnialne.*

Dowód: Załóżmy, że $k < \ell$. Wybierzmy $m \geq k, \ell$ i niech S będzie m-podstawieniem. Wtedy $S(x) = T_p$, przy czym $p > k, \ell$. Zatem

$$M[S] \twoheadrightarrow_\beta \lambda\vec{u}\, f.\, fP_1[S] \ldots P_k[S]\vec{u} \qquad \text{oraz} \qquad N[S] \twoheadrightarrow_\beta \lambda\vec{v}\, g.\, gQ_1[S] \ldots Q_\ell[S]\vec{v},$$

gdzie wektory $\vec{u}$ i $\vec{v}$ są odpowiednio długości $p-k$ i $p-\ell$. Wektor $\vec{u}$ jest dłuższy i możemy przyjąć $\vec{u} = \vec{v}\, g\, \vec{w}$. A więc $M[S] \twoheadrightarrow_\beta \lambda\vec{v}\, g\, \vec{w}\, f.\, fP_1[S] \ldots P_k[S]\vec{v}\, g\, \vec{w}$. Niech $\vec{L} = L_1 \ldots L_{p-k+1}$, gdzie $L_{p-k+1} = \lambda y_1 \ldots y_p.\, \mathbf{true}$, $L_{p-\ell+1} = \lambda z_1 \ldots z_{p-k+\ell}.\, \mathbf{false}$, oraz $L_j = \mathbf{I}$ w przeciwnym przypadku. Wtedy

$$M[S]\vec{L} \twoheadrightarrow_\beta L_{p-k+1}P_1[S] \ldots P_k[S]\mathbf{I} \ldots \mathbf{I}L_{p-\ell+1}\mathbf{I} \ldots \mathbf{I} \twoheadrightarrow_\beta \mathbf{true},$$

$$N[S]\vec{L} \twoheadrightarrow_\beta L_{p-\ell+1}Q_1[S] \ldots Q_\ell[S]\mathbf{I} \ldots \mathbf{I}L_{p-k+1} \twoheadrightarrow_\beta \mathbf{false}.$$ ∎

Twierdzenie Böhma wynika bezpośrednio z następnego lematu.

Lemat 8.6 *Jeśli M i N są postaciami β-normalnymi i $M \neq_\eta N$, to M i N są rozróżnialne.*

Dowód: Indukcja ze względu na sumę długości termów M i N, liczoną bez nawiasów i kropek, ale za to z lambdami, np. term $\lambda xy.\, x$ ma długość 4, a term $x(yz)$ ma długość 3.

Jeśli oba termy M i N są abstrakcjami, to teza wynika z założenia indukcyjnego i lematu 8.2. Jeśli $M = \lambda x\, M'$ jest abstrakcją, ale N nie, to weźmy $x \notin \mathrm{FV}(M) \cup \mathrm{FV}(N)$ i rozpatrzmy termy $M = \lambda x\, M'$ i $N' = Nx$. Wtedy $M' \neq_\eta Nx$, a ponieważ suma długości termów M' i Nx jest mniejsza o jeden od sumy długości termów M i N (z powodu usunięcia lambdy), więc możemy zastosować założenie indukcyjne do M' i Nx, a następnie odwołać się do lematu 8.3.

Załóżmy więc, że ani M ani N nie jest abstrakcją. Jeśli termy M i N mają inne zmienne czołowe, to stosuje się lemat 8.4, jeśli zmienna czołowa jest ta sama, ale liczba argumentów jest inna, to użyjemy lematu 8.5. Pozostaje przypadek, gdy jedno i drugie jest takie samo. A więc $M = xP_1 \ldots P_k$ i $N = xQ_1 \ldots Q_k$. Skoro $M \neq_\eta N$, to jest takie $i \leq k$, że $P_i \neq_\eta Q_i$. Z założenia indukcyjnego jest takie m, że termy P_i i Q_i są m-rozróżnialne, a więc także m'-rozróżnialne gdy $m' \geq m$. Weźmy $m' \geq m, k$ i dowolne m'-podstawienie S, takie że $S(x) = T_p$ jest określone. Wtedy $P_i[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true}$ i $Q_i[S]\vec{L} \twoheadrightarrow_\beta \mathbf{false}$, dla pewnego $\vec{L}$. Ponadto:

$$M[S] = T_pP_1[S] \ldots P_k[S] \twoheadrightarrow_\beta \lambda\vec{u}f.\, fP_1[S] \ldots P_k[S]\vec{u};$$

$$N[S] = T_pQ_1[S] \ldots Q_k[S] \twoheadrightarrow_\beta \lambda\vec{u}f.\, fQ_1[S] \ldots Q_k[S]\vec{u},$$

gdzie wektor $\vec{u}$ jest długości $p-k$. Stąd

$$M[S]\mathbf{I} \ldots \mathbf{I}\pi_i^p\vec{L} \twoheadrightarrow_\beta \pi_i^pP_1[S] \ldots P_k[S]\mathbf{I} \ldots \mathbf{I}\vec{L} \twoheadrightarrow_\beta P_i[S]\vec{L} \twoheadrightarrow_\beta \mathbf{true}$$

i podobnie $N[S]\mathbf{I} \ldots \mathbf{I}\pi_i^p\vec{L} \twoheadrightarrow_\beta \mathbf{false}$. ∎

Rozwiązalność

Pierwsza intuicja związana z rachunkiem lambda (rozumianym jako abstrakcyjny język programowania) jest taka: „wartością" termu jest jego postać normalna. A zatem term, który nie ma postaci normalnej jest pozbawiony wartości. Każdy taki term jest równie dobry, albo

raczej równie zły, i nie powinno być przeszkód w utożsamieniu ich wszystkich ze sobą. Niestety, utożsamienie na przykład termów $\lambda x.x\mathbf{K}\Omega$ i $\lambda x.x\mathbf{S}\Omega$ daje w wyniku sprzeczną teorię — wystarczy oba zaaplikować do $\mathbf{K}$. Dostajemy konkretne i różne rezultaty: $\mathbf{K}$ i $\mathbf{S}$.

Term bez postaci normalnej może więc czasem objawiać dobrze określone działanie, czyli w pewnym sensie mieć „wartość". Można uważać, że tą wartością jest czołowa postać normalna, o ile istnieje. Przypomnijmy, że czołowa postać normalna to każdy term postaci $\lambda\vec{x}.z\vec{R}$. Term M *ma czołową postać normalną* N gdy $M =_\beta N$ i N jest w czołowej postaci normalnej.

Lemat 8.7 *Jeśli term MN ma czołową postać normalną to M też.*

Dowód: Jeśli term M nie ma czołowej postaci normalnej, to ciąg redukcji czołowych $M = M_0 \stackrel{\mathrm{h}}{\to} M_1 \stackrel{\mathrm{h}}{\to} \cdots$ jest nieskończony. Jeśli żadne M_i nie jest abstrakcją, to mamy też nieskończoną redukcję $MN = M_0N \stackrel{\mathrm{h}}{\to} M_1N \stackrel{\mathrm{h}}{\to} \cdots$ Niech więc M_i będzie pierwszą abstrakcją, tj. $M_i = \lambda u.N_i \stackrel{\mathrm{h}}{\to} \lambda u.N_{i+1} \stackrel{\mathrm{h}}{\to} \lambda u.N_{i+2} \stackrel{\mathrm{h}}{\to} \cdots$ gdzie $N_i \stackrel{\mathrm{h}}{\to} N_{i+1} \stackrel{\mathrm{h}}{\to} N_{i+2}\cdots$ Wtedy $MN \stackrel{\mathrm{h}}{\twoheadrightarrow} M_iN \stackrel{\mathrm{h}}{\to} N_i[u := N] \stackrel{\mathrm{h}}{\to} N_{i+1}[u := N] \stackrel{\mathrm{h}}{\to} \cdots$, bo $N_i \stackrel{\mathrm{h}}{\to} N_{i+1}$ implikuje $N_i[u := N] \stackrel{\mathrm{h}}{\to} N_{i+1}[u := N]$. ∎

Powiemy, że term zamknięty M jest *rozwiązalny* jeżeli $M\vec{P} =_\beta \mathbf{I}$ dla pewnych kombinatorów $\vec{P}$. Term ze zmiennymi wolnymi $\vec{x}$ jest *rozwiązalny* jeżeli rozwiązalne jest jego *domknięcie*, czyli kombinator $\lambda\vec{x}.M$

Wniosek 8.8 *Term jest rozwiązalny wtedy i tylko wtedy, gdy ma czołową postać normalną.*

Dowód: Bez straty ogólności można założyć, że mowa o termie zamkniętym.

Jeśli $M =_\beta \lambda x_1x_2\ldots x_n.x_iR_1\ldots R_m$, to oczywiście $MP_1P_2\ldots P_n =_\beta \mathbf{I}$, gdzie $P_i = \lambda y_1\ldots y_m.\mathbf{I}$.

Na odwrót, jeżeli $M\vec{P} =_\beta \mathbf{I}$, to w istocie $M\vec{P} \twoheadrightarrow_\beta \mathbf{I}$. Skoro więc $M\vec{P}$ ma czołową postać normalną $\mathbf{I}$, to i M musi mieć czołową postać normalną (lemat 8.7 plus indukcja). ∎

Jeśli termy nierozwiązalne (niezdolne do dobrze określonego zachowania) uznamy za pozbawione wartości, to znaczy, że postulujemy teorię równościową, w której te wszystkie termy są równe. Jakie rozwiązalne termy powinny być równe w tej teorii? Te, których zachowanie jest jednakowe. Zdolność do sensownego zachowania przejawia się istnieniem czołowej postaci normalnej, ale nie możemy identyfikować termów za pomocą ich czołowych postaci normalnych, bo term rozwiązalny może ich mieć nawet nieskończenie wiele. Nie możemy się też posłużyć beta-równością, bo chcemy aby $x\Omega$ było w naszej teorii równe $x(\Omega y)$. Pozostaje *obserwować* zachowanie termów w różnych sytuacjach. Jeśli jest ono zawsze takie samo, to powinniśmy dane termy utożsamić.

Niech M i N będą dowolnymi termami i niech $\mathrm{FV}(M) \cup \mathrm{FV}(N) = \vec{x}$. Powiemy, że M i N są *obserwacyjnie równoważne*, i napiszemy $M \equiv N$, gdy dla dowolnego kombinatora P,

$$P(\lambda\vec{x}.M) \text{ jest rozwiązalny} \iff P(\lambda\vec{x}.N) \text{ jest rozwiązalny.}$$

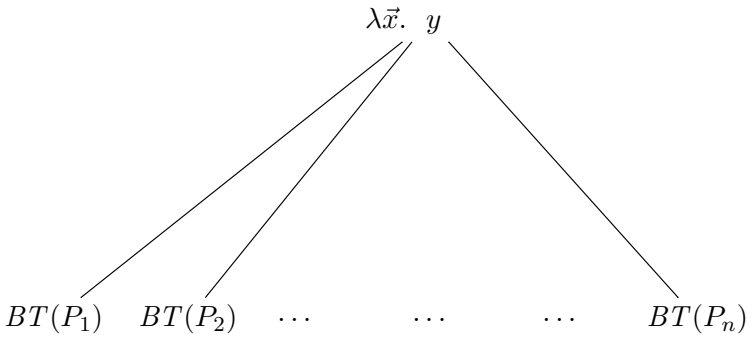
Nietrudno zauważyć, że $\equiv$ jest rozszerzeniem ekstensjonalnej teorii równościowej $\lambda\beta\eta$, i że $M \equiv N$ dla dowolnych nierozwiązalnych M i N (ćwiczenie 3). Ponadto, z twierdzenia Böhma wynika, że dla termów w postaci normalnej zachodzi równoważność

$$M \equiv N \qquad \Longleftrightarrow \qquad M =_{\beta\eta} N.$$

Drzewa Böhma

Drzewo Böhma termu M, oznaczane przez $BT(M)$, definiujemy przez ko-indukcję.

- Jeśli term M jest nierozwiązalny, to $BT(M)$ składa się tylko z jednego wierzchołka, oznaczonego przez $\mathbf{\Omega}$.
- Jeśli term M ma czołową postać normalną $\lambda\vec{x}.yP_1 \dots P_n$, to drzewo $BT(M)$ jest zbudowane jak na Obrazku 11.



Obrazek 11: Drzewo $BT(\lambda\vec{x}.yP_1 \dots P_n)$

Drzewo Böhma to uogólnienie postaci normalnej. Zwykła postać normalna to skończone drzewo Böhma bez wystąpień $\mathbf{\Omega}$.

Chcemy teraz uogólnić pojęcie eta-redukcji i eta-równości na drzewa Böhma. Oczywiście $B \to_\eta B'$ oznacza jeden krok redukcji. Jednak definicja eta-równości powinna uwzględniać to, że nieskończone drzewa mogą mieć nieskończenie wiele eta-redeksów. Musimy więc pozwolić na to, że eta-równoważne drzewa różnią się w nieskończenie wielu miejscach.

Oczywiście drzewo Böhma można uważać za „granicę" ciągu termów skończonych, w których może występować stała $\mathbf{\Omega}$. Ściślej, powiemy, że ciąg drzew T_n jest *zbieżny* do drzewa B jeżeli dla dowolnego n istnieje takie m, że wszystkie drzewa $T_m, T_{m+1}, \dots$ są do głębokości n identyczne z drzewem B. Ciąg skończony jest zaś zbieżny do swojego ostatniego elementu.

Powiemy, że drzewa Böhma B i B' są *η-równoważne* ($B \approx_\eta B'$), gdy istnieją (skończone lub nieskończone) ciągi *eta-ekspansji:*

$$B = B_0 \;{}_\eta\!\leftarrow B_1 \;{}_\eta\!\leftarrow B_2 \;{}_\eta\!\leftarrow B_3 \;{}_\eta\!\leftarrow \cdots \qquad \text{oraz} \qquad B' = B'_0 \;{}_\eta\!\leftarrow B'_1 \;{}_\eta\!\leftarrow B'_2 \;{}_\eta\!\leftarrow B'_3 \;{}_\eta\!\leftarrow \cdots$$

zbieżne do tego samego drzewa. Przebrnąwszy przez tę ostatnią definicję, możemy sformułować twierdzenie Wadswortha:

Twierdzenie 8.9 (Wadsworth, 1975) *Termy M i N są obserwacyjnie równoważne wtedy i tylko wtedy, gdy $BT(M) \approx_\eta BT(N)$.*

Dowód z lewej do prawej polega na zastosowaniu techniki „*Böhm-out*". W przeciwną stronę stosuje się metody semantyczne.

Ćwiczenia

1. Skonstruować takie $\mathbf{X}$, że $\mathbf{X}(\lambda yz.y(yz(yy))z) \twoheadrightarrow_{\beta\eta} \mathbf{K}$ oraz $\mathbf{X}(\lambda yz.y(yz(yz))z) \twoheadrightarrow_{\beta\eta} \mathbf{S}$.
2. Udowodnić, ze jeśli M nie ma czołowej postaci normalnej, to żaden term postaci $M[x := P]$ nie ma czołowej postaci normalnej. *Wskazówka:* Jeśli $M \xrightarrow{\text{h}} N$, to $M[x := P] \xrightarrow{\text{h}} N[x := P]$.
3. Udowodnić, ze jeśli M nie ma czołowej postaci normalnej, ale PM ma czołową postać normalną, to każdy term postaci PN ma czołową postać normalną. *Wskazówka:* Jak wygląda czołowa postać normalna termu P?
4. Tradycyjna definicja obserwacyjnej równoważności posługuje się pojęciem *kontekstu* czyli termu z „dziurą". Umieszczenie termu M w kontekscie $C[\ ]$ (czyli włożenie go do „dziury" $[\ ]$) daje w wyniku term $C[M]$, w którym zmienne wolne M mogą zostać związane abstrakcjami w $C[\ ]$. Pokazać, że $M \equiv N$ wtedy i tylko wtedy, gdy dla dowolnego kontekstu $C[\ ]$,
$$C[M] \text{ jest rozwiązalny} \iff C[N] \text{ jest rozwiązalny.}$$
5. Załóżmy, że kombinatory M i N są obserwacyjnie równoważne, i że $PM =_{\beta\eta} \lambda x_1 \ldots x_n.\, x_i A_1 \ldots A_k$, dla pewnych $A_1, \ldots, A_k$. Udowodnić, że $PN =_{\beta\eta} \lambda x_1 \ldots x_n.\, x_i B_1 \ldots B_k$ dla pewnych $B_1, \ldots, B_k$.
6. Znaleźć term, którego drzewo Böhma to jedna nieskończona gałąź postaci $\lambda x_1.x_1(\lambda x_2.x_2(\lambda x_3 \ldots$ *Wskazówka: użyć kombinatora* $\mathbf{Y}$.
7. Skonstruować term M o takiej własności: drzewo Böhma $BT(M)$ jest nieskończone i każdy jego wierzchołek ma więcej synów niż miał jego ojciec.
Rozwiązanie: $M = \mathbf{Y}F\mathbf{1}$, gdzie $F = \lambda V \lambda n \lambda x.n(\lambda y.y(V(\mathbf{succ}\, n))x)$.
8. Czy istnieje taki term M, że drzewo $BT(M)$ ma kształt $\lambda x_0.x_1(\lambda x_2.x_3(\lambda x_4. \ldots$?
9. Niech $\mathbf{J} = \mathbf{Y}(\lambda fxy.\, x(fy))$. Opisać drzewo $BT(\mathbf{J})$. Czy $BT(\mathbf{J}) \approx_\eta BT(\mathbf{I})$?

9 Rachunek kombinatorów

Zdefiniujemy teraz system CL (beztypowego) rachunku kombinatorów, zwany także (z przyczyn historycznych) *logiką kombinatoryczną*, chociaż ta druga nazwa jest nieco myląca. Zbiór termów $\mathcal{C}$ definiujemy jako najmniejszy zbiór o własnościach:

- Zmienne przedmiotowe należą do $\mathcal{C}$;
- Stałe K i S należą do $\mathcal{C}$;
- Jeśli $F, G \in \mathcal{C}$, to $(FG) \in \mathcal{C}$.

Konwencje nawiasowe są takie same jak dla termów rachunku lambda. Podstawienie w CL definiujemy podobnie jak dla rachunku lambda, ale łatwiej, bo nie ma zmiennych związanych.

- $x[x := F] = F$ oraz $y[x := F] = y$ dla $y \neq x$;
- $\mathsf{K}[x := F] = \mathsf{K}$ oraz $\mathsf{S}[x := F] = \mathsf{S}$;

- $GH[x := F] = G[x := F]H[x := F]$.

Relację *słabej redukcji* w zbiorze $\mathcal{C}$ definiujemy jako najmniejszą relację $\rightarrow_w$ o własnościach:

- $\mathsf{K}AB \rightarrow_w A$ dla dowolnych A, B;
- $\mathsf{S}ABC \rightarrow_w AC(BC)$ dla dowolnych A, B, C;
- Jeśli $A \rightarrow_w B$, to $AC \rightarrow_w BC$ oraz $CA \rightarrow_w CB$, dla dowolnych A, B, C.

Jak zwykle, symbol $\twoheadrightarrow_w$ oznacza domknięcie przechodnio-zwrotne relacji $\rightarrow_w$, a $=_w$ (słaba równość), to najmniejsza relacja równoważności zawierająca $\rightarrow_w$. Termy rachunku kombinatorów, które nie zawierają zmiennych, nazywamy oczywiście *kombinatorami*. Zamknięte termy rachunku lambda przejęły tę nazwę od swoich odpowiedników w $\mathcal{C}$. Oto kilka ważnych kombinatorów i związane z nimi redukcje:

- Niech $\mathsf{I} = \mathsf{SKK}$. Wtedy $\mathsf{I}F \rightarrow_w \mathsf{K}F(\mathsf{K}F) \rightarrow_w F$ dla dowolnego F.
- Term $\Omega = \mathsf{SII}(\mathsf{SII})$ redukuje się sam do siebie.
- Niech $\mathsf{W} = \mathsf{SS}(\mathsf{KI})$. Wtedy $\mathsf{W}FG \twoheadrightarrow_w FGG$ dla dowolnych F, G.
- Niech $\mathsf{B} = \mathsf{S}(\mathsf{KS})\mathsf{K}$. Wtedy $\mathsf{B}FGH \twoheadrightarrow_w F(GH)$, dla dowolnych F, G, H.
- Niech $\mathsf{C} = \mathsf{S}(\mathsf{BBS})(\mathsf{KK})$. Wtedy $\mathsf{C}FGH \twoheadrightarrow_w FHG$ dla dowolnych F, G, H.
- Niech $\mathsf{B}' = \mathsf{CB}$. Wtedy $\mathsf{B}'FGH \twoheadrightarrow_w G(FH)$.
- Niech $\mathsf{D} = \mathsf{C}(\mathsf{BC}(\mathsf{B}(\mathsf{CI})\mathsf{K}))$. Wtedy $\mathsf{D}FGH \twoheadrightarrow_w H(\mathsf{K}G)F$.
- Niech $2 = \mathsf{SB}(\mathsf{SB}(\mathsf{KI}))$. Wtedy $2FG \twoheadrightarrow_w F(FG)$.

Uwaga: Termy K, S, KS, SK, I, B, 2, W są w postaci w-normalnej, a termy B', C nie są. Czasem wszystkie te kombinatory są jednak traktowane jak stałe.

System CL, jako system redukcyjny, ma własności podobne do rachunku lambda. Na przykład prawdziwe jest twierdzenie Churcha-Rossera, a takie własności jak normalizacja, silna normalizacja, czy równość, są nierozstrzygalne. System CL ma bowiem podobną siłę wyrazu — można w nim na przykład zdefiniować liczebniki i reprezentować funkcje częściowo rekurencyjne. W istocie, logika kombinatoryczna została wynaleziona przez Curry'ego i Schönfinkela niezależnie od rachunku lambda i mniej więcej w tym samym czasie, ale w gruncie rzeczy w podobnym celu.

Podobieństwo pomiędzy CL i rachunkiem lambda można wyrazić definiując translacje

$$(\ )_\Lambda : \mathcal{C} \rightarrow \Lambda \qquad\qquad (\ )_\mathcal{C} : \Lambda \rightarrow \mathcal{C}$$

Pierwsza z nich jest łatwa:

- $(x)_\Lambda = x$ dla $x \in V$;

- $(\mathsf{K})_\Lambda = \lambda xy.x$;
- $(\mathsf{S})_\Lambda = \lambda xyz.xz(yz)$;
- $(FG)_\Lambda = (F)_\Lambda (G)_\Lambda$.

Następujący łatwy fakt wyraża poprawność tej translacji ze względu na redukcje.

Fakt 9.1

- *Jeśli* $F \twoheadrightarrow_w G$, *to* $(F)_\Lambda \twoheadrightarrow_\beta (G)_\Lambda$.
- *Jeśli* $F =_w G$, *to* $(F)_\Lambda =_\beta (G)_\Lambda$.

Fakt 9.1 można odczytać tak: translację $(\ )_\Lambda$ można uważać za przekształcenie z $\mathcal{C}/_{=_w}$ do $\Lambda/_{=_\beta}$, czyli za morfizm z teorii równościowej CL do teorii równościowej Λ.

Aby określić translację $(\ )_\mathcal{C} : \Lambda \to \mathcal{C}$ musimy przede wszystkim zdefiniować w CL konstrukcję (zwaną *kombinatoryczną abstrakcją*), która zachowuje się tak jak abstrakcja. Można to zrobić na kilka sposobów, na przykład tak:

- $\lambda^* x.F = \mathsf{K}F$, gdy $x \notin \mathrm{FV}(F)$;
- $\lambda^* x.x = \mathsf{I}$;
- $\lambda^* x.FG = \mathsf{S}(\lambda^* x.F)(\lambda^* x.G)$, w przeciwnym przypadku.

Dowód poniższego faktu pozostawiamy jako ćwiczenie.

Fakt 9.2 $(\lambda^* x.F)G \twoheadrightarrow_w F[x := G]$.

Wniosek 9.3 (kombinatoryczna zupełność) *Dla dowolnych* x *i* F *istnieje takie* H, *że dla dowolnego* G *zachodzi* $HG \twoheadrightarrow_w F[x := G]$.

Teraz możemy zdefiniować translację $(\ )_\mathcal{C} : \Lambda \to \mathcal{C}$.

- $(x)_\mathcal{C} = x$ dla $x \in V$;
- $(MN)_\mathcal{C} = (M)_\mathcal{C}(N)_\mathcal{C}$;
- $(\lambda x.M)_\mathcal{C} = \lambda^* x.(M)_\mathcal{C}$.

Fakt 9.4 *Dla dowolnego* $M \in \Lambda$ *zachodzi* $((M)_\mathcal{C})_\Lambda =_\beta M$.

Dowód: Najpierw należy pokazać $(\lambda^* x.F)_\Lambda =_\beta \lambda x\,(F)_\Lambda$, przez indukcję ze względu na F. Potem też indukcja ze względu na M. ∎

A zatem translacja $(\ )_\Lambda$ wyznacza przekształcenie „na” z ilorazu $\mathcal{C}/_{=_w}$ do $\Lambda/_{=_\beta}$. Poniższy wniosek stwierdza, że termy **K** i **S** stanowią *bazę* dla rachunku lambda.

Wniosek 9.5 *Każdy zamknięty lambda-term można otrzymać (z dokładnością do $=_\beta$) z termów* **K** *i* **S** *przez stosowanie aplikacji.*

Dowód: Natychmiast z Faktu 9.4. ∎

Niestety, operacja $(\ )_\mathcal{C}$ nie ma tak dobrych własności jak translacja $(\ )_\Lambda$. Nie można jej uważać za morfizm teorii równościowych.

Fakt 9.6

- *Złożenie $((\ )_\Lambda)_\mathcal{C}$ nie jest identycznością. Na przykład $((\mathsf{K}_\Lambda)_\mathcal{C} = \mathsf{S(KK)I} \neq_w \mathsf{K}$.*
- *Z równości $M =_\beta N$ nie wynika $(M)_C =_w (N)_C$. Na przykład $\lambda x.\mathbf{KI}x \to_\beta \lambda x\,\mathbf{I}$, ale $(\lambda x.\mathbf{KI}x)_\mathcal{C} = \mathsf{S(K(S(KK)II))I} =_w \mathsf{S(K(KI))I} \neq_w \mathsf{KI} = (\lambda x\,\mathbf{I})_\mathcal{C}$.*

A zatem „słaba" równość jest w istocie *silniejsza* niż $=_\beta$. Przyczyną tego jest to, że kombinatoryczna abstrakcja λ^* nie spełnia warunku słabej ekstensjonalności ze względu na $=_w$:

$$\frac{M = N}{\lambda x.M = \lambda x.N,} \qquad (\xi)$$

W istocie

$$\lambda^* x.\mathsf{KI}x = \mathsf{S(K(KI))I} \neq_w \mathsf{KI} = \lambda^* x.\mathsf{I},$$

chociaż $\mathsf{KI}x \to_w \mathsf{I}$. Można to poprawić kosztem wprowadzenia ekstensjonalności. Niech $=_{ext}$ bedzie najmniejszą relacją równoważności w $\mathcal{C}$, taką że:

- Jeśli $G =_w H$, to $G =_{ext} H$;
- Jeśli $Gx =_{ext} Hx$ i $x \notin \mathrm{FV}(G) \cup \mathrm{FV}(H)$, to $G =_{ext} H$;
- Jeśli $G =_{ext} G'$, to $GH =_{ext} G'H$ i $HG =_{ext} HG'$.

Fakt 9.7

- *Dla dowolnych $G, H \in \mathcal{C}$, warunki $G =_{ext} H$ i $(G)_\Lambda =_{\beta\eta} (H)_\Lambda$ są równoważne.*
- *Dla dowolnych $M, N \in \Lambda$, warunki $M =_{\beta\eta} N$ i $(M)_\mathcal{C} =_{ext} (N)_\mathcal{C}$ są równoważne.*
- *Równanie $((G)_\Lambda)_\mathcal{C} =_{ext} G$ zachodzi dla dowolnego $G \in \mathcal{C}$.*

Paradoks Curry'ego

Skoro lambda-abstrakcja może być zdefiniowana w rachunku CL, to siła wyrazu rachunku kombinatorów jest podobna do siły wyrazu rachunku lambda. Możemy na przykład zdefiniować kombinatory punktu stałego:

$$\mathsf{Y} = \mathsf{WS(BWB)} \qquad\qquad \Theta = \mathsf{WI(B(SI)(WI))}.$$

Dlatego rachunek kombinatorów jest zbyt silny na to, aby mógł być prawdziwą *logiką kombinatoryczną*. Zobaczmy jak dodanie logiki do systemu CL może prowadzić do sprzeczności.

Zdefiniujemy system NCL „naiwnej logiki kombinatorycznej", dodając do składni nową stałą P, która ma przedstawiać implikację. Zamiast $\mathsf{P}FG$ będziemy pisać $F \Rightarrow G$. W ten sposób dowolną zdaniową formułę implikacyjną można uważać za term rachunku kombinatorów. Aby można było udowodnić taką formułę, reguły wnioskowania systemu NCL pozwalają wyprowadzać nie tylko równości między termami, ale też same termy. System ma trzy schematy aksjomatów:

$$F = F, \qquad F \Rightarrow F, \qquad (F \Rightarrow (F \Rightarrow G)) \Rightarrow (F \Rightarrow G),$$

oraz następujące reguły:

$$\frac{F = G}{FH = GH} \qquad \frac{F = G}{HF = HG} \qquad \frac{F = G}{G = F} \qquad \frac{F = G,\ G = H}{F = H}$$

$$\frac{F, \quad F = G}{G}\ (\text{Conv}) \qquad\qquad \frac{F, \quad F \Rightarrow G}{G}\ (\text{FP})$$

Ostatnia reguła to dobrze znana reguła odrywania. Oprócz niej mamy dwa niewinne aksjomaty rachunku zdań i dosyć oczywiste własności równań, w tym regułę konwersji: zdanie równe zdaniu udowodnionemu należy uważać za udowodnione. Niestety,

Fakt 9.8 (Paradoks Curry'ego) *System* NCL *jest logicznie sprzeczny: każdy term (w tym każda formuła) ma dowód.*

Dowód: Weźmy dowolny term F i niech $N = \mathsf{Y}(\lambda^* x.\, x \Rightarrow F)$. Nietrudno sprawdzić, że równanie $N = N \Rightarrow F$ ma dowód w systemie NCL (term Y to kombinator punktu stałego).

Zatem $N \Rightarrow (N \Rightarrow F) = N \Rightarrow N$ w NCL, a ponieważ $N \Rightarrow N$ jest aksjomatem, więc stosując regułę (Conv) udowodnimy, że $N \Rightarrow (N \Rightarrow F)$. Używając (MP) i trzeciego aksjomatu, wyprowadzimy stąd $N \Rightarrow F$, ale przecież mamy już $N = N \Rightarrow F$, więc udowodniliśmy także N. Ale z $N \Rightarrow F$ i N wynika F przez modus ponens, a F było dowolnym termem. ∎

Ćwiczenia

1. Niech $\mathbf{B} = \lambda xyz.x(yz)$ i niech $\mathbf{C} = \lambda xyz.xzy$. Pokazać, że każdy $\lambda\mathbf{I}$-term można otrzymać (z dokładnością do $=_\beta$) z termów $\mathbf{S}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{I}$, poprzez stosowanie aplikacji. *Wskazówka: zdefiniować konstrukcję $\lambda^\circ x.F$ w ten sposób, że:*
   - $\lambda^\circ x.FG = \mathsf{C}(\lambda^\circ x.F)G$, *gdy* $x \in \mathrm{FV}(F)$ *oraz* $x \notin \mathrm{FV}(G)$;
   - $\lambda^\circ x.FG = \mathsf{B}F(\lambda^\circ x.G)$, *gdy* $x \notin \mathrm{FV}(F)$ *oraz* $x \in \mathrm{FV}(G)$.
2. Term nazwiemy *afinicznym*, gdy każdy jego podterm postaci $\lambda x.M$ zawiera co najwyżej jedno wolne wystąpienie zmiennej x. Pokazać, że każdy afiniczny lambda-term można otrzymać (z dokładnością do $=_\beta$) z termów $\mathbf{B}$, $\mathbf{C}$, $\mathbf{K}$ poprzez stosowanie aplikacji.
3. Term nazwiemy *liniowym*, gdy każdy jego podterm postaci $\lambda x.M$ zawiera dokładnie jedno wolne wystąpienie zmiennej x. Pokazać, że każdy liniowy lambda-term można otrzymać (z dokładnością do $=_\beta$) z termów $\mathbf{B}$, $\mathbf{C}$, $\mathbf{I}$ poprzez stosowanie aplikacji.

Problem otwarty:

Zdefiniować taką (obliczalną) translację $T : \Lambda \to \mathcal{C}$, że

$$M =_\beta N \qquad \Longleftrightarrow \qquad T(M) =_w T(N).$$

10 Modele

Zaczniemy od pewnej ogólnej teorii, potrzebnej dla uściślenia pojęcia modelu dla rachunku lambda. Naszym zamiarem jest interpretacja każdego lambda-termu M w pewnej konkretnej dziedzinie A, jako pewnego elementu $[\![M]\!] \in A$. Ponieważ mamy w termach zmienne wolne, więc nasza interpretacja może zależeć od *wartościowania* $v : V \to A$ (gdzie V oznacza zbiór wszystkich zmiennych), co zaznaczymy pisząc $[\![M]\!]_v$. Indeks $_v$ pomijamy, gdy v jest znane, lub M nie ma zmiennych wolnych. Poniższa definicja formułuje nasze podstawowe oczekiwania w stosunku do takiej interpretacji.

Niech $\cdot$ będzie dwuargumentową operacją w zbiorze A i niech $[\![\ ]\!] : \Lambda \times A^V \to A$. Strukturę $\mathcal{A} = \langle A, \cdot, [\![\ ]\!] \rangle$ nazywamy *lambda-interpretacją*, jeżeli spełnione są warunki (a) – (d) poniżej. Stosujemy notację $[\![M]\!]_v$ na oznaczenie wartości $[\![\ ]\!](M, v)$. Symbol $v[x \mapsto a]$ oznacza wartościowanie różniące się od v tylko tym, że $v[x \mapsto a](x) = a$.

(a) Jeśli x jest zmienną, to $[\![x]\!]_v = v(x)$;

(b) $[\![PQ]\!]_v = [\![P]\!]_v \cdot [\![Q]\!]_v$;

(c) $[\![\lambda x.P]\!]_v \cdot a = [\![P]\!]_{v[x \mapsto a]}$, dla dowolnego $a \in A$;

(d) Jeśli $v|_{\mathrm{FV}(P)} = u|_{\mathrm{FV}(P)}$, to $[\![P]\!]_v = [\![P]\!]_u$.

Na pierwszy rzut oka może się wydawać, że warunki (a) – (d) stanowią indukcyjną *definicję* funkcji $[\![\ ]\!]$. Tak nie jest, bo warunek (c) nie określa *którym elementem* modelu jest $[\![\lambda x.P]\!]_v$ a postuluje tylko jego zachowanie przy aplikacji. (Elementu, który tak się zachowuje, może w ogóle nie być.)

Jeśli $a, b \in A$, to napiszemy $a \approx b$, gdy dla dowolnego $c \in A$ zachodzi równość $a \cdot c = b \cdot c$. Oznacza to, że zachowanie a i b jako funkcji jest takie samo. Interpretacja jest *ekstensjonalna* wtedy i tylko wtedy, gdy z warunku $a \approx b$ wynika $a = b$.

Lambda-interpretację $\mathcal{A} = \langle A, \cdot, [\![\ ]\!] \rangle$ nazywamy

- *lambda-algebrą*, gdy warunek $M =_\beta N$ implikuje $[\![M]\!]_v = [\![N]\!]_v$ dla dowolnego v (co zapiszemy $\mathcal{A} \models M = N$);
- *lambda-modelem*, gdy warunek $[\![\lambda x.M]\!]_v \approx [\![\lambda x.N]\!]_v$ implikuje $[\![\lambda x.M]\!]_v = [\![\lambda x.N]\!]_v$.

A więc lambda-algebra to poprawna interpretacja beta-równości. Natomiast lambda-model to interpretacja *słabo ekstensjonalna*, tj. taka, w której ekstensjonalność zachodzi dla abstrakcji. Aby zrozumieć o co chodzi, zauważmy tu, że $[\![\lambda x.M]\!]_v \approx [\![\lambda x.N]\!]_v$ oznacza tyle samo co $[\![M]\!]_{v[x \mapsto a]} = [\![N]\!]_{v[x \mapsto a]}$ dla wszystkich a. A zatem warunek słabej ekstensjonalności w definicji

lambda-modelu mówi, że znaczenie abstrakcji $[\![\lambda x.M]\!]_v$ jest zdeterminowane przez funkcję, która każdemu a przypisuje element $[\![M]\!]_{v[x\mapsto a]}$. Nieformalnie: wartość $[\![\lambda x.M]\!]$ zależy tylko od wartości $[\![M]\!]$, podobnie jak $[\![MN]\!]$ zależy tylko od $[\![M]\!]$ i $[\![N]\!]$. Oznacza to tyle, że abstrakcja jest operacją semantyczną (ma sens w modelu, niezależny od składni) tak samo jak aplikacja.

Przykład 10.1 Szczególnym przypadkiem interpretacji jest interpretacja zbudowana z samych termów. Niech $\mathfrak{M}(\lambda) = \langle \Lambda/_{=_\beta}, \cdot, [\![\ ]\!] \rangle$, gdzie $[M]_\beta \cdot [N]_\beta = [MN]_\beta$ oraz $[\![M]\!]_v$ jest (z dokładnością do $=_\beta$) wynikiem *jednoczesnego* podstawienia termu $v(x)$ na każdą zmienną wolną x termu M. Przez $\mathfrak{M}^0(\lambda)$ oznaczymy analogiczną interpretację, której dziedzinę tworzą klasy termów zamkniętych. Podobnie $\mathfrak{M}(\lambda\eta)$ i $\mathfrak{M}^0(\lambda\eta)$ oznacza struktury w których zbiór termów (odp. zbiór kombinatorów) podzielono przez $=_{\beta\eta}$. Wszystkie te struktury są lambda-algebrami, ale $\mathfrak{M}^0(\lambda)$ i $\mathfrak{M}^0(\lambda\eta)$ nie są lambda-modelami.

Następujący lemat mówi o własności, której oczekujemy od każdej sensownej semantyki. Znaczenie podstawienia $M[x := N]$ powinno być takie samo jak znaczenie termu M przy odpowiednio zmienionym wartościowaniu.

Lemat 10.2 *W dowolnym lambda-modelu zachodzi tożsamość* $[\![M[x := N]]\!]_v = [\![M]\!]_{v[x\mapsto [\![N]\!]_v]}$.

Dowód: Udowodnimy najpierw, że teza zachodzi przy dodatkowym założeniu $x \notin \mathrm{FV}(N)$. Postępujemy przez indukcję ze względu na M. Przypadki zmiennej i aplikacji są łatwe, niech więc $M = \lambda y\, P$, gdzie $x \neq y \notin \mathrm{FV}(N)$. Chcemy udowodnić równość $[\![\lambda y.\, P[x := N]]\!]_v = [\![\lambda y\, P]\!]_{v[x\mapsto [\![N]\!]_v]}$. Zauważmy, że $[\![\lambda y.\, P[x := N]]\!]_v = [\![\lambda y.\, P[x := N]]\!]_{v[x\mapsto [\![N]\!]_v]}$, bo założyliśmy, że $x \notin \mathrm{FV}(N)$, zatem naszą tezę możemy napisać w postaci $[\![\lambda y.\, P[x := N]]\!]_{v[x\mapsto [\![N]\!]_v]} = [\![\lambda y.\, P[x := N]]\!]_{v[x\mapsto [\![N]\!]_v]}$, gdzie po obu stronach równania występuje to samo wartościowanie.

Teraz możemy skorzystać ze słabej ekstensjonalności. Dla dowolnego a, korzystając z założenia indukcyjnego o termie P, dostaniemy

$$\begin{aligned}[\![(\lambda y.\, P)[x := N]]\!]_{v[x\mapsto [\![N]\!]_v]} \cdot a = [\![\lambda y.P[x := N]]\!]_v \cdot a = [\![P[x := N]]\!]_{v[y\mapsto a]} = \\ = [\![P]\!]_{v[y\mapsto a][x\mapsto [\![N]\!]_{v[y\mapsto a]}]} = [\![P]\!]_{v[y\mapsto a][x\mapsto [\![N]\!]_v]} = [\![\lambda y.\, P]\!]_{v[x\mapsto [\![N]\!]_v]} \cdot a.\end{aligned}$$

Ponieważ mamy do czynienia z lambda-modelem, więc z tego wynika

$$[\![\lambda y.\, P[x := N]]\!]_v = [\![\lambda y.\, P[x := N]]\!]_{v[x\mapsto [\![N]\!]_v]} = [\![\lambda y\, P]\!]_{v[x\mapsto [\![N]\!]_v]}.$$

Niech teraz $x \in \mathrm{FV}(N)$. Ustalmy nową zmienną z (tj. taką, że $x \neq z \notin \mathrm{FV}(M), \mathrm{FV}(N)$) i niech $w = v[z \mapsto [\![N]\!]_v]$. Wtedy z Wniosku 1.5 mamy $M[x := N] = M[x := z][z := N]$, a ponieważ $z \notin \mathrm{FV}(N)$ i $x \notin \mathrm{FV}(z)$, więc z pierwszej części wynika

$$[\![M[x := N]]\!]_v = [\![M[x := z]]\!]_w = [\![M]\!]_{w[x\mapsto [\![z]\!]_w]} = [\![M]\!]_{v[x\mapsto [\![z]\!]_w]} = [\![M]\!]_{v[x\mapsto [\![N]\!]_v]}.$$

Przedostatnia równość bierze się stąd, że $z \notin \mathrm{FV}(M)$. ∎

Fakt 10.3 *Każdy lambda-model jest lambda-algebrą.*

Dowód: Przez indukcję ze względu na definicję beta-równości pokazujemy że jeśli $M =_\beta N$, to $[\![M]\!]_v = [\![N]\!]_v$.

Najpierw zauważmy, że $[\![(\lambda x.P)Q]\!]_v = [\![\lambda x.P]\!]_v \cdot [\![Q]\!]_v = [\![P]\!]_{v[x\mapsto [\![Q]\!]_v]} = [\![P[x := Q]]\!]_v$ na mocy lematu 10.2.

Niech teraz $M = \lambda x.P$ i $N = \lambda x.Q$, gdzie $P =_\beta Q$. Z założenia indukcyjnego wynika, że $[\![P]\!]_u = [\![Q]\!]_u$, przy każdym u, w szczególności gdy $u = v[x \mapsto a]$. A zatem $[\![M]\!]_v \cdot a = [\![P]\!]_{v[x\mapsto a]} = [\![Q]\!]_{v[x\mapsto a]} = [\![N]\!]_v \cdot a$ dla dowolnego a. Stąd $[\![M]\!]_v = [\![N]\!]_v$.

Pozostałe przypadki są rutynowe. ∎

Twierdzenie 10.4 (o pełności) *Następujące warunki są równoważne:*

1) $M =_\beta N$*;*

2) $\mathcal{A} \models M = N$ *dla dowolnej lambda-algebry* $\mathcal{A}$*;*

3) $\mathcal{A} \models M = N$ *dla dowolnego lambda-modelu* $\mathcal{A}$*.*

Dowód: Implikacja (1) ⇒ (2) wynika wprost z definicji, a implikacja (2) ⇒ (3) z Lematu 10.3. Natomiast implikacja (3) ⇒ (1) wynika stąd, że $\mathfrak{M}(\lambda)$ jest lambda-modelem, a termy równe w $\mathfrak{M}(\lambda)$ muszą być β-równe (rozpatrzmy trywialne wartościowanie $v(x) = x$). ∎

Modele częściowo uporządkowane

Na początek przypomnijmy kilka definicji. Niech $\langle A, \leq\rangle$ będzie porządkiem częściowym.

- Podzbiór B zbioru A jest *skierowany* wtedy i tylko wtedy, gdy dla dowolnych $a, b \in B$ istnieje takie $c \in B$, że $a, b \leq c$.
- Zbiór A jest *zupełnym* porządkiem częściowym (cpo) wtedy i tylko wtedy, gdy każdy jego skierowany podzbiór ma kres górny.

Oczywiście każdy łańcuch jest zbiorem skierowanym. W szczególności elementy dowolnego ciągu wstępującego $a_0 \leq a_1 \leq a_2 \leq \ldots$ tworzą zbiór skierowany. Także zbiór pusty jest zbiorem skierowanym. Z definicji porządku zupełnego wynika więc istnienie elementu najmniejszego $\sup \varnothing$, tradycyjnie oznaczanego przez $\bot$.

Niech teraz $\langle A, \leq\rangle$ i $\langle B, \leq\rangle$ będą porządkami częściowymi.

- Funkcja $f : A \to B$ jest *monotoniczna* wtedy i tylko wtedy, gdy dla dowolnych $x, y \in A$ nierówność $x \leq y$ implikuje $f(x) \leq f(y)$.
- Jeśli $\langle A, \leq\rangle$ i $\langle B, \leq\rangle$ są zupełnymi porządkami częściowymi to funkcja $f : A \to B$ jest *ciągła* wtedy i tylko wtedy, gdy f zachowuje kresy górne niepustych zbiorów skierowanych, tj. dla dowolnego skierowanego i niepustego podzbioru $X \subseteq A$ istnieje $\sup f(X)$ i zachodzi równość $f(\sup X) = \sup f(X)$.

Fakt 10.5 (ćwiczenie)

1) Każda funkcja ciągła jest monotoniczna.

2) Złożenie funkcji ciągłych jest ciągłe.

Zbiór wszystkich funkcji ciągłych z A do B oznaczamy przez $[A \to B]$. Ten zbiór, uporządkowany warunkiem:

$$f \leq g \quad \text{wtedy i tylko wtedy, gdy} \quad \forall a \in A(f(a) \leq g(a)),$$

jest zupełnym porządkiem częściowym. Podobnie można nadać strukturę cpo zbiorowi $A \times B$ za pomocą zwykłego uporządkowania „po współrzędnych":

$$\langle a, b\rangle \leq \langle a', b'\rangle \quad \text{wtedy i tylko wtedy, gdy} \quad a \leq a' \text{ i } b \leq b'.$$

Lemat 10.6 *Funkcja $f : D_1 \times D_2 \to D$ jest ciągła wtedy i tylko wtedy gdy jest ciągła ze względu na obie współrzędne, tj. dla dowolnych $a \in D_1$ i $b \in D_2$ oraz dowolnych skierowanych $A \subseteq D_1$ i $B \subseteq D_2$ zachodzi $f(\sup A, b) = \sup f(A, b)$ oraz $f(a, \sup B) = \sup f(a, B)$.*

Dowód: Implikacja z lewej do prawej jest oczywista. Dla dowodu implikacji odwrotnej rozpatrzmy niepusty skierowany zbiór $X \subseteq D_1 \times D_2$ i niech $A = \pi_1(X)$ oraz $B = \pi_2(X)$. Oczywiście $X \subseteq A \times B$, chociaż niekoniecznie na odwrót. Ale jeśli $\langle a, b\rangle \in A \times B$, to zawsze istnieje takie $\langle a', b'\rangle \in X$, że $\langle a, b\rangle \leq \langle a', b'\rangle$. Istotnie, mamy wtedy pary $\langle a, b''\rangle, \langle a'', b\rangle \in X$. Skoro zbiór X jest skierowany, to $\langle a, b''\rangle, \langle a'', b\rangle \leq \langle a', b'\rangle$ dla pewnego $\langle a', b'\rangle \in X$. Porządek jest po współrzędnych, więc $\langle a, b\rangle \leq \langle a', b'\rangle$.

Nietrudno zauważyć, że $\sup X = \sup(A \times B) = \langle \sup A, \sup B\rangle$. Niech $\langle a_0, b_0\rangle$ będzie tym kresem. Należy sprawdzić, że $f(a_0, b_0) = \sup f(X)$. Oczywiście $f(a_0, b_0) \geq \sup f(X)$, przypuśćmy więc, że $c \geq f(a, b)$ dla wszystkich $(a, b) \in X$. Wtedy także $c \geq f(a, b)$ dla wszystkich $(a, b) \in A \times B$. Jeśli ustalimy dowolne $b \in B$, to na mocy ciągłości ze względu na pierwszą współrzędną, otrzymamy $c \geq f(a_0, b) = \sup f(A, b)$. Dalej z ciągłości ze względu na drugą współrzędną wynika także $c \geq f(a_0, b_0)$. ∎

Lemat 10.7 *Operacja aplikacji $Ap : D_1 \times [D_1 \to D_2] \to D_2$, określona przez $Ap(d, f) = f(d)$, jest funkcją ciągłą.*

Dowód: Na mocy lematu 10.6 wystarczy sprawdzić ciągłość ze względu na współrzędne. Dla skierowanego $A \subseteq D_1$ i ustalonego f mamy $\sup Ap(A, f) = \sup f(A) = f(\sup A) = Ap(\sup A, f)$, bo f jest ciągła. A jeśli $F \subseteq [D_1 \to D_2]$ jest skierowany oraz $a \in D_1$ to $\sup Ap(a, F) = \sup\{f(a) \mid f \in F\} = (\sup F)(a) = Ap(a, \sup F)$ z definicji $\sup F$. ∎

Lemat 10.8 *Operacja abstrakcji $Abs : [(D_1 \times D_2) \to D] \to [D_1 \to [D_2 \to D]]$, dana przez $Abs(f)(d)(e) = f(d, e)$, jest dobrze określona i ciągła.*

Dowód: Dla dowolnego d, funkcja $Abs(f)(d)$ jest złożeniem funkcji ciągłej f z funkcją ciągłą $g(e) = \langle d, e\rangle$, zatem jest ciągła.

Dla $A \subseteq D_1$ mamy $Abs(f)(\sup A)(e) = f(\sup A, e) = \sup f(A, e) = \sup Abs(f)(A)(e)$ czyli $Abs(f)(\sup A) = \sup Abs(f)(A)$, co oznacza, że funkcja $Abs(f)$ jest ciągła.

Wreszcie gdy F jest skierowanym podzbiorem $[(D_1 \times D_2) \to D]$, to $Abs(\sup F)(d)(e) = (\sup F)(d, e) = \sup\{f(d, e) \mid f \in F\} = \sup\{Abs(f)(d)(e) \mid f \in F\} = \sup Abs(F)(d)(e)$, czyli sama funkcja Abs też jest ciągła. ∎

Refleksywne cpo

Kluczowa definicja jest taka. Zupełny porządek częściowy D jest *refleksywny*, gdy przestrzeń funkcyjna $[D \to D]$ jest retraktem D, tj. gdy istnieją przekształcenia ciągłe $F : D \to [D \to D]$ i $G : [D \to D] \to D$, takie, że $F \circ G = \mathrm{id}_{[D\to D]}$. Zauważmy, że wtedy $[D \to D]$ jest izomorficzne z pewnym podzbiorem D (obrazem funkcji G).

Na takim cpo możemy określić lambda-interpretację $\mathcal{D} = \langle D, \cdot, [\![\ ]\!]\rangle$, gdzie dla $a, b \in D$

$$a \cdot b = F(a)(b),$$

a znaczenie termów jest określone tak:

- Jeśli x jest zmienną, to $[\![x]\!]_v = v(x)$;
- $[\![PQ]\!]_v = [\![P]\!]_v \cdot [\![Q]\!]_v$;
- $[\![\lambda x.P]\!]_v = G(\xi_P^{v,x})$.

Powyżej, symbol $\xi_P^{v,x}$ oznacza funkcję określoną równaniem $\xi_P^{v,x}(a) = [\![P]\!]_{v[x\mapsto a]}$, którą nieformalnie możemy zapisać tak $\lambda\!\!\lambda a.[\![P]\!]_{v[x\mapsto a]}$.

Nietrudno zauważyć, że wtedy zachodzą następujące warunki:

- $G(f) \cdot a = f(a)$, dla dowolnego $f \in [D \to D]$ i $a \in D$;
- $[\![\lambda x.P]\!]_v \cdot a = [\![P]\!]_{v[x\mapsto a]}$, dla dowolnego $a \in D$;
- Jeśli $v|_{\mathrm{FV}(P)} = u|_{\mathrm{FV}(P)}$, to $[\![P]\!]_v = [\![P]\!]_u$,

a zatem faktycznie mamy lambda-interpretację. Nietrudno też przeoczyć pewien drobiazg: poprawność powyższej definicji nie jest wcale oczywista i wymaga dowodu. Operacja G jest bowiem określona tylko dla funkcji ciągłych.

Lemat 10.9 *Dla dowolnych M i v, funkcja $\xi_M^{v,x}$ (czyli $\lambda\!\!\lambda a.[\![M]\!]_{v[x\mapsto a]}$) jest ciągła.*

Dowód: Indukcja ze względu na M. Dla $M = PQ$, funkcja $\xi_M^{v,x} = \lambda\!\!\lambda a.[\![M]\!]_{v[x\mapsto a]}$ jest złożeniem postaci $Ap \circ h$, gdzie funkcja h dana wzorem $h(a) = \langle F(\xi_P^{v,x}(a)), \xi_Q^{v,x}(a)\rangle$ jest ciągła. Jeśli $M = \lambda y\, N$ to z założenia indukcyjnego funkcja f, która parze $\langle a, b\rangle$ przypisuje wartość $[\![N]\!]_{v[x\mapsto a][y\mapsto b]}$ jest ciągła ze względu na obie współrzędne, a zatem ciągła na mocy lematu 10.6. Natomiast funkcja $\xi_M^{v,x} = \lambda\!\!\lambda a.G(\lambda\!\!\lambda b.[\![N]\!]_{v[x\mapsto a][y\mapsto b]})$ jest złożeniem $G \circ Abs(f)$. ∎

Twierdzenie 10.10 *Jeśli D jest refleksywnym cpo, to lambda-interpretacja $\mathcal{D} = \langle D, \cdot, [\![\ ]\!] \rangle$ jest lambda-modelem.*

Dowód: Pozostaje sprawdzić, że mamy słabą ekstensjonalność. Wystarczy w tym celu zauważyć, że warunek $[\![\lambda x.M]\!]_v \approx [\![\lambda x.N]\!]_v$ implikuje $\xi_M^{v,x} = \xi_N^{v,x}$. ∎

Fakt 10.11 *Model jak wyżej jest ekstensjonalny wtedy i tylko wtedy, gdy $G \circ F = \mathrm{id}_D$.*

Ćwiczenia

1. Pokazać, że $\mathfrak{M}^0(\lambda)$, $\mathfrak{M}(\lambda)$, $\mathfrak{M}(\lambda\eta)$ i $\mathfrak{M}^0(\lambda\eta)$ są lambda-algebrami.
2. Pokazać, że $\mathfrak{M}(\lambda)$ i $\mathfrak{M}(\lambda\eta)$ są lambda-modelami.
3. Znaleźć w książce Barendregta, że $\mathfrak{M}^0(\lambda)$ ani $\mathfrak{M}^0(\lambda\eta)$ nie są lambda-modelami.
4. (Meyer) Niech $\mathtt{1} = [\![\mathbf{1}]\!]$. Pokazać, że lambda-algebra jest lambda-modelem wtedy i tylko wtedy, gdy z warunku $a \approx b$ wynika $\mathtt{1} \cdot a = \mathtt{1} \cdot b$.
5. Czy z tego, że $\mathcal{A} \models \lambda x.Mx = M$ dla wszystkich M wynika ekstensjonalność interpretacji $\mathcal{A}$? Inaczej: czy interpretacja $\beta\eta$-konwersji musi być ekstensjonalna? *Wskazówka: Użyć zadania 3.*
6. Pokazać, że jeśli D jest refleksywnym cpo i $f : D \to D$ jest ciągła, to istnieje takie $a \in D$, że $f(x) = a \cdot x$ dla dowolnego x.
7. Czy funkcja odwrotna do ciągłej bijekcji musi być ciągła?
8. Jeśli $\mathrm{FV}(M) = \{x_1, ..., x_n\}$, to $[\![M]\!]$ można uważać za funkcję z D^n do D. Pokazać, że ta funkcja jest zawsze ciagla.

11 Model $\mathcal{D}_\infty$

Skonstruujemy teraz konkretny przykład modelu — pewne refleksywne cpo, zwane modelem Scotta $\mathcal{D}_\infty$. Zacznijmy od prostej definicji.

Niech A i B będą cpo. *Projekcją* z B na A nazywamy parę funkcji ciągłych

$$\varphi : A \to B \quad \text{i} \quad \psi : B \to A,$$

spełniającą warunki

$$\psi \circ \varphi = \mathrm{id}_A \quad \text{oraz} \quad \varphi \circ \psi \leq \mathrm{id}_B.$$

Zauważmy, że wtedy $\varphi(\bot_A) = \bot_B$, bo $\varphi(\bot_A) \leq \varphi(\psi(\bot_B)) \leq \bot_B$.[8] Gdy mamy taką projekcję, to w szczególności A jest retraktem B. Identyfikując element $d \in A$ z jego obrazem $\varphi(d)$, możemy wtedy uważać zbiór A za podzbiór B. Przykładem projekcji jest para funkcji

$$\varphi_0 : D \to [D \to D] \quad \text{i} \quad \psi_0 : [D \to D] \to D, \text{ określona tak:}$$
$$\varphi_0(d)(a) = d, \quad \text{oraz} \quad \psi_0(f) = f(\bot)$$

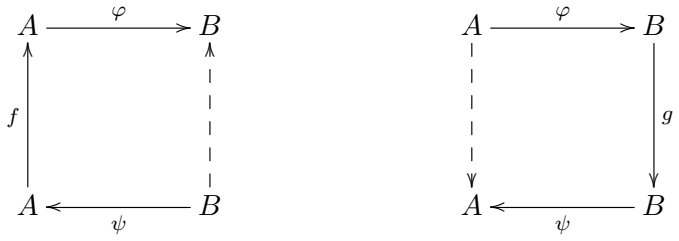
dla dowolnych $a, d \in D$ i $f \in [D \to D]$. Mamy tu zanurzenie zbioru D w przestrzeń funkcyjną, przy którym każdy element D jest utożsamiany z funkcją stałą.

[8] Ogólniej: $\varphi(x)$ to najmniejszy element y o własności $\psi(y) = x$.

Jeśli mamy projekcję (φ, ψ) z B na A, to możemy „podnieść" tę projekcję na poziom funkcji, czyli określić nową projekcję (φ^*, ψ^*) z przestrzeni $[B \to B]$ na $[A \to A]$. Dla $f : A \to A$ i $g : B \to B$ przyjmujemy

$$\varphi^*(f) = \varphi \circ f \circ \psi \quad \text{oraz} \quad \psi^*(g) = \psi \circ g \circ \varphi,$$

co ilustruje obrazek:



Lemat 11.1 *Jeśli (φ, ψ) jest projekcją z B na A, to para (φ^*, ψ^*) jest projekcją z $[B \to B]$ na $[A \to A]$.*

Dowód: Ćwiczenie. ∎

Przypuśćmy teraz, że mamy jakiekolwiek cpo D. Określimy ciąg zbiorów D_n zaczynając od $D_0 = D$ i indukcyjnie przyjmując $D_{n+1} = [D_n \to D_n]$. Oczywiście wszystkie D_n są cpo.

Dalej definiujemy przez indukcję ciąg projekcji (φ_n, ψ_n) z D_{n+1} na D_n, przyjmując

$$(\varphi_{n+1}, \psi_{n+1}) = (\varphi_n^*, \psi_n^*).$$

Mamy więc transmisję w obie strony:

$$D_0 \xrightarrow{\varphi_0} D_1 \xrightarrow{\varphi_1} D_2 \xrightarrow{\varphi_2} \cdots$$

$$D_0 \xleftarrow{\psi_0} D_1 \xleftarrow{\psi_1} D_2 \xleftarrow{\psi_2} \cdots$$

Każdy ciąg $(x_n)_{n\in\mathbb{N}}$ złożony z elementów $x_n \in D_n$ i spełniający dla dowolnego n warunek $x_n = \psi_n(x_{n+1})$ nazywamy *nicią*. Tak wygląda nić:

$$x_0 \xleftarrow{\psi_0} x_1 \xleftarrow{\psi_1} x_2 \xleftarrow{\psi_2} \cdots$$

Dla nici stosujemy notację $x = (x_n)_{n\in\mathbb{N}}$.

Zbiór wszystkich nici oznaczamy przez D_∞. Można go uporządkować po współrzędnych:

$$x \leq y \quad \text{wtedy i tylko wtedy, gdy} \quad \forall n \in \mathbb{N}(x_n \leq y_n).$$

Lemat 11.2 *Zbiór D_∞ jest zupełnym porządkiem częściowym.*

Dowód: Jeśli $X \subseteq D_\infty$ jest skierowany, to każdy ze zbiorów $X_n = \{x_n \mid x \in X\}$ jest skierowany, zatem istnieją kresy $y_n = \sup X_n$. Wtedy dla dowolnego n:

$$\psi_n(y_{n+1}) = \psi_n(\sup X_{n+1}) = \sup \psi_n(X_{n+1}) = \sup X_n = y_n,$$

bo funkcja ψ_n jest ciągła oraz $\psi_n(X_{n+1}) = X_n$. A zatem ciąg $y = (y_n)_{n\in\mathbb{N}}$ jest nicią (należy do D_∞). Pozostaje łatwe sprawdzenie, że $y = \sup X$. ∎

Dla dowolnych $n \leq m$, mamy oczywiście projekcję $(\varphi_{nm}, \psi_{nm})$ z D_m na D_n, czyli złożenie

$$\varphi_{nm} = \varphi_{m-1} \circ \cdots \circ \varphi_n, \quad \psi_{nm} = \psi_n \circ \cdots \circ \psi_{m-1}.$$

Lemat 11.3 *Dla $n \leq m$ zachodzi $\varphi^*_{nm} = \varphi_{n+1,m+1}$ oraz $\psi^*_{nm} = \psi_{n+1,m+1}$.*

Można też określić projekcje $(\varphi_{n\infty}, \psi_{n\infty})$ z D_∞ na D_n. Dla $x \in D_n$ i $y \in D_\infty$:

$$(\varphi_{n\infty}(x))_i = \begin{cases} \psi_{in}(x), & \text{gdy } i < n; \\ x, & \text{gdy } i = n; \\ \varphi_{ni}(x), & \text{gdy } i > n. \end{cases} \qquad \psi_{n\infty}(y) = y_n.$$

Od tej pory możemy uważać, że zachodzą inkluzje

$$D_0 \subseteq D_1 \subseteq D_2 \subseteq \cdots \subseteq D_\infty,$$

zadane przez powyższe projekcje. W szczególności każdy element $x \in D_n$ utożsamiamy z nicią prawie wszędzie równą x:

$$x_0 \xleftarrow{\psi_0} x_1 \xleftarrow{\psi_1} x_2 \xleftarrow{\psi_2} \cdots \xleftarrow{\psi_{n-2}} x_{n-1} \xleftarrow{\psi_{n-1}} x \xleftarrow{\psi_n} x \xleftarrow{\psi_{n+1}} x \xleftarrow{\psi_{n+2}} \cdots$$

bo przecież x uważamy za element wszystkich D_m dla $m \geq n$. Zauważmy, że mamy tu nierówności $x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_{n-1} \leq x_n = x$.

Lemat 11.4

1. *Każda nić x jest ciągiem wstępującym $(x_0 \leq x_1 \leq x_2 \leq \ldots)$ i przy tym $x = \sup x_n$;*
2. *Najmniejszy element jest zawsze ten sam: $\bot_{D_0} = \bot_{D_n} = \bot_{D_\infty}$.*

Dowód: Ćwiczenie. ∎

Lemat 11.5 *Aplikacja w D_∞ określona wzorem*

$$x \cdot y = \sup\{x_{n+1}(y_n) \mid n \in \mathbb{N}\}$$

jest operacją ciągłą.

Dowód: Najpierw trzeba pokazać, że kres istnieje, elementy $x_{n+1}(y_n) \in D_n$ nie muszą bowiem tworzyć nici. Jest to jednak ciąg wstępujący, co wynika z takich związków pomiędzy elementami zbioru D_n:

$$\begin{aligned} \varphi_{n-1}(x_n(y_{n-1})) &= \varphi_{n-1}(\psi_{n-1}(x_{n+1}(\varphi_{n-1}(y_{n-1})))) \leq \\ &\quad x_{n+1}(\varphi_{n-1}(y_{n-1})) = x_{n+1}(\varphi_{n-1}(\psi_{n-1}(y_n))) \leq x_{n+1}(y_n). \end{aligned}$$

Dowód ciągłości korzysta z lematu 10.6. Ciągłość ze względu na x liczymy tak: Z jednej strony $\sup_{x\in X} x \cdot y = \sup_{x\in X} \sup_{n\in\mathbb{N}} x_{n+1}(y_n)$, a z drugiej strony

$$(\sup X)\cdot y = \sup_{n\in\mathbb{N}}(\sup X)_{n+1}(y_n) = \sup_{n\in\mathbb{N}}(\sup X_{n+1})(y_n) = \sup_{n\in\mathbb{N}}\sup_{x\in X} x_{n+1}(y_n),$$

gdzie $X_{n+1} = \{x_{n+1} \mid x \in X\}$. A przecież to jest to samo. Ciągłość ze względu na y jest nawet łatwiejsza. Mamy bowiem $x\cdot\sup Y = \sup_n(x_{n+1}((\sup Y)_n)) = \sup_n\sup_y x_{n+1}(y_n)$ oraz $\sup(x\cdot Y) = \sup_y\sup_n x_{n+1}(y_n)$. ∎

Teraz trochę o tym, jak działa aplikacja.

Lemat 11.6

1. *Jeśli $x \in D_{n+1}$, to $x\cdot y = x(y_n)$, a jeśli dodatkowo $y \in D_n$, to $x\cdot y = x(y)$;*
2. *Jeśli $y \in D_n$ to $(x\cdot y)_n = x_{n+1}(y)$.*
3. *Zawsze $(x\cdot\bot)_0 = x_0$.*
4. *Jeśli $x \in D_0$, to $x\cdot y = x = x\cdot\bot$.*

Dowód: 1. W części pierwszej należy pokazać, że $x\cdot y = \sup_m x_{m+1}(y_m) = x_{n+1}(y_n)$. W tym celu zauważmy, że dla $m < n$ mamy

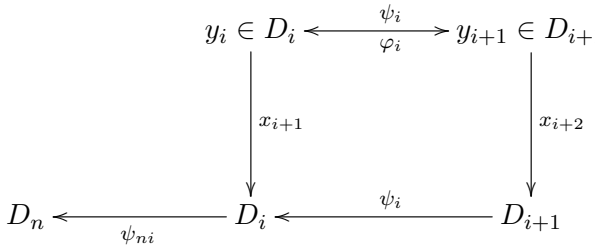
$$x_{m+1}(y_m) = \psi_{m+1,n+1}(x)(\psi_{mn}(y_n)) = \psi^*_{mn}(x_{n+1})(\psi_{mn}(y_n)) \le \psi_{mn}(x_{n+1}(y_n)),$$

bo $x_{n+1} = x$, $y_n = y$, oraz $\psi^*_{mn}(x_{n+1}) = \psi_{mn}\circ x_{n+1}\circ\varphi_{mn}$. Natomiast dla $m \ge n$

$$x_{m+1}(y_m) = \varphi_{n+1,m+1}(x)(y_m) = \varphi^*_{nm}(x)(y_m) = (\varphi_{n,m}\circ x\circ\psi_{n,m})(y_m) = \varphi_{n,m}(x(y_n))$$

bo $\varphi^*_{nm}(x) = \varphi_{nm}\circ x\circ\psi_{nm}$ oraz $\psi_{n,m}(y_m) = y_n$.

2. W części drugiej, dla dowolnego $i \ge n$ mamy $y_{i+1} = \varphi(y_i)$, więc $x_{i+1}(y_i) = \psi_i(x_{i+2}(y_{i+1}))$, co implikuje $(x_{i+2}(y_{i+1}))_n = (x_{i+1}(y_i))_n = x_{n+1}(y)$. A wszystko to widać na takim obrazku:



3. Część trzecia wynika z drugiej, dla $y = \bot \in D_0$, a część czwarta wprost z definicji. ∎

Lemat 11.7 *Aplikacja w D_∞ jest ekstensjonalna: jeśli $x\cdot z = y\cdot z$ zachodzi dla dowolnego z, to elementy x i y są równe.*

Dowód: W istocie zachodzi implikacja

$$\textit{jeśli } \forall z \in D_\infty(x\cdot z \le y\cdot z), \textit{ to } x \le y.$$

Nierówność $x_0 \le y_0$ wynika z lematu 11.6(3), bo mamy $x\cdot\bot \le y\cdot\bot$. Dalej, z lematu 11.6(2),

$$x_{n+1}(z) = (x \cdot z)_n \leq (y \cdot z)_n = y_{n+1}(z),$$

dla dowolnego n i dowolnego $z \in D_n$, a więc $x_{n+1} \leq y_{n+1}$. ∎

Lemat 11.8 *Niech $f \in D_{n+1}$ i $g \in D_{n+2}$ będą takie, że $\varphi_n(f(a)) \leq g(\varphi_n(a))$ dla $a \in D_n$. Wtedy $f \leq g$ w zbiorze D_∞.*

Dowód: Należy sprawdzić, jakie nici wyznaczają f i g. Ponieważ $\varphi_n(f(a)) \leq g(\varphi_n(a))$ więc $f(a) = \psi_n(\varphi_n(f(a))) \leq \psi_n(g(\varphi_n(a)) = \psi_{n+1}(g)(a)$, czyli $f \leq \psi_{n+1}(g)$ w zbiorze D_{n+1}. Inaczej mówiąc $(f)_{n+1} \leq (g)_{n+1}$, a reszta wynika z monotoniczności funkcji φ_i i ψ_i. ∎

Twierdzenie 11.9 *Zbiór D_∞ jest refleksywnym cpo, co więcej, jest on izomorficzny z przestrzenią funkcyjną $[D_\infty \to D_\infty]$.*

Dowód: Funkcja $F : D_\infty \to [D_\infty \to D_\infty]$ jest określona w oczywisty sposób:

$$F(x)(y) = x \cdot y.$$

Z lematów 11.5 i 11.7 wynika, że F jest ciągła i różnowartościowa. Aby sprawdzić, że jest to surjekcja, weźmy dowolną funkcję $f \in [D_\infty \to D_\infty]$ i niech $f^{(n)} : D_n \to D_n$ będzie jej „aproksymacją" do D_n, tj. niech $f^{(n)}(y) = f(y)_n$ dla $y \in D_n$.

Ciąg $f^{(n)}$ nie musi być nicią, ale jest wstępujący. Istotnie, zauważmy że elementy $y \in D_n$ oraz $\varphi_n(y) \in D_{n+1}$ wyznaczają tę samą nić, zatem z punktu widzenia D_∞ są po prostu równe. A więc $f(y) = f(\varphi_n(y))$ w D_∞ skąd $f^{(n)}(y) = f(y)_n \leq f(\varphi_n(y))_{n+1} = f^{(n+1)}(\varphi_n(y))$ zachodzi dla $y \in D_n$. Ściślej, mamy nierówność $\varphi_n(f^{(n)}(y)) \leq f^{(n+1)}(\varphi_n(y))$ pomiędzy elementami zbioru D_{n+1}, co na mocy lematu 11.8 oznacza $f^{(n)} \leq f^{(n+1)}$.

Skoro ciąg $f^{(n)}$ jest wstępujący, to ma kres $a = \sup_n f^{(n)}$. Trzeba teraz sprawdzić, że $F(a) = f$. Weźmy dowolne $y \in D_\infty$. Ponieważ ciągi $f^{(n)}$ i y_n są wstępujące, więc

$$F(a)(y) = a \cdot y = (\sup_n f^{(n)}) \cdot y = \sup_n (f^{(n)} \cdot y) = \sup_n f^{(n)}(y_n) = \sup_n f(y_n)_n = f(y)$$

Jeśli $f, g \in [D_\infty \to D_\infty]$ oraz $f \leq g$ to oczywiście także $\sup_n f^{(n)} \leq \sup_n g^{(n)}$. Oznacza to, że funkcja $G : [D_\infty \to D_\infty] \to D_\infty$, odwrotna do F jest monotoniczna. A zatem funkcja F jest izomorfizmem porządków, w szczególności F i G są ciągłe. ∎

Wniosek 11.10 *Zbiór D_∞ wyznacza ekstensjonalny lambda-model $\mathcal{D}_\infty = \langle D_\infty, \cdot, [\![\ ]\!] \rangle$.*

Twierdzenie 11.11 (Hyland, Wadsworth) *Termy M i N są obserwacyjnie równoważne wtedy i tylko wtedy, gdy $\mathcal{D}_\infty \models M = N$.*

Dowód: Można go znaleźć w książce Barendregta. ∎

Ćwiczenia

1. Uzupełnić szczegóły dowodów.
2. Jakie nici są interpretacjami termów **K**, ω, itd. w modelu $\mathcal{D}_\infty$?
3. Niech $D_0 = \{\bot, \top\}$. Ile elementów mają D_1 i D_2?

12 Model $\mathcal{P}_\omega$

Model $\mathcal{P}_\omega$ też jest pomysłu Scotta. Symbol P_ω oznacza po prostu zbiór $\mathbf{P}(\mathbb{N})$ uporządkowany przez zwykłą inkluzję. Konstrukcja tego modelu opiera się na tym, że krata $\langle P_\omega, \subseteq\rangle$ jest *algebraiczna* tj. każdy element jest sumą swoich skończonych podzbiorów. Konsekwencją tego jest to, że funkcja ciągła nad P_ω jest zdeterminowana przez swoje zachowanie na zbiorach skończonych.

Lemat 12.1 *Funkcja $f : P_\omega \to P_\omega$ jest ciągła wtedy i tylko wtedy, gdy*

$$f(a) = \bigcup\{f(e) \mid e \text{ skończony oraz } e \subseteq a\},$$

dla dowolnego $a \in P_\omega$.

Ponieważ zbiorów skończonych jest przeliczalnie wiele, funkcji ciągłych jest continuum i można informację o takiej funkcji reprezentować za pomocą jednego elementu P_ω. W tym celu potrzebna jest tylko funkcja pary, na przykład:

$$(m, n) = \frac{(n+m)(n+m+1)}{2} + m,$$

i kodowanie zbiorów skończonych, na przykład:

$$e_n = \{k_0, k_1, \ldots, k_{r-1}\}, \quad \text{dla } n = \sum_{i<r} 2^{k_i}.$$

Teraz można określić funkcje

$$\mathit{graph} : [P_\omega \to P_\omega] \to P_\omega, \qquad \mathit{fun} : P_\omega \to [P_\omega \to P_\omega]$$

następująco. Dla $f \in [P_\omega \to P_\omega]$ i $a, b \in P_\omega$ przyjmujemy

$$\begin{aligned} \mathit{graph}(f) =& \{(n, m) \mid m \in f(e_n)\};\\ \mathit{fun}(a)(x) =& \{m \mid \exists n \in \mathbb{N}(e_n \subseteq x \wedge (n, m) \in a\}. \end{aligned}$$

Lemat 12.2 *Funkcje graph i fun są ciągłe, a w dodatku $\mathit{fun} \circ \mathit{graph} = \mathrm{id}_{[P_\omega \to P_\omega]}$.*

Dowód: Ćwiczenie. ∎

A zatem P_ω jest refleksywnym cpo.

Wniosek 12.3 *Struktura $\mathcal{P}_\omega = \langle P_\omega, \cdot, [\![\ ]\!] \rangle$, w której $x \cdot y = fun(x)(y)$, jest lambda-modelem.*

Zauważmy jednak, że funkcje *graph* i *fun* nie są bijekcjami, bo każdy niepusty zbiór postaci $graph(g)$ musi być nieskończony. Istotnie, jeśli należy do niego para (n, m) to także $(k, m) \in graph(g)$, dla $e_n \subseteq e_k$. Zatem model $\mathcal{P}_\omega$ nie jest ekstensjonalny (fakt 10.11).

Fakt 12.4 *Model $\mathcal{P}_\omega$ nie jest modelem $\beta\eta$-konwersji, tj. $\mathcal{P}_\omega \not\models x = \lambda y.xy$.*

Dowód: Niech $v(x) = a \neq \bot$. Wtedy $[\![\lambda y.xy]\!]_v = graph(\lambda\!\!\lambda b.\, [\![xy]\!]_{v[y \mapsto b]}) = graph(\lambda\!\!\lambda b.\, a \cdot b)$. Natomiast $[\![x]\!]_v = a$. Jeśli a jest niepustym zbiorem skończonym, to $[\![\lambda y.xy]\!]_v$, jest zbiorem nieskończonym, w szczególności $[\![\lambda y.xy]\!]_v \neq a$. ∎

Okazuje się, że teoria modelu $\mathcal{P}_\omega$ to dokładnie teoria drzew Böhma.

Twierdzenie 12.5 (Hyland) *Dla dowolnych termów M, N*

$$\mathcal{P}_\omega \models M = N \quad \Longleftrightarrow \quad BT(M) = BT(N)$$

Dowód: Można go znaleźć w książce Barendregta. ∎

Ćwiczenia

1. Uzupełnić szczegóły dowodów.
2. Czy $m \in fun(a)(e_n)$ wtedy i tylko wtedy, gdy $(n, m) \in a$?
3. Jakie zbiory są interpretacjami termów $\mathbf{I}$, $\mathbf{K}$, ω, itd. w modelu $\mathcal{P}_\omega$?
4. Pokazać, że $\mathcal{P}_\omega \not\models \mathbf{1} = \mathbf{I}$.
5. Funkcja ciągła z $\mathcal{P}_\omega$ w $\mathcal{P}_\omega$ jest jednoznacznie wyznaczona przez swoje wartości na zbiorach skończonych. Czy jest jednoznacznie wyznaczona przez swoje wartości na zbiorze pustym i singletonach?

13 Typy proste

Zaczynamy od składni typów. Przyjmijmy, że mamy pewien niepusty[9] zbiór *typów atomowych*, który oznaczymy przez TV. Typy atomowe nazywamy też *zmiennymi typowymi*. *Typy* (ściślej: *typy proste*) definiujemy indukcyjnie:

- Typy atomowe $p, q, r, \ldots$ są typami;
- Jeśli σ i τ są typami, to $\sigma \to \tau$ jest typem.

[9] Czasami zakłada się, że jest tylko jeden atom **0**. My zrobimy tak w rozdziale 17, ale zwykle przyjmujemy, że atomów jest nieograniczona ilość.

Intuicja jest oczywiście taka: typ postaci $\sigma \to \tau$ to typ operatora, który argumentom typu σ przypisuje obiekty typu τ. Zbiór wszystkich typów oznaczymy przez T. Stosujemy taką konwencję, że strzałka jest łączna w prawo, tj. napis $\sigma \to \tau \to \rho$ oznacza $\sigma \to (\tau \to \rho)$. Zauważmy, że każdy typ można zapisać jako $\sigma_1 \to \cdots \to \sigma_n \to p$, gdzie p jest typem atomowym.

W rachunku lambda z typami przypisuje się typy lambda-termom. Ponieważ typ termu może zależeć od typów jego zmiennych wolnych, potrzebna jest taka definicja:

Otoczenie typowe to częściowa funkcja ze zbioru zmiennych przedmiotowych w zbiór typów. Taką funkcję utożsamiamy ze zbiorem par postaci $x : \tau$ gdzie x jest zmienną przedmiotową a τ jest typem. Napis $\Gamma(x : \tau)$ oznacza to samo co $\Gamma[x \mapsto \tau]$, a więc $\Gamma(x : \tau)$ powstaje z Γ przez dodanie deklaracji $x : \tau$, poprzedzone usunięciem deklaracji $x : \Gamma(x)$, jeśli $x \in \mathrm{Dom}(\Gamma)$. Zauważmy, że jeśli $\Gamma \subseteq \Gamma'$, to $\Gamma(x : \tau) \subseteq \Gamma'(x : \tau)$.

Rachunek lambda z typami prostymi (w wersji Curry'ego) to system wnioskowania $\lambda_\to$, w którym wyprowadza się osądy (asercje) postaci $\Gamma \vdash M : \tau$ czytane: *„term M ma typ τ w otoczeniu Γ"*. (Takie systemy wnioskowania nazywamy *systemami przypisania typów*.)

System $\lambda_\to$ ma trzy reguły:

$$\text{(Var)}\quad \Gamma(x:\sigma) \vdash x : \sigma$$

$$\text{(Abs)}\quad \frac{\Gamma(x:\sigma) \vdash M : \tau}{\Gamma \vdash (\lambda x\, M) : \sigma \to \tau} \qquad \text{(App)}\quad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (MN) : \tau}$$

Jeśli $\Gamma \vdash M : \tau$, dle pewnych Γ i τ, to mówimy, że term M jest *typowalny*. Zauważmy, że nie każdy term jest typowalny, na przykład term xx nie ma typu w żadnym otoczeniu. Co więcej, termy typowalne mają wiele typów, na przykład term $\mathbf{I}$ ma w każdym otoczeniu wszystkie typy postaci $\tau \to \tau$.

Oto podstawowe własności systemu $\lambda_\to$.

Lemat 13.1 (o generowaniu)

1. *Jeśli* $\Gamma \vdash MN : \sigma$, *to* $\Gamma \vdash M : \tau \to \sigma$ *i* $\Gamma \vdash N : \tau$ *dla pewnego* τ.
2. *Jeśli* $\Gamma \vdash x : \sigma$, *gdzie* x *jest zmienną, to* $\sigma = \Gamma(x)$.
3. *Jeśli* $\Gamma \vdash \lambda x\, M : \sigma$, *to* $\sigma = \rho \to \tau$ *dla pewnych* ρ *i* τ, *takich że* $\Gamma(x:\rho) \vdash M : \tau$.

Dowód: Natychmiastowy. ∎

Lemat 13.2 (osłabianie) *Jeśli* $\Gamma \vdash M : \sigma$ *oraz* $\Gamma \subseteq \Gamma'$, *to* $\Gamma' \vdash M : \sigma$.

Dowód: Łatwa indukcja ze względu na wyprowadzenie $\Gamma \vdash M : \sigma$. ∎

Lemat 13.3 *Jeśli* $\Gamma(x : \sigma) \vdash M : \tau$ *oraz* $\Gamma \vdash N : \sigma$, *to* $\Gamma \vdash M[x := N] : \tau$.

Dowód: Indukcja ze względu na M. Korzystamy z lematów o generowaniu i osłabianiu. ∎

Wniosek 13.4 (poprawność redukcji) [10] *Jeśli* $\Gamma \vdash M : \tau$ *oraz* $M \to_{\beta\eta} N$, *to* $\Gamma \vdash N : \tau$.

Dowód: Indukcja ze względu na definicję $\to_{\beta\eta}$. Przypadek $M = (\lambda x.P)Q \to_\beta P[x{:=}Q] = N$ wynika z lematu 13.3. ∎

13.1 Wariant Churcha

Alternatywny sposób wprowadzenia typów do rachunku lambda polega na rozszerzeniu składni lambda termów o informację typową. Mówimy wtedy o rachunku z typami w *wersji Churcha*. W skrajnej („ortodoksyjnej") wersji przyjmuje się, że dla każdego typu σ dany jest osobny zbiór TV_σ zmiennych typu σ, po czym definiuje się termy dowolnych typów indukcyjnie:[11]

– Zmienna typu σ jest termem typu σ.
– Jeśli M jest termem typu $\sigma \to \tau$ oraz N jest termem typu σ, to (MN) jest termem typu τ.
– Jeśli x jest zmienną typu σ oraz M jest termem typu τ, to $(\lambda x\, M)$ jest termem typu $\sigma \to \tau$.

W ten sposób dla każdego typu σ otrzymujemy zbiór T_σ termów typu σ. Zbiory T_σ są parami rozłączne – każdy term ma dokładnie jeden typ. Podtermy termu $M \in T_\sigma$ też mają jednoznacznie ustalone typy, co często podkreślamy, stosując adnotacje typowe, tj. pisząc np. $((\lambda x^\sigma.N^\tau)^{\sigma\to\tau}P^\sigma)^\tau \in T_\tau$.

Istnieje też „nieortodoksyjny" wariant systemu w wersji Churcha. W tym wariancie zmienne nie mają z góry ustalonych typów. Typy zmiennych wolnych są określone przez otoczenie, jak u Curry'ego, a typy zmiennych związanych przez adnotacje typowe, które są obowiązkową częścią składni. Mamy więc zmienne, aplikacje (MN) i abstrakcje postaci $(\lambda x{:}\sigma.\, M)$, przy czym za dobrze określone *termy* uważamy tylko te wyrażenia, którym można poprawnie przypisać typ regułami wnioskowania:

$$\text{(Var)}\quad \Gamma(x:\sigma) \vdash x : \sigma$$

$$\text{(Abs)}\quad \frac{\Gamma(x:\sigma) \vdash M : \tau}{\Gamma \vdash (\lambda x{:}\sigma.\, M) : \sigma \to \tau} \qquad \text{(App)}\quad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (MN) : \tau}$$

Delikatność tej definicji polega na tym, że to samo wyrażenie może być poprawnym termem w jednym otoczeniu a w innym nie. Zauważmy też, że w ustalonym otoczeniu typ termu może być tylko jeden (jak w „ortodoksyjnej" wersji Churcha), ale to samo wyrażenie może mieć różne typy w różnych otoczeniach (jak w wersji Curry'ego).

Uwaga: Dla systemów w wersji Churcha powinniśmy na nowo zdefiniować operację podstawienia, relację redukcji itd. Na szczęście w oczywisty sposób, np. ortodoksyjna reguła beta-redukcji ma postać

$$(\lambda x^\sigma.\, M)N^\sigma \Rightarrow M[x^\sigma := N].$$

[10] *Subject reduction property*
[11] Jak zwykle, utożsamia się wyrażenia pozostające w relacji alfa-konwersji.

„Deterministyczny" charakter reguł systemu $\lambda_\to$ powoduje, że różnice pomiędzy trzema wariantami systemu są w istocie nieznaczne. Ortodoksyjny Church to w gruncie rzeczy to samo co nieortodoksyjny Church w jednym ustalonym otoczeniu. A relację między (nieortodoksyjnym) Churchem i Currym ustala operacja *wycierania typów* zdefiniowana tak:

– $|x| = x$;

– $|MN| = |M||N|$;

– $|\lambda x{:}\sigma.\, M| = \lambda x\, |M|$.

Oto najważniejsze własności wycierania typów. Dla czytelności, termy Churcha są tutaj oznaczone czcionką prostą, a termy Curry'ego jak zwykle kursywą.

Twierdzenie 13.5

1. *Jeśli* $\Gamma \vdash \mathrm{M} : \tau$, *to* $\Gamma \vdash |\mathrm{M}| : \tau$.
2. *Jeśli* $\Gamma \vdash M : \tau$, *to istnieje taki term* M, *że* $|\mathrm{M}| = M$ *oraz* $\Gamma \vdash \mathrm{M} : \tau$.
3. *Jeśli* $\mathrm{M} \to_\beta \mathrm{N}$, *to* $|\mathrm{M}| \to_\beta |\mathrm{N}|$.
4. *Jeśli* $|\mathrm{M}| \to_\beta N$, *to istnieje takie* N, *że* $|\mathrm{N}| = N$ *oraz* $\mathrm{M} \to_\beta \mathrm{N}$.

Dowód: Łatwa indukcja. ∎

Twierdzenie 13.5 pokazuje, że systemy Churcha i Curry'ego właściwie niczym sie nie różnią. W istocie term w stylu Churcha to to samo, co term w stylu Curry'ego wraz z odpowiednim wyprowadzeniem typu. Istotnie, aby dany term mógł mieć przypisany typ w otoczeniu Γ, konieczne jest jednoznaczne przypisanie typów wszystkim jego podtermom. Dlatego dobrze otypowany term Curry'ego można sobie wyobrażać jako term z adnotacjami typowymi. na przykład można napisać $(\lambda x^\sigma.N^\tau)P^\sigma : \tau$, żeby zaznaczyć jaki typ użyty został dla zmiennej x w wyprowadzeniu osądu $(\lambda x.N)P : \tau$.

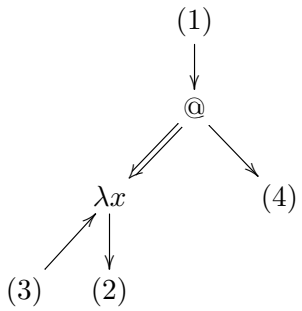
W dalszym ciągu będziemy w dość nieformalny sposób posługiwać się takimi adnotacjami, „mieszając" ze sobą formalizm Churcha i Curry'ego i wybierając zawsze ten, który jest w danym momencie wygodniejszy.

Ćwiczenia

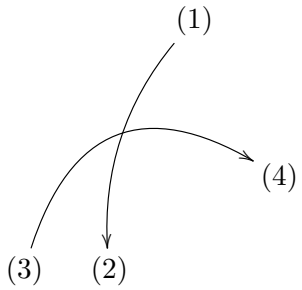
1. Które z następujących termów są typowalne: $\mathbf{K}$, $\mathbf{S}$, $\mathbf{2}$, $\mathbf{\Omega}$, $\mathbf{2K}$, $\lambda x.\, \mathbf{K}(\mathbf{K}\, x)$?
2. Zdefiniować relację alfa-konwersji i operację podstawienia dla wyrażeń w stylu Churcha (w obu wariantach). Następnie zdefiniować beta- i eta-redukcję i udowodnić, że zachowują one typy termów.
3. Pokazać, że oba rachunki w wersji Churcha mają własność Churcha-Rossera, powtarzajac konstrukcję z rozdziału 3.
4. Pokazać, że jeśli $\Gamma \vdash M : \tau$ i $\Gamma \vdash M : \sigma$, w nieortodoksyjnej wersji Churcha, to $\tau = \sigma$.
5. Czy jeśli $\Gamma \vdash |\mathrm{M}| : \tau$, to $\Gamma \vdash \mathrm{M} : \tau$ i czy z $|\mathrm{M}| \to_\beta |\mathrm{N}|$ wynika $\mathrm{M} \to_\beta \mathrm{N}$ (por. twierdzenie 13.5(2,4))?
6. Czy w twierdzeniu 13.5(2) można zamienić słowo „istnieje" na „istnieje dokładnie jeden"?
7. Czy dla η-redukcji zachodzi twierdzenie podobne do 13.5?

14 Normalizacja

Udowodnimy teraz twierdzenie o normalizacji dla termów z typami: każdy typowalny term ma postać normalną. Zaczniemy od rozwiązania ćwiczenia 1 z rozdziału 4: jakie nowe β-redeksy mogą powstać w termie na skutek wykonania jednego kroku β-redukcji? Jeśli term przedstawimy w postaci grafu, to każdy redeks jest wyznaczony przez krawędź skierowaną w lewo od @ do λ (obrazek 12). Usunięcie tej krawędzi powoduje powstanie bezpośrednich połączeń od wierzchołka (1) do (2) oraz od (3) do (4), jak widać na obrazku 13. Uwaga: Obrazki 12 i 13 są nieco uproszczone. W istocie zamiast jednego wierzchołka (3) mamy ich tyle ile jest wystąpień zmiennej x w podtermie zaczepionym w punkcie (2).



Obrazek 12: Beta-redeks przed redukcją ...



Obrazek 13: ...i po redukcji.

Powstanie nowego redeksu jest możliwe, gdy na skutek skrócenia połączeń pomiędzy (1) i (2) oraz (3) i (4) powstanie krawędź łącząca bezpośrednio wierzchołek typu @ z wierzchołkiem typu λ. Może się to zdarzyć na 3 sposoby:

1. W pozycji (1) jest wierzchołek typu @, a w pozycji (2) jest wierzchołek typu λ. Mamy wtedy podterm postaci $(\lambda x \lambda y Q)NP$, który redukuje się do $(\lambda y Q[x := N])P$.

2. W pozycji (3) jest wierzchołek typu @, a w pozycji (4) jest wierzchołek typu λ, tj. podterm $(\lambda x \ldots xP \ldots)(\lambda yQ)$ redukuje się do $\ldots (\lambda yQ)P \ldots$

3. Wierzchołki (2) i (3) są sklejone w jeden, w pozycji (1) jest wierzchołek typu @, a w pozycji (4) jest wierzchołek typu λ. Tak jest w przypadku podtermu $(\lambda x\, x)(\lambda yQ)P$, który redukuje się do $(\lambda yQ)P$.

Oczywiście na skutek redukcji może także dojść do zwielokrotnienia redeksów już istniejących.

Twierdzenie 14.6 *Jeśli* $\Gamma \vdash M : \tau$ *to* M *ma postać normalną.*

Dowód: Rozważamy pewne ustalone wyprowadzenie osądu $\Gamma \vdash M : \tau$. To wyprowadzenie jednoznacznie określa przypisanie typów dla wszystkich termów, które można otrzymać z termu M w drodze redukcji. A zatem możemy bez obawy nieporozumienia stosować adnotacje typowe w formie indeksów w stylu Churcha. *Rangą* redeksu $(\lambda x^{\sigma}.\, P^{\tau})Q^{\sigma}$ nazwiemy długość typu $\sigma \to \tau$, czyli typu przypisanego abstrakcji $(\lambda x.\, P)$. *Ranga* termu to maksymalna ranga występujących w nim redeksów.

Dowód twierdzenia przebiega przez indukcję ze względu na dwa parametry (n, m), gdzie n jest rangą termu M a m jest liczbą redeksów rangi n występujących w M. Wystarczy udowodnić, że na skutek redukcji odpowiednio wybranego redeksu w M para (n, m) musi się (leksykograficznie) zmniejszyć. Wybrać należy redeks rangi n położony (zaczynający się) najdalej na prawo. Przy takiej redukcji zwielokrotnieniu mogą ulec tylko redeksy mniejszej rangi (położone na prawo). Pozostaje zauważyć, że nowe redeksy powstające w wyniku naszej redukcji też muszą być rangi mniejszej od n. Mamy bowiem trzy wcześniej wspomniane możliwości:

1. $(\lambda x^{\alpha} \lambda y^{\beta}.Q^{\gamma})N^{\alpha}P^{\beta} \to (\lambda y^{\beta} Q[x := N]^{\gamma})P^{\beta}$.
2. $(\lambda x^{\alpha\to\gamma} \ldots xP^{\alpha} \ldots)^{\beta}(\lambda y^{\alpha} Q^{\gamma}) \to \ldots (\lambda y^{\alpha} Q^{\gamma})P^{\alpha} \ldots$
3. $(\lambda x^{\alpha\to\beta}\, x^{\alpha\to\beta})(\lambda y^{\alpha} Q^{\beta})P^{\alpha} \to (\lambda y^{\alpha} Q^{\beta})P^{\alpha}$.

W każdym z przypadków nowy redeks jest mniejszej rangi niż redeks wyeliminowany:

1. Abstrakcję typu $\alpha \to \beta \to \gamma$ zastąpiono przez abstrakcję typu $\beta \to \gamma$.
2. Abstrakcję typu $(\alpha \to \gamma) \to \beta$ zastąpiono przez abstrakcję typu $\alpha \to \gamma$.
3. Abstrakcję typu $(\alpha \to \beta) \to \alpha \to \beta$ zastąpiono przez abstrakcję typu $\alpha \to \beta$. ∎

Silna normalizacja

Twierdzenie o normalizacji można wzmocnić: term rachunku lambda z typami redukuje się zawsze do postaci normalnej, niezależnie od przyjętej strategii. Inaczej mówiąc, wszystkie termy z typami mają własność silnej normalizacji (SN). Moralny sens tego faktu jest taki:

statyczna kontrola typów gwarantuje terminację obliczenia niezależnie od kolejności ewaluacji. Program z typami może się zapętlić tylko wtedy, gdy jakiś jego składnik jawnie na to pozwala (np. zawiera on pętlę, rekursję itp.).

Metoda dowodu silnej normalizacji (*computability method*), którą zastosujemy, pochodzi od Williama Taita i bywa stosowana także dla dowodu innych własności języków z typami. W istocie polega ona na konstrukcji pewnego syntaktycznego modelu dla typów prostych. Dla dowolnego typu τ, określimy pewną interpretację typu τ w zbiorze termów, a mianowicie zbiór termów *stabilnych*[12] dla typu τ, który oznaczymy przez $[\![\tau]\!]$.

- $[\![p]\!] := \mathrm{SN}$, gdy p jest atomem;
- $[\![\tau \to \sigma]\!] := \{M \mid \forall N(N \in [\![\tau]\!] \Rightarrow MN \in [\![\sigma]\!])\}$.

Lemat 14.7 *Dla dowolnego typu τ:*

1) $[\![\tau]\!] \subseteq \mathrm{SN}$*;*

2) Jeśli $N_1, \ldots, N_k \in \mathrm{SN}$ *to* $xN_1 \ldots N_k \in [\![\tau]\!]$*. W szczególności zmienne są stabilne.*

Dowód: Indukcja ze względu na τ. Niech na początek τ będzie atomem. Warunek (1) wynika wprost z definicji. Warunek (2) właściwie też, bo oczywiście $xN_1 \ldots N_k \in \mathrm{SN}$.

Teraz niech $\tau = \sigma \to \rho$ i niech $M \in [\![\sigma \to \rho]\!]$. Weźmy dowolną zmienną x. Z założenia indukcyjnego (2) mamy $x \in [\![\sigma]\!]$, więc z definicji $Mx \in [\![\rho]\!]$. A więc $Mx \in \mathrm{SN}$, z założenia indukcyjnego (1). Tym bardziej $M \in \mathrm{SN}$. Pokazaliśmy (1).

W punkcie (2) mamy pokazać, że $xN_1 \ldots N_k P \in [\![\rho]\!]$ dla każdego $P \in [\![\sigma]\!]$. Ale $P \in \mathrm{SN}$ z założenia indukcyjnego (1), więc teza wynika z założenia indukcyjnego (2) dla ρ. ∎

Lemat 14.8 *Jeśli* $M[x{:=}N_0]N_1 \ldots N_k \in \mathrm{SN}$ *oraz* $N_0 \in \mathrm{SN}$*, to także* $(\lambda x.M)N_0N_1 \ldots N_k \in \mathrm{SN}$.

Dowód: Przypuśćmy, że term $P_0 = (\lambda x.M)N_0N_1 \ldots N_k$ ma nieskończony ciąg redukcji $P_0 \to P_1 \to P_2 \to \cdots$ Ponieważ wszystkie termy $M, N_0, \ldots, N_k$ są silnie normalizowalne (ćwiczenie 6.1), więc prędzej czy później będzie zredukowany redeks czołowy. Ściślej, dla pewnego n otrzymamy:

$$P_n = (\lambda x.M')N_0'N_1' \ldots N_k' \to M'[x := N_0']N_1' \ldots N_k' = P_{n+1},$$

gdzie $M \twoheadrightarrow M'$ oraz $N_i \twoheadrightarrow N_i'$ dla $i \leq k$. Ponieważ $M[x := N_0]N_1 \ldots N_k \twoheadrightarrow P_{n+1}$, więc mamy sprzeczność z założeniem. ∎

Lemat 14.9 *Jeśli* $M[x{:=}N_0]N_1 \ldots N_k \in [\![\tau]\!]$ *oraz* $N_0 \in \mathrm{SN}$*, to także* $(\lambda x.M)N_0N_1 \ldots N_k \in [\![\tau]\!]$.

[12] Inna, często spotykana terminologia: termy *obliczalne* lub *redukowalne*.

Dowód: Indukcja ze względu na τ. Przypadek typu atomowego to dokładnie lemat 14.8. Niech więc $\tau = \sigma \to \rho$, i niech $M[x := N_0]N_1 \dots N_k \in [\![\tau]\!]$. Mamy sprawdzić czy dla $P \in [\![\sigma]\!]$ zachodzi $(\lambda x.M)N_0N_1 \dots N_kP \in [\![\rho]\!]$. Ale skoro $M[x := N_0]N_1 \dots N_k \in [\![\tau]\!] = [\![\sigma \to \rho]\!]$ więc $M[x := N_0]N_1 \dots N_kP \in [\![\rho]\!]$ i wystarczy zastosować założenie indukcyjne dla ρ (przyjmując $N_{k+1} = P$). ∎

Następujący lemat można uważać za twierdzenie o poprawności dla semantyki zadanej przez zbiory termów stabilnych.

Lemat 14.10 *Jeśli* $\Gamma \vdash M : \tau$ *oraz* $N_i \in [\![\Gamma(x_i)]\!]$ *dla* $i \leq n$, *to* $M[x_1{:=}N_1, \dots, x_n{:=}N_n] \in [\![\tau]\!]$.

Dowód: Indukcja ze względu na M. Jeśli M jest zmienną x_i, to teza wynika z lematu o generowaniu. Jeśli M jest inną zmienną, to korzystamy z lematu 14.7(2).

W przypadku gdy $M = PQ$, z lematu o generowaniu mamy $\Gamma \vdash P : \zeta \to \tau$ i $\Gamma \vdash Q : \zeta$ dla pewnego typu ζ. Z założenia indukcyjnego $P \in [\![\zeta \to \tau]\!]$ i $Q \in [\![\zeta]\!]$ więc $M = PQ \in [\![\tau]\!]$ wprost z definicji $[\![\zeta \to \tau]\!]$.

Niech teraz $M = \lambda y.P$. Typ τ jest wtedy postaci $\sigma \to \rho$, i do tego wiemy jeszcze, że $\Gamma(y : \sigma) \vdash P : \rho$. Wystarczy oczywiście pokazać, że $M[\vec{x} := \vec{N}] \in [\![\sigma \to \rho]\!]$. Weźmy więc jakiś term $Q \in [\![\sigma]\!]$ i rozpatrzmy aplikację $M[\vec{x} := \vec{N}]Q = (\lambda y.P)[\vec{x} := \vec{N}]Q = (\lambda y.P[\vec{x} := \vec{N}])Q$. Przyjęliśmy tu oczywiście, że zmiennna y nie jest wolna w żadnym z termów $\vec{N}$. Oznacza to w szczególności, że $P[\vec{x} := \vec{N}][y := Q] = P[\vec{x} := \vec{N}, y := Q]$. Ten ostatni term jest stabilny dla ρ, co wiemy z założenia indukcyjnego. Zatem z lematu 14.9 wynika, że $M[\vec{x} := \vec{N}]Q$ też jest stabilny dla ρ. Pokazaliśmy więc ostatecznie, że $M[\vec{x} := \vec{N}]$ jest stabilny dla $\sigma \to \rho$. ∎

Twierdzenie 14.11 (silna normalizacja) *Jeśli* $\Gamma \vdash M : \tau$ *dla pewnych* Γ *i* τ *to term* M *jest silnie normalizowalny.*

Dowód: Wystarczy przyjąć $n = 0$ w treści lematu 14.10. ∎

Ćwiczenia

1. Udowodnić silną normalizację dla termów w stylu Churcha.
2. Udowodnić twierdzenie Churcha-Rossera dla termów w stylu Churcha, korzystając z twierdzeń 3.1 i 13.5 oraz ćwiczenia 1.

15 Izomorfizm Curry'ego-Howarda

Typy proste można utożsamiać z formułami zdaniowymi zbudowanymi z atomów (zmiennych zdaniowych) za pomocą samej implikacji. Ta analogia nie jest przypadkowa, bo reguły wyprowadzania typów odpowiadają regułom wnioskowania:

$$(\text{Ax})\ \Gamma, \sigma \vdash \sigma$$

$$(\mathrm{W}\to)\ \frac{\Gamma, \sigma \vdash \tau}{\Gamma \vdash \sigma \to \tau} \qquad (\mathrm{E}\to)\ \frac{\Gamma \vdash \sigma \to \tau \quad \Gamma \vdash \sigma}{\Gamma \vdash \tau}$$

Reguły te otrzymaliśmy przez *wytarcie* z reguł systemu $\lambda_\to$ wszystkiego co dotyczy termów i pozostawienie tylko informacji o typach. Oczywiście symbol Γ, wystepujący w regułach wnioskowania, oznacza po prostu zbiór formuł. Logika określona tymi regułami to *minimalna logika implikacyjna*.

Jeśli Γ jest otoczeniem typowym, to przez $|\Gamma|$ oznaczymy zbiór formuł $\{\Gamma(x) \mid x \in \mathrm{Dom}(\Gamma)\}$. Następujące twierdzenie wyraża w uproszczeniu odpowiedniość nazywaną często *izomorfizmem Curry'ego-Howarda*.

Twierdzenie 15.12

- *Jeśli* $\Gamma \vdash M : \tau$ *w systemie* $\lambda_\to$*, to* $|\Gamma| \vdash \tau$ *w logice minimalnej.*
- *Jeśli* $\Gamma \vdash \tau$ *w logice minimalnej, to istnieje takie otoczenie typowe* Δ*, że* $|\Delta| = \Gamma$ *oraz* $\Delta \vdash M : \tau$ *dla pewnego* M *w systemie* $\lambda_\to$*.*

Dowód: Łatwa indukcja. ∎

W sensie ogólnym, izomorfizm Curry'ego-Howarda to właśnie ścisła odpowiedniość pomiędzy

- formułami i typami;
- dowodami i termami (programami).

Powstaje pytanie czy logika minimalna to to samo co implikacyjny fragment logiki klasycznej. Otóż nie. Następująca klasyczna tautologia, zwana *prawem Peirce'a*, nie jest twierdzeniem logiki minimalnej:

$$\pi = ((p \to q) \to p) \to p$$

Można to łatwo stwierdzić, korzystając z prostej obserwacji: jeśli istnieje term M typu τ, to istnieje też taki term w postaci normalnej. Pozostaje więc zbadać, jak może wyglądać zamknięty term typu π w postaci normalnej i przekonać się, że nijak.

Na czym polega różnica? Logika minimalna jest *konstruktywna*. Implikacja w tej logice jest traktowana jak typ pewnej funkcji. Dowód formuły postaci $\alpha \to \beta$ musi polegać na wskazaniu metody przekształcenia każdego potencjalnego dowodu ("konstrukcji") założenia α w poprawny dowód formuły β. Nie ma innego kryterium prawdy oprócz dowodu, w szczególności nie wystarczy odwołanie do dwuwartościowej semantyki.

Kombinatory z typami

Jak zauważyliśmy, termy rachunku lambda z typami prostymi odpowiadają ściśle dowodom w minimalnej logice implikacyjnej. Chodzi tu o dowody w systemie *naturalnej dedukcji*. Pojęcie dowodu formalnego często jednak kojarzy się nam nie z naturalną dedukcją, ale z podejściem Hilberta. To podejście jest bowiem zwykle prezentowane w podręcznikach logiki. Dla

klasycznego implikacyjnego rachunku zdań, hilbertowski system dowodzenia można oprzeć na jednej regule wnioskowania (*modus ponens*)

$$\frac{\alpha,\ \alpha \to \beta}{\beta}$$

i trzech następujących schematach aksjomatów.

- $\alpha \to \beta \to \alpha$;
- $(\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$;
- $((\alpha \to \beta) \to \alpha) \to \alpha$.

Jak już wiemy, trzeci z tych aksjomatów, prawo Peirce'a, nie jest twierdzeniem logiki minimalnej, ale pierwsze dwa tak. Są to mianowicie typy znanych nam termów zamkniętych:

- $\mathbf{K} = \lambda xy.x$;
- $\mathbf{S} = \lambda xyz.xz(yz)$.

Okazuje się, że jeśli z powyższego systemu odrzucimy trzeci aksjomat a pozostawimy dwa pierwsze, to otrzymamy system równoważny logice minimalnej. Piszemy $\Gamma \vdash_H \varphi$ gdy formuła φ ma dowód w takim systemie Hilberta ze zbioru dodatkowych założeń Γ. Równoważność naszego systemu Hilberta i naturalnej dedukcji wynika z następującego twierdzenia:

Twierdzenie 15.13 (o dedukcji) *Warunki* $\Gamma \vdash_H \varphi \to \psi$ *i* $\Gamma, \varphi \vdash_H \psi$ *są równoważne.*

Dowód: Oczywiście wszyscy znają dowód twierdzenia o dedukcji, ale przypomnijmy go[13] z powodów metodologicznych. Dowód ten w swojej trudniejszej części — z prawej do lewej — przebiega przez indukcję ze względu na długość dowodu formuły ψ ze zbioru założeń Γ, φ.

Rozpatrzmy najpierw kilka łatwych przypadków.

Pierwszy łatwy przypadek mamy wtedy, gdy w dowodzie nie użyto w ogóle założenia φ. Tak jest w szczególności wtedy, kiedy ψ jest aksjomatem lub $\psi \in \Gamma$, a dowód składa się tylko z tej jednej formuły. Inaczej mówiąc mamy w istocie dowód $\Gamma \vdash_H \psi$. Wtedy dowód formuły $\varphi \to \psi$ jest otrzymany przez oderwanie ψ od aksjomatu $\psi \to (\varphi \to \psi)$.

Jeśli $\psi = \varphi$, to wystarczy udowodnić, że $\vdash_H \varphi \to \varphi$, co zostawiamy jako ćwiczenie.

Pozostaje główny krok indukcyjny, gdy dowód formuły ψ ze zbioru Γ, φ otrzymano przez odrywanie. Znaczy to, że $\Gamma, \varphi \vdash \vartheta \to \psi$ i $\Gamma, \varphi \vdash \vartheta$ dla pewnej formuły ϑ, i że odpowiednie dowody są krótsze. Z założenia indukcyjnego otrzymujemy, że formuły $\varphi \to (\vartheta \to \psi)$ i $\varphi \to \vartheta$ mają dowody ze zbioru Γ. Aby otrzymać dowód dla $\varphi \to \psi$, należy te dwie formuły kolejno oderwać od drugiego aksjomatu w postaci $(\varphi \to (\vartheta \to \psi)) \to ((\varphi \to \vartheta) \to (\varphi \to \psi))$. ∎

[13] W istocie nasz dowód nieznacznie różni się od tego, który wystepuje w większości podręczników.

Wniosek 15.14 *Osąd* $\Gamma \vdash \varphi$ *jest wyprowadzalny w systemie naturalnej dedukcji wtedy i tylko wtedy, gdy* $\Gamma \vdash_H \varphi$.

Dowód: W obie strony mamy prostą indukcję. Twierdzenie 15.13 jest nam potrzebne w dowodzie części ($\Rightarrow$) w przypadku wprowadzania implikacji. ∎

Dla wnioskowania w stylu Hilberta też można mówić o odpowiedniości Curry'ego-Howarda, ale nie chodzi już teraz o rachunek lambda. Dowody w stylu Hilberta otrzymujemy wyłącznie przez stosowanie reguły *modus ponens*, która jak wiemy odpowiada aplikacji. Aksjomaty logiczne systemu hilbertowskiego można uważać za stałe odpowiednich typów. Dokładniej, dla dowolnych typów α, β, γ możemy postulować stałe $\mathsf{K}_{\alpha\beta}$ i $\mathsf{S}_{\alpha\beta\gamma}$, którym przypisujemy w dowolnym otoczeniu typy:

- $\vdash \mathsf{K}_{\alpha\beta} : \alpha \to \beta \to \alpha$;
- $\vdash \mathsf{S}_{\alpha\beta\gamma} : (\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma$.

Inna możliwość, to założyć, że w systemie są tylko dwie stałe K i S, którym można (w dowolnym otoczeniu) przypisać każdy z typów odpowiedniej postaci. W ten sposób otrzymamy *rachunek kombinatorów z typami prostymi*, a wniosek 15.14 będziemy teraz mogli odczytać tak:

(∗) *Typy niepuste w rachunku lambda i rachunku kombinatorów z typami prostymi są takie same.*

Uderzające jest podobieństwo pomiędzy dowodem twierdzenia 15.13 i definicją kombinatorycznej abstrakcji λ^* (ćwiczenie 8). Ciekawe jest to, że ta analogia sięga znacznie dalej niż tylko zgodność typów. Można powiedzieć, że twierdzenie o dedukcji ma sens wykraczający poza język formuł logicznych.

Typologia ogólna: spójniki logiczne

Odpowiedniość pomiędzy formułami i typami byłaby niepełna, gdyby nie rozciągała się na spójniki logiczne inne niż implikacja. W istocie nietrudno jest zinterpretować w tym duchu zarówno koniunkcję jak i alternatywę. Zacznijmy od przypomnienia reguł naturalnej dedukcji dla tych spójników.

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ (W\wedge) \qquad \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi}\ (E\wedge)\ \frac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ (W\vee)\ \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} \qquad \frac{\Gamma \vdash \varphi \vee \psi \quad \Gamma, \varphi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta}\ (E\vee)$$

Reguły te możemy objaśniać w myśl tzw. *interpretacji BHK* (od nazwisk: Brouwer, Heyting, Kołmogorow).

- Dowód koniunkcji składa się z dowodów jej składowych, jest więc (uporządkowaną) parą dowodów;

- Dowód alternatywy to dowód jednego z jej członów (ze wskazaniem którego).

A zatem koniunkcja odpowiada iloczynowi kartezjańskiemu (rekordowi) a alternatywa sumie prostej (wariantowi). Wprowadzanie koniunkcji to tworzenie pary, a eliminacja koniunkcji to rzutowanie.

$$\frac{\Gamma \vdash M : \varphi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N\rangle : \varphi \wedge \psi}\ (W\wedge) \qquad \frac{\Gamma \vdash M : \varphi \wedge \psi}{\Gamma \vdash \pi_1(M) : \varphi}\ (E\wedge)\ \frac{\Gamma \vdash M : \varphi \wedge \psi}{\Gamma \vdash \pi_2(M) : \psi}$$

Wprowadzanie alternatywy odpowiada tworzeniu obiektu wariantowego. Eliminacja alternatywy to instrukcja wyboru.

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \mathbf{inl}(M) : \varphi \vee \psi}\ (W\vee) \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathbf{inr}(M) : \varphi \vee \psi}$$

$$\frac{\Gamma \vdash M : \varphi \vee \psi \quad \Gamma(x : \varphi) \vdash P : \vartheta \quad \Gamma(y : \psi) \vdash Q : \vartheta}{\Gamma \vdash \mathbf{case}\ M\ \mathbf{of}\ [x]P, [y]Q : \vartheta}\ (E\vee)$$

Dla tak rozszerzonego jezyka możemy teraz postulować następujące redukcje.

Beta redukcje:

$$\pi_1(\langle M, N\rangle) \to M,\ \pi_2(\langle M, N\rangle) \to N;$$
$$\mathbf{case}\ \mathbf{inl}(P)\ \mathbf{of}\ [x]M, [y]N \quad \to \quad M[x := P];$$
$$\mathbf{case}\ \mathbf{inr}(Q)\ \mathbf{of}\ [x]M, [y]N \quad \to \quad N[y := Q].$$

Eta redukcje:

$$\langle \pi_1(M), \pi_2(M)\rangle \to M; \qquad (\mathbf{case}\ M\ \mathbf{of}\ [x]\mathbf{inl}\ x, [y]\mathbf{inr}\ y) \to M.$$

Mówimy tu o beta i eta redukcjach przez analogię z redukcjami dla implikacji. Beta redeks odpowiada wprowadzeniu spójnika, po którym natychmiast następuje jego eliminacja. Natomiast postulat eta-redukcji odpowiada założeniu, że każde wyrażenie danego typu opisuje pewien *kanoniczny obiekt* tego typu (funkcję, parę, wariant — ogólnie rezultat operacji wprowadzenia).

Pozostaje sprawa negacji, którą jednak możemy zinterpretować jako specyficzną implikację:

$$\neg\alpha = \alpha \to \bot$$

Symbol $\bot$ oznacza fałsz, który oczywiście nie może mieć dowodu. Nie ma więc kanonicznych obiektów typu $\bot$ — jest to typ pusty. Mamy jednak regułę eliminacji fałszu:

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \varepsilon_\varphi(M) : \varphi}\ (E\bot)$$

Symbol ε_φ oznacza cud typu φ. Skoro mamy *rzecz, której nie ma*, to znaczy, że możemy z niej zrobić co zechcemy.

Reguły wprowadzania i eliminacji spójników tworzą system naturalnej dedukcji dla zdaniowej *logiki intuicjonistycznej*. Logika minimalna to jej fragment implikacyjny. Zauważmy, że nie ma wśród naszych reguł zasady wnioskowania przez zaprzeczenie, np. takiej:

$$\frac{\Gamma, \neg\varphi \vdash \bot}{\Gamma \vdash \varphi}$$

Dlatego twierdzeniami logiki intuicjonistycznej nie jest żadna z formuł postaci $p \vee \neg p$, $\neg\neg p \to p$, $(\neg p \to \neg q) \to q \to p$, prawo Peirce'a, ani wiele innych praw logiki klasycznej.

Ćwiczenia

1. Udowodnić twierdzenie 15.12.
2. Udowodnić że nie istnieje kombinator typu $\pi = ((p \to q) \to p) \to p$.
3. Pokazać, że **K** jest jedynym termem zamkniętym w postaci normalnej, który ma typ $s \to t \to s$ (gdzie s i t są zmiennymi typowymi). Pokazać analogiczną własność dla **S**.
4. Wyprowadzić $\vdash_H \alpha \to \alpha$. *Wskazówka: Zbudować ze stałych* K *i* S *term takiego typu.*
5. Sformułować reguły przypisania typów dla rachunku kombinatorów z typami prostymi i udowodnić stwierdzenie (*).
6. Udowodnić silną normalizację dla rachunku kombinatorów z typami prostymi.
7. Znaleźć typy dla kombinatorów $\mathsf{W}, \mathsf{B}, \mathsf{B}'$ i C.
8. Udowodnić, że w rachunku kombinatorów z typami prostymi warunek $\Gamma, x : \tau \vdash M : \sigma$ implikuje $\Gamma \vdash \lambda^* x.M : \tau \to \sigma$.

16 Problemy decyzyjne

Zgodnie z izomorfizmem Curry'ego-Howarda, twierdzenia intuicjonistycznej logiki implikacyjnej to dokładnie typy zamkniętych termów rachunku lambda (lub logiki kombinatorycznej). Ponieważ każdy term typowalny ma postać normalną, więc dla ustalenia, czy istnieje term danego typu wystarczy szukać takiego termu w postaci normalnej. Prowadzi to do następującego *algorytmu Ben-Yellesa*, który rozwiązuje nieco ogólniej postawione zadanie:

- Dane są otoczenie Γ i typ τ;
- Szukamy takiego termu M, że $\Gamma \vdash M : \tau$;

Zadanie to można rozbić na dwa przypadki:

(1) Jeśli $\tau = \tau_1 \to \tau_2$, to można zakładać, że M musi być abstrakcją postaci $\lambda x.M'$. (Jeśli term M nie jest abstrakcją, to zamiast M można wziąć term $\lambda x.Mx$, gdzie x jest nowe.) Należy więc znaleźć taki term M' aby $\Gamma, x : \tau_1 \vdash M' : \tau_2$.

(2) Jeśli τ jest zmienną typową s, to term M musi być aplikacją postaci $xM_1 \dots M_k$, gdzie $\Gamma(x) = \sigma_1 \to \dots \to \sigma_k \to s$. Należy więc znaleźć termy $M_1, \dots, M_k$ spełniające warunki $\Gamma \vdash M_1 : \sigma_1, \dots, \Gamma \vdash M_k : \sigma_k$.

Ponieważ w przypadku (2) możemy mieć do wyboru więcej niż jedną zmienną o typie kończącym się na s, więc nasz algorytm jest w istocie niedeterministycznym algorytmem rekurencyjnym, lub jak kto woli — algorytmem alternującym.[14]

Oczywiście, jeśli „inhabitant" typu τ istnieje, to prędzej czy później znajdziemy go tą metodą. Ale może być tak, że nasz algorytm się zapętli — weźmy na przykład typ $(s \to s) \to s$, albo $((s \to t) \to t) \to s \to t$. Jeśli więc rozwiązanie zadania dla danych Γ i τ prowadzi ponownie do tego samego zadania, przerywamy obliczenie z wynikiem negatywnym. Aby uniknąć sytuacji, w której pojawia się nieskończenie wiele różnych zadań, musimy też nieco poprawić działanie algorytmu w przypadku (1).

(1a) Jeśli $\tau = \tau_1 \to \tau_2$, oraz w otoczeniu Γ nie ma zmiennej typu τ_1, to szukamy takiego M' aby $\Gamma, x : \tau_1 \vdash M' : \tau_2$.

(1b) Jeśli $\tau = \tau_1 \to \tau_2$, oraz w otoczeniu Γ jest zmienna typu τ_1, to szukamy takiego M' aby $\Gamma \vdash M' : \tau_2$.

Poprawność przypadku (1b) wynika z następującej prostej obserwacji:

$$\textit{Jeśli } \Gamma(y) = \tau_1 \textit{ oraz } \Gamma, x : \tau_1 \vdash M' : \tau_2, \textit{ to } \Gamma \vdash M'[x := y] : \tau_2.$$

Inaczej mówiąc, wystarczy po jednej zmiennej każdego typu. Liczba typów, które mogą się pojawić w obliczeniu jest proporcjonalna do rozmiaru zadania, a otoczenie stale rośnie. Zatem głębokość rekursji jest wielomianowa (kwadratowa), a stąd wynika, że cały algorytm jest w klasie PSPACE. I nie da się go istotnie ulepszyć.

Twierdzenie 16.1 (Statman) *Implikacyjna logika intuicjonistyczna jest* PSPACE*-zupełna. A zatem problem niepustości dla typów prostych jest też* PSPACE*-zupełny.*

Dowód: Redukujemy problem kwantyfikowanych formuł Boole'owskich (QBF), czyli klasyczną logikę zdaniową drugiego rzędu, do intuicjonistycznego implikacyjnego rachunku zdań.[15]

Niech Φ będzie formułą w języku QBF. Bez straty ogólności możemy założyć, że jest to zdanie, i że negacja występuje w Φ tylko w kontekstach postaci $\neg p$, gdzie p jest zmienną zdaniową. Dla higieny załóżmy jeszcze, że wszystkie zmienne związane w Φ są różne.

Dla dowolnej zmiennej zdaniowej p, która występuje w formule Φ, niech s_p i $s_{\neg p}$ będą nowymi zmiennymi typowymi. Podobnie, dla każdej podformuły φ formuły Φ wybierzmy nowe zmienne typowe s_φ i $s_{\neg\varphi}$. Przez Γ_Φ oznaczmy zbiór złożony z następujących formuł:

- $(s_p \to s_\psi) \to (s_{\neg p} \to s_\psi) \to s_\varphi$, dla każdej podformuły φ postaci $\forall p\psi$;
- $(s_p \to s_\psi) \to s_\varphi$ i $(s_{\neg p} \to s_\psi) \to s_\varphi$, dla każdej podformuły φ postaci $\exists p\psi$;
- $s_\psi \to s_\vartheta \to s_\varphi$, dla każdej podformuły φ postaci $\psi \wedge \vartheta$;

[14] Krok egzystencjalny to wybór zmiennej x, krok uniwersalny to wybór jednego z termów M_i.
[15] Jak widać, logika minimalna wcale nie jest taka minimalna.

- $s_\psi \to s_\varphi$ i $s_\vartheta \to s_\varphi$, dla każdej podformuły φ postaci $\psi \vee \vartheta$.

Jeśli teraz v jest wartościowaniem zerojedynkowym o skończonej dziedzinie, to Γ_v oznacza Γ_Φ rozszerzone, dla wszystkich $p \in \mathrm{Dom}(v)$, o zmienne

- s_p, gdy $v(p) = 1$;
- $s_{\neg p}$, gdy $v(p) = 0$,

Teraz przez łatwą indukcję ze względu na długość podformuły φ, dowodzimy, że

$$\Gamma_v \vdash s_\varphi \quad \text{wtedy i tylko wtedy, gdy} \quad v(\varphi) = 1,$$

jeśli v jest określone na zmiennych wolnych (ale nie związanych) formuły φ. W szczególności formuła Φ jest prawdziwa wtedy i tylko wtedy, gdy $\Gamma_\Phi \vdash s_\Phi$. Pozostaje zauważyć, że warunek $\Gamma_\Phi \vdash s_\Phi$ to to samo co $\gamma_1 \to \cdots \to \gamma_n \to s_\Phi$ dla odpowiednich γ_i, oraz, że całą konstrukcję można przeprowadzić w pamięci logarytmicznej. ∎

Rekonstrukcja typu

Oczywiście nie każdy term jest typowalny w systemie $\lambda_\to$, nawet jeśli jest w postaci normalnej. Przykładem jest choćby $\lambda x.xx$. Problemem *typowalności* (typability) (albo problemem *rekonstrukcji typu*) dla danego systemu przypisania typów nazywamy taki problem decyzyjny:

Czy dany term M jest typowalny?

Zbliżonym zagadnieniem jest problem *sprawdzenia typu* (type checking):

Dane są M, Γ i τ. Czy $\Gamma \vdash M : \tau$?

Problem typowalności dla rachunku z typami prostymi jest równoważny (ze względu na redukcje w LOGSPACE) problemowi unifikacji dla języka pierwszego rzędu z jedną operacją binarną $\to$, i dlatego mamy następujący fakt:

Twierdzenie 16.2 *Problem typowalności dla $\lambda_\to$ jest zupełny w klasie* P. ∎

Problem sprawdzenia typu dla $\lambda_\to$ też jest w klasie P. Inny dowód rozstrzygalności obu problemów polega na sprowadzeniu ich do szukania cyklu w pewnym skończonym grafie.

Konsekwencją redukcji typowania do unifikacji jest następująca *własność typu głównego*. Powiemy, że para (Γ, τ) jest *instancją* pary (Γ_0, τ_0), gdy (Γ, τ) jest otrzymane z (Γ_0, τ_0) przez podstawienie pewnych typów w miejsce zmiennych typowych.

Twierdzenie 16.3 *Jeśli term M jest typowalny, to istnieje takie otoczenie Γ_0 i typ τ_0, że dla dowolnych Γ, τ:*

$$\Gamma \vdash M : \tau \qquad \Leftrightarrow \qquad \textit{para } (\Gamma, \tau) \textit{ jest instancją pary } (\Gamma_0, \tau_0).$$

Para (Γ_0, τ_0), o której mowa w twierdzeniu, jest nazywana *parą główną* dla M, a typ τ_0 to oczywiście *typ główny* tego termu.

Równość

Rozpatrzmy term $\mathbf{2}\cdots\mathbf{2}xy$, w którym występuje n dwójek. Nietrudno się przekonać, że postacią normalną tego termu jest $x(x(\cdots(xy)\cdots))$, gdzie liczbą wystąpień zmiennej x jest

$$2^{2^{\cdot^{\cdot^{2}}}\}n}$$

Najprostszy algorytm sprawdzający, czy dwa termy tego samego typu są beta-równe, polega na obliczeniu i porównaniu ich postaci normalnych. Jak widać z powyższego przykładu rozmiar postaci normalnej może być bardzo duży w stosunku do rozmiaru danego termu. Nawet jeśli uwzględnimy rozmiary użytych *typów* sytuacja zmieni się niewiele. Zauważmy bowiem, że typy, których potrzebujemy aby wywnioskować

$$x : t \to t, y : t \vdash \mathbf{2}\cdots\mathbf{2}xy : t$$

są „zaledwie" wykładniczego rozmiaru (każda kolejna dwójka ma dwa razy dłuższy typ). A zatem nasz naiwny algorytm ma nieelementarną złożoność. Co gorsza, nie można go istotnie poprawić.

Twierdzenie 16.4 (Statman) *Problem równości termów w rachunku lambda z typami prostymi nie jest problemem elementarnym (nie istnieje algorytm rozwiązujący ten problem w czasie* $2^{2^{\cdot^{\cdot^{2^n}}}\}k}$ *dla żadnego ustalonego* k*).* ∎

Co innego jednak porównać dwa termy, a co innego znaleźć term, lub termy spełniające zadane równanie. Rozwiązywanie równań z niewiadomymi dowolnych typów skończonych nazywamy *unifikacją wyższego rzędu*. Aby ściśle sformułować problem unifikacji przyjmijmy, że dane jest otoczenie Γ i para termów (M, N), które w tym otoczeniu mają ten sam typ. Wśród zmiennych deklarowanych w Γ wyróżnimy *niewiadome* $x_1, \ldots, x_k$ a wszystkie pozostałe zmienne nazwiemy *parametrami*. *Rozwiązaniem* równania $M = N$ o niewiadomych $x_1, \ldots, x_k$ nazywamy dowolne termy $P_1, \ldots, P_k$ spełniające warunki

- $\Gamma \vdash P_i : \Gamma(x_i)$ dla $i = 1, \ldots, k$;
- $M[x_1 := P_1, \ldots, x_k := P_k] =_{\beta\eta} N[x_1 := P_1, \ldots, x_k := P_k]$.

Problem unifikacji wyższego rzędu to następujący problem decyzyjny: czy istnieje rozwiązanie danego równania? Zwykle zakłada się, że wszystkie typy występujące w zadaniu są zbudowane z jednego atomu. Nie zmienia to istotnie stopnie trudności zadania. Jeśli typy wszystkich niewiadomych są postaci $t \to t \to \cdots \to t \to t$ lub t, gdzie t jest typem atomowym, to mówimy o unifikacji *drugiego rzędu*. Zwykłą unifikację nazywamy też unifikacją *pierwszego rzędu* (niewiadome mają typ atomowy).

Przykład 16.5

Oto dwa przykłady unifikacji drugiego rzędu z parametrami $f : t \to t \to t$, $a : t$, $b : t$ i niewiadomymi $G : t \to t \to t$, $F : t \to t$.

- $Ga(fba) = fa(Gab)$;
- $fa(Fa) = F(faa)$.

Pierwszy przykład nie ma rozwiązania, drugi ma ich nieskończenie wiele. Rozwiązaniami są wszystkie termy postaci $\lambda x.fa(fa(fa(\cdots(fax)\cdots)))$.

Twierdzenie 16.6 (Goldfarb) *Unifikacja drugiego rzędu jest nierozstrzygalna.* ∎

Dowód twierdzenia Goldfarba polega na redukcji Dziesiątego Problemu Hilberta do unifikacji drugiego rzędu. Dla dowolnego wielomianu konstruuje się takie równanie unifikacyjne, które ma rozwiązanie wtedy i tylko wtedy, gdy dany wielomian ma zero całkowite.

Szczególnym przypadkiem unifikacji jest *dopasowanie* (matching). *Problem dopasowania wyższego rzędu* polega na rozwiazaniu równania, w którym niewiadome występuja tylko po lewej stronie. Od dawna wiadomo było, że dopasowanie rzędu czwartego jest rozstrzygalne, aż wreszcie w 2007 roku C. Stirling ogłosił, że problem jest rozstrzygalny w ogólności. Jeśli w zadaniu dopasowania zmienić $=_{\beta\eta}$ na $=_{\beta}$, to problem okazuje się być nierozstrzygalny. Udowodnił to kilka... (już dwadzieścia!) lat temu Ralph Loader.

Funkcje definiowalne

Jak pamiętamy, w beztypowym rachunku lambda można reprezentować wszystkie funkcje rekurencyjne, a nawet częściowo rekurencyjne (twierdzenie 7.9). W rachunku lambda z typami prostymi nie należy się spodziewać podobnego wyniku, bo równość termów jest rozstrzygalna. W istocie klasa funkcji definiowalnych w rachunku $\lambda_{\to}$ jest dość uboga. Ale musimy zacząć od odpowiedniej definicji.

Niech σ będzie dowolnym typem. Przez ω_σ oznaczamy typ $(\sigma \to \sigma) \to \sigma \to \sigma$. Jeśli σ jest nieistotne (zwykle jest to ustalony atom), to piszemy po prostu ω. Oczywiście wszystkie liczebniki Churcha mają typ ω, więc naturalna jest następująca definicja:

Funkcja $f : \mathbb{N}^k \to \mathbb{N}$ jest *β-definiowalna* (lub po prostu *definiowalna*) w rachunku lambda z typami prostymi, w typie ω_σ, jeżeli istnieje term zamknięty F, spełniający następujące warunki:

- $\vdash F : \omega_\sigma \to \cdots \to \omega_\sigma \to \omega_\sigma$;
- Dla dowolnych $n_1, \ldots, n_k \in \mathbb{N}$, jeżeli $f(n_1, \ldots, n_k) = m$ to $F\mathbf{n}_1 \ldots \mathbf{n}_k =_\beta \mathbf{m}$.

Zamieniając w powyższej definicji znak $=_\beta$ na $=_{\beta\eta}$ otrzymujemy klasę funkcji *$\beta\eta$-definiowalnych*.

Następnik, dodawanie i mnożenie są przykładami funkcji definiowalnych w $\lambda_{\to}$ (zob. przykład 7.3). Możemy też zdefiniować funkcję warunkową

$$\mathit{ifzero}(p, q, r) = \begin{cases} q, & \text{jeśli } p = 0; \\ r, & \text{w przeciwnym przypadku.} \end{cases}$$

za pomocą termu

$$\mathbf{ifzero} = \lambda pqr\lambda fx.p(\lambda y.rfx)(qfx).$$

Jeśli zostaniemy przy równości β to więcej zdefiniować nam się nie uda.

Twierdzenie 16.7 (Schwichtenberg) *Dla dowolnego σ klasa funkcji β-definiowalnych w $\lambda_\to$ w typie ω_σ to dokładnie klasa* wielomianów warunkowych*, tj. najmniejsza klasa funkcji nad $\mathbb{N}$, zamknięta ze względu na składanie i zawierająca:*

- *rzutowania;*
- *dodawanie;*
- *mnożenie;*
- *funkcje stałe 0 i 1;*
- *funkcję warunkową ifzero.* ∎

Z twierdzenia Schwichtenberga wynika, w szczególności, że wybór typu σ nie ma znaczenia. Kiedyś uważano, że tak samo jest w przypadku $\beta\eta$-definiowalności, ale Mateusz Zakrzewski pokazał, że np. funkcja

$$ifeven(p,q,r) = \begin{cases} q, & \text{jeśli } p \text{ jest parzyste;} \\ r, & \text{w przeciwnym przypadku,} \end{cases}$$

która nie jest wielomianem warunkowym, jest $\beta\eta$-definiowalna (gdy liczebniki interpretujemy w odpowiednio dobranym typie ω_σ). Patrz ćwiczenie 10.

Nawet jednak z użyciem $\beta\eta$-konwersji, tak proste funkcje jak poprzednik, odejmowanie i potęgowanie nie są definiowalne. Poprzednik i potęgę uda nam się zdefiniować jeśli jeszcze bardziej osłabimy nasze wymagania. Powiemy, że funkcja $f : \mathbb{N}^k \to \mathbb{N}$ jest *skośnie definiowalna* w $\lambda_\to$, jeśli istnieje term F spełniający warunki:

- $\vdash F : \omega_{\sigma_1} \to \cdots \to \omega_{\sigma_k} \to \omega_\sigma$;
- Dla dowolnych $n_1, \ldots, n_k \in \mathbb{N}$, jeżeli $f(n_1, \ldots, n_k) = m$ to $F\mathbf{n}_1 \ldots \mathbf{n}_k =_\beta \mathbf{m}$.

Okazuje się, że poprzednik i potęgowanie są definiowalne skośnie, ale już odejmowanie nie jest. Klasę funkcji arytmetycznych definiowalnych w skończonych typach może istotnie powiekszyć dopiero dodanie do systemu „prawdziwego” operatora iteracji. W ten sposób otrzymujemy tzw. System **T** Gödla.

Ćwiczenia

1. Czy założenie o higienicznym rozdzieleniu zmiennych wolnych i związanych jest istotne w dowodzie twierdzenia 16.1?
2. Udowodnić twierdzenie 16.2.
3. Niech τ będzie typem, w którym występują tylko zmienne typowe $s_1, \ldots, s_n$ i niech Γ będzie takim otoczeniem, że $\Gamma(x_i) = s_i$ dla $i = 1, \ldots n$. Skonstruować taki term M_τ, że $\Gamma \vdash M : \sigma$ wtedy i tylko wtedy, gdy $\sigma = \tau$.
4. Niech $E = \{\tau_1 = \sigma_1, \ldots \tau_k = \sigma_k\}$ będzie instancją problemu unifikacji dla języka pierwszego rzędu z jedną operacją binarną $\to$ o niewiadomych $s_1, \ldots, s_n$. Skonstruować (w pamieci logarytmicznej) taki term M, który jest typowalny wtedy i tylko wtedy, gdy E ma rozwiązanie. *Wskazówka: Skorzystać z poprzedniego zadania.*

5. Zbadać rozwiązalność unifikacji z przykładu 16.5.
6. (Łatwe) Niech $c_n = \lambda x.fa(fa(fa(\cdots(fax)\cdots)))$, gdzie f wystepuje n razy. Skonstruować takie równanie drugiego rzędu, którego rozwiązaniami są dokładnie trójki termów postaci c_n, c_m, c_{n+m}.
7. (Trudne) Skonstruować takie równanie drugiego rzędu, którego rozwiązaniami są dokładnie trójki termów postaci $c_n, c_m, c_{n\cdot m}$.
8. (Łatwe, jeśli umiemy rozwiązać poprzednie dwa zadania.) Udowodnić twierdzenie 16.6.
9. Udowodnić, że poprzednik i potęgowanie są niejednostajnie definiowalne w skończonych typach. Dlaczego są kłopoty z odejmowaniem?
10. Niech $F = \lambda nfxa_1a_2a_3.\, n(\lambda yz_1z_2z_3.\, yz_2z_3z_1)(\lambda z_1z_2z_3.\, z_1)(xa_1a_2a_3)(fxa_1a_2a_3)(f(fx)a_1a_2a_3)$. Pokazać, że $F : \omega_\tau \to \omega_\tau$ dla pewnego τ. Jaką funkcję z $\mathbb{N}$ do $\mathbb{N}$ definiuje term F?
11. Wzorując się na poprzednim ćwiczeniu, zdefiniować funkcję $n \mapsto \min(n, 3)$.

17 Semantyka typów prostych

W teorii typów prostych często przyjmuje się dwa założenia, które istotnie upraszczają pewne konstrukcje. Po pierwsze, zamiast równości $=_\beta$ rozważa się ekstensjonalną równość $=_{\beta\eta}$, po drugie zakłada się, że wszystkie typy są zbudowane z jednego tylko typu bazowego. My też teraz przyjmiemy te założenia. A więc typy definiujemy tak:

- Stała 0 jest typem;
- Jeśli σ i τ są typami, to $\sigma \to \tau$ jest typem.

Niech T oznacza zbiór wszystkich takich typów. Jeśli teraz $\{D_\sigma\}_{\sigma\in T}$ jest rodziną niepustych zbiorów, to możemy rozważać *wartościowania*, które zmiennym typu σ przypisują elementy D_σ. (Wygodnie jest zakładać, że nasze termy są w ortodoksyjnym stylu Churcha.) Rozważamy wielosortowe struktury postaci $\mathcal{A} = \langle \{D_\sigma\}_{\sigma\in T}, \{\cdot_{\sigma\tau}\}_{\sigma,\tau\in T}, [\![\,]\!]^{\mathcal{A}} \rangle$, gdzie dla dowolnych $\sigma, \tau \in T$:

- D_σ jest niepustym zbiorem;
- $d \cdot_{\sigma\tau} e \in D_\tau$ dla $d \in D_{\sigma\to\tau},\ e \in D_\sigma$;
- $[\![M]\!]^{\mathcal{A}}_v \in D_\sigma$, dla dowolnego M typu σ.

Oczywiście zamiast $\cdot_{\sigma\tau}$ będziemy pisać zwykłą kropkę, a zamiast $[\![\,]\!]^{\mathcal{A}}$ napiszemy $[\![\,]\!]$. Taka struktura jest (ekstensjonalnym) *modelem*, gdy spełnia następujące warunki:

- Jeśli $d, d' \in D_{\sigma\to\tau}$ oraz $d \cdot e = d' \cdot e$ dla dowolnego $e \in D_\sigma$, to $d = d'$.
- Jeśli x jest zmienną, to $[\![x]\!]_v = v(x)$;
- $[\![PQ]\!]_v = [\![P]\!]_v \cdot [\![Q]\!]_v$;
- $[\![\lambda x^\sigma P]\!]_v \cdot a = [\![P]\!]_{v[x\mapsto a]}$, dla dowolnego $a \in D_\sigma$.

Założenie ekstensjonalności powoduje, że powyższe warunki w istocie jednoznacznie definiują funkcję $[\![\ ]\!]$ dla danego $\langle \{D_\sigma\}_{\sigma\in T}, \{\cdot_{\sigma\tau}\}_{\sigma,\tau\in T}\rangle$, o ile taka funkcja istnieje (tj. istnieją wszystkie elementy potrzebne do zinterpretowania abstrakcji). Z ekstensjonalności wynika też przez łatwą indukcję pominięty w definicji[16] warunek:

- Jeśli $v|_{\mathrm{FV}(P)} = u|_{\mathrm{FV}(P)}$, to $[\![P]\!]_v = [\![P]\!]_u$.

Następujący lemat jest odpowiednikiem lematu 10.2. Dowodzimy go przez indukcję ze względu na budowę termów

Lemat 17.1 *W dowolnym modelu zachodzi tożsamość* $[\![M[x := N]]\!]_v = [\![M]\!]_{v[x\mapsto [\![N]\!]_v]}$.

Piszemy $\mathcal{A}, v \models M = N$, gdy $[\![M]\!]_v = [\![N]\!]_v$. Dalej, $\mathcal{A} \models M = N$ oznacza, że $\mathcal{A}, v \models M = N$ dla dowolnego v, natomiast napis $\models M = N$ mówi, że jest tak w każdym modelu. Zaczynamy od łatwego twierdzenia o poprawności.

Fakt 17.2 *Jeśli* $M =_{\beta\eta} N$ *to* $\models M = N$.

Dowód: Indukcja ze względu na definicję $=_{\beta\eta}$, z użyciem lematu 17.1. Istotne równości:

- $[\![(\lambda x.P)Q]\!]_v = [\![\lambda x.P]\!]_v \cdot [\![Q]\!]_v = [\![P]\!]_{v[x\mapsto [\![Q]\!]_v]} = [\![P[x := Q]]\!]_v$.
- $[\![\lambda x.Px]\!]_v \cdot a = [\![Px]\!]_{v[x\mapsto a]} = [\![P]\!]_{v[x\mapsto a]} \cdot a = [\![P]\!]_v \cdot a$, gdy $x \notin \mathrm{FV}(P)$. ∎

Nas, tak naprawdę, interesują tylko modele, w których $D_{\sigma\to\tau}$ to po prostu zbiór wszystkich funkcji z D_σ do D_τ. Jeśli $D_0 = A$, to taki model oznaczamy przez $\mathfrak{M}(A)$. Dla dowodu twierdzenia o pełności potrzebujemy jednak jeszcze jednego przykładu. Przez $\mathfrak{M}_\eta$ oznaczymy model, w którym dziedzina D_σ składa się ze wszystkich termów typu σ, branych z dokładnością do $\beta\eta$-konwersji (tj. w istocie z klas abstrakcji). W modelu $\mathfrak{M}_\eta$ znaczenie termów jest określone przez podstawienie, tj. dla $\mathrm{FV}(M) = \{x_1, \ldots, x_n\}$ mamy

$$[\![M]\!]_v = M[x_1, \ldots, x_n := v(x_1), \ldots, v(x_n)].$$

Nietrudno teraz pokazać twierdzenie odwrotne do Faktu 17.2, czyli najłatwiejszą wersję twierdzenia o pełności.

Fakt 17.3 *Jeśli* $\mathfrak{M}_\eta \models M = N$ *to* $M =_{\beta\eta} N$. *Zatem* $\models M = N$ *implikuje* $M =_{\beta\eta} N$.

W dalszym ciągu pokażemy twierdzenie o pełności dla modeli postaci $\mathfrak{M}(A)$. Wymaga to jednak pewnych przygotowań.

Definicja 17.4 *Częściowy epimorfizm* z modelu $\mathcal{A} = \langle \{D_\sigma\}_{\sigma\in T}, \{\cdot_{\sigma\tau}\}_{\sigma,\tau\in T}, [\![\ ]\!]\rangle$ do modelu $\mathcal{B} = \langle \{E_\sigma\}_{\sigma\in T}, \{\cdot_{\sigma\tau}\}_{\sigma,\tau\in T}, [\![\ ]\!]\rangle$, to rodzina częściowych surjekcji $\phi_\sigma : D_\sigma \overset{\text{na}}{\multimap\!\!\to} E_\sigma$, zachowująca

[16] Por. analogiczną definicję dla modeli bez typów.

aplikację, tj. spełniająca warunek $\phi_\tau(d \cdot e) = \phi_{\sigma\to\tau}(d) \cdot \phi_\sigma(e)$. Ściślej, żądamy aby wartość $\phi_{\sigma\to\tau}(d)$ była określona i równa h wtedy i tylko wtedy, gdy dla każdego $e \in \mathrm{Dom}(\phi_\sigma)$ określone jest $\phi_\tau(d\cdot e)$ i zachodzi równość[17] $\phi_\tau(d\cdot e) = h \cdot \phi_\sigma(e)$. Poniżej zamiast $\phi_\sigma(d)$ często będziemy pisać $\overline{d}$. Na przykład zamiast $\phi_\tau(d \cdot e) = \phi_{\sigma\to\tau}(d) \cdot \phi_\sigma(e)$ napiszemy $\overline{d \cdot e} = \overline{d} \cdot \overline{e}$.

Lemat 17.5 *Jeśli $\{\phi_\sigma\}_{\sigma\in T}$ jest częściowym epimorfizmem z $\mathcal{A}$ do $\mathcal{B}$ oraz $\overline{v}(x) = \overline{v(x)}$ dla dowolnego x, to dla każdego M zachodzi $[\![M]\!]^{\mathcal{B}}_{\overline{v}} = \overline{[\![M]\!]^{\mathcal{A}}_v}$, w szczególności $\overline{[\![M]\!]^{\mathcal{A}}_v}$ jest określone.*

Dowód: Indukcja ze względu na M. Dla zmiennych teza wynika natychmiast z definicji $\overline{v}$. Dla aplikacji mamy $[\![PQ]\!]^{\mathcal{B}}_{\overline{v}} = [\![P]\!]^{\mathcal{B}}_{\overline{v}} \cdot [\![Q]\!]^{\mathcal{B}}_{\overline{v}} = \overline{[\![P]\!]^{\mathcal{A}}_v} \cdot \overline{[\![Q]\!]^{\mathcal{A}}_v} = \overline{[\![P]\!]^{\mathcal{A}}_v \cdot [\![Q]\!]^{\mathcal{A}}_v} = \overline{[\![PQ]\!]^{\mathcal{A}}_v}$. W przypadku abstrakcji pamiętajmy, że każdy element modelu $\mathcal{B}$ jest postaci $\overline{a}$. Mamy teraz

$$[\![\lambda x P]\!]^{\mathcal{B}}_{\overline{v}} \cdot \overline{a} = [\![P]\!]^{\mathcal{B}}_{\overline{v}[x\mapsto\overline{a}]} = [\![P]\!]^{\mathcal{B}}_{\overline{v[x\mapsto a]}} = \overline{[\![P]\!]^{\mathcal{A}}_{v[x\mapsto a]}} = \overline{[\![\lambda x P]\!]^{\mathcal{A}}_v \cdot a} = \overline{[\![\lambda x P]\!]^{\mathcal{A}}_v} \cdot \overline{a},$$

dla dowolnego $\overline{a}$, i stosujemy ekstensjonalność. Czytelnikowi pozostawiamy sprawdzenie, że $\overline{[\![M]\!]^{\mathcal{A}}_v}$ jest zawsze określone. ∎

Wniosek 17.6 *Jeśli $\mathcal{A} \models M = N$ i istnieje częściowy epimorfizm z $\mathcal{A}$ do $\mathcal{B}$ to $\mathcal{B} \models M = N$.*

Dowód: Bo każde wartościowanie w $\mathcal{B}$ ma postać $\overline{v}$. ∎

Lemat 17.7 *Jeśli w modelu $\mathcal{B} = \langle \{E_\sigma\}_{\sigma\in T}, \{\cdot_{\sigma\tau}\}_{\sigma,\tau\in T}, [\![\ ]\!] \rangle$ dziedzina E_0 jest przeliczalna, to dowolną surjekcję z $\mathbb{N}$ do E_0 można rozszerzyć do częściowego epimorfizmu z $\mathfrak{M}(\mathbb{N})$ do $\mathcal{B}$.*

Dowód: Oznaczmy przez D_σ dziedziny w modelu $\mathfrak{M}(\mathbb{N})$. Definiujemy $\phi_\sigma : D_\sigma \overset{\text{na}}{\multimap\!\!\to} E_\sigma$ przez indukcję ze względu na σ, przyjmując daną surjekcję z $\mathbb{N}$ na E_0 jako ϕ_0. Załóżmy, że ϕ_σ i ϕ_τ są określone. Dla dowolnego $h \in E_{\sigma\to\tau}$, istnieją funkcje częściowe $d : D_\sigma \to D_\tau$ o własności $\phi_\tau(d(e)) = h \cdot \phi_\sigma(e)$ (inaczej $\overline{d(e)} = h \cdot \overline{e}$) dla dowolnego $e \in D_\sigma$. Dla każdej takiej funkcji d należy przyjąć $\overline{d} = h$. ∎

Twierdzenie 17.8 (H. Friedman, 1975) *Warunki $M =_{\beta\eta} N$ i $\mathfrak{M}(\mathbb{N}) \models M = N$ są równoważne.*

Dowód: Na mocy lematu 17.7 mamy częściowy epimorfizm z $\mathfrak{M}(\mathbb{N})$ do modelu $\mathfrak{M}_\eta$. Jeśli więc $\mathfrak{M}(\mathbb{N}) \models M = N$ to $\mathfrak{M}_\eta \models M = N$ (lemat 17.5) a stąd $M =_{\beta\eta} N$. ∎

[17] Jedyność h wynika z ekstensjonalności.

Twierdzenie Statmana

Pokażemy teraz twierdzenie o pełności dla modeli postaci $\mathfrak{M}(A)$, gdzie A jest skończone. Przypomnijmy, że typ bazowy 0 jest *rzędu* 0, a typy postaci $0 \to \cdots \to 0 \to 0$ są *rzędu* 1. O zmiennej typu τ powiemy, że jest *rzędu* 0 lub *rzędu* 1, gdy takiego rzędu jest typ τ.

Lemat 17.9 *Załóżmy, że term $Q : 0$ jest w postaci normalnej, i że wszystkie jego zmienne wolne są rzędu co najwyżej* 1. *Wtedy istnieje taka stała k, że dla dowolnego $N : 0$ zachodzi:*

$$\textit{Jeśli}\ \ \mathfrak{M}(k) \models Q = N \ \ \textit{to}\ \ Q =_{\beta\eta} N.$$

Dowód: Zauważmy najpierw, że term Q nie może zawierać żadnych abstrakcji (jest zbudowany jak zwykły term „algebraiczny" w którym zmienne rzędu 1 odgrywają rolę symboli funkcyjnych).

Ponumerujmy wszystkie podtermy termu Q, które są typu 0, ustawiając je w ciąg skończony $t_1, t_2, t_3, \ldots, t_{k-1}$ (bez powtórzeń). Możemy teraz określić wartościowanie v w modelu $\mathfrak{M}(\mathbb{N})$ w ten sposób, że $[\![t_i]\!]_v = i$ dla każdego $i = 1, \ldots, k-1$, oraz $[\![N]\!]_v = 0$ dla każdego innego termu $N : 0$ w postaci normalnej. Wtedy wartość $[\![Q]\!]_v$ jest jedną z liczb $1, \ldots, k-1$, powiedzmy $k-1$. Przyporządkowanie

$$\phi_0(i) = \begin{cases} i, & \text{jeśli } i = 1, \ldots, k-1; \\ 0, & \text{w przeciwnym przypadku,} \end{cases}$$

jest surjekcją z $\mathbb{N}$ do $k = \{0, \ldots, k-1\}$, a zatem rozszerza się do częściowego epimorfizmu z modelu $\mathcal{A} = \mathfrak{M}(\mathbb{N})$ do modelu $\mathcal{B} = \mathfrak{M}(k)$. Mamy teraz $\overline{[\![Q]\!]}_v^{\mathcal{A}} = \overline{k-1} = k-1$, i jeśli $N \neq_{\beta\eta} Q$, to z lematu 17.5 wynika $[\![N]\!]_v^{\mathcal{B}} = \overline{[\![N]\!]}_v^{\mathcal{A}} \neq k-1 = \overline{[\![Q]\!]}_v^{\mathcal{A}} = [\![Q]\!]_v^{\mathcal{B}}$, czyli $\mathfrak{M}(k), v \not\models Q = N$. ∎

Uogólnienie lematu 17.9 na typy wyższych rzędów wymaga dwóch lematów o charakterze syntaktycznym. Mówimy, że term $M : \sigma_1 \to \cdots \to \sigma_n \to 0$ jest *w postaci Statmana*, gdy

$$M = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.x(M_1 x_1 \ldots x_n) \ldots (M_m x_1 \ldots x_n),$$

gdzie $M_1, \ldots, M_m$ są w postaci Statmana, oraz $x_1, \ldots, x_n \notin \mathrm{FV}(M_i)$. Łatwo widzieć, że każdy term jest beta-eta-równy termowi w postaci Statmana.

Lemat 17.10 *Dla wszystkich typów σ istnieją termy U^σ typu σ, których zmienne wolne są rzędu co najwyżej jeden, i które mają taką własność: Jeśli dwa termy zamknięte M, N typu $\sigma_1 \to \cdots \to \sigma_n \to 0$ w postaci Statmana są różnej długości, to $MU^{\sigma_1} \ldots U^{\sigma_n} \neq_{\beta\eta} NU^{\sigma_1} \ldots U^{\sigma_n}$.*

Dowód: Termy U^σ definiujemy przez indukcję, wraz z termami $V^\sigma : \sigma \to 0$. Najpierw bierzemy $U^0 = z_0$ (gdzie z_0 jest nową zmienną) oraz $V^0 = \lambda x^0 z_0$. Dla typu σ postaci $\sigma = \sigma_1 \to \cdots \to \sigma_n \to 0$ przyjmujemy

$$U^\sigma = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.z_\sigma(V^{\sigma_1} x_1) \ldots (V^{\sigma_n} x_n);$$

$$V^\sigma = \lambda x^\sigma.\, xU^{\sigma_1} \ldots U^{\sigma_n},$$

gdzie zmienna z_σ jest znowu świeża (inna dla różnych σ).

Przypuśćmy teraz, że $M, N : \sigma_1 \to \cdots \to \sigma_n \to 0$ oraz $MU^{\sigma_1} \ldots U^{\sigma_n} =_{\beta\eta} NU^{\sigma_1} \ldots U^{\sigma_n}$. Jeśli $M = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.x_i(M_1x_1 \ldots x_n) \ldots (M_mx_1 \ldots x_n)$, to lewa strona równości jest $\beta\eta$-równa termowi $U^{\sigma_i}(M_1U^{\sigma_1} \ldots U^{\sigma_n}) \ldots (M_mU^{\sigma_1} \ldots U^{\sigma_n})$. Przyjmując $\sigma_i = \tau_1 \to \cdots \to \tau_m \to 0$, mamy dalej $MU^{\sigma_1} \ldots U^{\sigma_n} =_{\beta\eta} z_{\sigma_i}(V^{\tau_1}(M_1U^{\sigma_1} \ldots U^{\sigma_n})) \ldots (V^{\tau_m}(M_mU^{\sigma_1} \ldots U^{\sigma_n}))$, co z kolei jest $\beta\eta$-równe wyrażeniu $z_{\sigma_i}(M_1U^{\sigma_1} \ldots U^{\sigma_n}\vec{U}_1) \ldots (M_mU^{\sigma_1} \ldots U^{\sigma_n}\vec{U}_m)$, w którym wektory $\vec{U}_1, \ldots, \vec{U}_m$ są takie jak w termach $V^{\tau_1}, \ldots, V^{\tau_m}$.

Podobnie, jeśli $N = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.x_j(N_1x_1 \ldots x_n) \ldots (N_rx_1 \ldots x_n)$, to $NU^{\sigma_1} \ldots U^{\sigma_n}$ zredukuje się do termu $z_{\sigma_j}(N_1U^{\sigma_1} \ldots U^{\sigma_n}\vec{U}^1) \ldots (N_rU^{\sigma_1} \ldots U^{\sigma_n}\vec{U}^r)$, gdzie $\sigma_j = \rho_1 \to \cdots \to \rho_r \to 0$.

Skoro te termy są $\beta\eta$-równe, to przede wszystkim musi się zgadzać zmienna czołowa $z_{\sigma_j} = z_{\sigma_i}$. A więc $\sigma_j = \sigma_i$ skąd $m = r$. Dalej $M_lU^{\sigma_1} \ldots U^{\sigma_n}\vec{U}_l =_{\beta\eta} N_lU^{\sigma_1} \ldots U^{\sigma_n}\vec{U}_l$, dla $l \leq m$ i możemy zastosować indukcję, bo termy M_i są krótsze niż M. (Uwaga: teraz już wiemy, że wektor $\vec{U}_l$ jest po obu stronach taki sam.) ∎

Lemat 17.11 *Jeśli M, N są zamkniętymi termami typu $\sigma = \sigma_1 \to \cdots \to \sigma_n \to 0$, oraz $M \neq_{\beta\eta} N$, to istnieją termy $V_1^{\sigma_1}, \ldots V_n^{\sigma_n}$, których zmienne wolne są rzędu co najwyżej jeden, takie że $MV_1 \ldots V_n \neq_{\beta\eta} NV_1 \ldots V_n$.*

Dowód: Indukcja ze względu na M. Można założyć, że M i N są w postaci Statmana:

$$M = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.x_i(M_1x_1 \ldots x_n) \ldots (M_mx_1 \ldots x_n);$$
$$N = \lambda x_1^{\sigma_1} \ldots x_n^{\sigma_n}.x_j(N_1x_1 \ldots x_n) \ldots (N_rx_1 \ldots x_n).$$

Jeśli $i \neq j$, to sprawa jest prosta: wystarczy wybrać $V_i = \lambda y_1 \ldots y_m.\, z_1$ i $V_j = \lambda y_1 \ldots y_r.\, z_2$, gdzie z_1 i z_2 to dwie różne zmienne. Niech więc $i = j$ (oraz $m = r$). Skoro $M \neq_{\beta\eta} N$, to $M_\ell \neq_{\beta\eta} N_\ell$ dla pewnego $\ell \leq m$. Niech $\sigma_i = \tau_1 \to \cdots \to \tau_m \to 0$, oraz $\tau_\ell = \rho_1 \to \cdots \to \rho_d \to 0$. Z założenia indukcyjnego są termy $W_1^{\sigma_1}, \ldots, W_n^{\sigma_n}, W_{n+1}^{\rho_1}, \ldots, W_{n+d}^{\rho_d}$, o zmiennych wolnych rzędu co najwyżej 1, takie że $M_\ell W_1 \ldots W_{n+d} \neq_{\beta\eta} N_\ell W_1 \ldots W_{n+d}$.

Dla $k \neq i$ przyjmujemy $V_k = W_k$. Aby określić V_i, przyjmijmy, że $W_i = \lambda y_1 \ldots y_m.W$. Wtedy

$$V_i = \lambda y_1 \ldots y_m.zW(y_\ell W_{n+1} \ldots W_{n+d}),$$

gdzie z jest nową zmienną typu $0 \to 0 \to 0$. Wówczas

$$MV_1 \ldots V_n =_{\beta\eta} V_i(M_1V_1 \ldots V_n) \ldots (M_mV_1 \ldots V_n) =_{\beta\eta} zW'(M_\ell V_1 \ldots V_n W_{n+1} \ldots W_{n+d}),$$

gdzie $W' = W[y_1, \ldots, y_m := M_1V_1 \ldots V_n, \ldots, M_mV_1 \ldots V_n]$. Podobnie zredukuje się term $NV_1 \ldots V_n$, jeśli więc $MV_1 \ldots V_n =_{\beta\eta} NV_1 \ldots V_n$, to w szczególności

$$M_\ell V_1 \ldots V_n W_{n+1} \ldots W_{n+d} =_{\beta\eta} N_\ell V_1 \ldots V_n W_{n+1} \ldots W_{n+d}.$$

Wektor $V_1 \ldots V_n$ różni się od wektora $W_1 \ldots W_n$ tylko na i-tej współrzędnej, a wolna zmienna z występuje tylko w termie V_i. Podstawiając w termie V_i na miejsce zmiennej z kombinator **K** otrzymamy więc fałszywą równość $M_\ell W_1 \ldots W_{n+d} =_{\beta\eta} N_\ell W_1 \ldots W_{n+d}$. ∎

Twierdzenie 17.12 (R. Statman, 1982) *Dla dowolnego termu M istnieje taka liczba k (efektywnie obliczalna z M), że jeśli N jest dowolnym termem, to:*

$$M =_{\beta\eta} N \quad \textit{wtedy i tylko wtedy gdy} \quad \mathfrak{M}(k) \models M = N.$$

Dowód: Załóżmy, że M jest typu σ. Z lematów 17.10 i 17.11 wynika, że istnieją takie termy $L_1, \ldots, L_m : \sigma \to 0$, że dla dowolnego N mamy

$$\text{Jeśli } \; M \neq_{\beta\eta} N \; \text{ to } \; L_i M \neq_{\beta\eta} L_i N, \text{ dla pewnego } i = 1, \ldots, m.$$

Ponadto typy zmiennych wolnych w $L_1, \ldots, L_m$ są rzędu co najwyżej 1. Niech Q będzie postacią normalną termu $z(L_1 M) \ldots (L_m M)$, gdzie z jest nową zmienną. Do termu Q można zastosować lemat 17.9, weźmy więc odpowiednie k. Jeśli $\mathfrak{M}(k) \models M = N$, to wtedy także $\mathfrak{M}(k) \models Q = z(L_1 N) \ldots (L_m N)$, skąd $Q =_{\beta\eta} z(L_1 N) \ldots (L_m N)$ i dalej $L_i M =_{\beta\eta} L_i N$ dla wszystkich i. W konsekwencji $M =_{\beta\eta} N$. ∎

Wniosek 17.13 (S. Sołowiow, 1981) *Termy M i N są beta-eta-równe wtedy i tylko wtedy, gdy są równe we wszystkich modelach skończonych.*

Istotne w twierdzeniu Statmana jest to, że liczba k nie zależy od N. Przykładem zastosowania tw. Statmana jest niemożność niejednostajnego reprezentowania równości liczb naturalych w rachunku z typami prostymi. (Podobnie można pokazać, że nie można niejednostajnie reprezentować odejmowania.) Nie jest mi znany syntaktyczny dowód tego faktu.

Wniosek 17.14 *Nie istnieje term $E : \omega_\tau \to \omega_\sigma \to \omega_\rho$, taki, że dla dowolnych $p, q \in \mathbb{N}$:*

$$E\mathbf{p}^{\omega_\tau}\mathbf{q}^{\omega_\sigma} =_{\beta\eta} \mathbf{0}^{\omega_\rho} \quad \textit{wtedy i tylko wtedy, gdy} \quad p = q.$$

Dowód: Dobieramy k do $M = \mathbf{0}^{\omega_\rho}$ z twierdzenia Statmana. Dziedzina D_{ω_τ} w modelu $\mathfrak{M}(k)$ jest skończona, więc istnieją takie liczby $p \neq q$, że $[\![\mathbf{p}^{\omega_\tau}]\!] = [\![\mathbf{q}^{\omega_\tau}]\!]$. Wtedy $[\![E\mathbf{p}^{\omega_\tau}\mathbf{q}^{\omega_\sigma}]\!] = [\![E]\!][\![\mathbf{p}^{\omega_\tau}]\!][\![\mathbf{q}^{\omega_\sigma}]\!] = [\![E]\!][\![\mathbf{q}^{\omega_\tau}]\!][\![\mathbf{q}^{\omega_\sigma}]\!] = [\![E\mathbf{q}^{\omega_\tau}\mathbf{q}^{\omega_\sigma}]\!] = [\![\mathbf{0}^{\omega_\rho}]\!]$, czyli $\mathfrak{M}(k) \models E\mathbf{p}^{\omega_\tau}\mathbf{q}^{\omega_\sigma} = \mathbf{0}^{\omega_\rho}$. Zatem $p = q$ wbrew założeniu. ∎

Nierozstrzygalność definiowalności

Przez pewien czas otwartym problemem była tzw. hipoteza Plotkina o rozstrzygalności definiowalności w skończonych modelach. Hipoteza okazała się nieprawdziwa.

Twierdzenie 17.15 (R. Loader, 1993) *Następujący problem jest nierozstrzygalny:*

Dany jest skończony zbiór X, typ τ i element $d \in D_\tau(X)$.
Czy istnieje taki kombinator M typu τ, że $[\![M]\!] = d$?

Ogólniejsza wersja tego problemu jest taka: dane są elementy $e_1 \in D_{\sigma_1}(X), \ldots, e_n \in D_{\sigma_n}(X)$ i pytamy, czy istnieje takie M, że $[\![M]\!]_v = d$, gdzie $v(x_1) = e_1, \ldots, v(x_n) = e_n$. Mówimy wtedy, że d jest *definiowalne z* $e_1, \ldots e_n$. Otóż wystarczy udowodnić nierozstrzygalność problemu w tej wersji. Faktycznie, gdyby problem definiowalności był rozstrzygalny, to moglibyśmy przeglądać wszystkie funkcje $f : D_{\sigma_1}(X) \to \cdots \to D_{\sigma_n}(X) \to D_\tau(X)$ spełniające warunek $f e_1 \ldots e_n = d$ i sprawdzać, czy któraś z nich jest definiowalna.

Dowód Loadera polega na redukcji nierozstrzygalnego problemem słów dla systemów półthue'owskich nad alfabetem $\{a, b\}$. Przypomnijmy, że system półthue'owski to skończony zbiór reguł postaci $C \Rightarrow D$, gdzie $C, D \subseteq \{a, b\}^*$, i że reguła $C \Rightarrow D$ pozwala słowo xCy przepisać w jednym kroku w słowo xDy. Problem słów to pytanie czy dane słowo w można w wielu krokach przepisać w słowo v.

Sprowadzimy to pytanie do uogólnionego problemu definiowalności w modelu o dziedzinie bazowej $X = \{a, b, L, R, *, 1, 0\}$. W tym celu zakodujemy każde słowo $w = o_1 \dots o_n$ jako funkcję $\overline{w} : D_0^n \to D_0$, określoną tak:

– $\overline{w}(* \cdots * o_i * \cdots *) = 1$, gdy o_i znajduje się na pozycji i;

– $\overline{w}(* \cdots * LR * \cdots *) = 1$;

– $\overline{w}(\dots) = 0$, w pozostałych przypadkach.

Dla $S \subseteq D_0^n$ niech $\chi[S] : D_0^n \to D_0$ będzie funkcją określoną tak:

$$\chi[S](\vec{o}) = \begin{cases} 1, & \text{jeśli } \vec{o} \in S; \\ 0, & \text{w przeciwnym przypadku.} \end{cases}$$

Teraz każdą regułę $F = (C \Rightarrow D)$ zakodujemy jako funkcję $\overline{F} : (D_0^m \to D_0) \to (D_0^n \to D_0)$, gdzie $m = |C|$ i $n = |D|$. Funkcja $\overline{F}$ zachowuje się tak:

– $\overline{F}(\chi[\{* \cdots *\}]) = \chi[\{* \cdots *\}]$;

– $\overline{F}(\chi[\{R * \cdots *\}]) = \chi[\{R * \cdots *\}]$;

– $\overline{F}(\chi[\{* \cdots * L\}]) = \chi[\{* \cdots * L\}]$;

– $\overline{F}(\overline{C}) = \overline{D}$;

– $\overline{F}(g) = \chi[\varnothing]$, dla każdej innej funkcji g.

Główna własność tej konstrukcji jest taka: słowo v można otrzymać w skończonej liczbie kroków ze słowa w wtedy i tylko wtedy, gdy kod $\overline{v}$ jest definiowalny w modelu z kodu $\overline{w}$ i kodów reguł systemu.

Dowód implikacji z lewej do prawej jest łatwiejszy. Przypuśćmy, że $w = w_1 C w_2$ przepisujemy w jednym kroku w słowo $v = w_1 D w_2$, z pomocą reguły $F = (C \Rightarrow D)$. Jeśli term W definiuje element $\overline{w}$, to $\overline{v}$ można zdefiniować termem

$$V = \lambda \vec{x} \vec{y} \vec{z}.\, \overline{F}(\lambda \vec{u}.\, W \vec{x} \vec{u} \vec{z}) \vec{y}, \tag{*}$$

gdzie $|\vec{y}| = |C|$, $|\vec{u}| = |D|$, $|\vec{x}| = |w_1|$ i $|\vec{z}| = |w_2|$.

Przyjemność sprawdzenia poprawności tej definicji pozostawiona jest czytelnikowi. Jeszcze większą przyjemność powinno mu sprawić przekonanie się, że nie ma innej metody zdefiniowania elementu postaci $\overline{v}$ jak składanie operacji (*), co dowodzi trudniejszej implikacji z prawej do lewej.

Ćwiczenia

1. Pokazać, że $\mathfrak{M}_\eta$ jest modelem, i że $\mathfrak{M}_\eta \models M = N$ zachodzi wtedy i tylko wtedy gdy $M =_{\beta\eta} N$.
2. Rodzinę relacyj $\{\sim_\sigma\}_{\sigma \in T}$, odpowiednio w $\{D_\sigma\}_{\sigma \in T}$, nazywamy *relacją logiczną*, jeżeli dla dowolnych typów σ i τ i dowolnych $d, d' \in D_{\sigma \to \tau}$ zachodzi równoważność (zamiast $\sim_\sigma$ piszemy $\sim$):

$$d \sim d' \quad \text{wtedy i tylko wtedy, gdy} \quad \forall e, e' \in D_\sigma (e \sim e' \to d \cdot e \sim d' \cdot e').$$

Udowodnić, że jeśli $v(x) \sim w(x)$ dla dowolnego $x \in \mathrm{FV}(M)$, to $[\![M]\!]_v \sim [\![M]\!]_w$.

3. Jeśli $\{\phi_\sigma\}_{\sigma \in T}$ jest częściowym epimorfizmem, oraz
$$d \sim d' \quad \text{wtedy i tylko wtedy, gdy} \quad \overline{d} = \overline{d'}$$
to $\{\sim_\sigma\}_{\sigma \in T}$ jest relacją logiczną.[18]
4. Zdefiniować wartościowanie v, o którym mowa w dowodzie lematu 17.9.
5. Uzasadnić istnienie termów L_i, o których mowa w dowodzie twierdzenia 17.12.
6. Zdefiniować pojęcie modelu dla rachunku z wieloma atomami typowymi. Czy twierdzenie Statmana pozostaje prawdziwe?
7. Uzupełnić szczegóły dowodu twierdzenia 17.15.

18 Semantyka w stylu Curry'ego

Zajmiemy się teraz semantyczną interpretacją osądów typowych w stylu Curry'ego. Musimy się w tym celu odwołać do semantyki beztypowej, bo mamy przecież do czynienia z termami „czystego" rachunku lambda. Przypuśćmy więc, że mamy lambda-model $\mathcal{D} = \langle D, \cdot, [\![\ ]\!] \rangle$. Podzbiory tego modelu reprezentują własności jego elementów, a także własności termów. Mówimy więc, że element $a \in D$ „ma własność σ", gdy $a \in \sigma \subseteq D$. Podobnie, powiemy że term zamknięty M ma własność σ, gdy $[\![M]\!] \in \sigma$. W przypadku termów ze zmiennymi wolnymi będzie to oczywiście zależało od wartościowania. Naturalna jest następująca definicja:

$$\sigma \Rightarrow \tau := \{a \in D \mid \forall b \in D (b \in \sigma \to a \cdot b \in \tau)\}.$$

Własność $\sigma \Rightarrow \tau$ to poprawność ze względu na prewarunek σ i postwarunek τ. Oczywista jest tu analogia z typem funkcyjnym. Nadamy jej ścisły sens, interpretując typy jako podzbiory modelu. Funkcję $\xi : TV \to \mathbf{P}(D)$ nazwiemy *wartościowaniem typowym*, a znaczenie $[\![\tau]\!]_\xi$ typu τ przy wartościowaniu ξ zdefiniujemy tak:

- $[\![s]\!]_\xi = \xi(s)$, gdy s jest zmienną typową;
- $[\![\sigma \to \tau]\!]_\xi = [\![\sigma]\!]_\xi \Rightarrow [\![\tau]\!]_\xi$.

Piszemy $\mathcal{D}, v, \xi \models M : \sigma$, gdy $[\![M]\!]_v \in [\![\sigma]\!]_\xi$. Powiemy wtedy, że term M ma typ σ w modelu $\mathcal{D}$ przy wartościowaniu v i wartościowaniu typowym ξ.

Podobnie, napis $\mathcal{D}, v, \xi \models \Gamma$ oznacza, że dla wszystkich $(x : \tau) \in \Gamma$ zachodzi $v(x) \in [\![\tau]\!]_\xi$.

Semantycznym odpowiednikiem osądu $\Gamma \vdash M : \sigma$ jest więc następująca własność:

Dla dowolnego modelu $\mathcal{D}$ i dowolnych v, ξ, warunek $\mathcal{D}, v, \xi \models \Gamma$ implikuje $\mathcal{D}, v, \xi \models M : \sigma$,

zapisywana $\Gamma \models M : \sigma$. Zachodzi teraz taki fakt:

Twierdzenie 18.1 (o poprawności) *Jeśli $\Gamma \vdash M : \sigma$, to $\Gamma \models M : \sigma$.*

[18] Uwaga: Relacja, o której tu mowa, jest *częściową relacją równoważności* tj. jest symetryczna i przechodnia, ale nie musi być zwrotna. W ogólności, relacja logiczna nie musi też być symetryczna ani przechodnia.

Dowód: Indukcja ze względu na wyprowadzenie $\Gamma \vdash M : \sigma$. Jest tu kilka przypadków, w zależności od tego jakiej reguły użyto w tym wyprowadzeniu jako ostatniej. Rozpatrzmy przypadek wyprowadzenia kończącego się zastosowaniem reguły (Abs). Wtedy σ jest postaci $\tau \to \rho$, a term M jest abstrakcją $\lambda x.N$. Natomiast konkluzję $\Gamma \vdash \lambda x.N : \tau \to \rho$ otrzymano z przesłanki $\Gamma, x : \tau \vdash N : \rho$.

Przypuśćmy, że $\mathcal{D}, v, \xi \models \Gamma$. Mamy pokazać, że $\mathcal{D}, v, \xi \models \lambda x.N : \tau \to \rho$, innymi słowy, że $[\![\lambda x.N]\!]_v \in [\![\tau \to \rho]\!]_\xi = [\![\tau]\!]_\xi \Rightarrow [\![\rho]\!]_\xi$. Niech więc $a \in [\![\tau]\!]_\xi$. Wtedy $\mathcal{D}, v[x \mapsto a], \xi \models \Gamma, x : \tau$, więc $[\![\lambda x.N]\!]_v \cdot a = [\![N]\!]_{v[x \mapsto a]} \in [\![\rho]\!]_\xi$, bo z założenia indukcyjnego $\Gamma, x : \tau \models N : \rho$. Pozostałe przypadki są natychmiastowe. ∎

Twierdzenie odwrotne (o pełności) nie zachodzi. Na przykład $\models \lambda x.\, \mathbf{K}x(\lambda y.\, yy) : p \to p$ oraz $\models \lambda x.(\lambda y.\, yy)\mathbf{I}x : p \to p$ chociaż $\not\vdash \lambda x.\, \mathbf{K}x(\lambda y.\, yy) : p \to p$ i $\not\vdash \lambda x.(\lambda y.\, yy)\mathbf{I}x : p \to p$. Przyczyną jest oczywiście to, że równość $=_\beta$, a więc też równość w modelu, nie zachowuje typów. Twierdzenie o pełności otrzymamy dla bardziej „elastycznego" systemu przypisania typów.

Ćwiczenia

1. Oznaczmy przez ω całą dziedzinę $\mathcal{D}$. Udowodnić, że dla dowolnych podzbiorów modelu $\mathcal{D}$ zachodzą związki:
$$\sigma \subseteq \omega, \quad \omega \subseteq \omega \Rightarrow \omega, \quad \sigma \cap \tau \subseteq \sigma, \quad \sigma \cap \tau \subseteq \tau, \quad \sigma \subseteq \sigma \cap \sigma$$
$$(\sigma \Rightarrow \tau) \cap (\sigma \Rightarrow \rho) \subseteq \sigma \Rightarrow \tau \cap \rho$$
Jeśli $\sigma \subseteq \sigma'$ i $\tau \subseteq \tau'$, to $\sigma \cap \tau \subseteq \sigma' \cap \tau'$ oraz $\sigma' \Rightarrow \tau \subseteq \sigma \Rightarrow \tau'$.
2. Jeśli model $\mathcal{D}$ jest częściowo uporządkowany, to naturalne jest żądanie, aby własności były zbiorami zamkniętymi w górę (jeśli $a \in \sigma$ i $a \leq b$, to $b \in \sigma$) lub były postaci $\uparrow a = \{b \mid a \leq b\}$. Pokazać, że wtedy $\sigma \Rightarrow \tau$ jest zamknięte w górę, oraz że $(\uparrow a \Rightarrow \uparrow b) = \uparrow f$ dla pewnego f.
3. Niech $\mathcal{D}$ będzie kratą. Jeśli każde z σ_i i τ_i jest postaci $\uparrow a$ to, dla skończonych I, z warunku $\bigcap\{\sigma_i \Rightarrow \tau_i \mid i \in I\} \subseteq \sigma \Rightarrow \tau$ wynika $\bigcap\{\tau_i \mid \sigma \subseteq \sigma_i\} \subseteq \tau$.

19 Typy iloczynowe

Zdefiniujemy teraz rachunek lambda z typami *iloczynowymi*,[19] które w odróżnieniu od typów prostych, zbudowane są z pomocą dwóch konstruktorów $\to$ i $\cap$. Zaczynamy od składni typów.

- Stała ω jest typem;
- Typy atomowe są typami;
- Jeśli σ i τ są typami, to $\sigma \to \tau$ i $\sigma \cap \tau$ są typami.

Zbiór wszystkich typów oznaczymy przez $\mathcal{T}_{\cap\omega}$. Przyjmujemy konwencję, że iloczyn ma wyższy priorytet niż strzałka. Iloczyn, jak się zaraz okaże, można w gruncie rzeczy uważać za łączny i pomijać nawiasy.

[19] Dawniej używane określenie „typy koniunkcyjne" wyszło z użycia, bo jest mylące. Obecnie uważamy je za niepoprawne. Odpowiednikiem logicznej koniunkcji nie jest iloczyn typów, ale produkt kartezjański.

W zbiorze $\mathcal{T}_{\cap\omega}$ określamy relację $\leq$ jako najmniejszy quasiporządek[20] spełniający takie warunki:

$$\sigma \leq \omega, \quad \omega \leq \omega \to \omega, \quad \sigma \cap \tau \leq \sigma, \quad \sigma \cap \tau \leq \tau, \quad \sigma \leq \sigma \cap \sigma$$

$$(\sigma \to \tau) \cap (\sigma \to \rho) \leq \sigma \to \tau \cap \rho$$

$$\text{Jeśli } \sigma \leq \sigma' \text{ i } \tau \leq \tau', \text{ to } \sigma \cap \tau \leq \sigma' \cap \tau' \text{ oraz } \sigma' \to \tau \leq \sigma \to \tau'.$$

Definicja powyżej motywowana jest własnościami operacji $\Rightarrow$ i $\cap$ (ćwiczenie 1 do rozdziału 18). Piszemy $\sigma \equiv \tau$, gdy zachodzi zarówno $\sigma \leq \tau$ jak i $\tau \leq \sigma$. Czasem takie typy są po prostu utożsamiane.

Określimy teraz reguły przypisania (wyprowadzania) typów iloczynowych dla termów rachunku lambda. Następujące aksjomaty i reguły tworzą system wnioskowania o typach, który nazwiemy *systemem* BCD od nazwisk Barendregt, Coppo i Dezani.

$$\text{(Var) } \Gamma\,(x:\sigma) \vdash x : \sigma \qquad (\omega)\ \Gamma \vdash M : \omega$$

$$\text{(Abs) } \frac{\Gamma\,(x:\sigma) \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \text{(App) } \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

$$(\cap\text{I}) \ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \cap \tau} \qquad (\leq) \ \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \tau}\ (\sigma \leq \tau)$$

Reguła ($\cap$I) jest zwana regułą *wprowadzania iloczynu*. Reguła (Abs) jest często nazywana regułą *wprowadzania strzałki*, a reguła (App) regułą *eliminacji strzałki*. *Eliminacja iloczynu*, to taki szczególny przypadek reguły ($\leq$):

$$(\cap\text{E}) \ \frac{\Gamma \vdash M : \sigma_1 \cap \sigma_2}{\Gamma \vdash M : \sigma_i}$$

Jeśli osąd $\Gamma \vdash M : \tau$ ma wyprowadzenie w systemie BCD, to piszemy $\Gamma \vdash_{\text{BCD}} M : \tau$ lub po prostu $\Gamma \vdash M : \tau$. Jeśli $\Gamma = \varnothing$, to zamiast $\Gamma \vdash M : \tau$ piszemy $\vdash M : \tau$, lub wręcz $M : \tau$.

Przykład: Następujące osądy są wyprowadzalne w systemie BCD:[21]

- $\vdash \lambda x.xx : t \cap (t \to s) \to s$;
- $\vdash \mathbf{2} : (t \to s) \cap (s \to r) \to (t \to r)$;
- $\vdash \mathbf{K} : t \to s \to t$;
- $\vdash \mathbf{S} : (t' \to s \to r) \to (t'' \to s) \to (t' \cap t'') \to r$.

Zauważmy, że typowanie termu zależy tylko od jego zmiennych wolnych.

[20] Quasiporządek to relacja zwrotna i przechodnia.

[21] Także wtedy, gdy zamiast reguły ($\leq$) używamy tylko słabszej reguły ($\cap$E).

Lemat 19.1 *Niech* $\Gamma \vdash M : \tau$ *i niech* $\Gamma|_{\mathrm{FV}(M)} = \{(x : \Gamma(x)) \mid x \in \mathrm{FV}(M) \text{ i } \Gamma(x) \text{ określone}\}$. *Wtedy* $\Gamma|_{\mathrm{FV}(M)} \vdash M : \tau$.

Dowód: Dowód jest przez łatwą indukcję ze względu na wyprowadzenie $\Gamma \vdash M : \tau$. ∎

Poprawność redukcji

Naszym celem jest teraz pokazanie, że typy termów zachowują się przy redukcjach. Niestety musimy zacząć od nieprzyjemnych lematów. Notacja $\bigcap S$, gdzie $S = \{\rho_1, \ldots, \rho_n\}$ oznacza iloczyn $\rho_1 \cap \cdots \cap \rho_n$.

Lemat 19.2 *Jeżeli* $\omega \not\equiv \rho$ *oraz* $\bigcap\{\sigma_i \to \tau_i \mid i \in I\} \leq \rho$ *to* $\rho = \bigcap\{\xi_j \to \zeta_j \mid j \in J\}$ *dla pewnego* J *i pewnych* ξ_j, ζ_j.

Dowód: Indukcja ze względu na definicję nierówności. ∎

Lemat 19.3 *Jeżeli* $\bigcap\{\sigma_i \to \tau_i \mid i \in I\} \leq \sigma \to \tau$, *oraz* $\sigma \to \tau \not\equiv \omega$, *to zbiór* $\{\tau_i \mid \sigma \leq \sigma_i\}$ *jest niepusty, oraz* $\bigcap\{\tau_i \mid \sigma \leq \sigma_i\} \leq \tau$.

Dowód: Indukcja ze względu na definicję nierówności. ∎

Lemat 19.4 *Jeśli* $\Gamma \vdash M : \sigma$ *i* $\tau \leq \Gamma(x)$, *to* $\Gamma(x : \tau) \vdash M : \sigma$.

Dowód: Łatwa indukcja ze względu na wyprowadzenie $\Gamma \vdash M : \sigma$. ∎

Lemat 19.5 (o generowaniu)

1. *Jeśli* $\Gamma \vdash MN : \sigma$, *to* $\Gamma \vdash M : \tau \to \sigma$ *i* $\Gamma \vdash N : \tau$ *dla pewnego* τ.
2. *Jeśli* $\Gamma \vdash x : \sigma$, *gdzie* x *jest zmienną, to* $\Gamma(x) \leq \sigma$.
3. *Jeśli* $\Gamma \vdash \lambda x.M : \sigma$ *i* $\sigma \neq \omega$ *to* $\sigma = \bigcap\{\sigma_i \to \tau_i \mid i \in I\}$ *dla pewnego* I *i pewnych* σ_i, τ_i, *przy czym* $\tau_i \not\equiv \omega$.
4. *Jeśli* $\Gamma \vdash \lambda x.M : \sigma \to \tau$, *to* $\Gamma\,(x{:}\sigma) \vdash M : \tau$.

Dowód: (1) Dowód jest przez indukcję ze względu na rozmiar wyprowadzenia $\Gamma \vdash MN : \sigma$ (liczbę wierzchołków w dowodzie). Mamy kilka przypadków w zależności od tego, która reguła została użyta w wyprowadzeniu jako ostatnia. Jeśli jest to reguła (App), to teza jest oczywista. Przypuśćmy, że jest to reguła ($\leq$). To znaczy, że konkluzję $\Gamma \vdash MN : \sigma$ otrzymaliśmy z wcześniej wyprowadzonego osądu $\Gamma \vdash MN : \sigma'$. Dowód tej ostatniej jest

mniejszy, więc możemy zastosować założenie indukcyjne. Otrzymujemy $\Gamma \vdash M : \tau \to \sigma'$ i $\Gamma \vdash N : \tau$. Ponieważ $\tau \to \sigma' \leq \tau \to \sigma$, więc $\Gamma \vdash M : \tau \to \sigma$ tak jak chcemy. Jeśli ostatnią zastosowaną regułą była reguła ($\cap$I), to mamy $\sigma = \sigma_1 \cap \sigma_2$ oraz $\Gamma \vdash MN : \sigma_1$ i $\Gamma \vdash MN : \sigma_2$, przy czym rozmiary tych wyprowadzeń są mniejsze. Z założenia indukcyjnego $\Gamma \vdash M : \tau_1 \to \sigma_1$, i $\Gamma \vdash M : \tau_2 \to \sigma_2$ oraz $\Gamma \vdash N : \tau_1$ i $\Gamma \vdash N : \tau_2$. Stąd wnioskujemy, że $\Gamma \vdash N : \tau_1 \cap \tau_2$ oraz $\Gamma \vdash M : (\tau_1 \to \sigma_1) \cap (\tau_2 \to \sigma_2)$. Ponieważ

$$(\tau_1 \to \sigma_1) \cap (\tau_2 \to \sigma_2) \leq (\tau_1 \cap \tau_2 \to \sigma_1) \cap (\tau_1 \cap \tau_2 \to \sigma_2) \leq (\tau_1 \cap \tau_2 \to \sigma_1 \cap \sigma_2),$$

więc ostatecznie $\Gamma \vdash M : \tau_1 \cap \tau_2 \to \sigma_1 \cap \sigma_2$ i też jest dobrze.

Ostatnią regułą nie może być ani reguła (Abs) ani (Var). Pozostaje więc tylko ta możliwość, że całe wyprowadzenie polega na przywołaniu aksjomatu (ω). Ale wtedy $\Gamma \vdash N : \omega$ i $\Gamma \vdash M : \omega$. Ponieważ $\omega \leq \omega \to \omega$, więc mamy też $\Gamma \vdash M : \omega \to \omega$.

(2) Dowód jest podobny, a nawet prostszy.

(3) Postępujemy przez podobną indukcję, korzystając z lematu 19.2. Zauważmy tu tylko, że warunek $\tau_i \not\equiv \omega$ wynika stąd, że $\sigma \to \omega \equiv \omega$ dla dowolnego σ. „Niedobre" człony iloczynu można więc pomijać.

(4) Dowód jest znowu przez indukcję ze względu na wyprowadzenie. Nieoczywisty przypadek jest tylko jeden: gdy konkluzję $\Gamma \vdash \lambda x.M : \sigma \to \tau$ otrzymano z $\Gamma \vdash \lambda x.M : \rho$ za pomocą reguły ($\leq$). Z części 3 wynika, że typ ρ jest postaci $\bigcap\{\sigma_i \to \tau_i \mid i \in I\}$. Możemy zastosować założenie indukcyjne do każdego członu tego iloczynu, otrzymując $\Gamma(x : \sigma_i) \vdash M : \tau_i$ dla wszystkich $i \in I$. Na mocy lematu 19.4 zachodzi też $\Gamma(x : \sigma) \vdash M : \tau_i$ dla wszystkich $i \in I$, dla których $\sigma \leq \sigma_i$. Stąd dalej wnioskujemy, że $\Gamma(x : \sigma) \vdash M : \bigcap\{\tau_i \mid \sigma \leq \sigma_i\}$ i do pełni szczęścia brakuje nam tylko nierówności $\bigcap\{\tau_i \mid \sigma \leq \sigma_i\} \leq \tau$. Ponieważ jednak $\rho \leq \sigma \to \tau$, więc nierówność ta wynika z lematu 19.3. ∎

Lemat 19.6 *Jeśli* $\Gamma(x : \sigma) \vdash M : \tau$ *oraz* $\Gamma \vdash N : \sigma$, *to* $\Gamma \vdash M[x := N] : \tau$.

Dowód: Dowód jest przez indukcję ze względu na budowę termu M i wykorzystuje lemat o generowaniu. Na przykład jeśli $M = PQ$ to $\Gamma, x : \sigma \vdash M : \tau$ implikuje $\Gamma, x : \sigma \vdash P : \rho \to \tau$ i $\Gamma, x : \sigma \vdash Q : \rho$. Można wiec zastosować założenie indukcyjne do P i Q, otrzymując $\Gamma \vdash P[x := N] : \rho \to \tau$ i $\Gamma \vdash Q[x := N] : \rho$, skąd łatwo wynika teza. Pozostałe przypadki są podobne (ćwiczenie). ∎

Twierdzenie 19.7 (poprawność redukcji) *Jeśli* $\Gamma \vdash M : \tau$ *oraz* $M \to_{\beta\eta} N$, *to* $\Gamma \vdash N : \tau$.

Dowód: Dowód jest przez indukcję ze względu na definicję redukcji. Przypadki nietrywialne mają miejsce gdy M jest beta- lub eta-redeksem.

Jeśli $M = (\lambda x.P)Q \to_\beta P[x := Q] = N$, to z lematu 19.5 otrzymujemy $\Gamma \vdash \lambda x.P : \sigma \to \tau$ i $\Gamma \vdash Q : \sigma$. Wtedy $\Gamma, x : \sigma \vdash P : \tau$ (znowu z lematu o generowaniu) i pozostaje użyć lematu 19.6.

Niech więc $M = \lambda x.Nx \to_\eta N$ (gdzie $x \notin \mathrm{FV}(M)$). Bez straty ogólności można założyć, że $\tau \not\equiv \omega$. Z lematu o generowaniu typ τ jest więc iloczynem postaci $\bigcap\{\sigma_i \to \tau_i \mid i \in I\}$

i na dodatek $\Gamma, x : \sigma_i \vdash Nx : \tau_i$ dla wszystkich $i \in I$. Stosując dalej lemat o generowaniu otrzymamy $\Gamma, x : \sigma_i \vdash N : \xi_i \to \tau_i$ oraz $\Gamma, x : \sigma_i \vdash x : \xi_i$. Wtedy $\sigma_i \leq \xi_i$, skąd wynika, że $\Gamma, x : \sigma_i \vdash N : \sigma_i \to \tau_i$. Z lematu 19.1 wnioskujemy, że $\Gamma \vdash N : \sigma_i \to \tau_i$, a skoro tak jest dla wszystkich i, to wreszcie $\Gamma \vdash N : \tau$. ∎

System typów iloczynowych BCD jest w tym wyjątkowy, że zachodzi następujące twierdzenie (nieprawdziwe dla większości innych systemów):

Twierdzenie 19.8 *Jeżeli $M \to_\beta N$, to warunki $\Gamma \vdash M : \tau$ i $\Gamma \vdash N : \tau$ są równoważne.*[22]

Dowód: Implikacja z lewej do prawej to oczywiście twierdzenie 19.7. Dla dowodu implikacji odwrotnej potrzebna nam taka własność:

$$\textit{Jeśli } \Gamma \vdash M[x := N] : \tau, \textit{ to istnieje taki typ } \sigma, \textit{ że } \Gamma \vdash N : \sigma \textit{ i } \Gamma, x : \sigma \vdash M : \tau,$$

czyli twierdzenie odwrotne do lematu 19.6. Dowód tej własności przebiega przez indukcję ze względu na budowę termu M, z pomocą lematu o generowaniu (lemat 19.5). Przypadek $M = y \neq x$ wymaga odwołania się do stałej ω. ∎

Wniosek 19.9 *Jeżeli $M =_\beta N$, to termy M i N mają te same typy w każdym otoczeniu.*

Ćwiczenia

1. Czy twierdzenie 19.7 pozostanie prawdziwe, gdy z definicji $\leq$ usuniemy warunek $\omega \leq \omega \to \omega$?
2. Wskazać przykłady termów, które nie mają żadnego typu prostego, ale mają typy iloczynowe, nie zawierające (a) stałej ω, (b) stałej ω ani iloczynu.
3. Czy z tego, że term M ma typ w otoczeniu Γ wynika, że $\Gamma(x)$ jest określone dla wszystkich zmiennych $x \in \mathrm{FV}(M)$?
4. Czy twierdzenie 19.8 uogólnia się dla η-redukcji?
5. Sformułować i udowodnić lemat o generowaniu dla systemu, w którym regułę $(\leq)$ zastąpiono regułą $(\cap\mathrm{E})$.
6. Typy iloczynowe *według Scotta* zbudowane są z dwóch stałych typowych ω i κ za pomocą spójników $\to$ i $\cap$ (nie ma zmiennych typowych). Relacja $\leq$ jest poprawiona przez dodanie nowego aksjomatu $\kappa = \omega \to \kappa$, reguły wnioskowania pozostają te same. Pokazać, że dla tak określonego systemu zachodzą lematy 19.2–19.6 i twierdzenia 19.7-19.8.

20 Model z filtrów

Wracamy do semantyki typów, ale tym razem iloczynowych. Znaczenie typów przy wartościowaniu typowym ξ definiujemy jak poprzednio, dodając jeszcze dwie klauzule.

[22] *Subject conversion property*

- $[\![s]\!]_\xi = \xi(s)$, gdy s jest zmienną typową;
- $[\![\omega]\!]_\xi = D$;
- $[\![\sigma \cap \tau]\!]_\xi = [\![\sigma]\!]_\xi \cap [\![\tau]\!]_\xi$;
- $[\![\sigma \to \tau]\!]_\xi = [\![\sigma]\!]_\xi \Rightarrow [\![\tau]\!]_\xi$.

Terminologia i notacja pozostaje taka jak dla typów prostych. Twierdzenie o poprawności pozostaje prawdziwe.

Lemat 20.1 *Jeśli $\sigma \leq \tau$, to $[\![\sigma]\!]_\xi \subseteq [\![\tau]\!]_\xi$ dla dowolnego ξ.*

Dowód: Ćwiczenie. ∎

Twierdzenie 20.2 (o poprawności) *Jeśli $\Gamma \vdash M : \sigma$, to $\Gamma \models M : \sigma$.*

Dowód: Podobny do dowodu twierdzenia 18.1. Przypadek reguły $(\leq)$ wynika wprost z lematu 20.1. ∎

Teraz zabieramy się za dowód pełności. Podzbiór $F \subseteq \mathcal{T}$ nazywamy *filtrem*, jeżeli spełnione są takie warunki:

- F jest niepusty;
- Jeżeli $\sigma, \tau \in F$, to $\sigma \cap \tau \in F$;
- Jeżeli $\sigma \in F$ i $\sigma \leq \tau$, to $\tau \in F$.

Przykładem filtru jest każdy zbiór postaci $\sigma\uparrow = \{\tau \mid \sigma \leq \tau\}$, nazywany *filtrem głównym*. W zbiorze $\mathbf{F}$ wszystkich filtrów określamy operację aplikacji:

$$F_1 \cdot F_2 = \{\tau \mid \sigma \to \tau \in F_1 \text{ dla pewnego } \sigma \in F_2\}.$$

Wartościowanie $v : V \to \mathbf{F}$ przypisuje każdej zmiennej x pewien filtr $v(x)$. Nieformalnie, jest to filtr złożony z typów „dozwolonych" dla x. Mówimy, że otoczenie Γ jest *zgodne* z wartościowaniem v, gdy $\Gamma(x) \in v(x)$ dla wszystkich x, dla których $\Gamma(x)$ jest określone.

Definiujemy teraz znaczenie termu M w zbiorze $\mathbf{F}$ (przy wartościowaniu v), jako zbiór typów „dozwolonych" dla M:

$$[\![M]\!]_v = \{\sigma \mid \Gamma \vdash M : \sigma, \text{ dla pewnego } \Gamma \text{ zgodnego z } v\}.$$

W następnym dowodzie (i nie tylko) przyda się następująca definicja. Jeśli Γ_1 i Γ_2 są otoczeniami, to przez $\Gamma_1 \& \Gamma_2$ oznaczamy otoczenie, którego dziedzina jest sumą dziedzin Γ_1 i Γ_2, i które jest określone tak:

$$(\Gamma_1 \& \Gamma_2) = \begin{cases} \Gamma_1(x) \cap \Gamma_2(x), & \text{jeśli } \Gamma_1(x) \text{ i } \Gamma_2(x) \text{ są określone;} \\ \Gamma_1(x), & \text{jeśli tylko } \Gamma_1(x) \text{ jest określone;} \\ \Gamma_2(x), & \text{jeśli tylko } \Gamma_2(x) \text{ jest określone.} \end{cases}$$

Lemat 20.3 *Struktura* $\mathcal{F} = \langle \mathbf{F}, \cdot, [\![\ ]\!] \rangle$ *jest lambda-modelem.*

Dowód: Naszym zadaniem jest sprawdzenie następujących warunków:

(a) Jeśli x jest zmienną, to $[\![x]\!]_v = v(x)$;

(b) $[\![PQ]\!]_v = [\![P]\!]_v \cdot [\![Q]\!]_v$;

(c) $[\![\lambda x.P]\!]_v \cdot F = [\![P]\!]_{v[x\mapsto F]}$, dla dowolnego $F \in \mathbf{F}$;

(d) Jeśli $v|_{\mathrm{FV}(P)} = u|_{\mathrm{FV}(P)}$, to $[\![P]\!]_v = [\![P]\!]_u$.

(e) Jeśli $[\![\lambda x.M]\!]_v \approx [\![\lambda x.N]\!]_v$ to $[\![\lambda x.M]\!]_v = [\![\lambda x.N]\!]_v$.

Zaczynamy od (a). Przypuśćmy, że $\sigma \in [\![x]\!]_v$. Z definicji oznacza to, że $\Gamma \vdash x : \sigma$ dla pewnego Γ zgodnego z v, czyli takiego, że $\Gamma(x) \in v(x)$. Z lematu o generowaniu wiemy, że wtedy $\sigma \geq \Gamma(x)$, więc tym bardziej $\sigma \in v(x)$. Mamy więc inkluzję $[\![x]\!]_v \subseteq v(x)$. Inkluzja odwrotna jest jeszcze łatwiejsza, bo dla $\sigma \in v(x)$ otoczenie $\{x : \sigma\}$ jest zgodne z v.

W punkcie (b) pokażemy inkluzję $[\![P]\!]_v \cdot [\![Q]\!]_v \subseteq [\![PQ]\!]_v$, która jest mniej oczywista. Załóżmy, że $\sigma \in [\![P]\!]_v \cdot [\![Q]\!]_v$. Jest więc takie $\zeta \in [\![Q]\!]_v$, że $\zeta \to \sigma \in [\![P]\!]_v$. Istnieją zatem otoczenia Γ_1 i Γ_2 zgodne z v i takie, że $\Gamma_1 \vdash P : \zeta \to \sigma$ oraz $\Gamma_2 \vdash Q : \zeta$. Nietrudno sprawdzić, że $\Gamma_0 = \Gamma_1 \& \Gamma_2$ jest zgodne z v, oraz że $\Gamma_0 \vdash P : \zeta \to \sigma$ i $\Gamma_0 \vdash Q : \zeta$. Stąd $\Gamma_0 \vdash PQ : \sigma$ i mamy $\sigma \in [\![PQ]\!]_v$.

W części (c) „trudniejsza" jest inkluzja z lewej do prawej. Niech $\sigma \in [\![\lambda x.\, P]\!]_v \cdot F$. Oznacza to, że $\Gamma \vdash \lambda x.P : \tau \to \sigma$ i $\tau \in F$, gdzie Γ jest zgodne z v. Na mocy lematu o generowaniu $\Gamma, x : \tau \vdash P : \sigma$. Ponieważ $\Gamma, x : \tau$ jest zgodne z $v[x \mapsto F]$, więc $\sigma \in [\![P]\!]_{v[x\mapsto F]}$ i dobrze.

Warunek (d) wynika wprost z lematu 19.1.

Możemy teraz już stwierdzić, że struktura filtrów jest lambda-interpretacją. Pozostaje sprawdzenie warunku (e). Niech więc $[\![\lambda x.M]\!]_v \approx [\![\lambda x.N]\!]_v$, czyli $[\![\lambda x.M]\!]_v \cdot F = [\![\lambda x.N]\!]_v \cdot F$ dla dowolnego filtru F. Przypuśćmy, że $\sigma \in [\![\lambda x.M]\!]_v$. Wtedy, na mocy lematu o generowaniu, $\sigma = \bigcap\{\sigma_i \to \tau_i \mid i \in I\}$ oraz $\Gamma, x : \sigma_i \vdash M : \tau_i$ dla $i \in I$, dla pewnego Γ, zgodnego z v.

Ponieważ otoczenie Γ, $x : \sigma_i$ jest zgodne z wartościowaniem $v[x \mapsto \sigma_i\!\uparrow]$, więc typ τ_i należy do filtru $[\![M]\!]_{v[x\mapsto\sigma_i\uparrow]} = [\![\lambda x\, M]\!]_v \cdot \sigma_i\!\uparrow = [\![\lambda x\, N]\!]_v \cdot \sigma_i\!\uparrow = [\![N]\!]_{v[x\mapsto\sigma_i\uparrow]}$. A więc mamy $\Gamma_i' \vdash N : \tau_i$, gdzie Γ_i' jest pewnym otoczeniem zgodnym z $v[x \mapsto \sigma_i\!\uparrow]$. Ale wtedy $\Gamma_i'(x) \geq \sigma_i$, więc tym bardziej $\Gamma_i'(x \mapsto \sigma_i) \vdash N : \tau_i$, czyli $\Gamma_i' \vdash \lambda x.N : \sigma_i \to \tau_i$. Ponieważ Γ_i' jest także zgodne z v, więc $\sigma_i \to \tau_i \in [\![\lambda x.N]\!]$. Ostatecznie $\sigma \in [\![\lambda x.N]\!]$, bo σ to iloczyn wszystkich $\sigma_i \to \tau_i$. ∎

Lemat 20.4 *W modelu* $\mathcal{F} = \langle \mathbf{F}, \cdot, [\![\ ]\!] \rangle$ *określamy wartościowanie typowe* $\xi(p) = \{F \mid p \in F\}$. *Wtedy* $[\![\tau]\!]_\xi = \{F \mid \tau \in F\}$, *dla dowolnego typu* τ.

Dowód: Dowód jest przez indukcję ze względu na τ. Przypadek iloczynu jest łatwy, bo zachodzi równoważność

$$(\sigma \in F) \wedge (\rho \in F) \iff \sigma \cap \rho \in F.$$

Niech więc $\tau = \rho \to \sigma$. Mamy wykazać, że typ $\rho \to \sigma$ należy do filtru F wtedy i tylko wtedy, gdy $F \in [\![\rho]\!]_\xi \Rightarrow [\![\sigma]\!]_\xi$.

Implikacja z lewej do prawej jest łatwa. Jeśli bowiem $G \in [\![\rho]\!]_\xi$, to z założenia indukcyjnego dla ρ wynika $\rho \in G$, z zatem $\sigma \in F \cdot G$ wprost z definicji aplikacji $F \cdot G$.

Na odwrót, przypuśćmy, że $F \in [\![\rho]\!]_\xi \Rightarrow [\![\sigma]\!]_\xi$. Ponieważ $\rho \in \rho\uparrow$ więc z założenia indukcyjnego dla ρ mamy $\rho\uparrow \in [\![\rho]\!]_\xi$, a stąd $F \cdot \rho\uparrow \in [\![\sigma]\!]_\xi$, czyli (na mocy założenia indukcyjnego dla σ) typ σ należy do $F \cdot \rho\uparrow$. Istnieje więc takie $\mu \in \rho\uparrow$, że $\mu \to \sigma \in F$. Wtedy jednak $\mu \geq \rho$, więc $\rho \to \sigma \geq \mu \to \sigma$, i typ $\rho \to \sigma$ też należy do F. ∎

Twierdzenie 20.5 (o pełności) *Warunki* $\Gamma \vdash M : \sigma$ *i* $\Gamma \models M : \sigma$ *są równoważne.*

Dowód: Przypuśćmy, że $\Gamma \models M : \sigma$. W modelu $\mathcal{F} = \langle \mathbf{F}, \cdot, [\![\ ]\!] \rangle$ definiujemy wartościowanie v, przyjmując $v(x) = \tau\uparrow$ dla dowolnego $(x : \tau) \in \Gamma$. Ponieważ wtedy $\tau \in v(x)$, więc $v(x) \in [\![\tau]\!]_\xi$ (lemat 20.4). Zatem $\mathcal{F}, v, \xi \models \Gamma$, a wobec tego $\mathcal{F}, v, \xi \models M : \sigma$, gdzie ξ jest takie jak w Lemacie 20.4. To oznacza, że $[\![M]\!]_v \in [\![\sigma]\!]_\xi$, czyli $\sigma \in [\![M]\!]_v$. Jest więc otoczenie Γ' zgodne z v, takie że $\Gamma' \vdash M : \sigma$. Ponieważ Γ' jest zgodne z v, więc $\Gamma'(x) \in v(x)$, czyli $\Gamma(x) \leq \Gamma'(x)$ dla wszystkich x. A więc tym bardziej $\Gamma \vdash M : \sigma$. ∎

Ćwiczenia

1. Udowodnić, że jeśli F_1 i F_2 są filtrami, to $F_1 \cdot F_2$ jest filtrem.
2. Udowodnić, że $[\![M]\!]_v$ zawsze jest filtrem.
3. Czy model z filtrów jest ekstensjonalny?
4. Przypuśćmy, że model D to model Scotta $\mathcal{D}_\infty$, w którym $D_0 = \{\bot, \top\}$. Zdefiniować $[\![\kappa]\!]$, tak aby zachodziło twierdzenie o pełności dla $\mathcal{D}_\infty$ i typów iloczynowych według Scotta (ćwiczenie 19.6).

21 Typy iloczynowe bez omegi

System BCD, o którym była mowa do tej pory, przypisywał każdemu bez wyjątku termowi „trywialny" typ ω. Jeżeli z tego systemu usuniemy aksjomat

$$\Gamma \vdash M : \omega,$$

to już nie każdy term będzie miał typ. Nabiera więc sensu następująca definicja. Term M jest *typowalny* wtedy i tylko wtedy, gdy $\Gamma \vdash M : \tau$, dla pewnych Γ i τ. W istocie, jak się okaże poniżej, termy typowalne za pomocą typów iloczynowych bez omegi, to dokładnie termy silnie normalizowalne.

Dla porządku zacznijmy od definicji. Zbiór typów $\mathcal{T}_\cap$ definiujemy tak samo jak zbiór $\mathcal{T}_{\cap\omega}$, ale z pominięciem omegi:

- Typy atomowe należą do $\mathcal{T}_\cap$;
- Jeśli σ i τ są w $\mathcal{T}_\cap$, to $\sigma \to \tau$ i $\sigma \cap \tau$ też należą do $\mathcal{T}_\cap$.

W zbiorze $\mathcal{T}_\cap$ określamy relację $\leq$ jako najmniejszy quasiporządek $\leq$ spełniający warunki

$$\sigma \cap \tau \leq \sigma, \quad \sigma \cap \tau \leq \tau, \quad \sigma \leq \sigma \cap \sigma$$

$$(\sigma \to \tau) \cap (\sigma \to \rho) \leq \sigma \to \tau \cap \rho$$

Jeśli $\sigma \leq \sigma'$ i $\tau \leq \tau'$, to $\sigma \cap \tau \leq \sigma' \cap \tau'$ oraz $\sigma' \to \tau \leq \sigma \to \tau'$.

Będziemy teraz przypisywać termom typy za pomocą tych samych reguł co poprzednio, ale z wyłączeniem aksjomatu (ω). Dla odróżnienia od systemu BCD ten nowy system nazwijmy *systemem* $\lambda_{\cap\leq}$. Oto aksjomat i reguły systemu $\lambda_{\cap\leq}$.

$$\text{(Var)}\ \Gamma, x : \sigma \vdash x : \sigma$$

$$\text{(Abs)}\ \frac{\Gamma, x{:}\sigma \vdash M : \tau}{\Gamma \vdash (\lambda x.M) : \sigma \to \tau} \qquad \text{(App)}\ \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (MN) : \tau}$$

$$(\cap\text{I})\ \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash M : \sigma \cap \tau} \qquad (\leq)\ \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \tau}\ (\sigma \leq \tau)$$

Osądy wyprowadzalne w systemie $\lambda_{\cap\leq}$ możemy zapisywać $\Gamma \vdash_\cap M : \sigma$, ale zwykle zamiast $\vdash_\cap$ napiszemy po prostu $\vdash$. O którym systemie mowa, powinno być wiadomo z kontekstu.

Dla systemu $\lambda_{\cap\leq}$ zachodzi następujący wariant lematu 19.1.

Lemat 21.1 *Jeśli* $\Gamma \vdash M : \tau$*, to* $\Gamma(x)$ *jest określone dla każdego* $x \in \mathrm{FV}(M)$*. Ponadto* $\Gamma|_{\mathrm{FV}(M)} \vdash M : \tau$*, gdzie* $\Gamma|_{\mathrm{FV}(M)} = \{(x : \Gamma(x)) \mid x \in \mathrm{FV}(M)\}$.

Dowód: Łatwa indukcja ze względu na rozmiar wyprowadzenia. ∎

Różnica pomiędzy treścią lematu 21.1 i lematu 19.1 bierze się oczywiście z braku aksjomatu (ω). Teraz, aby przypisać typ całemu termowi, trzeba zadeklarować typy wszystkich jego zmiennych wolnych.

Podstawowe własności systemu $\lambda_{\cap\leq}$ są podobne do własności systemu BCD. W szczególności pozostają prawdziwe lematy 19.2, 19.3, a także lemat 19.5 o generowaniu (z pominięciem tego, co odnosi się do omegi). Podobnie jak poprzednio, wyprowadzamy stąd lemat o podstawieniu (odpowiednik lematu 19.6) a następnie twierdzenie o poprawności redukcji. Dowody tych faktów pozostają niezmienione (ale prostsze, bo niepotrzebne są rozważania dotyczące omegi).

Dla systemu $\lambda_{\cap\leq}$ nie zachodzi jednak twierdzenie 19.8. Na przykład nietypowalny term $\mathbf{KK}\Omega$ redukuje się do typowalnego $\mathbf{K}$.

Silna normalizacja

Udowodnimy teraz twierdzenie o silnej normalizacji dla systemu $\lambda_{\cap\leq}$. Nasz dowód będzie niewielką adaptacją dowodu twierdzenia 14.11 o silnej normalizacji rachunku z typami prostymi. Definicja $[\![\tau]\!]$ ma teraz o jeden przypadek więcej:

- $[\![p]\!] := \mathrm{SN}$, gdy p jest atomem;
- $[\![\tau \to \sigma]\!] := \{M \mid \forall N(N \in [\![\tau]\!] \Rightarrow MN \in [\![\sigma]\!])\}$.
- $[\![\tau \cap \sigma]\!] := [\![\tau]\!] \cap [\![\sigma]\!]$.

Lematy 14.7 i 14.9 pozostają prawdziwe dla tej definicji, a ich dowody pozostają praktycznie bez zmian (przypadek iloczynu to natychmiastowe zastosowanie założenia indukcyjnego). Dowód lematu 14.10 ulega tylko dwu drobnym zmianom. Po pierwsze, w przypadku zmiennej musimy się odwołać do takiej dodatkowej własności:

Lemat 21.2 *Jeśli $\tau \leq \sigma$, to $[\![\tau]\!] \subseteq [\![\sigma]\!]$.*

Dowód: Indukcja ze względu na definicję relacji $\leq$. ∎

Po drugie, w przypadku abstrakcji zamiast zwykłego $\sigma \to \rho$ mamy iloczyn $\bigcap\{\sigma_i \to \rho_i \mid i \in I\}$, gdzie $\Gamma, y : \sigma_i \vdash P : \rho_i$ dla $i \in I$, i musimy pokazać, że $M[\vec{N}/\vec{x}]$ jest stabilny dla wszystkich typów $\sigma_i \to \rho_i$. Trzeba więc cierpliwie podopisywać wszędzie indeks i. Otrzymamy w końcu

Twierdzenie 21.3 (o silnej normalizacji) *System $\lambda_{\cap\leq}$ ma własność silnej normalizacji: każdy term typowalny jest SN.*

Ćwiczenia

1. Sprawdzić, że lematy 19.3–19.6 i twierdzenie 19.7 dla systemu BCD przenoszą się na system $\lambda_{\cap\leq}$.
2. Udowodnić lemat 21.2.
3. Rozpatrzmy system $\lambda_\cap$ powstały z systemu $\lambda_{\cap\leq}$ przez zamianę reguły $(\leq)$ na regułę

$$(\cap\mathrm{E})\ \frac{\Gamma \vdash M : \sigma_1 \cap \sigma_2}{\Gamma \vdash M : \sigma_i}$$

   Pokazać, że termy typowalne w obu systemach są takie same.
4. Czy w systemie $\lambda_\cap$ zachodzi twierdzenie o poprawności eta-redukcji?
5. System $\lambda_{\cap\eta}$ to rozszerzenie systemu $\lambda_\cap$ o dodatkową regułę

$$(\eta)\ \frac{\Gamma \vdash M : \sigma, \quad M \to_\eta N}{\Gamma \vdash N : \sigma}$$

   Pokazać, że w systemach $\lambda_{\cap\eta}$ i $\lambda_{\cap\leq}$ wyprowadzalne są dokładnie te same osądy typowe.

Twierdzenie Pottingera

Twierdzenie o silnej normalizacji zachodzi dla wielu systemów przypisania typów. Jest to wręcz uważane za dość podstawową własność takich systemów, a system BCD z regułą (ω) stanowi raczej wyjątek, uzasadniony swoimi semantycznymi korzeniami.

Spośród systemów o własności SN, typy iloczynowe stanowią mechanizm najsilniejszy w tym sensie, że twierdzenie o silnej normalizacji jest dla tego systemu odwracalne: każdy term o własności SN jest typowalny. Pierwszy dowód tego faktu opublikował w 1980 roku G. Pottinger. Ten dowód i inne późniejsze dowody zawierały różne usterki. Pierwszy poprawny dowód (niepublikowany) podała w latach dziewięćdziesiątych B. Venneri. Poniższy dowód, miejmy nadzieję, też jest poprawny.

Naturalna idea dowodu jest oczywiście taka: Zacząć od typowania postaci normalnych i dowodzić twierdzenia przez indukcję ze względu na długość redukcji termu do postaci normalnej. Ale zauważyliśmy już, że dla systemu $\lambda_{\cap\leq}$ nie zachodzi twierdzenie 19.8. W szczególności nie zachodzi twierdzenie odwrotne do lematu 19.6. Mamy jednak nieco słabszą własność.

Lemat 21.4 *Jeśli* $\Gamma \vdash M[x := N] : \tau$ *oraz* $\Gamma \vdash N : \rho$*, to istnieje taki typ* σ*, że* $\Gamma \vdash N : \sigma$ *oraz* $\Gamma, x : \sigma \vdash M : \tau$*,*

Dowód: Ćwiczenie. ∎

Lemat 21.5

1. *Term* $xN_1 \dots N_k$ *jest typowalny wtedy i tylko wtedy, gdy* $N_1, \dots, N_k$ *są typowalne.*
2. *Abstrakcja* $\lambda x\, N$ *jest typowalna wtedy i tylko wtedy, gdy term* N *jest typowalny.*

Dowód: Udowodnimy implikację z prawej do lewej w części (1). Załóżmy, że $\Gamma_i \vdash N_i : \tau_i$ dla $i = 1, \dots, k$. Wtedy $\Gamma_1 \,\&\, \cdots \,\&\, \Gamma_k \,\&\, \{x : \tau_1 \to \cdots \tau_k \to p\} \vdash xN_1 \dots N_k : p$. Pozostałe części są oczywiste. ∎

Wniosek 21.6 *Każdy term w postaci normalnej jest typowalny.*

Dowód: Indukcja ze względu na budowę termu z pomocą lematu 21.5. ∎

Twierdzenie 21.7 (Pottinger) *Term jest typowalny w systemie* $\lambda_{\cap\leq}$ *wtedy i tylko wtedy, gdy jest silnie normalizowalny.*

Dowód: Implikacja z lewej do prawej to twierdzenie 21.3 o silnej normalizacji. Dowód implikacji odwrotnej przebiega przez indukcję ze względu na maksymalną możliwą długość ciągu redukcji danego termu M, którą oznaczymy przez $\delta(M)$. Jeśli $\delta(M) = 0$, czyli mamy do czynienia z postacią normalną, to stosujemy wniosek 21.6. W kroku indukcyjnym zakładamy, że każdy term N o własności $\delta(N) < n$ jest typowalny i dowodzimy, że jeśli $\delta(M) = n$ to M jest typowalny. Dowód jest przez „lokalną" indukcję ze względu na budowę termu M.

Niech $M \to_L M'$. Wtedy $\delta(M') < n$, zatem M' jest typowalny.

Przypadek 1: $M = \lambda x\, N \to_L \lambda x\, N' = M'$. Term M' jest typowalny, więc na mocy lematu 21.5(2) także N' jest typowalny. Z „lokalnego" założenia indukcyjnego term N jest typowalny, więc abstrakcja $M = \lambda x\, N$ też jest typowalna.

Przypadek 2: $M = xN_1 \dots N_i \dots N_k \to_L xN_1 \dots N_i' \dots N_k = M'$, gdzie $N_i \to_L N_i'$. Skoro term M' jest typowalny, to także termy $N_1, \dots, N_i', \dots, N_k$ są typowalne. Z założenia indukcyjnego wynika, że N_i jest typowalny, i z lematu 21.5(1) dostajemy typowalność M.

Przypadek 3: Przyszedł czas na przypadek redeksu czołowego:

$$M = (\lambda x.P)QN_1 \dots N_k \to_L P[x := Q]N_1 \dots N_k = M'.$$

Stosując k-krotnie pierwszą część lematu o generowaniu, wnioskujemy, że $\Gamma' \vdash N_i : \rho_i$ oraz $\Gamma' \vdash P[x := Q] : \rho_1 \to \dots \to \rho_k \to \tau$ dla pewnych ρ_i i τ. Ponadto term Q jest silnie normalizowalny, jako podterm silnie normalizowalnego termu M. Ponieważ M ma redeks czołowy, więc najdłuższa możliwa redukcja termu M jest dłuższa przynajmniej o jeden krok od każdej redukcji termu Q. Zatem $\delta(Q) < \delta(M)$, skąd Q jest termem typowalnym, np. $\Gamma'' \vdash Q : \zeta$. Niech $\Gamma = \Gamma' \& \Gamma''$. Wtedy nadal $\Gamma \vdash P[x := Q] : \rho_1 \to \dots \to \rho_k \to \tau$ oraz Q ma typ w Γ. Z lematu 21.4 dostajemy $\Gamma \vdash Q : \sigma$ oraz $\Gamma, x : \sigma \vdash P : \rho_1 \to \dots \to \rho_k \to \tau$, dla pewnego typu σ. Stąd $\Gamma \vdash \lambda x.P : \sigma \to \rho_1 \to \dots \to \rho_k \to \tau$, i dalej $\Gamma \vdash M : \tau$. ∎

Wniosek 21.8 *Problem typowalności dla typów iloczynowych jest nierozstrzygalny.*

Dowód: Wynika to wprost z twierdzenia 21.7 i z nierozstrzygalności silnej normalizacji (wniosek 7.11). ∎

Wniosek 21.9 *Problem sprawdzenia typu dla typów iloczynowych też jest nierozstrzygalny.*

Dowód: Redukujemy problem typowalności do problemu sprawdzenia typu. Niech $\vec{x}$ będą wszystkimi zmiennymi wolnymi termu M. Term M jest typowalny wtedy i tylko wtedy, gdy $x : s \vdash \mathbf{K}x(\lambda\vec{x}M) : s$. ∎

Problem niepustości typu

Dualny do problemu typowalności jest *problem niepustości typu*, nazywany też problemem *inhabitacji*, i formułowany tak:

Czy istnieje term zamknięty danego typu?

Nieco ogólniejsze (ale równoważne) sformułowanie jest takie:

Czy dla danych Γ *i* τ *istnieje term* M*, który ma typ* τ *w otoczeniu* Γ*?*

Pokażemy szkic dowodu, że problem inhabitacji dla typów iloczynowych jest nierozstrzygalny. Jest to nowy dowód, podany przez S. Salvatiego w 2009 roku. Polega on na redukcji problemu definiowalności w skończonych modelach (twierdzenie 17.15). Poniżej jest mowa o systemie $\lambda_{\cap\leq}$, ale podobne rozumowanie stosuje się dla innych wariantów.

Główny pomysł polega na kodowaniu każdego elementu d w skończonym modelu $\mathfrak{M}(A)$ jako typu iloczynowego τ_d. Robimy to tak:[23]

- Jeśli $d \in D_0$ to $\tau_d = d$.
- Jeśli $d \in D_{\alpha\to\beta}$ to $\tau_d = \bigcap_{e\in D_\alpha}(\tau_e \to \tau_{de})$.

Lemat 21.1 *Jeśli $\tau_d \le \tau_e$ to $d = e$.*

Dowód: Indukcja ze względu na łączny rozmiar typów τ_d i τ_e. Jeśli któryś z nich jest atomem to łatwo pokazać, że drugi też musi być atomem i to tym samym. W przeciwnym razie oba typu są iloczynami implikacji i mamy $d \in D_{\alpha\to\beta}$ oraz $e \in D_{\gamma\to\delta}$. Wtedy

$$\tau_d = \bigcap_{a\in D_\alpha} (\tau_a \to \tau_{da}) \le \tau_b \to \tau_{eb},$$

gdzie b jest dowolnym elementem D_γ. Ponieważ lemat 19.3 zachodzi też bez omegi, więc

$$\bigcap_{a\in S} \tau_{da} \le \tau_{eb},$$

gdzie $S = \{a \mid \tau_b \le \tau_a\}$ jest zbiorem niepustym. Z założenia indukcyjnego mamy $a = b$, dla $a \in S$ (w szczególności S jest singletonem) oraz $db = eb$. Z dowolności b wnioskujemy, że funkcje d i e należą do tej samej dziedziny i są równe. ∎

Lemat 21.2 *Załóżmy, że każda deklaracja w Γ jest postaci $x : \tau_d$, dla pewnego $d \in \mathfrak{M}(A)$, i że M jest w postaci normalnej. Wówczas*

1. *Jeśli $\Gamma \vdash M : \tau_d$ i $\Gamma \vdash M : \tau_e$, dla pewnych $d, e \in D_\alpha$, to $d = e$.*
2. *Jeśli $\Gamma \vdash xM_1 \dots M_n : \rho$, gdzie $\Gamma(x) = \tau_c$ i $c \in D_\alpha$, to $\alpha = \alpha_1 \to \dots \to \alpha_m \to 0$, dla pewnego $m \ge n$, i na dodatek:*
   - *dla każdego $i \le n$ istnieje takie $e_i \in D_{\alpha_i}$, że $\Gamma \vdash M_i : \tau_{e_i}$;*
   - *dla $d = ce_1 \dots e_n$ zachodzi $\Gamma \vdash M : \tau_d$, oraz $\tau_d \le \rho$.*

Dowód: Jednoczesna indukcja ze względu na M. Przypadek $M = \lambda x.\, M'$ jest łatwy. Typy τ_d i τ_e nie mogą być atomowe, więc d, e muszą być funkcjami, tj. $\alpha = \beta \to \gamma$. Wtedy $\Gamma, x : \tau_c \vdash M' : \tau_{dc}$ oraz $\Gamma, x : \tau_c \vdash M' : \tau_{ec}$, dla wszystkich $c \in D_\beta$. Z założenia indukcyjnego wynika, że $dc = ec$, dla wszystkich c, skąd $d = e$.

W przypadku $M = xM_1 \dots M_n$ stosujemy pomocniczą indukcję ze względu na n. Dla $n = 0$, teza wynika łatwo z lematu o generowaniu i lematu 21.1.

Niech więc $n > 0$ i niech $M' = xM_1 \dots M_{n-1}$, czyli $M = M'M_n$. Jeśli $\Gamma \vdash M : \rho$ to $\Gamma \vdash M' : \sigma \to \rho$, oraz $\Gamma \vdash M_n : \sigma$. Z (pomocniczego) założenia indukcyjnego dostajemy

[23] Dla odróżnienia, typy proste oznaczamy poniżej przez $\alpha, \beta, \gamma, \delta$, a typy iloczynowe przez τ, σ, ρ.

$\Gamma \vdash M' : \tau_d$, gdzie $\tau_d \leq \sigma \to \rho$. Typ τ_d nie może być atomem (w szczególności $n-1 < m$), a stąd otrzymujemy $\bigcap_{a\in S} \tau_{da} \leq \rho$, gdzie $S = \{a \mid \sigma \leq \tau_a\}$ jest niepuste. Dla wszystkich $a \in S$ mamy $\Gamma \vdash M : \tau_a$, więc z głównego założenia indukcyjnego wynika, że $S = \{e\}$. Stąd $\tau_{de} \leq \rho$, co kończy dowód części drugiej. Z niej i z lematu 21.1 wynika część pierwsza. ∎

Dla danego wartościowania v w modelu $\mathfrak{M}(A)$, przyjmijmy $\Gamma_v(x) = \tau_{v(x)}$.

Lemat 21.3 *Dla dowolnego termu M w postaci normalnej:*

$$[\![M]\!]_v = d \qquad \textit{wtedy i tylko wtedy, gdy} \qquad \Gamma_v \vdash M : \tau_d.$$

Dowód: Indukcja ze względu na M, z lewej do prawej zupełnie rutynowa. W kierunku odwrotnym, jeśli M jest abstrakcją $\lambda x.\, M'$, to $\Gamma, x : \tau_a \vdash M' : \tau_{da}$, dla wszystkich a w odpowiedniej dziedzinie D_α. Z założenia indukcyjnego, $[\![M']\!]_{v[x\mapsto a]} = da$, dla każdego a, skąd $[\![M]\!]_v = d$. Jeśli $M = xM_1 \dots M_n$, gdzie $\Gamma(x) = \tau_c$, to stosujemy lemat 21.2(2) i indukcję. Wnioskujemy, żę $[\![M_i]\!] = e_i$, aplikacja $ce_1 \dots e_n$ jest możliwa, i daje w wyniku wartość $[\![M]\!]_v$. Zatem $\tau_{ce_1\dots e_n} \leq \tau_d$, skąd $d = ce_1 \dots e_n = [\![M]\!]_v$. ∎

Twierdzenie 21.4 *Problem inhabitacji dla typów iloczynowych jest nierozstrzygalny.*

Dowód: Aby sprawdzić czy element d modelu $\mathfrak{M}(A)$ jest definiowalny, wystarczy zbadać, czy typ τ_d jest niepusty. Na mocy twierdzenia 17.15 niepustość nie może być rozstrzygalna. ∎

Ćwiczenia

1. Udowodnić wniosek 21.6. *Wskazówka:* term M jest w postaci normalnej, wtedy i tylko wtedy, gdy zachodzi jedna z trzech możliwości: (a) M jest zmiennną, (b) $M = \lambda x.N$, gdzie N jest w postaci normalnej, (c) $M = x\vec{N}$, gdzie $\vec{N}$ są w postaci normalnej.
2. Czy w dowodzie twierdzenia 21.7 można wzmocnić założenie indukcyjne do warunku:
   Jeśli $\delta(M) = n$ oraz $M \to_L M'$ i przy tym $\Gamma \vdash M' : \tau$, to $\Gamma \vdash M : \tau$?
3. Czy w dowodzie twierdzenia 21.7, w założeniu indukcyjnym można zastąpić warunek $M \to_L M'$ przez $M \to M'$?

22 Logika drugiego rzędu

Przez logikę drugiego rzędu zazwyczaj rozumie się system powstały przez rozszerzenie rachunku predykatów (logiki pierwszego rzędu) o kwantyfikatory przebiegające zbiory i relacje. Z konstruktywnego punktu widzenia, celowość takiego rozszerzenia jest wątpliwa. Interpretacja BHK nadaje bowiem taki sens kwantyfikatorowi ogólnemu:

Konstrukcją (dowodem) formuły $\forall R\, W(R)$ jest metoda przekształcenia dowolnego znaczenia $\mathbf{R}$ zmiennej R w konstrukcję stwierdzenia $W(\mathbf{R})$.

Aby jednak wykonać konstrukcję dla $W(\mathbf{R})$ trzeba na ogół *znać* sam zbiór $\mathbf{R}$, tj. umieć go zdefiniować, a to nie zawsze jest możliwe, bo każda nieskończona dziedzina ma nieprzeliczalnie wiele podzbiorów. Dlatego konstruktywne rozumienie kwantyfikatora drugiego rzędu jest inne: zmienna relacyjna przebiega predykaty definiowalne formułami. W intuicjonistycznej logice drugiego rzędu przyjmuje się więc (z grubsza[24]) takie reguły:

$$\frac{\Gamma \vdash \forall X\ \varphi(X)}{\Gamma \vdash \varphi(\sigma)} \qquad \frac{\Gamma \vdash \varphi(X)}{\Gamma \vdash \forall X\ \varphi(X)}\ (X \text{ nie ma w } \Gamma)$$

Uwaga: Systemy klasycznej logiki drugiego rzędu, w których kwantyfikatory przebiegają dowolne zbiory, nie są efektywnie aksjomatyzowalne. Tak nie jest w przypadku intuicjonistycznej logiki drugiego rzędu, która jest aksjomatyzowalna i dlatego rekurencyjnie przeliczalna.

Jak powiedzieliśmy, w logice drugiego rzędu obok kwantyfikatorów relacyjnych występują także elementy zwykłego rachunku predykatów: wyrażenia i kwantyfikatory indywiduowe. Okazuje się jednak, że te ostatnie nie zawsze są istotne, a wiele własności logiki drugiego rzędu można badać w oderwaniu od jej aspektu indywiduowego. Jeżeli w formułach rachunku predykatów drugiego rzędu zaniedbać wszystko to, co odnosi się do indywiduów, to otrzymamy formuły zdaniowe z kwantyfikatorami. Na przykład rozpatrzmy formułę $N(x)$, która definiuje liczby naturalne:

$$\forall R(\forall z(R(z) \to R(sz)) \to R(0) \to R(x))$$

Usunięcie z niej informacji o indywiduach daje kwantyfikowaną formułę zdaniową:

$$\forall R((R \to R) \to R \to R),$$

w której jednoargumentowa zmienna relacyjna R została zastąpiona zmienną zdaniową. W podobny sposób można każdą formułę przekształcić w formułę zdaniową z kwantyfikatorami. Okazuje się, że ten zabieg, znacznie upraszczając składnię, nie zmienia wielu istotnych własności naszej logiki.

Zdaniowa logika drugiego rzędu

Na razie zatem będziemy się zajmować tylko zdaniową logiką drugiego rzędu, i to wyłącznie fragmentem implikacyjno-uniwersalnym. Zaczynamy od składni. *Formułami* nazywamy zmienne zdaniowe $p, q, r, \ldots$ i wyrażenia $(\sigma \to \tau)$, $\forall p\, \sigma$, gdzie σ i τ są formułami. Przez $\mathrm{FVT}(\varphi)$ oznaczamy zbiór zmiennych zdaniowych występujących wolno w formule φ. Definiujemy go tak: $\mathrm{FVT}(p) = \{p\}$, $\mathrm{FVT}(\sigma \to \tau) = \mathrm{FVT}(\sigma) \cup \mathrm{FVT}(\tau)$, oraz $\mathrm{FVT}(\forall p\, \sigma) = \mathrm{FVT}(\sigma) - \{p\}$. Podstawienie formuły σ do formuły φ w miejsce wolnych wystąpień zmiennej p oznaczamy przez $\varphi[p := \sigma]$ (lub przez $\varphi[\sigma/p]$). Podstawienie jest definiowane przez indukcję:

- $p[p := \sigma] = \sigma$ oraz $q[p := \sigma] = q$;
- $(\psi \to \tau)[p := \sigma] = \psi[p := \sigma] \to \tau[p := \sigma]$;
- $(\forall p\, \psi)[p := \sigma] = \forall p\, \psi$;

[24] Pomijamy tu ścisłą definicję wyrażenia $\varphi(\sigma)$.

- $(\forall q\,\psi)[p := \sigma] = \forall q\,\psi[p := \sigma]$, gdy $q \notin \mathrm{FVT}(\sigma)$;

Utożsamiamy (alfa-konwersja) formuły różniące się tylko wyborem zmiennych związanych. Reguły naturalnej dedukcji dla naszej logiki są takie:

$$(\mathrm{Ax})\ \Gamma, \varphi \vdash \varphi$$

$$(\to \mathrm{I})\ \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi} \qquad (\to \mathrm{E})\ \frac{\Gamma \vdash \varphi \to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$$

$$(\forall \mathrm{I})\ \frac{\Gamma \vdash \varphi}{\Gamma \vdash \forall p\,\varphi}\ (p \notin \mathrm{FVT}(\Gamma)) \qquad (\forall \mathrm{E})\ \frac{\Gamma \vdash \forall p\,\varphi}{\Gamma \vdash \varphi[p := \vartheta]}$$

Uwaga: Znaczeniem zmiennej zdaniowej może być dowolna formuła, zob. regułę (∀E). Tę własność czasem nazywa się *full comprehension*. Znaczenie formuły $\varphi = \forall p\,\psi$ jest wyznaczone przez wszystkie możliwe podstawienia $\psi[p := \sigma]$. Ale ponieważ w roli σ może wystąpić samo φ, więc znaczenie formuły w sposób uwikłany zależy od niej samej. Inaczej mówiąc, nasza logika jest *impredykatywna*. Konsekwencją tej własności jest utrudnione definiowanie i dowodzenie własności przez zwykłą indukcję ze względu na „budowę formuły".

Twierdzenie 22.1 (Löb, Gabbay/Sobolew) *Intuicjonistyczna logika zdaniowa drugiego rzędu jest nierozstrzygalna (nierozstrzygalne jest pytanie czy dana formuła jest twierdzeniem).*

Powyższy wynik kontrastuje ze znanym twierdzeniem z teorii złożoności: klasyczna logika kwantyfikowanych formuł boole'owskich jest PSPACE-zupełna. Zauważmy jednak, że w logice klasycznej, opartej na zerojedynkowej semantyce, każdy kwantyfikator można wyeliminować (kosztem wykładniczego wydłużenia formuły). Klasyczny rachunek zdań jest *funkcjonalnie zupełny:* każdą funkcję zerojedynkową można zdefiniować formułą. W przypadku intuicjonistycznym tak nie jest. Na przykład formuły $p \to \forall q(q \vee \neg q)$ i $\neg\neg p \to \exists q((p \to \neg q \vee q) \to p)$ nie są równoważne żadnej formule zdaniowej $\varphi(p)$ bez kwantyfikatorów.

Ćwiczenia

1. Pokazać, że formuła $\forall p(q \vee \neg p) \to q \vee \forall p\,\neg p$ jest twierdzeniem zdaniowej logiki intuicjonistycznej drugiego rzędu (reguły dla alternatywy są takie same jak zwykle). Czy każda formuła postaci $\forall p(q \vee \varphi(p)) \to q \vee \forall p\,\varphi(p)$ będzie twierdzeniem?
2. Pokazać, że formuła $\forall r((p \to r) \to (q \to r) \to r)$ jest w zdaniowej logice intuicjonistycznej drugiego rzędu równoważna alternatywie $p \vee q$.
3. Jak zdefiniować reguły naturalnej dedukcji dla zdaniowego kwantyfikatora szczegółowego?

23 System F

Zgodnie z interpretacją BHK, konstrukcją dla $\forall p\,\varphi$ jest metoda, która dla dowolnej formuły σ pozwala otrzymać konstrukcję dla $\varphi[p := \sigma]$. Taka konstrukcja jest więc funkcją określoną

na zbiorze formuł, ktora dla każdego σ przyjmuje wartość ze zbioru konstrukcji $\varphi[p := \sigma]$. Zbiór takich konstrukcji odpowiada (nieformalnie) produktowi $\Pi_{p\in\mathbf{Prop}}\varphi[p := \sigma]$. Jeśli więc utożsamiać formuły i typy to formuła uniwersalna jest typem produktowym. Patrząc na to z innej strony, możemy powiedzieć, że term typu $\forall p\,\varphi$ reprezentuje procedurę polimorficzną z (formalnym) parametrem typowym p. Przekazanie do takiej procedury parametru aktualnego σ ustala typ bieżącej aktywacji jako $\varphi[p := \sigma]$.

System rachunku lambda, w którym typy są formułami zdaniowymi drugiego rzędu nazywamy *polimorficznym rachunkiem lambda*, *rachunkiem lambda drugiego rzędu*, *polimorficznym rachunkiem lambda drugiego rzędu* lub *systemem* **F**. Poniższe reguły są zarówno regułami przypisania typów, jak też definicją składni. Wyrażenie $\Gamma \vdash M : \sigma$ czytamy „w otoczeniu Γ term M ma typ σ", gdzie *otoczenie* jest rozumiane jak zwykle. Przez $\mathrm{FVT}(\Gamma)$ oznaczamy sumę $\bigcup\{\mathrm{FVT}(\gamma) \mid (x : \gamma) \in \Gamma,\ \text{dla pewnego } x\}$.

$$\text{(Var)}\ \Gamma(x : \varphi) \vdash x : \varphi$$

$$\text{(Abs)}\ \frac{\Gamma(x:\varphi) \vdash M : \psi}{\Gamma \vdash (\lambda x{:}\varphi\, M) : \varphi \to \psi} \qquad \text{(App)}\ \frac{\Gamma \vdash M : \varphi \to \psi \quad \Gamma \vdash N : \varphi}{\Gamma \vdash (MN) : \psi}$$

$$\text{(Gen)}\ \frac{\Gamma \vdash M : \varphi}{\Gamma \vdash (\Lambda p\, M) : \forall p\,\varphi}\ (p \notin \mathrm{FVT}(\Gamma)) \qquad \text{(Inst)}\ \frac{\Gamma \vdash M : \forall p\,\varphi}{\Gamma \vdash M\sigma : \varphi[p := \sigma]}$$

Składnię systemu **F** zdefiniowaliśmy *w stylu Churcha*, tj. tak, że typy stanowią integralną część termów. Przy ustalonym otoczeniu, deklarującym typy zmiennych wolnych, każdy term ma dokładnie jeden typ, który można bez trudu ustalić.

W termach mogą występować zarówno zmienne przedmiotowe jak typowe (zdaniowe). Operatorami wiążącymi zmienne są obie lambdy i kwantyfikator występujący w typach. Symbol FVT( ) odnosi się do zmiennych wolnych zdaniowych, a symbol FV do zmiennych wolnych przedmiotowych. Definicje są indukcyjne:

- $\mathrm{FV}(x) = \{x\}$;
- $\mathrm{FV}(\lambda x{:}\varphi\, M) = \mathrm{FV}(M) - \{x\}$;
- $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$;
- $\mathrm{FV}(\Lambda p\, M) = \mathrm{FV}(M)$;
- $\mathrm{FV}(M\sigma) = \mathrm{FV}(M)$;
- $\mathrm{FVT}(x) = \varnothing$;
- $\mathrm{FVT}(\lambda x{:}\varphi\, M) = \mathrm{FVT}(\varphi) \cup \mathrm{FVT}(M)$;
- $\mathrm{FVT}(MN) = \mathrm{FVT}(M) \cup \mathrm{FVT}(N)$;
- $\mathrm{FVT}(\Lambda p\, M) = \mathrm{FVT}(M) - \{p\}$;
- $\mathrm{FVT}(M\sigma) = \mathrm{FVT}(M) \cup \mathrm{FVT}(\sigma)$.

Operacja podstawienia występuje w dwóch wersjach: podstawienie typu (formuły) za zmienną typową (zdaniową) i podstawienie termu za zmienną przedmiotową. Definiujemy je, zakładając, że zmienne związane są tak dobrane, aby nie doszło do konfuzji, tj. stosujemy w razie potrzeby domyślną wymianę zmiennej związanej.

- $x[x := P] = P$ oraz $y[x := P] = y$;
- $(MN)[x := P] = M[x := P]N[x := P]$;
- $(\lambda x{:}\varphi\, M)[x := P] = \lambda x{:}\varphi\, M$;
- $(\lambda y{:}\varphi\, M)[x := P] = \lambda y{:}\varphi.\, M[x := P]$, gdzie $y \notin \mathrm{FV}(P)$;
- $(M\tau)[x := P] = M[x := P]\tau$;
- $(\Lambda p\, M)[x := P] = \Lambda p\, M[x := P]$, gdzie $p \notin \mathrm{FVT}(P)$;
- $x[p := \sigma] = x$;
- $(MN)[p := \sigma] = M[p := \sigma]N[p := \sigma]$;
- $(\lambda x{:}\varphi\, M)[p := \sigma] = \lambda x{:}\varphi[p := \sigma].\, M[p := \sigma]$;
- $(M\tau)[p := \sigma] = M[p := \sigma]\tau[p := \sigma]$;
- $(\Lambda p\, M)[p := \sigma] = \Lambda p\, M$;
- $(\Lambda q\, M)[p := \sigma] = \Lambda q\, M[p := \sigma]$, gdzie $q \notin \mathrm{FVT}(\sigma)$.

Przez $\Gamma[p := \sigma]$ oznaczymy otoczenie Γ, w którym każda deklaracja $(x : \tau)$ została zastąpiona przez $(x : \tau[p := \sigma])$.

Lemat 23.1

(1) Jeśli $\Gamma \vdash M : \varphi$, *to* $\Gamma[p := \sigma] \vdash M[p := \sigma] : \varphi[p := \sigma]$.

(2) Jeśli $\Gamma, x : \tau \vdash M : \varphi$ *oraz* $\Gamma \vdash P : \tau$ *to* $\Gamma \vdash M[x := P] : \varphi$.

Dowód: Indukcja ze względu na budowę M. Część (2) wymaga lematu podobnego do lematu 13.1. ∎

Poniżej mamy kilka przykładów termów i ich typów (zamiast $\lambda x : \tau$ piszemy często λx^{τ}):

- $\vdash \Lambda q \lambda x^{\forall p(p\to p)}.\, x(q \to q)(xq) : \forall q(\forall p(p \to p) \to q \to q)$;
- $\vdash \Lambda p.\lambda f^{p\to p}\lambda x^{p}.f(fx) : \forall p((p \to p) \to (p \to p))$;
- $\vdash \lambda f^{\forall p(p\to q\to p)}\Lambda p\lambda x^{p}.f(q{\to}p)(fpx) : \forall p(p \to q \to p) \to \forall p(p \to q \to q \to p)$.

Twierdzenie 23.2 *Dla dowolnego typu* τ *następujące warunki są równoważne:*

1) Istnieje term zamknięty typu τ*;*
2) Formuła τ *jest twierdzeniem intuicjonistycznej logiki zdaniowej drugiego rzędu.*

Dowód: Oczywiste. ∎

Definicja 23.3 Przez beta redukcję w systemie **F** rozumiemy najmniejszą relację $\to_\beta$ w zbiorze termów zawierającą:

- $(\lambda x{:}\tau.M)N \longrightarrow_\beta M[x := N]$;
- $(\Lambda\alpha.M)\tau \longrightarrow_\beta M[\alpha := \tau]$,

i zamkniętą ze względu na konteksty, tj. $M \to_\beta N$ implikuje $MP \to_\beta NP$, $PM \to_\beta PN$, $\lambda x{:}\tau.M \to_\beta \lambda x{:}\tau.N$, $\Lambda p\, M \to_\beta \Lambda p\, N$ oraz $M\tau \to_\beta N\tau$, o ile odpowiednie wyrażenia są poprawnymi termami.

Jak zwykle w takich przypadkach, symbol $\twoheadrightarrow_\beta$ oznacza domknięcie przechodnio-zwrotne relacji redukcji $\to_\beta$. Indeks $_\beta$ często jest pomijany. Najważniejsze własności redukcji są takie:

Twierdzenie 23.4

1) *Jeśli* $\Gamma \vdash M : \tau$ *oraz* $M \twoheadrightarrow_\beta N$ *to* $\Gamma \vdash N : \tau$.
2) *Własność Churcha-Rossera: Jeżeli* $M \twoheadrightarrow_\beta N_1$ *oraz* $M \twoheadrightarrow_\beta N_2$ *to* $N_1 \twoheadrightarrow_\beta P$ *i* $N_2 \twoheadrightarrow_\beta P$, *dla pewnego* P.
3) *Silna normalizacja: Nie istnieje nieskończony ciąg redukcji rozpoczynający się od jakiegokolwiek (poprawnego) termu.*

Dowód: Część (1) wynika z lematu 23.1. Części (2) i (3) zostawiamy na razie bez dowodu. ∎

Uwaga: Dla termów systemu **F** można też rozważać relację eta-redukcji, zadaną regułami:

- $\lambda x{:}\tau.Mx \longrightarrow_\beta M$ (gdy $x \notin \mathrm{FV}(M)$);
- $\Lambda p.Mp \longrightarrow_\beta M$ (gdy $p \notin \mathrm{FVT}(M)$).

Liczby naturalne

Niech $\omega = \forall p((p \to p) \to (p \to p))$. Liczby naturalne są reprezentowane przez termy zamknięte typu ω.

$$\mathbf{n} = \Lambda p \lambda f : (p \to p) \lambda x : p.f(f(\cdots f(x)))$$

Na przykład $\mathbf{2} = \Lambda p.\lambda f^{p\to p} x^p.f(fx)$.

Mówimy, że funkcja $f : \mathbb{N}^k \to \mathbb{N}$ jest *definiowalna* w systemie **F**, gdy istnieje taki term $M : \omega^k \to \omega$, że dla dowolnych $n_1, \cdots n_k$ zachodzi $M\mathbf{n_1} \ldots \mathbf{n_k} \twoheadrightarrow_\beta \mathbf{f}(\mathbf{n_1}, \ldots, \mathbf{n_k})$.

Na przykład dodawanie, mnożenie i potęgowanie definiujemy tak:

- $\mathbf{A}dd = \lambda mn.\Lambda p.\lambda fx.mpf(npfx)$;
- $\mathbf{M}ult = \lambda mn.\Lambda p.\lambda fx.mp(npf)x$;
- $\mathbf{E}xp = \lambda mn.\Lambda p.\lambda fx.m(p \to p)(np)fx$.

Można też zdefiniować poprzednik i to typu $\omega \to \omega$. A więc odejmowanie i równość też są definiowalne (por. wniosek 17.14).

Spójniki zdaniowe

W logice zdaniowej drugiego rzędu fałsz można zdefiniować tak:

$$\bot = \forall p\, p$$

Nietrudno zauważyć, że z fałszu wszystko wynika. Jeśli $M : \bot$, to $M\tau : \tau$.

Można też zdefiniować pozostałe spójniki zdaniowe. Przypomnijmy odpowiednie reguły naturalnej dedukcji wraz z przypisaniem termów:

Koniunkcja:
$$\frac{\Gamma \vdash M : \varphi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N\rangle : \varphi \wedge \psi}\ (\mathrm{I}\wedge) \qquad \frac{\Gamma \vdash M : \varphi \wedge \psi}{\Gamma \vdash \pi_1(M) : \varphi}\ (\mathrm{E}\wedge)\ \frac{\Gamma \vdash M : \varphi \wedge \psi}{\Gamma \vdash \pi_2(M) : \psi}$$

Alternatywa:
$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \mathbf{inl}_{\varphi\vee\psi}(M) : \varphi \vee \psi}\ (\mathrm{I}\vee)\ \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathbf{inr}_{\varphi\vee\psi}(M) : \varphi \vee \psi}$$

$$\frac{\Gamma \vdash M : \varphi \vee \psi \quad \Gamma(x : \varphi) \vdash P : \vartheta \quad \Gamma(y : \psi) \vdash Q : \vartheta}{\Gamma \vdash \mathbf{case}\ M\ \mathbf{of}\ [x]P, [y]Q : \vartheta}\ (\mathrm{E}\vee)$$

Redukcje związane z koniunkcją i alternatywą są następujące:

$$\begin{aligned} \pi_1(\langle M, N\rangle) &\to M, \\ \pi_2(\langle M, N\rangle) &\to N; \\ \mathbf{case}\ \mathbf{inl}(P)\ \mathbf{of}\ [x]M, [y]N &\to M[x := P]; \\ \mathbf{case}\ \mathbf{inr}(Q)\ \mathbf{of}\ [x]M, [y]N &\to N[y := Q]. \end{aligned}$$

Definicję alternatywy w systemie **F** otrzymamy, jeśli spróbujemy wyrazić w logice drugiego rzędu pojęcie sumy jako najmniejszego zbioru zawierającego oba składniki:

$$x \in A \cup B \Leftrightarrow \forall P(A \subseteq P \to B \subseteq P \to x \in P),$$

lub inaczej:

$$(A \cup B)(x) \Leftrightarrow \forall P(\forall y(A(y) \to P(y)) \to \forall y(B(y) \to P(y)) \to P(x)).$$

Po usunięciu informacji o indywiduach otrzymujemy definicję alternatywy:

$$\sigma \vee \tau = \forall p((\sigma \to p) \to (\tau \to p) \to p),$$

gdzie zmienna p jest nowa, tj. $p \notin \mathrm{FVT}(\sigma) \cup \mathrm{FVT}(\tau)$. Pozostaje sprawdzić, że nasza definicja jest dobra. Wystarczy w tym celu zdefiniować **inl**, **inr** i **case** w taki sposób, żeby reguły (I$\vee$) i (E$\vee$) były poprawne. Robimy to tak:

- $\mathbf{inl}_{\sigma\vee\tau} M = \Lambda p \lambda u^{\sigma\to p} \lambda v^{\tau\to p}.uM$;
- $\mathbf{inr}_{\sigma\vee\tau} M = \Lambda p \lambda u^{\sigma\to p} \lambda v^{\tau\to p}.vM$;
- **case** M **of** $[x]P^{\vartheta}, [y]Q^{\vartheta} = M\vartheta(\lambda x^{\sigma}.P)(\lambda y^{\tau}Q)$.

Nietrudno sprawdzić, że typowanie jest poprawne, co w szczególności oznacza, że nasza definicja spełnia reguły wnioskowania dla alternatywy. W istocie spełniony jest silniejszy warunek: także redukcje są zgodne z oczekiwaniami, chociaż mogą wymagać więcej niż jednego kroku:

$$\begin{aligned} \mathbf{case}\ \mathbf{inl}(P)\ \mathbf{of}\ [x]M, [y]N &\quad \twoheadrightarrow \quad M[x := P]; \\ \mathbf{case}\ \mathbf{inr}(Q)\ \mathbf{of}\ [x]M, [y]N &\quad \twoheadrightarrow \quad N[y := Q]. \end{aligned}$$

Koniunkcję definiujemy tak:

$$\sigma \wedge \tau = \forall p((\sigma \to \tau \to p) \to p),$$

gdzie zmienna p jest nowa (patrz wyżej), a parę i rzuty tak:

- $\langle M, N\rangle = \Lambda p \lambda y^{\sigma\to\tau\to p}.yMN$;
- $\Pi_1(M) = M\sigma(\lambda x^{\sigma}\lambda y^{\tau}.x)$;
- $\Pi_2(M) = M\tau(\lambda x^{\sigma}\lambda y^{\tau}.y)$.

Definicja koniunkcji jest wzorowana na równoważności:

$$x \in A \cap B \Leftrightarrow \forall P(\forall y(y \in A \to y \in B \to y \in P) \to x \in P).$$

Łatwo widzieć, że ta definicja spełnia zarówno reguły typowania jak i redukcji.

Kwantyfikator szczegółowy

Zdaniowy kwantyfikator szczegółowy drugiego rzędu ma takie reguły naturalnej dedukcji:

$$(\exists\mathrm{I})\ \frac{\Gamma \vdash \sigma[p := \tau]}{\Gamma \vdash \exists p.\,\sigma}$$

$$(\exists\mathrm{E})\ \frac{\Gamma \vdash \exists p.\,\sigma \quad \Gamma \vdash \rho}{\Gamma, \sigma \vdash \rho}\ (p \notin \mathrm{FVT}(\Gamma) \cup \mathrm{FVT}(\rho))$$

Interpretujemy go tak: typ $\exists p.\,\sigma$ jest typem abstrakcyjnym, w którym zmienna p reprezentuje typ prywatny. Na przykład, jeśli

$$\mathit{stos}(p) = p \times (p \to p) \times (p \to \omega) \times (\omega \times p \to p),$$

to typ $\exists p\, stos(p)$ jest abstrakcyjnym typem struktury stosowej, w której mamy stos pusty jako stałą oraz operacje *pop*, *top* i *push*.)

Operacja **pack** tworzy obiekt typu abstrakcyjnego ukrywając prywatną część typu:

$$\text{(pack)}\ \frac{\Gamma \vdash M : \sigma[p := \tau]}{\Gamma \vdash \mathbf{pack}\ M, \tau\ \mathbf{to}\ \exists p.\, \sigma : \exists p.\, \sigma}$$

Taki obiekt może być użyty tam, gdzie implementacja typu prywatnego nie ma znaczenia:

$$\text{(unpack)}\ \frac{\Gamma \vdash M : \exists p.\, \sigma \quad \Gamma(x : \sigma) \vdash N : \rho}{\Gamma \vdash \mathbf{let}\ M\ \mathbf{be}\ x, p\ \mathbf{in}\ N : \rho}\ (p \notin \mathrm{FVT}(\Gamma) \cup \mathrm{FVT}(\rho))$$

Następująca reguła redukcji mówi o tym, co się wtedy dzieje:

$$\mathbf{let}\ (\mathbf{pack}\ M, \tau\ \mathbf{to}\ \exists p.\, \sigma)\ \mathbf{be}\ x, p\ \ \mathbf{in}\ N \longrightarrow_\beta N[p := \tau][x := M].$$

Definicja kwantyfikatora szczegółowego w systemie **F** przypomina definicję alternatywy, i może być objaśniona w podobny sposób. Mamy tu „wytarcie" definicji sumy uogólnionej wyrażonej w języku drugiego rzędu. Taka suma to najmniejszy zbiór zawierający wszystkie składniki. Poniżej, zmienna q musi być nowa, tj. $q \notin \mathrm{FVT}(\sigma) \cup \{p\}$.

$$\exists p\, \sigma = \forall q(\forall p(\sigma \to q) \to q).$$

Zauważmy, że powyższa definicja jest wzmocnieniem definicji klasycznej przez prawo de Morgana:

$$\neg\forall p \neg\sigma = \forall p(\sigma \to \bot) \to \bot$$

Teraz możemy zdefiniować **pack** i **let**:

- $\mathbf{pack}\ M, \tau\ \mathbf{to}\ \exists p.\, \sigma = \Lambda q \lambda z^{\forall p(\sigma \to q)}.z\tau M$;
- $\mathbf{let}\ M\ \mathbf{be}\ x, p\ \mathbf{in}\ N^\rho = M\rho(\Lambda p \lambda x^\sigma.N)$.

i sprawdzić, że wszystko działa jak trzeba.

Ćwiczenia

1. Zdefiniować w systemie **F** funkcję wykładniczą, poprzednik, dzielenie całkowite, etc.
2. Zdefiniować w systemie **F** funkcję Ackermanna.
3. Czy da się zdefiniować *każdą* funkcję obliczalną?

24 System F w stylu Curry'ego

Wariant systemu **F** w stylu Curry'ego określony jest przez następujące reguły przypisania typów dla termów „czystego" rachnku lambda. (Typy i otoczenia typowe określone są tak samo jak dla wersji Churcha.)

$$\Gamma(x : \tau) \vdash x : \tau$$

$$\text{(APP)}\ \frac{\Gamma \vdash N : \tau \to \sigma \quad \Gamma \vdash M : \tau}{\Gamma \vdash NM : \sigma} \qquad \frac{\Gamma(x : \tau) \vdash M : \sigma}{\Gamma \vdash \lambda x.M : \tau \to \sigma}\ \text{(ABS)}$$

$$\text{(GEN)}\ \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \forall p\, \sigma}\ (p \notin \mathrm{FVT}(\Gamma)) \qquad \frac{\Gamma \vdash M : \forall p\, \sigma}{\Gamma \vdash M : \sigma[p := \tau]}\ \text{(INST)}$$

Reguły dla abstrakcji i aplikacji są takie same jak dla rachunku z typami prostymi w stylu Curry'ego. Reguły wprowadzania i eliminacji kwantyfikatora ogólnego (zwane odpowiednio regułami *generalizacji* and *instancjacji*) reprezentują ideę *polimorfizmu implicite*: obiekt o typie uniwersalnym $\forall p\, \tau$ ma wszystkie typy postaci $\tau[p := \sigma]$.

Definicja 24.1 Operacja *wycierania typów* jest określona równaniami:

$$\begin{aligned} |x| &= x \\ |MN| &= |M||N| \\ |\lambda x{:}\sigma.\, M| &= \lambda x.\, |M| \\ |\Lambda p.\, M| &= |M| \\ |M\tau| &= |M| \end{aligned}$$

Fakt 24.2 *Osąd $\Gamma \vdash M : \tau$ jest wyprowadzalny w stylu Curry'ego wtedy i tylko wtedy, gdy istnieje taki term M_0 w stylu Churcha, że $|M_0| = M$, oraz $\Gamma \vdash M_0 : \tau$.*

Dowód: Łatwy. ∎

O termie N w stylu Churcha można myśleć jak o wyprowadzeniu typu dla termu $|N|$. Następujące twierdzenie w istocie stwierdza, że każda redukcja w $|N|$ odpowiada pewnej redukcji w N.

Twierdzenie 24.3 *Jeśli $\Gamma \vdash M : \tau$ w styly Curry'ego, oraz $M \to_\beta M'$ to $\Gamma \vdash M' : \tau$.*

Zanim zajmiemy się dowodem twierdzenia 24.3, zauważmy, że sprawa wcale nie jest taka oczywista.

Przykład 24.4 Rozpatrzmy następujące przypisanie typu:

$$x : p \to \forall q\, (q \to q) \vdash \lambda y.\, xy : p \to q \to q.$$

Chociaż mamy eta-redukcję $\lambda y.\, xy \to_\eta x$, to jednak:

$$x : p \to \forall q\, (q \to q) \nvdash x : p \to q \to q.$$

Zauważmy, że termowi $\lambda y.\, xy$ odpowiada w naszym przykładzie term Churcha $\lambda y{:}p.\, xyq$, który nie jest eta-redeksem. A więc przyczyną tego, że system **F** w stylu Curry'ego nie jest zamknięty ze względu na η-redukcje jest to, że wycieranie typów odsłania „nowe" redeksy. John Mitchell pokazał, że określając odpowiednią relację $\sqsubseteq$ i rozszerzając system o regułę

$$\frac{\Gamma \vdash M : \tau, \quad \tau \sqsubseteq \sigma}{\Gamma \vdash M : \sigma,}$$

otrzymuje się rachunek zamknięty ze względu na η-redukcje.

Dowód twierdzenia 24.3

Na rozgrzewkę mamy takie proste obserwacje:

Lemat 24.5 *Dla termów w stylu Churcha:*

1. $|M[p := \tau]| = |M|$ *oraz* $|M[x := N]| = |M|[x := |N|]$;

2. *Jeśli* $M \to_\beta N$ *to* $|M| \to_\beta |N|$.

Dowód twierdzenia 24.3 wymaga oczywiście odpowiedniego lematu o generowaniu. Aby taki lemat sformułować potrzebujemy pewnej pomocniczej relacji na typach. Piszemy $\sigma \preceq \tau$, gdy typ σ jest postaci $\forall \vec{p}\,\sigma'$, gdzie $\forall \vec{p}$ jest ciągiem kwantyfikatorów uniwersalnych, typ σ' nie zaczyna się od kwantyfikatora, oraz zachodzi równość $\tau = \forall \vec{q}.\sigma'[\vec{p} := \vec{\rho}]$, dla pewnych typów $\vec{\rho}$, i pewnych zmiennych $\vec{q}$, nie występujących wolno w σ. Wtedy $\Gamma \vdash M : \sigma$ implikuje $\Gamma \vdash M : \tau$ i wystarczą do tego reguły (GEN) i (INST). Mamy też nieco słabsze twierdzenie odwrotne:

Lemat 24.6 *Jeśli osąd* $\Gamma \vdash M : \tau$ *otrzymano z* $\Gamma \vdash M : \sigma$ *za pomocą ciągu zastosowań reguł* (GEN) *i* (INST)*, to* $\forall \vec{p}\,\sigma \preceq \forall \vec{q}\,\tau$*, gdzie* $\vec{p} = \mathrm{FVT}(\sigma) - \mathrm{FVT}(\Gamma)$ *oraz* $\vec{q} = \mathrm{FVT}(\tau) - \mathrm{FVT}(\Gamma)$.

Dowód: Indukcja ze względu na liczbę kroków. ∎

Lemat 24.7 *Jeśli* $\Gamma \vdash M : \tau$ *to* $\Gamma[p := \rho] \vdash M : \tau[p := \rho]$.

Dowód: Indukcja ze względu na rozmiar wyprowadzenia $\Gamma \vdash M : \tau$ ∎

Lemat 24.8 (o generowaniu)

1. *Jeśli* $\Gamma \vdash x : \tau$ *to* $\Gamma(x) \preceq \tau$.
2. *Jeśli* $\Gamma \vdash MN : \tau$ *to* $\Gamma \vdash M : \varrho \to \sigma$ *oraz* $\Gamma \vdash N : \varrho$ *dla pewnych* ϱ *i* σ*, takich że* $\forall \vec{p}\,\sigma \preceq \tau$*, gdzie* $\vec{p} = \mathrm{FVT}(\sigma) - \mathrm{FVT}(\Gamma)$.
3. *Jeśli* $\Gamma \vdash \lambda x\, M : \tau$*, to* $\tau = \forall \vec{p}(\varrho \to \sigma)$*, gdzie* $\Gamma(x : \varrho) \vdash M : \sigma$*, a* $\vec{p}$ *nie są wolne w* Γ.

Dowód: Wyprowadzenie typu dla zmiennej zaczyna się zawsze od aksjomatu $\Gamma \vdash x : \Gamma(x)$, po którym następuje ciąg zastosowań „stacjonarnych" reguł (GEN) i (INST). Teza wynika więc bezpośrednio z lematu 24.6. Podobnie jest dla aplikacji. W przypadku abstrakcji trzeba dodatkowo skorzystać z lematu 24.7. ∎

Poniższe reguły tworzą system przypisania typów, który jest „syntaktycznie zorientowany" w takim sensie: ostatnia reguła użyta w wyprowadzeniu typu dla termu M jest wyznaczona przez postać termu M.

(VAR') $\Gamma \vdash x : \tau$ jeśli $(x : \sigma)$ is in Γ i $\sigma \preceq \tau$

(APP') $$\frac{\Gamma \vdash M : \tau \to \rho, \quad \Gamma \vdash N : \tau}{\Gamma \vdash (MN) : \sigma}$$ jeśli $\forall \vec{p}\, \rho \preceq \sigma$, gdzie $\vec{p} = \mathrm{FVT}(\rho) - \mathrm{FVT}(\Gamma)$

(ABS') $$\frac{\Gamma(x:\tau) \vdash M : \sigma}{\Gamma \vdash \lambda x.M : \forall \vec{p}(\tau \to \sigma)}$$ jeśli $\vec{p} \cap \mathrm{FVT}(\Gamma) = \varnothing$

Konsekwencją lematu o generowaniu jest:

Lemat 24.9 *Osąd $\Gamma \vdash M : \sigma$ jest wyprowadzalny w* **F** *wtedy i tylko wtedy, gdy jest wyprowadzalna w systemie jak wyżej.*

Dowód twierdzenia: Przypuśćmy teraz, że $\Gamma \vdash (\lambda x.M)N : \sigma$. Na mocy lematu 24.9 mamy $\Gamma, x{:}\tau \vdash M : \rho$ oraz $\Gamma \vdash N : \tau$, dla pewnych τ i ρ. Na dodatek $\forall \vec{p}\, \rho \preceq \sigma$, gdzie $\vec{p} = \mathrm{FVT}(\rho) - \mathrm{FVT}(\Gamma)$. Stąd nietrudno wywnioskować $\Gamma \vdash M[x := N] : \rho$ a to z kolei implikuje $\Gamma \vdash M[x := N] : \sigma$.

Termy typowalne i nie

Używając polimorfizmu, można przypisywać typy rozmaitym termom, które nie są typowalne w typach prostych. Na przykład termy $\lambda x.\, xx$ i **2K** są teraz typowalne. Jak się zaraz okaże, termy typowalne mają własność SN, ale nie na odwrót: kontrprzykładem jest silnie normalizowalny term

$$(\lambda zy.\, y(z\mathbf{I})(z\mathbf{K}))(\lambda x.\, xx),$$

który nie jest typowalny w systemie **F**. Przyczyną jest niemożność znalezienia takiego typu dla $(\lambda x.\, xx)$, który „pasowałby" zarówno do argumentu **I** jak i **K**. Innym przykładem jest **22K**. Zauważmy jednak, że term **2**(**2K**) jest typowalny.

Niestety, zachodzi następujące twierdzenie:

Twierdzenie 24.10 (Wells) *Problem typowalności*[25] *i problem sprawdzenia typu*[26] *są dla systemu* **F** *nierozstrzygalne.*

Ćwiczenia

1. Pokazać, że każda postać normalna jest typowalna w systemie **F**.
2. Wyprowadzić typy dla termów $\lambda x.\, xx$, **2K** oraz **2**(**2K**).
3. Czy w części 2 lematu 24.8 można żądać aby $\sigma \preceq \tau$?

[25] Dany term M, czy istnieją takie Γ i τ, że $\Gamma \vdash M : \tau$?
[26] Dane są Γ, M i τ. Czy $\Gamma \vdash M : \tau$?

25 Silna normalizacja

Naiwna próba uogólnienia metody Taita na typy polimorficzne polega na zdefiniowaniu:

$$[\![\forall p\,\tau]\!] = \bigcap\{[\![\tau[p := \sigma]]\!] \mid \sigma \text{ — dowolny typ}\}.$$

Ale ta definicja nie jest dobrze ufundowana: nie da się zdefiniować $[\![\forall p\,\tau]\!]$ bez wcześniejszego zdefiniowania np. $[\![\tau[p := \forall p\,\tau]]\!]$, co z kolei wymaga definicji $[\![\forall p\,\tau]\!]$. Pomysł Girarda, który wykorzystamy, zwany „metodą kandydatów", polega na tym, aby nie definiować zbiorów $[\![\tau]\!]$ w sposób „absolutny", w szczególności nie przyjmować z góry $[\![p]\!] := \mathrm{SN}$ dla zmiennych. Zamiast tego należy określić warunki jakie zbiory $[\![\tau]\!]$ powinny spełniać i rozważać dowolne wybory zbiorów $[\![p]\!]$, jak gdyby to były „wartościowania" zmiennych typowych.

Powiemy więc, że zbiór termów X jest *kandydatem*[27] (albo, że jest *nasycony*, jest *rodziną normalną*), jeżeli X ma trzy własności z lematów 14.7–14.9:

1) $X \subseteq \mathrm{SN}$.
2) Jeśli $N_1, \ldots, N_k \in \mathrm{SN}$ to $xN_1 \ldots N_k \in X$.
3) Jeśli $M[x := N_0]N_1 \ldots N_k \in X$, oraz $N_0 \in \mathrm{SN}$ to $(\lambda x.M)N_0N_1 \ldots N_k \in X$.

Niech $\mathcal{G}$ będzie rodziną wszystkich kandydatów. Nietrudno zauważyć, że $\mathrm{SN} \in \mathcal{G}$ a zatem $\mathcal{G} \neq \varnothing$. *Wartościowaniem* w $\mathcal{G}$ nazwiemy dowolną funkcję $\xi : U \to \mathcal{G}$. Teraz dla dowolnego wartościowania ξ i dowolnego typu τ definiujemy zbiór $[\![\tau]\!]_\xi$:

$$\begin{array}{lcl} [\![p]\!]_\xi & = & \xi(p); \\ [\![\sigma \to \tau]\!]_\xi & = & \{M \mid \forall N(N \in [\![\sigma]\!]_\xi \Rightarrow MN \in [\![\tau]\!]_\xi)\}; \\ [\![\forall p.\sigma]\!]_\xi & = & \bigcap_{X \in \mathcal{G}} [\![\sigma]\!]_{\xi(p \mapsto X)}. \end{array}$$

Lemat 25.1 *Dla dowolnych ξ i σ zachodzi $[\![\sigma]\!]_\xi \in \mathcal{G}$.*

Dowód: Dowód jest przez indukcję ze względu na τ. Jeśli τ jest zmienną to $[\![\tau]\!]_\xi = \mathrm{SN} \in \mathcal{G}$ i dobrze. Przypadek $\tau = \forall p\,\sigma$ wynika z prostej obserwacji: iloczyn rodziny kandydatów jest kandydatem. Niech więc $\tau = \sigma \to \rho$ i niech $M \in [\![\sigma \to \rho]\!]_\xi$. Weźmy zmienną $x : \sigma$. Z założenia indukcyjnego (2) mamy $x \in [\![\sigma]\!]_\xi$, więc z definicji $Mx \in [\![\rho]\!]_\xi$. A więc $Mx \in \mathrm{SN}$, z założenia indukcyjnego (1). Tym bardziej $Mx \in \mathrm{SN}$. Pokazaliśmy (1).

W punkcie (2) mamy pokazać, że $xN_1 \ldots N_kP \in [\![\rho]\!]_\xi$, dla każdego $P \in [\![\sigma]\!]_\xi$. Ale $P \in \mathrm{SN}$ z założenia indukcyjnego (1), więc teza wynika z założenia indukcyjnego (2) dla ρ.

Zostaje warunek (3). Niech więc $M[x := N_0]N_1 \ldots N_k \in [\![\sigma \to \rho]\!]_\xi$, oraz $N_0 \in \mathrm{SN}$. Mamy pokazać, że dla $P \in [\![\sigma]\!]_\xi$ musi zachodzić $(\lambda x.M)N_0N_1 \ldots N_kP \in [\![\rho]\!]_\xi$. Ale to wynika z założenia indukcyjnego (3) dla ρ. ∎

[27] Oryginalna nazwa: *candidat de reductibilité*

Lemat 25.2

- $[\![\sigma[p := \tau]]\!]_\xi = [\![\sigma]\!]_{\xi(p \mapsto ([\![\tau]\!]_\xi)}$.
- $[\![\sigma]\!]_\xi = [\![\sigma]\!]_{\xi'}$, *gdy* $\xi(p) = \xi'(p)$ *dla* $p \in \mathrm{FVT}(\sigma)$.

Dowód: Indukcja ze względu na σ. ∎

Lemat 25.3 *Niech* $\Gamma \vdash M : \tau$ *oraz* $\mathrm{FV}(M) = \{x_1, \ldots, x_k\}$ *i przy tym* $N_i \in [\![\Gamma(x_i)]\!]_\xi$, *dla* $i = 1, \ldots, k$. *Wtedy* $M[x_1 := N_1, \ldots, x_k := N_k] \in [\![\tau]\!]_\xi$.

Dowód: Dowód jest przez indukcję ze względu na wyprowadzenie $\Gamma \vdash M : \tau$, przez przypadki w zależności od ostatniej użytej reguły.

(VAR) Przypuśćmy, że $\Gamma \vdash x_i : \Gamma(x_i)$. Wtedy term $M[x_1 := N_1, \ldots, x_k := N_k]$ to po prostu N_i, więc teza wynika wprost z założenia.

(ABS) Przypuśćmy, że $\Gamma \vdash \lambda x.M : \sigma \to \rho$ otrzymano z przesłanki $\Gamma, x : \sigma \vdash M : \rho$. Niech $P \in [\![\sigma]\!]_\xi$. Mamy pokazać, że $((\lambda x.M)[x_1 := N_1, \ldots, x_k := N_k])P \in [\![\rho]\!]_\xi$. Ponieważ możemy założyć, że x nie jest wolne w termach N_i, więc mamy

$$(\lambda x.M)[x_1 := N_1, \ldots, x_k := N_k] = \lambda x.M[x_1 := N_1, \ldots, x_k := N_k].$$

Z założenia indukcyjnego

$$M[x_1 := N_1, \ldots, x_k := N_k][x := P] = M[x_1 := N_1, \ldots, x_k := N_k, x := P] \in [\![\rho]\!]_\xi,$$

więc teza wynika z warunku (3) definicji kandydata.

(APP) Przypuśćmy, że $\Gamma \vdash MN : \tau$ otrzymano z przesłanek $\Gamma \vdash M : \sigma \to \tau$ i $\Gamma \vdash N : \sigma$. Z założenia indukcyjnego otrzymujemy, że $M[x_1 := N_1, \ldots, x_k := N_k] \in [\![\sigma \to \tau]\!]_\xi$ a także $N[x_1 := N_1, \ldots, x_k := N_k] \in [\![\sigma]\!]_\xi$. Stąd $(MN)[x_1 := N_1, \ldots, x_k := N_k] \in [\![\tau]\!]_\xi$.

(GEN) Przypuśćmy, że $\Gamma \vdash M : \forall p\tau$ otrzymano z przesłanki $\Gamma \vdash M : \tau$, i przy tym p nie należy do $\mathrm{FVT}(\Gamma)$. Niech $M' = M[x_1 := N_1, \ldots, x_k := N_k]$, gdzie $N_i \in [\![\Gamma(x_i)]\!]_\xi$, dla $i = 1, \ldots, k$. Z założenia indukcyjnego $M' \in [\![\tau]\!]_\nu$ przy dowolnym ν, spełniającym warunek $N_i \in [\![\Gamma(x_i)]\!]_\nu$, dla $i = 1, \ldots, k$. Skoro $p \notin \mathrm{FVT}(\Gamma(x_i))$ to $[\![\Gamma(x_i)]\!]_\xi = [\![\Gamma(x_i)]\!]_{\xi(p \mapsto X)}$, dla dowolnych i, X. Zatem $M' \in [\![\tau]\!]_{\xi(p \mapsto X)}$ dla dowolnego $X \in \mathcal{G}$, a więc $M' \in [\![\forall p\, \tau]\!]_\xi$.

(INST) Przypuśćmy, że $\Gamma \vdash M : \sigma[p := \tau]$ otrzymano z $\Gamma \vdash M : \forall p\, \sigma$. Z założenia indukcyjnego $M[x_1 := N_1, \ldots, x_k := N_k] \in [\![\forall p \sigma]\!]_\xi$, a więc także $M[x_1 := N_1, \ldots, x_k := N_k] \in [\![\sigma]\!]_{\xi(p \mapsto [\![\tau]\!]_\xi)} = [\![\sigma[p := \tau]]\!]_\xi$ i już. ∎

Twierdzenie 25.4 *System* **F** *w wersji Curry'ego ma własność silnej normalizacji.*

Dowód: Jeśli $\Gamma \vdash M : \tau$ dla pewnych Γ i τ to z poprzedniego lematu wynika, że dla dowolnego ξ mamy $M \in [\![\tau]\!]_\xi$, bo $x \in [\![\Gamma(x)]\!]_\xi$ dla dowolnej zmiennej x. Teza wynika więc stąd, że $[\![\tau]\!]_\xi \subseteq \mathrm{SN}$. ∎

Twierdzenie 25.5 *System* **F** *w wersji Churcha ma własność silnej normalizacji.*

Dowód: Przypuśćmy, że mamy nieskończony ciąg redukcji

$$M = M_0 \to M_1 \to M_2 \to \cdots$$

Nietrudno zauważyć, że wtedy

$$|M| = |M_0| \succ |M_1| \succ |M_2| \succ \cdots$$

gdzie $|M_i|$ są jak w definicji 24.1, a symbol $\succ$ oznacza zawsze $\to$ lub $=$. Ponieważ termy typowalne w systemie **F** w wersji Curry'ego mają własność SN (twierdzenie 25.4) więc poczynając od pewnego miejsca, termy $|M_n|$ muszą być identyczne. To znaczy, że prawie wszystkie redukcje w naszym ciągu są „typowe", tj. postaci $(\Lambda p.N)\tau \longrightarrow_q N[p := \tau]$. Skoro każdorazowo znika jedno wystąpienie symbolu Λ, to nasz ciąg nie może być nieskończony. ∎

Twierdzenie 25.6 *System* **F** *ma własność Churcha-Rossera.*[28]

Dowód: Własność Churcha-Rossera wynika z lematu Newmana (silna normalizacja wraz ze słabą własnością Churcha-Rossera implikuje własność Churcha-Rossera). ∎

Ćwiczenia

1. Czy twierdzenie 25.6 wynika w oczywisty sposób z twierdzenia Churcha-Rossera dla beztypowego rachunku lambda?
2. Czy równość dwóch termów systemu **F** w stylu Churcha jest rozstrzygalna? A w stylu Curry'ego?

26 System $\mathbf{F}_\omega$

O typie τ, który ma zmienną wolną p, można myśleć jak o operacji $\lambda p\,\tau$, która zaaplikowana do argumentu σ daje w wyniku pewien typ $\tau(\sigma)$. Ten sposób myślenia został zalegalizowany w systemie $\mathbf{F}_\omega$, zwanym polimorficznym rachunkiem lambda wyższego rzędu. Taką operację $\lambda p\,\tau$ nazywamy tutaj *konstruktorem typowym*. Raz dopuściwszy operacje na typach, nie widzimy nic zdrożnego we wprowadzeniu także operacji na konstruktorach, operacji na takich operacjach itd.

Formalizację powyższego zaczynamy od pojęcia *rodzaju* (ang. *kind*). Mamy stałą $*$ (rodzaj wszystkich typów) a ponadto jeśli ∇_1 i ∇_2 są rodzajami, to $\nabla_1 \Rightarrow \nabla_2$ jest rodzajem. Dla każdego rodzaju ∇, definiujemy *konstruktory* tego rodzaju przez indukcję:

- *Zmienna konstruktorowa* rodzaju ∇ jest konstruktorem rodzaju ∇.

[28] W obu wersjach.

- Jeśli φ jest konstruktorem rodzaju $\nabla_1 \Rightarrow \nabla_2$ i τ jest konstruktorem rodzaju∇_1, to $\varphi\tau$ jest konstruktorem rodzaju ∇_2.
- Jeśli α jest zmienną konstruktorową rodzaju ∇_1 i τ jest konstruktorem rodzaju ∇_2, to $\lambda\alpha\,\tau$ jest konstruktorem rodzaju $\nabla_1 \Rightarrow \nabla_2$.
- Jeśli α jest zmienną konstruktorową dowolnego rodzaju i τ jest konstruktorem rodzaju $*$, to $\forall\alpha\,\tau$ jest konstruktorem rodzaju $*$.
- Jeśli τ i σ są konstruktorami rodzaju $*$, to $\tau \to \sigma$ jest konstruktorem rodzaju $*$.

Piszemy $\tau : \nabla$ na oznaczenie tego, że τ jest konstruktorem rodzaju ∇. Konstruktory rodzaju $*$ nazywamy *typami*. Jak dla **F** podstawienie (konstruktora na zmienną konstruktorową) definiujemy przez indukcję.

- $\alpha[\alpha := \varphi] = \varphi$;
- $\beta[\alpha := \varphi] = \beta$;
- $(\vartheta\psi)[\alpha := \varphi] = \vartheta[\alpha := \varphi]\psi[\alpha := \varphi]$;
- $(\sigma \to \rho)[\alpha := \varphi] = \sigma[\alpha := \varphi] \to \rho[\alpha := \varphi]$;
- $(\forall\alpha\,\sigma)[\alpha := \varphi] = \forall\alpha\,\sigma$;
- $(\forall\beta\,\sigma)[\alpha := \varphi] = \forall\beta\,\sigma$, jeśli $\alpha \notin FV(\sigma)$;
- $(\forall\beta\,\sigma)[\alpha := \varphi] = \forall\beta\,\sigma[\alpha := \varphi]$, jeśli $\alpha \in FV(\sigma)$ i $\beta \notin FV(\varphi)$;
- $(\lambda\alpha\,\vartheta)[\alpha := \varphi] = \lambda\alpha\,\vartheta$;
- $(\lambda\beta\,\vartheta)[\alpha := \varphi] = \lambda\beta\,\vartheta$, jeśli $\alpha \notin FV(\vartheta)$;
- $(\lambda\beta\,\vartheta)[\alpha := \varphi] = \lambda\beta\,\vartheta[\alpha := \varphi]$, jeśli $\alpha \in FV(\vartheta)$ i $\beta \notin FV(\varphi)$;

Zauważmy, że mamy teraz dwie operacje wiążące zmienne konstruktorowe i typowe: lambdę i kwantyfikator. Alfa-konwersja jest określona warunkami

- $\forall\alpha\,\tau =_\alpha \forall\beta\,(\tau[\alpha := \beta])$, gdzie β nie jest wolne w τ;
- $\lambda\alpha\,\tau =_\alpha \lambda\beta\,(\tau[\alpha := \beta])$, gdzie β nie jest wolne w τ; β does not occur free in τ;
- $\tau \to \rho =_\alpha \tau' \to \rho'$ i $\forall\alpha\,\tau =_\alpha \forall\alpha\,\tau'$ gdzie $\tau =_\alpha \tau'$ i $\rho =_\alpha \rho'$;
- $\tau\rho =_\alpha \tau'\rho'$ i $\lambda\alpha\,\tau =_\alpha \lambda\alpha\,\tau'$, gdzie $\tau =_\alpha \tau'$ i $\rho =_\alpha \rho'$;

Oczywiście utożsamiamy alfa-równoważne konstruktory.

Na konstruktorach określa się beta-redukcję:

$$(\beta) \qquad (\lambda\alpha\ \sigma)\tau \Longrightarrow \sigma[\alpha := \tau].$$

Z silnej normalizacji dla typów skończonych łatwo wywnioskować, że każdy konstruktor jest silnie normalizowalny. Własność Churcha-Rossera też zachodzi. Przez $nf(\sigma)$ oznaczymy postać normalną typu σ.

Termy Churcha

System $\mathbf{F}_\omega$ w wersji Churcha różni się od systemu $\mathbf{F}$ tym, że zamiast aplikacji i abstrakcji typowej mamy aplikację i abstrakcję konstruktorową. Jeśli więc $M : \forall\alpha\,\sigma$, gdzie $\alpha : \nabla$, to dopuszczamy termy postaci $M\varphi : nf(\sigma[\alpha := \varphi])$, gdzie φ jest dowolnym konstruktorem rodzaju ∇. A jeśli zmienna konstruktorowa $\alpha : \nabla$ nie występuje wolno w typie żadnej zmiennej wolnej termu M, to możemy utworzyć polimorficzną abstrakcję $\Lambda\alpha\,M$ typu $\forall\alpha\,M$. Tak określony rachunek ma własność Churcha-Rossera i własność silnej normalizacji.

Termy Curry'ego

System $\mathbf{F}_\omega$ w wersji Curry'ego jest zadany następującymi regułami przypisania typów.

$$\text{VAR} \qquad \Gamma \vdash x : \sigma \qquad \text{gdy} \quad \Gamma(x) = \sigma$$

$$\text{APP} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash (M\ N) : \tau}$$

$$\text{ABS} \qquad \frac{\Gamma(x : \sigma) \vdash M : \tau}{\Gamma \vdash (\lambda x\ M) : \sigma \to \tau}$$

$$\text{INST} \qquad \frac{\Gamma \vdash M : \forall\alpha\,\sigma}{\Gamma \vdash M : nf(\sigma[\alpha := \tau])}$$

$$\text{GEN} \qquad \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \forall\alpha\,\sigma} \quad \alpha \notin \mathrm{FV}(\Gamma)$$

Zamiast normalizacji typu w regule (INST) można przyjąć dodatkową regułę konwersji:

$$\text{CNV} \qquad \frac{\Gamma \vdash M : \sigma}{\Gamma \vdash M : \tau} \qquad \text{gdy} \quad \sigma =_\beta \tau$$

Przykład

Rozpatrzmy następujący term

$$M = (\lambda x\ xx\mathbf{K})2,$$

gdzie $\mathbf{K} = \lambda xy.\,x$ i $2 = \lambda fy.\,f(fy)$. Ten term jest typowalny w $\mathbf{F}_\omega$, a jedyne nietrywialne rodzaje jakich potrzebujemy to $* \Rightarrow *$ i $* \Rightarrow * \Rightarrow *$. Uwaga: ten przykład jest trudniejszy niż term $22\mathbf{K}$, bo mamy tylko jedną dwójkę.

Poniżej p i q są zmiennymi typowymi, δ, γ i β są zmiennymi konstruktorowymi rodzaju $* \Rightarrow *$, a ξ jest zmienną konstruktową rodzaju $* \Rightarrow * \Rightarrow *$.

Niech

$$\kappa = \forall p\delta\,(p \to q \to p),$$

i niech

$$\mathcal{L} = \forall p\gamma\,(\xi p(\forall p\delta\,(p \to \gamma p)) \to \xi(\beta p)(\forall p\delta\,(p \to \gamma(\gamma p)))).$$

Rozpatrzmy otoczenie złożone z deklaracji:

$$\{y : \xi p\kappa,\ f : \mathcal{L}\}$$

W tym otoczeniu wyprowadzimy

$$fy : \xi(\beta p)(\forall p\delta\,(p \to q \to q \to p)),$$

bo zmienna γ w typie f może być zastąpiona konstruktorem $\lambda p.\ q \to p$. Inna możliwość to podstawienie na γ konstruktora $\lambda p.\ q \to q \to p$, oraz podstawienie βp na p. To pozwala wyprowadzić

$$f(fy) : \xi(\beta(\beta p))\sigma,$$

gdzie

$$\sigma = \forall p\delta\,(p \to q \to q \to q \to q \to p).$$

Zauważmy, że p i γ nie są wolne w $\mathcal{L}$, więc w pustym otoczeniu mamy:

$$2 : \forall \xi p\beta\,(\mathcal{L} \to \forall p\gamma\,(\xi p\kappa \to \xi(\beta(\beta p))\sigma).$$

Oznaczmy powyższy typ przez τ. Chcemy znaleźć typ dla termu $xx\mathbf{K}$ w otoczeniu, w którym zmienna x ma typ τ. Typem $\mathbf{K}$ będzie oczywiście κ. Typ drugiego wystąpienia x otrzymamy jako instancję τ: podstawimy na ξ rzutowanie typowe $\lambda p_1 p_2.\, p_1$. Otrzymamy typ $\forall p\beta\,(\forall p\gamma\,(p \to \beta p) \to \forall p\gamma\,(p \to \beta(\beta p)))$, alfa-równoważny typowi

$$\tau_2 = \forall p\gamma\,(\forall p\delta\,(p \to \gamma p) \to (\forall p\delta\,(p \to \gamma(\gamma p)))).$$

Pierwsze wystąpienie x otrzyma typ otrzymany z τ przez podstawienie drugiego rzutowania $\lambda p_1 p_2.\, p_2$, a mianowicie typ $\forall p\beta\,(\tau_2 \to \forall p\gamma(\kappa \to \forall p\delta\,(p \to q \to q \to q \to q \to p)))$. Pozostaje pozbyć się z tego typu początkowych kwantyfikatorów (za pomocą trywialnego podstawienia) i mamy taki typ dla x:

$$\tau_1 = \tau_2 \to \forall p\gamma(\kappa \to \sigma).$$

Poprawne typowanie dla $xx\mathbf{K}$ jest już łatwe.

Drugi przykład

Teraz (już bez dowodu) przykład termu, który ma własność silnej normalizacji, ale nie jest typowalny w $\mathbf{F}_\omega$. Oto on:

$$(\lambda x.\, z(x1)(x1'))(\lambda y.\, yyy)$$

Powyżej 1 oznacza $\lambda fu.\, fu$, a $1'$ oznacza $\lambda vg.\, gv$.

Nierozstrzygalność

Fakt 26.1 *Następujące problemy są nierozstrzygalne dla systemu* $\mathbf{F}_\omega$*:*

- *Sprawdzanie typu: dane* Γ, M, τ*, pytamy czy* $\Gamma \vdash M : \tau$*?*
- *Typowalność: dane* M*, pytamy czy istnieją takie* Γ, τ*, że* $\Gamma \vdash M : \tau$*?*

Szkic dowodu części pierwszej: Skorzystamy z nierozstrzygalności problemu unifikacji drugiego rzędu (twierdzenie 16.6). Wiadomo, że problem ten jest już nierozstrzygalny dla sygnatury złożonej z jednego dwuargumentowego symbolu funkcyjnego $\to$ (w notacji infiksowej) i dwóch stałych α i β. Co więcej, bez straty ogólności można ograniczyć się do pojedynczych równań postaci $\tau = \rho$, gdzie niewiadome (dla uproszczenia wszystkie niewiadome są dwuargumentowymi symbolami funkcyjnymi) występujące w τ i ρ tworzą rozłączne zbiory. (Zauważmy bowiem, że dla dwuargumentowych niewiadomych ζ i ζ', równania $\zeta\alpha\beta = \zeta'\alpha\beta$ i $\zeta\beta\alpha = \zeta'\beta\alpha$ „wymuszają" równość $\zeta = \zeta'$, tj. odpowiednie współrzędne każdego rozwiązania muszą być równe).

Załóżmy więc, ze dane jest równanie $\tau = \rho$, gdzie niewiadome $\zeta_1, \ldots, \zeta_k$ występują w τ, a niewiadome $\zeta_{k+1}, \ldots, \zeta_n$ występują w ρ. Pytanie o rozwiązanie tego równania sprowadzamy do pytania o to, czy termowi $\mathbf{K}y(x(xy))$ można przypisać typ p w otoczeniu złożonym z deklaracji $(y : \forall pp)$ i $(x : \forall\zeta_1, \ldots, \zeta_n(\tau \to \rho))$. Jest tak wtedy i tylko wtedy, gdy możliwa jest taka instancjacja zmiennych ζ_i, przy której możliwe jest składanie x ze sobą, tj. wtedy, gdy nasze równanie ma rozwiązanie. ∎

Twierdzenie 26.2 *Problem niepustości typu w systemie* $\mathbf{F}_\omega$ *jest nierozstrzygalny.*

Dla dowodu twierdzenia 26.2 przypomnimy pojęcie *automatu dwulicznikowego*. Można go zdefiniować jako trójkę uporządkowaną $\mathcal{M} = \langle Q, q_0, q_f, \delta\rangle$, gdzie Q to oczywiście zbiór stanów, q_0 i q_f to stan początkowy i końcowy, a funkcja przejścia δ przypisuje każdemu stanowi (oprócz q_f) pewną *instrukcję*. Instrukcje mogą być takie (dla $i = 1, 2$ oraz $p, p' \in Q$):

1. $c_1 := c_1 + 1$; **goto** p;
2. $c_2 := c_2 + 1$; **goto** p;
3. $c_1 := c_1 - 1$; **goto** p;
4. $c_2 := c_2 - 1$; **goto** p;
5. **if** $c_1 = 0$ **then goto** p **else goto** p';
6. **if** $c_2 = 0$ **then goto** p **else goto** p'.

Konfiguracja automatu to trójka $\langle q, m, n\rangle$, gdzie $q \in Q$ oraz $m, n \in \mathbb{N}$. Konfiguracja *początkowa* ma postać $\langle q_0, 0, 0\rangle$, a konfiguracje postaci $\langle q_f, m, n\rangle$ są *końcowe*. Relacja przejścia $\to_\mathcal{M}$ między konfiguracjami jest określona tak:

- $\langle q, m, n\rangle \to_\mathcal{M} \langle p, m + 1, n\rangle$, gdy $\delta(q)$ jest postaci (1);

- $\langle q, m, n\rangle \to_{\mathcal{M}} \langle p, m, n+1\rangle$, gdy $\delta(q)$ jest postaci (2);
- $\langle q, m+1, n\rangle \to_{\mathcal{M}} \langle p, m, n\rangle$, gdy $\delta(q)$ jest postaci (3);
- $\langle q, m, n+1\rangle \to_{\mathcal{M}} \langle p, m, n\rangle$, gdy $\delta(q)$ jest postaci (4);
- $\langle q, 0, n\rangle \to_{\mathcal{M}} \langle p, 0, n\rangle$, gdy $\delta(q)$ jest postaci (5);
- $\langle q, m+1, n\rangle \to_{\mathcal{M}} \langle p', m+1, n\rangle$, gdy $\delta(q)$ jest postaci (5);
- $\langle q, m, 0\rangle \to_{\mathcal{M}} \langle p, m, 0\rangle$, gdy $\delta(q)$ jest postaci (6);
- $\langle q, m, n+1\rangle \to_{\mathcal{M}} \langle p', m, n+1\rangle$, gdy $\delta(q)$ jest postaci (6);

Jak widać, wykonanie instrukcji (1) i (2) jest niemożliwe w przypadku gdy odpowiedni licznik ma wartość zero. Jak zwykle $\twoheadrightarrow_{\mathcal{M}}$ oznacza domknięcie przechodnio-zwrotne relacji $\to_{\mathcal{M}}$. Mówimy, że maszyna $\mathcal{M}$ *zatrzymuje się*, gdy $\langle q_0, 0, 0\rangle \twoheadrightarrow_{\mathcal{M}} \langle q_f, m, n\rangle$, dla pewnych m, n.

Twierdzenie 26.3 *Problem stopu dla automatów dwulicznikowych (czy dany automat się zatrzymuje?) jest nierozstrzygalny.*

Szkic dowodu: Naturalnym uogólnieniem pojęcia automatu dwulicznikowego jest automat z dowolną liczbą liczników. Nietrudno pokazać, że obliczenie dowolnej maszyny Turinga może być symulowane przez pewien automat licznikowy (przy pomocy odpowiedniego kodowania zawartości taśmy, położenia głowicy itd. jako liczb naturalnych). Pozostaje więc pokazać, jak zastąpić np. pięć liczników przez dwa.

Najpierw zauważmy, że używając drugiego licznika do celów pomocniczych możemy pomnożyć wartość licznika pierwszego przez ustaloną liczbę, na przykład 7. W tym celu najpierw zmniejszamy licznik drugi aż do zera, a potem powtarzamy taką czynność: odejmując 1 od licznika pierwszego, dodajemy 7 (tj. siedmiokrotnie dodajemy jedynkę) do licznika drugiego. Tak postępujemy aż do wyzerowania pierwszego licznika. Podobnie możemy dzielić przez 7, a także sprawdzić, czy wartość licznika jest podzielna przez 7.

Piątkę liczb naturalnych (zawartość pięciu liczników i, j, k, l, m) reprezentujemy za pomocą liczby $2^i 3^j 5^k 7^l 11^m$, którą przechowujemy jako wartość jednego z dwóch liczników. Drugi licznik służy do celów pomocniczych. Powiększeniu lub pomniejszeniu o 1 licznika l odpowiada teraz pomnożenie lub podzielenie naszej liczby przez 7. Możemy także sprawdzić, czy $l = 0$. W ten sposób obliczenie używające pięciu liczników zrealizujemy z pomocą dwóch. ∎

Przejdźmy do rzeczy. Piszemy $\Gamma \vdash \tau$, gdy $\Gamma \vdash M : \tau$, dla pewnego M. W szczególności $\vdash \tau$ oznacza, że typ τ jest *niepusty*, tj. istnieje zamknięty term M typu τ. Interesuje nas *problem niepustości typu*, czyli pytanie „Czy istnieje term zamknięty danego typu?". Łatwo widzieć, że ten problem jest równoważny zadaniu „Czy $\Gamma \vdash \tau$, dla danych Γ, τ?"

Problem stopu dla automatów dwulicznikowych sprowadzimy do pytania: *Dane* Γ *i* τ*, czy* $\Gamma \vdash \tau$*?* Załóżmy, że dany jest automat $\mathcal{M} = \langle Q, q_0, q_N, \delta\rangle$, gdzie $Q = \{q_0, \ldots, q_N\}$. Na początek ustalmy takie zmienne konstruktorowe:

$$\mathtt{Q}_i : * \Rightarrow * \Rightarrow *, \text{ dla dowolnego } i = 1, \ldots, N;$$

$$\mathtt{f} : * \Rightarrow *; \quad \mathtt{0} : *, \quad \mathtt{OK} : *.$$

Teraz zbudujemy pewne otoczenie Γ, reprezentujące automat $\mathcal{M}$. Z każdą instrukcją $\delta(q)$ związżemy jedną lub dwie zmienne deklarowane w Γ. Typy tych zmiennych są następujące:

- $\forall xy(\mathtt{Q}_i xy \to \mathtt{Q}_j(\mathtt{f}x)y)$, gdy $\delta(q_i) =$ „$c_1 := c_1 + 1;$ **goto** q_j;”
- $\forall xy(\mathtt{Q}_i xy \to \mathtt{Q}_j x(\mathtt{f}y))$, gdy $\delta(q_i) =$ „$c_2 := c_2 + 1;$ **goto** q_j;”
- $\forall xy(\mathtt{Q}_i(\mathtt{f}x)y \to \mathtt{Q}_j xy)$, gdy $\delta(q_i) =$ „$c_1 := c_1 - 1;$ **goto** q_j;”
- $\forall xy(\mathtt{Q}_i x(\mathtt{f}y) \to \mathtt{Q}_j xy)$, gdy $\delta(q_i) =$ „$c_2 := c_2 - 1;$ **goto** q_j;”
- $\forall y(\mathtt{Q}_i \mathtt{0}y \to \mathtt{Q}_j \mathtt{0}y)$ oraz $\forall xy(\mathtt{Q}_i(\mathtt{f}x)y \to \mathtt{Q}_k(\mathtt{f}x)y)$,
  gdy $\delta(q_i) =$ „**if** $c_2 = 0$ **then goto** q_j **else goto** q_k;”
- $\forall x(\mathtt{Q}_i x\mathtt{0} \to \mathtt{Q}_j x\mathtt{0})$ oraz $\forall xy(\mathtt{Q}_i x(\mathtt{f}y) \to \mathtt{Q}_k x(\mathtt{f}y))$,
  gdy $\delta(q_i) =$ „**if** $c_2 = 0$ **then goto** q_j **else goto** q_k.”

Na koniec do otoczenia Γ dokładamy jeszcze deklarację zmiennych takich typów:

$$\mathtt{Q}_0\mathtt{00} \quad \text{oraz} \quad \forall xy(\mathtt{Q}_N xy \to \mathtt{OK})$$

Oczywiście nazwy zadeklarowanych zmiennych tak naprawdę są bez znaczenia.

Na potrzeby tego dowodu wprowadźmy jeszcze oznaczenie $\underline{m} = \mathtt{f}^k(\mathtt{0})$. Oczywiście, $\underline{m} : *$.

Lemat 26.4 *Jeśli* $\langle q_i, m, n\rangle \twoheadrightarrow_{\mathcal{M}} \langle q_N, m', n'\rangle$ *to* $\Gamma, \mathtt{Q}_i\, \underline{m}\, \underline{n} \vdash \mathtt{OK}$.

Dowód: Indukcja ze względu na długość obliczenia rozpoczynającego się od konfiguracji $\langle q_i, m, n\rangle$. Jeśli to już jest konfiguracja końcowa, to należy wykorzystać zmienną typu $\forall xy(\mathtt{Q}_N xy \to \mathtt{OK})$. W kroku indukcyjnym należy zauważyć, że jeśli $\langle q_i, m, n\rangle \to_{\mathcal{M}} \langle q_j, m', n'\rangle$ to $\Gamma, \mathtt{Q}_i\, \underline{m}\, \underline{n} \vdash \mathtt{Q}_j\, \underline{m'}\, \underline{n'}$. ∎

Lemat 26.5 *Jeśli* $\Gamma \vdash \mathtt{Q}_j\, \underline{m}\, \underline{n}$, *to* $\langle q_0, 0, 0\rangle \twoheadrightarrow_{\mathcal{M}} \langle q_j, m, n\rangle$.

Dowód: Istnieje taki term M, że $\Gamma \vdash M : \mathtt{Q}_j\, \underline{m}\, \underline{n}$. Można zakładać, że M jest w postaci normalnej. Dowód jest przez indukcję ze względu na rozmiar M. Z powodu swojego typu, M nie może być abstrakcją. Musi to być zmienna lub aplikacja. Jedyny typ w Γ, który zaczyna się od konstruktora reprezentującego stan, to zmienna $\mathtt{Q}_0\mathtt{00}$. Ale jeśli to ma być typ M, to znaczy, że $m = n = j = 0$ i teza zachodzi w sposób trywialny.

Jeśli M jest aplikacją, to ma postać $z\vec{N}$, gdzie z jest zmienną deklarowaną w Γ, a $\vec{N}$ jest ciągiem typów i termów. Typ zmiennej z ma postać $\forall x(\cdots \to \mathtt{Q}_j uv)$ lub $\forall xy(\cdots \to \mathtt{Q}_j uv)$ i odpowiada zastosowaniu pewnej instrukcji automatu $\mathcal{M}$. Przypuśćmy na przykład, że z jest zmienną typu $\forall xy(\mathtt{Q}_i x(\mathtt{f}y) \to \mathtt{Q}_j xy)$, odpowiadającego instrukcji $\delta(q_i) = c_2 := c_2 - 1;$ **goto** q_j. Wtedy $M = z\, \underline{m}\, \underline{n} M'$ gdzie $\Gamma \vdash M' : \mathtt{Q}_i\, \underline{m}(\mathtt{f}\, \underline{n})$. Z założenia indukcyjnego wnioskujemy, że $\langle q_0, 0, 0\rangle \twoheadrightarrow_{\mathcal{M}} \langle q_i, m, n+1\rangle$. Ponadto oczywiście $\langle q_i, m, n+1\rangle \to_{\mathcal{M}} \langle q_j, m, n\rangle$, więc ostatecznie $\langle q_0, 0, 0\rangle \twoheadrightarrow_{\mathcal{M}} \langle q_j, m, n\rangle$, jak miało być. Podobnie postępujemy w przypadku pozostałych instrukcji. ∎

Dowód twierdzenia 26.2: Z lematów 26.4 i 26.5 łatwo otrzymujemy równoważność:

Automat $\mathcal{M}$ zatrzymuje się wtedy i tylko wtedy, gdy $\Gamma \vdash \texttt{OK}$.

To kończy dowód twierdzenia 26.2. ∎

Uwaga: Zauważmy, że w dowodzie wykorzystaliśmy tylko zmienne konstruktorowe rodzaju $*$, $* \Rightarrow *$ i $* \Rightarrow * \Rightarrow *$, przy czym tylko zmienne rodzaju $*$ były wiązane kwantyfikatorami.

Ćwiczenia

1. Udowodnić, że wszystkie termy postaci $22\ldots2\mathbf{K}$ są typowalne w $\mathbf{F}_\omega$.
2. Czy term $(\lambda x\, xx)(\lambda x\, xx)$ jest typowalny w $\mathbf{F}_\omega$?

27 Dodatki

Dodatek A: Inny dowód twierdzenia 7.9

Ten dowód polega na reprezentacji obliczeń (deterministycznej) maszyny Turinga w rachunku lambda. Przyjmijmy tu, że maszyna Turinga to krotka postaci $\mathcal{M} = \langle Q, A, \mathsf{B}, q_0, q_f, \delta\rangle$, gdzie Q to zbiór stanów, A to alfabet, $\mathsf{B} \in A$ jest symbolem pustym, q_0 i q_f to stan początkowy i końcowy, i wreszcie $\delta : Q - \{q_f\} \times A \to A \times \{-1, 0, 1\} \times Q$ to funkcja przejścia. Równość $\delta(q, a) = \langle b, x, p\rangle$ interpretujemy tak: widząc literę a w stanie q, maszyna pisze b, przesuwa głowicę o x klatek w prawo i przechodzi w stan p. Dla uproszczenia zakładamy, że maszyna nigdy nie wpisuje pustego symbolu w miejsce „prawdziwej" litery, tj.

$$\text{Jeśli } \delta(q, a) = \langle \mathsf{B}, x, p\rangle \text{ to } a = \mathsf{B} \text{ i } x = -1.$$

Konfiguracje maszyny zdefiniujemy jak zwykle jako trójki postaci $\langle w, q, v\rangle$, gdzie $q \in Q$ oraz $w, v \in (A - \{\mathsf{B}\})^*$. Ale interpretujemy je trochę niestandardowo: zawartością taśmy nie jest słowo wv ale słowo $w^R v$. Głowica, jak zwykle, „patrzy" na pierwszy symbol słowa v.

Relację zmiany konfiguracji $\mathcal{C}_1 \to \mathcal{C}_2$ definiujemy z grubsza jak zwykle, pamiętając jednak o „odwrotnej" interpretacji taśmy. Na przykład jeśli $\delta(q, a) = \langle b, +1, p\rangle$ i $\delta(q, \mathsf{B}) = \langle a, +1, p\rangle$, to $\langle w, q, av\rangle \to \langle bw, p, v\rangle$ oraz $\langle w, q, \varepsilon\rangle \to \langle aw, r, \varepsilon\rangle$. Nie określamy co się ma stać w konfiguracji $\langle \varepsilon, q, av\rangle$ dla $\delta(q, a) = \langle b, -1, r\rangle$ (próba wyjścia poza taśmę).

Załóżmy teraz, że $z \in A - \{\mathsf{B}\}$. Konfigurację $\mathcal{C}[n] = \langle \varepsilon, q_0, z^n\rangle$ nazywamy *początkową*, a konfiguracje $\langle w, q_f, v\rangle$ to konfiguracje *końcowe z wynikiem* $|w|$. Mówimy, że maszyna *zatrzymuje się* dla wejścia n z wynikiem k, jeżeli $\mathcal{C}[n] \twoheadrightarrow \mathcal{C}$ dla pewnej konfiguracji końcowej $\mathcal{C}$ z wynikiem k.

W ten sposób maszyna Turinga *oblicza* funkcję częściową $\varphi : \mathbb{N} \rightharpoonup \mathbb{N}$, gdzie

$$\varphi(n) = k \text{ wtedy i tylko wtedy, gdy } \mathcal{M} \text{ zatrzymuje się dla } n \text{ z wynikiem } k.$$

Aby uniknąć technicznych kłopotów, zakładamy, że maszyna $\mathcal{M}$ obliczająca funkcję φ nigdy nie usiłuje wyjść poza taśmę. (Ćwiczenie: pokazać, że dla każdego φ taka maszyna istnieje.)

Przez *superkonfigurację* maszyny rozumiemy czwórkę $\langle w, q, v, n\rangle$, gdzie $\langle w, q, v\rangle$ jest konfiguracją i $n = |w|$. Relacja $\to$ przenosi się w oczywisty sposób na superkonfiguracje.

Teraz przejdźmy wreszcie do lambda-termów.

Kodowanie maszyny w rachunku lambda: Elementy a_i skończonego zbioru $\{a_1, \ldots, a_k\}$ przedstawiamy w rachunku lambda jako rzuty $\mathbf{a}_i = \lambda x_1 \ldots x_k.x_i$. Zauważmy, że pozwala to na zdefiniowanie instrukcji wyboru `case` x `of` $P_1, \ldots, P_n$ po prostu jako aplikacji $xP_1 \ldots P_n$. Stany $q_0, q_1, ..., q_r$ i symbole $a_1, \ldots, a_m$ alfabetu maszyny też będziemy reprezentować za pomocą odpowiednich rzutów $\mathbf{q_0}, \mathbf{q_1}, ..., \mathbf{q_r}$ i $\mathbf{a_1}, \ldots, \mathbf{a_m}$.

Przyjmujemy teraz, że $nil = \langle \mathbf{blank}, \mathbf{I}\rangle$, gdzie **blank** jest rzutem odpowiadającym symbolowi B. oraz $a :: \ell = \langle a, \ell\rangle$, gdzie para jest zdefiniowana jak w rozdziale 7. Mamy funkcje $head = \lambda x.\, x\pi_1$ i $tail = \lambda x.\, x\pi_2$. Uwaga: przy naszych definicjach $head(nil) = \mathbf{blank}$,

Słowo w utożsamiamy z listą $\mathbf{w}$ jego liter, a liczby naturalne z liczebnikami Churcha, a więc superkonfiguracja $\langle w, q, v, n\rangle$ może być identyfikowana z termem $\langle \mathbf{w}, \mathbf{q}, \mathbf{v}, \mathbf{n}\rangle = \lambda x.\, x\mathbf{wqvn}$. Takie czwórki można „rozbierać" na części z pomocą rzutowań $\varpi_i = \lambda x_1 x_2 x_3 x_4.\, x_i$.

Aby napisać lambda-term przedstawiający zmianę superkonfiguracji maszyny, potrzebujemy operacji poprzednika.

Lemat A.6 *Funkcja p określona równaniami $p(n+1) = n$, $p(0) = 0$ jest lambda-definiowalna.*

Dowód: Niech $\mathbf{pred} = \lambda n.\,(n\mathbf{Step}\langle\mathbf{0},\mathbf{0}\rangle)\pi_2$, gdzie $\mathbf{Step} = \lambda p.\langle\mathbf{succ}(p\pi_1), p\pi_1\rangle$. Wtedy aplikacja $\mathbf{pred\,n}$ prowadzi do n-krotnej iteracji operacji **Step**. Otrzymujemy kolejno pary $\langle\mathbf{0},\mathbf{0}\rangle$, $\langle\mathbf{1},\mathbf{0}\rangle$, $\langle\mathbf{2},\mathbf{1}\rangle$, i tak dalej, aż do $\langle\mathbf{n},\mathbf{n-1}\rangle$. ∎

Napiszemy teraz term *Next* opisujący zmianę konfigracji w jednym kroku maszyny.

$$Next = \lambda t.\, t\varpi_2 R^0 \ldots R^r,$$

gdzie $R^0, \ldots R^r$ to instrukcje wyboru określające zachowanie maszyny w stanach $q_0, \ldots, q_r$:

$$R^i = head(t\varpi_3)R^i_1 \ldots R^i_m.$$

Zauważmy, że $\langle\mathbf{w},\mathbf{q},\mathbf{v},\mathbf{n}\rangle\varpi_2 = \mathbf{q}$ oraz, że $head(\langle\mathbf{w},\mathbf{q},\mathbf{v},\mathbf{n}\rangle\varpi_3)$ przedstawia pierwszą litera słowa kodowanego przez $\mathbf{v}$. A jeśli $\mathbf{v} = nil$, to dostaniemy tu **blank**.

Trzeba więc podefiniować składowe R^i_j. Załóżmy najpierw, że $a_j \neq \mathsf{B}$ (wtedy $t\varpi_3 \neq nil$, więc $tail(t\varpi_3)$ jest poprawnym kodem słowa) a stan q_i nie jest końcowy. Jeśli $\delta(q_i, a_j) = \langle b, 0, p\rangle$, to definicja jest łatwa: $R^i_j = \langle t\varpi_1, \mathbf{p}, \langle\mathbf{b}, (tail(t\varpi_3))\rangle, t\varpi_4\rangle$. Jeżeli $\delta(q_i, a_j) = \langle b, +1, p\rangle$, to możemy przyjąć $R^i_j = \langle\langle\mathbf{b}, (t\varpi_1)\rangle, \mathbf{p}, tail(t\varpi_3), \mathbf{succ}(t\varpi_4)\rangle$. Wreszcie dla $\delta(q_i, a_j) = \langle b, -1, p\rangle$ definiujemy $R^i_j = \langle tail(t\varpi_1), \mathbf{p}, \langle head(t\varpi_1), \langle\mathbf{b}, (tail(t\varpi_3))\rangle\rangle, \mathbf{pred}(t\varpi_4)\rangle$. Poprawność tej definicji wynika z założenia, że nasza maszyna nie zrobi kroku w lewo na lewym końcu taśmy.

Dla $a_j = \mathsf{B}$ należy zamienić $tail(t\varpi_3)$ na nil. Drugim wyjątkiem jest stan końcowy (załóżmy, że f jest po prostu jego numerem), dla którego funkcja δ jest nieokreślona, a jako R^f można przyjąć cokolwiek.

Potrzebujemy teraz jeszcze trzech funkcji pomocniczych. Pierwszą z nich jest funkcja

$$In = \lambda x.\,\langle nil, \mathbf{q}_0, x(\lambda y.\,\langle\mathbf{z}, y\rangle)nil, \mathbf{0}\rangle,$$

która każdemu liczebnikowi $\mathbf{n}$ przypisuje (super)konfigurację początkową $\mathcal{C}[n]$. Druga funkcja $Out = \lambda t.\, t\varpi_4$ odczytuje wynik, a trzecia to test na stan końcowy: $Koniec = \lambda t.\, t\varpi_2 d_1 \ldots d_r$, gdzie $d_f = \mathbf{true}$ i $d_i = \mathbf{false}$ dla $i \neq f$.

Teraz możemy już stwierdzić, że funkcja φ jest definiowana termem $\lambda x.W(In\,x)W$, gdzie

$$W = \lambda t.\,\mathbf{if}\ Koniec(t)\ \mathbf{then}\ \lambda w.Out(t)\ \mathbf{else}\ \lambda w.w(Next(t))w.$$

Uzasadnimy to podobnie jak w dowodzie twierdzenia 7.9.

Dodatek B: Inny dowód twierdzenia 14.11[29]

Definiujemy klasę termów $\mathcal{S}$ w następujący sposób:

- Jeśli $N_1, \ldots, N_k \in \mathcal{S}$, to $xN_1 \ldots N_k \in \mathcal{S}$;
- Jeśli $N \in \mathcal{S}$, to $\lambda x\, N \in \mathcal{S}$;
- Jeśli $Q \in \mathcal{S}$ oraz $P[x := Q]N_1 \ldots N_k \in \mathcal{S}$, to $(\lambda x\, P)QN_1 \ldots N_k \in \mathcal{S}$.

Symbol $|M|$ oznacza rozmiar (długość) termu M, podobnie $|\tau|$ to rozmiar typu τ. Jeśli $M \in \mathrm{SN}$ to $\ell(M)$ oznacza maksymalną długość ciągu β-redukcji dla termu M.

Lemat B.1 *Term M należy do $\mathcal{S}$ wtedy i tylko wtedy, gdy jest silnie normalizowalny.*

Dowód: ($\Rightarrow$) Indukcja ze względu na definicję $\mathcal{S}$. Pierwsze dwa przypadki są oczywiste, trzeci to w istocie lemat 14.8.

($\Leftarrow$) Indukcja ze względu na dwa parametry: pierwszy to $\ell(M)$, drugi to $|M|$. Przypuśćmy najpierw, że term $M \in \mathrm{SN}$ ma redeks czołowy, tj. $M = (\lambda x\, P)QN_1 \ldots N_k$. Wtedy $Q \in \mathcal{S}$ z założenia indukcyjnego (drugi parametr mniejszy, pierwszy nie większy). Założenie indukcyjne stosuje się także do termu $P[x := Q]N_1 \ldots N_k \in \mathcal{S}$ (pierwszy parametr mniejszy). Zatem oba te termy są w $\mathcal{S}$, skąd $M \in \mathcal{S}$.

Jeśli M nie ma redeksu czołowego, to albo M jest abstrakcją, albo ma postać $xN_1 \ldots N_k$. Oba te przypadki wynikają łatwo z założenia indukcyjnego. ∎

Powiemy, że *podstawienie $M[x := P]$ jest jednostajne*, gdy $\Gamma, x : \tau \vdash M : \sigma$ i $\Gamma \vdash P : \tau$, dla pewnych Γ i τ (tj. gdy podstawienie w stylu Churcha jest poprawne.)

Lemat B.2 *Jeśli podstawienie $M[x := P]$ jest jednostajne i $M, P \in \mathcal{S}$, to $M[x := P] \in \mathcal{S}$.*

Dowód: Załóżmy, że $\Gamma, x : \tau \vdash M : \sigma$ i $\Gamma \vdash P : \tau$. Dowód jest przez indukcję ze względu na trzy parametry: $|\tau|$, $\ell(M)$ oraz $|M|$ (rozważamy leksykograficzny porządek w $\mathbb{N}^3$). Mamy pięć przypadków.

Przypadek 1: $M = \lambda z\, N$. Wtedy $M[x := P] = \lambda z\, N[x := P] \in \mathcal{S}$, z założenia indukcyjnego dla N (pierwszy parametr bez zmian, drugi nie większy, trzeci mniejszy).

Przypadek 2: $M = yN_1 \ldots N_k$. Wtedy $M[x := P] = yN_1[x := P] \ldots N_k[x := P] \in \mathcal{S}$, z założenia indukcyjnego dla $N_1, \ldots, N_k$.

Przypadek 3: $M = x$. Wtedy $M[x := P] = P \in \mathcal{S}$ z założenia.

Przypadek 4: $M = xQN_1 \ldots N_k$. Wtedy $M[x := P] = PQ[x{:=}P]N_1[x{:=}P] \ldots N_k[x{:=}P]$. Skoro $M \in \mathcal{S}$, to $M \in \mathrm{SN}$, więc także $Q, N_1 \ldots N_k \in \mathrm{SN}$, czyli $Q, N_1 \ldots N_k \in \mathcal{S}$. Z założenia indukcyjnego łatwo wynika $Q[x := P], N_1[x := P], \ldots, N_k[x := P] \in \mathcal{S}$.

[29] Według René Davida.

Wystarczy pokazać, że $M[x := P] \in \mathrm{SN}$. Rozpatrzmy więc dowolny ciąg redukcji rozpoczynający się od $M[x := P]$. Jeśli ten ciąg składa się wyłącznie z redukcji odbywających się wewnątrz termów $P, Q[x := P], N_1[x := P], \ldots, N_k[x := P] \in \mathcal{S} = \mathrm{SN}$, to musi się skończyć. W przeciwnym razie mamy redukcję

$$PQ[x{:=}P]N_1[x{:=}P]\ldots N_k[x{:=}P] \twoheadrightarrow (\lambda y\, R)Q'N_1'\ldots N_k' \to R[y := Q']N_1'\ldots N_k' \twoheadrightarrow \cdots$$

gdzie $P \twoheadrightarrow \lambda y\, R$ oraz $Q[x := P] \twoheadrightarrow Q'$, a także $N_i[x := P] \twoheadrightarrow N_i'$ dla $i \leq k$. Redukcja zachowuje typy, więc $\Gamma \vdash \lambda y\, R : \tau$, co oznacza, że $\tau = \tau_1 \to \tau_2$ i term Q' musi mieć typ τ_1. Ale typ τ_1 jest krótszy niż τ, więc do (jednostajnego!) podstawienia $R[y := Q']$ można zastosować założenie indukcyjne. Stąd $R[y := Q'] \in \mathcal{S}$.

Teraz zauważmy, że term $R[y := Q']N_1'\ldots N_k'$ można (wprowadzając nową zmienną z typu τ_2) przedstawić w postaci

$$R[y := Q']N_1'\ldots N_k' = (zN_1'\ldots N_k')[z := R[y := Q']]$$

Ponieważ termy $N_1'\ldots N_k'$, jako redukty $N_1, \ldots, N_k \in \mathrm{SN}$, są silnie normalizowalne (czyli należą do $\mathcal{S}$), więc $zN_1'\ldots N_k' \in \mathcal{S}$. A ponieważ typ τ_2 jest krótszy niż τ, więc ponownie stosując założenie indukcyjne dostaniemy $(zN_1'\ldots N_k')[z := R[y := Q']] \in \mathcal{S} = \mathrm{SN}$. No to nasz ciąg redukcji musi się skończyć.

Przypadek 5: $M = (\lambda z R)QN_1\ldots N_k$. Term tej postaci może należeć do $\mathcal{S}$ tylko pod warunkiem, że $Q \in \mathcal{S}$ oraz $R[z := Q]N_1\ldots N_k \in \mathcal{S}$. Mamy wtedy

$$M[x := P] = (\lambda z R[x := P])Q[x := P]N_1[x := P]\ldots N_k[x := P]$$

i wystarczy pokazać, że $Q[x := P] \in \mathcal{S}$ (co łatwo wynika z założenia indukcyjnego), oraz, że $R[x := P][z := Q[x := P]]N_1[x := P]\ldots N_k[x := P] \in \mathcal{S}$. Ten ostatni term to nic innego niż $R[z := Q][x := P]N_1[x := P]\ldots N_k[x := P]$, czyli $(R[z := Q]N_1\ldots N_k)[x := P]$. Zauważmy jednak, że $M \twoheadrightarrow R[z := Q]N_1\ldots N_k$, więc $\ell(R[z := Q]N_1\ldots N_k) < \ell(M)$. Można więc zastosować założenie indukcyjne (tym razem pracuje drugi parametr). ∎

Dowód twierdzenia: Załóżmy, że $\Gamma \vdash M : \tau$. Przez indukcję ze względu na wyprowadzenie typu, dowodzimy, że $M \in \mathcal{S}$. Jeśli M jest zmienną, to teza jest oczywista. Dla abstrakcji teza wynika bezpośrednio z założenia indukcyjnego. W przypadku aplikacji $M = PQ$ stosujemy lemat B.2 do podstawienia $M = (xQ)[x := P]$, gdzie x jest nową zmienną. ∎

Dodatek C: System T

System Gödla **T** to w istocie rachunek lambda z typami prostymi, rozszerzony o typ **int** liczb naturalnych (wraz z „gotowymi" liczebnikami) oraz operator rekursji. Ściślej:

- Typy systemu **T** to stała typowa **int** i typy postaci $(\tau \to \sigma)$, gdzie τ i σ są typami.[30]
- Termy są jak w rachunku z typami prostymi w stylu Churcha, ale mogą zawierać stałe:
  - $\mathbf{0} : \mathbf{int}$;
  - $\mathbf{s} : \mathbf{int} \to \mathbf{int}$;
  - $\mathbf{R}_\tau : (\mathbf{int} \to \tau \to \tau) \to \tau \to (\mathbf{int} \to \tau)$, dla dowolnego τ.
- Oprócz zwykłej beta-redukcji w systemie są dwie reguły dla rekursora:
  - $\mathbf{R}_\tau MN(\mathbf{s}P) \Rightarrow MP(\mathbf{R}_\tau MNP)$;
  - $\mathbf{R}_\tau MN\mathbf{0} \Rightarrow N$.

  Relację redukcji oznaczamy przez $\to_{\mathbf{T}}$, znaki $\twoheadrightarrow_{\mathbf{T}}$ i $=_{\mathbf{T}}$ rozumiemy jak zwykle.

Liczebniki konstruujemy za pomocą stałych $\mathbf{0}$ i $\mathbf{s}$: przez $\overline{n}$ oznaczamy term $\mathbf{s}(\mathbf{s}(\cdots\mathbf{s}(\mathbf{0})\cdots))$, w którym $\mathbf{s}$ użyte zostało n razy. Możemy teraz mówić o definiowaniu funkcji. Powiemy, że funkcja $f : \mathbb{N}^k \to \mathbb{N}$ jest *definiowalna* w systemie **T**, jeżeli istnieje kombinator $F : \mathbf{int}^k \to \mathbf{int}$, taki że dla $f(n_1, \ldots, n_k) = m$ zachodzi $F\overline{n}_1 \ldots \overline{n}_k =_{\mathbf{T}} \overline{m}$.

Jako pierwszy przykład rozpatrzmy zastosowanie rekursora $\mathbf{R}_{\mathbf{int}}$ do termu $M = \lambda xz.\mathbf{s}z$. Dla dowolnych m, n otrzymamy $\mathbf{R}_{\mathbf{int}}M\overline{n}\overline{m} \twoheadrightarrow_{\mathbf{T}} \overline{n+m}$. Oto kilka dalszych przykładów:

- Funkcja warunkowa **if** $x = 0$ **then** y **else** z jest definiowalna wyrażeniem $\mathbf{R}_{\mathbf{int}}(\lambda uv.z)yx$.
- Funkcja poprzednika jest definiowalna wyrażeniem $\lambda x.\mathbf{R}_{\mathbf{int}}(\lambda uv.u)\mathbf{0}x$.
- Niech funkcja f będzie określona przez rekursję prostą
$$\begin{aligned} f(0, n_1, \ldots, n_k) &= g(n_1, \ldots, n_k); \\ f(s(m), n_1, \ldots, n_k) &= h(m, n_1, \ldots, n_k, f(m, n_1, \ldots, n_k)), \end{aligned}$$
  gdzie g, h są definiowalne odpowiednio przez termy G i H. Wtedy f jest definiowalne termem $F = \lambda x\vec{y}.\mathbf{R}_{\mathbf{int}}(\lambda uv.Hu\vec{y}v)(G\vec{y})x$. Jak widać, użycie rekursora $\mathbf{R}_{\mathbf{int}}$ odpowiada zwykłej rekursji prostej.
- Niech $f_0(n) = n+1$ i niech $f_{k+1}(n) = f_k^n(n)$, dla dowolnych k i n. Wszystkie funkcje f_k są definiowalne. Rzeczywiście, niech $M = \lambda fx.\mathbf{R}_{\mathbf{int}}(\lambda uv.fv)xx$. Jeśli teraz F_k definiuje funkcję f_k to term $F_{k+1} = MF_k$ definiuje f_{k+1}.
- Wreszcie niech $f_\omega(n) = f_n(n)$. Funkcja f_ω jest pewnym wariantem funkcji Ackermanna. Nie jest to funkcja pierwotnie rekurencyjna. Niemniej, jest definiowalna w systemie **T** za pomocą termu $F_\omega = \lambda x.\mathbf{R}_{\mathbf{int}\to\mathbf{int}}(\lambda\phi y.M\phi yy)\mathbf{s}xx$. Istotnie, operator $\lambda\phi y.M\phi yy$ przyłożony do F_k daje F_{k+1}. Iterujemy go n-krotnie, a wynik aplikujemy do n.

[30] Można dodać zmienne typowe, są nieszkodliwe, ale w tym kontekście mało przydatne.

- Rozpatrzmy pozaskończony ciąg funkcji f_α, w którym $f_{\alpha+1}(n) = f_\alpha^n(n)$, a dla liczby granicznej β przyjmujemy $f_\beta(n) = f_{\beta[n]}(n)$, gdzie $\beta[n]$ jest odpowiednio dobranym ciągiem rosnącym, takim że $\sup \beta[n] = \beta$. Na przykład $f_{\omega^\omega}(n) = f_{\omega^n}(n)$. Ta definicja ma sens dla $\alpha < \epsilon_0$, gdzie ϵ_0 jest najmniejszą liczbą porządkową o własności $\omega^{\epsilon_0} = \epsilon_0$, czyli granicą ciągu $\alpha_0 = \omega$, $\alpha_{n+1} = \omega^{\alpha_n}$. Każda z funkcji f_α jest definiowalna w $\mathbf{T}$, ale już dla $\alpha = \omega^\omega$ potrzebny jest rekursor $\mathbf{R}_{(\mathbf{int}\to\mathbf{int})\to(\mathbf{int}\to\mathbf{int})}$.

Arytmetyka pierwszego rzędu: Dziś można powiedzieć, że system $\mathbf{T}$ to pewien specyficzny język programowania. Ale wymyślono go po to, żeby udowodnić niesprzeczność arytmetyki Peana. Bo związek systemu $\mathbf{T}$ z arytmetyką pierwszego rzędu jest bardzo ścisły. Żeby go zauważyć, przypomnijmy sobie aksjomaty arytmetyki Peana w ich „tradycyjnej" postaci:

- $A_1 : \mathbf{int}(0)$.
- $A_2 : \forall x(\mathbf{int}(x) \to \mathbf{int}(\boldsymbol{s}x))$.
- $A_3 : \forall x(\mathbf{int}(x) \to \mathbf{int}(y) \to \boldsymbol{s}x = \boldsymbol{s}y \to x = y)$.
- $A_4 : \forall x(\mathbf{int}(x) \to \neg(\boldsymbol{s}x = 0))$.
- $A_5^\varphi : \forall x(\mathbf{int}(x) \to \varphi(x) \to \varphi(\boldsymbol{s}x)) \to \varphi(0) \to \forall x(\mathbf{int}(x) \to \varphi(x))$.

Tutaj $\mathbf{int}(x)$ jest jednoargumentowym predykatem, który czytamy "x jest liczbą naturalną". Aksjomaty A_1 i A_2 jawnie wprowadzają *kanoniczne obiekty* typu **int**: zero i nastepniki obiektów typu **int**. Aksjomaty A_3 i A_4 gwarantują, że każdorazowe zastosowanie operacji następnika tworzy nową liczbę naturalną, różną od wszystkich poprzednich. Wreszcie aksjomat A_5 to w istocie nie jeden aksjomat, ale *schemat* indukcji. Te same aksjomaty przyjmujemy dla intuicjonistycznej arytmetyki, którą nazywamy *arytmetyką Heytinga*. Jedynym aksjomatem zawierającym negację jest A_4, czasem zastępowany przez nieco słabszą „pozytywną" wersję:

- $A_4' : \forall x(\mathbf{int}(x) \to \boldsymbol{s}x = 0 \to \forall y(\mathbf{int}(y) \to y = 0))$.

Ponieważ ogólnie nie lubimy formuł negatywnych, więc my też poprzestańmy na A_4'. Nie tracimy na tym nic istotnego, jak długo zajmujemy się formułami pozytywnymi.

Wycieranie zależności: Dla takiego pozytywnego języka arytmetyki określimy operator *wycierania zależności* κ, który dowolnej formule φ arytmetyki pierwszego rzędu przypisuje formułę zdaniową $\kappa(\varphi)$. Otrzymujemy ją z φ przez „wytarcie" wyrażeń i kwanyfikatorów indywiduowych. Przy tym predykat **int** staje się po prostu wyróżnionym symbolem zdaniowym (stałą typową), a równość tłumaczymy na stałą logiczną $\top$ (typ jednostkowy). Jeśli bowiem ignorujemy indywidua, powinniśmy też ignorować równania pomiędzy nimi.

- $\kappa(\mathbf{int}(t)) = \mathbf{int}$.
- $\kappa(t = s) = \top$.
- $\kappa(\varphi \to \psi) = \kappa(\varphi) \to \kappa(\psi)$.
- $\kappa(\forall x\varphi) = \kappa(\exists x\varphi) = \kappa(\varphi)$.

Koniunkcję i alternatywę pominęliśmy, żeby nie komplikować sprawy. Zobaczmy co zostaje z aksjomatów Peana po wytarciu zależności:

- $\kappa(A_1) : \mathbf{int}$.
- $\kappa(A_2) : \mathbf{int} \to \mathbf{int}$.
- $\kappa(A_3) : \mathbf{int} \to \mathbf{int} \to \top \to \top)$.
- $\kappa(A'_4) : (\mathbf{int} \to \top) \to (\mathbf{int} \to \top)$.
- $\kappa(A_5^\varphi) : (\mathbf{int} \to \tau \to \tau) \to \tau \to \mathbf{int} \to \tau$, gdzie $\tau = \kappa(\varphi)$:

Formuły-typy $\kappa(A_3)$ i $\kappa(A'_4)$ nie są zbyt ciekawe: każda ma tylko trywialne inhabitanty. Ale typy pozostałych aksjomatów to dokładnie typy stałych systemu **T**. A skąd się wzięły reguły redukcji dla rekursora? Skoro rekursor ma taki typ jak aksjomat indukcji, to w myśl izomorfizmu Curry'ego-Howarda, użycie rekursora reprezentuje dowód przez indukcję.

Dowód indukcyjny tezy $\forall x(\mathbf{int}(x) \to \varphi(x))$ składa się z dwóch części: kroku bazowego D_0 (dowodu formuły $\varphi(0)$) i kroku indukcyjnego D_1 (dowodu dla $\forall x(\mathbf{int}(x) \to \varphi(x) \to \varphi(\boldsymbol{s}x))$). Możemy tu napisać $D_0 : \varphi(0)$ i $D_1 : \forall x(\mathbf{int}(x) \to \varphi(x) \to \varphi(\boldsymbol{s}x))$, a cały dowód indukcyjny oznaczyć jako $A_5^\varphi D_1 D_0 : \forall x(\mathbf{int}(x) \to \varphi(x))$. Możliwy sposób użycia takiego dowodu to zaaplikowanie go do konkretnej liczby naturalnej, a ściślej do wyrażenia arytmetycznego t oraz (uwaga!) *dowodu* $P : \mathbf{int}(t)$, uzasadniającego, że t przedstawia liczbę naturalną. Dostajemy wtedy dowód $A_5^\varphi D_1 D_0 t P : \mathbf{int}(t)$. W tej konwencji, dowodem formuły $\varphi(0)$ jest $A_5^\varphi D_1 D_0 0 A_1$, a dowodem formuły $\varphi(2)$ jest $A_5^\varphi D_1 D_0(\boldsymbol{s}(\boldsymbol{s}x))A_2 1(A_2 0(A_1))$, bo przecież $1 = \boldsymbol{s}0$ i $2 = \boldsymbol{s}(\boldsymbol{s}0)$.

Ale z takich dowodów indukcja może być wyeliminowana. Na przykład, nie ma potrzeby dowodzić $\forall x(\mathbf{int}(x) \to \varphi(x))$, jeśli tak naprawdę mamy wyprowadzić $\varphi(0)$. To znaczy, że powinniśmy mieć regułę upraszczania dowodów

$$A_5^\varphi D_1 D_0 0 A_1 \Rightarrow D_0.$$

Oczywiście można także wyeliminować np. z dowodu formuły $\varphi(4)$. Można ją zastąpić czterokrotną aplikacją kroku indukcyjnego do kroku bazowego. Taki dowód jest dłuższy, ale niewątpliwie bardziej elementarny. Oto reguła, która takie uproszczenie umożliwia:

$$A_5^\varphi D_1 D_0(\boldsymbol{s}t)(A_2 t P^{\mathbf{int}(t)})^{\mathbf{int}(\boldsymbol{s}t)} \Rightarrow D_1 t P(A_5^\varphi D_1 D_0 t P) : \varphi(\boldsymbol{s}t)$$

Skoro można wycierać zależności z formuł, można je też wycierać z dowodów. Usuńmy z powyższych reguł wszystko to co indywiduowe, a dla lepszego efektu zamieńmy A_1, A_2, A_5 odpowiednio na $\mathbf{0}, \mathbf{s}, \mathbf{R}$. Co dostajemy? Tak, reguły redukcji dla rekursora!

Morał: Termy systemu **T** to „schematy" czy „szkielety" (konstruktywnych) dowodów w języku arytmetyki, a reguły redukcji w **T** modelują normalizację dowodów w arytmetyce.[31]

Silna normalizacja

Dowód silnej normalizacji systemu **T** przeprowadzimy metodą Taita, tak jak dowód twierdzenia 14.11. Definicja stabilności jest w zasadzie taka sama.

- $[\![\mathbf{int}]\!] := \mathrm{SN}$;
- $[\![\tau \to \sigma]\!] := \{M : \tau \to \sigma \mid \forall N(N \in [\![\tau]\!] \to MN \in [\![\sigma]\!])\}$.

[31] Ale zauważmy, że liczebnik $\overline{n} = \mathbf{s}(\mathbf{s}(\cdots \mathbf{s}(\mathbf{0}) \cdots))$ nie powstaje z wyrażenia arytmetycznego $\boldsymbol{s}(\boldsymbol{s}(\ldots(0)\ldots))$, ale z dowodu $A_2\overline{n-1}(A_2\overline{n-2}(\cdots A_2(\mathbf{0}A_1)\cdots))$ stwierdzającego, że term $\underline{n}$ przedstawia liczbę naturalną.

Treść i dowody lematów 14.7–14.9 pozostają bez zmian. Potrzebujemy jeszcze jednego.

Lemat C.1 *Stałe systemu* **T** *są stabilne w odpowiednich typach:*

1. $\mathbf{0} \in [\![\mathbf{int}]\!]$;
2. $\mathbf{s} \in [\![\mathbf{int} \to \mathbf{int}]\!]$;
3. $R_\tau \in [\![(\mathbf{int} \to \tau \to \tau) \to \tau \to (\mathbf{int} \to \tau)]\!]$

Dowód: (3) Wystarczy udowodnić, że jeśli termy M, N, P są stabilne w odpowiednich typach, to term $R_\tau MNP$ jest stabilny w typie τ. Ponieważ P jest stabilny, to ma postać normalną $P_0 = \mathbf{s}^n(P_1)$, gdzie $n \geq 0$, a term P_1 nie zaczyna się od następnika. Działamy przez indukcję ze względu na n. Jeśli $\tau = \tau_1 \to \cdots \to \tau_k \to \mathbf{int}$, to wystarczy pokazać, że $R_\tau MNPN_1 \dots N_k$ jest silnie normalizowalny, jeśli tylko $N_1, \dots N_k$ są stabilne.

Rozpatrzmy dowolny ciąg redukcji zaczynający się od $R_\tau MNPN_1 \dots N_k$. Z powodu silnej normalizacji wszystkich składowych, musi się w tym ciągu pojawić redeks, tj. mamy $R_\tau MNPN_1 \dots N_k \twoheadrightarrow_{\mathbf{T}} R_\tau M'N'P'N'_1 \dots N'_k$, gdzie P' jest zerem lub zaczyna się od nastepnika. Jeśli $P' = \mathbf{0}$, to dalsza redukcja prowadzi do $N'N'_1 \dots N'_k$. Ten term można też otrzymać przez redukowanie stabilnego termu $NN_1 \dots N_k$, a więc należy on do SN. Jesli zaś $P' = \mathbf{s}P''$ to otrzymujemy term $X = M'P''(R_\tau M'N'P'')N'_1 \dots N'_k$. Aby pokazać, że $X \in \mathrm{SN}$ wystarczy sprawdzić silną normalizację termu $Z = MP''(R_\tau MNP'')N_1 \dots N_k$, który też redukuje się do X. To wynika z założenia indukcyjnego. Bo po pierwsze $P' \twoheadrightarrow \mathbf{s}P''$ więc $P'' \in \mathrm{SN} = [\![\mathbf{int}]\!]$. A po drugie postać normalna P'' ma na początku tylko $n-1$ następników.

A więc w każdym przypadku redukcja $R_\tau MNPN_1 \dots N_k$ prowadzi do termu w SN. ∎

Z powyższego wynika, że lemat 14.10 pozostaje prawdziwy dla systemu **T**, a zatem dostajemy:

Twierdzenie C.2 *System* **T** *ma własność silnej normalizacji.*

Twierdzenie C.3 *System* **T** *ma własność Churcha-Rossera.*

Dowód: Można to udowodnić korzystając z Twierdzenia C.2 i z Lematu Newmana. ∎

System T i arytmetyka: Korzystając z twierdzenia o silnej normalizacji dla systemu **T** można udowodnić, że każde twierdzenie arytmetyki Heytinga ma dowód w *postaci normalnej*, tj. dowód o szczególnie prostej strukturze. Pomijając wyjaśnienia co to dokładnie znaczy, powiedzmy tylko, że jedynym normalnym dowodem równości pomiędzy dwoma liczebnikami, tj. równości postaci $\boldsymbol{s}(\boldsymbol{s}(\dots \boldsymbol{s}(0)\dots)) = \boldsymbol{s}(\boldsymbol{s}(\dots \boldsymbol{s}(0)\dots))$ jest aksjomat A_1, co oznacza w szczególności, że po obu stronach musi być ten sam liczebnik. Nie ma więc dowodu formuły $0 = \boldsymbol{s}(0)$, a zatem intuicjonistyczna arytmetyka jest niesprzeczna. Klasyczna arytmetyka Peana też musi być niesprzeczna, bo można ją „zanurzyć" w arytmetyce Heytinga za pomocą tzw. translacji Kołmogorowa. Pamiętamy jednak z logiki, że niesprzeczności arytmetyki Peana nie da się udowodnić w arytmetyce Peana. Okazuje się, że jedyną częścią naszkicowanego

wyżej dowodu niesprzeczności, która nie daje się sformalizować w języku arytmetyki, jest właśnie twierdzenie C.2.

Własność silnej normalizacji systemu **T** można wypowiedzieć tak: *Dla każdego termu M istnieje takie n, że każdy skończony ciąg redukcji rozpoczynający się od M ma długość co najwyżej n.* Jeśli umówimy się, że termy i ich skończone ciągi są reprezentowane jako liczby naturalne za pomocą jakiegoś ustalonego kodowania, to powyższe zdanie można wyrazić w języku arytmetyki. Mamy więc konkretny przykład zdania w języku arytmetyki, które jest od niej niezależne. Jakiego aparatu logicznego wymaga więc dowód twierdzenia C.2?

W dowodzie definiowaliśmy dwuargumentowy predykat $M \in [\![\tau]\!]$, przez indukcję ze względu na typ τ. Definicja indukcyjna to jednak definicja „uwikłana". Nie umiemy natomiast zdefiniować tego predykatu „wprost", tj. nie umiemy napisać formuły $\varphi(M, \tau)$ równoważnej warunkowi $M \in [\![\tau]\!]$. Jeśli więc chcemy w języku logiki formalnej zapisać np. treść lematu 14.7(1) to musimy posłużyć się taką konstrukcją: *Dla dowolnego dwuargumentowego predykatu X, spełniającego warunki*

- $\forall M\,(X(M, \mathbf{int}) \Leftrightarrow M \in SN)$*;*
- $\forall M \forall \tau \forall \sigma (X(M, \tau \to \sigma) \Leftrightarrow \forall N (X(N, \tau) \Rightarrow X(MN, \sigma))\}$*,*

zachodzi $\forall M \forall \tau\,(X(M, \tau) \Rightarrow M \in \mathrm{SN})$. Ta konstrukcja używa jednak kwantyfikatora $\forall X$, gdzie X przebiega relacje, a nie indywidua. Takie kwantyfikatory są dozwolone w logice i arytmetyce drugiego rzędu. I tego się nie da poprawić: arytmetyka pierwszego rzędu nie wystarczy.

Twierdzenie C.4 *Twierdzenie o silnej normalizacji systemu* **T** *jest niezależne od arytmetyki Peana.*

Dodatek D: Jeszcze o modelu $\mathcal{D}_\infty$

Ponieważ model $\mathcal{D}_\infty$ jest izomorficzny z przestrzenią funkcji ciągłych $[\mathcal{D}_\infty \to \mathcal{D}_\infty]$ (a zatem także z przestrzenią $[[\mathcal{D}_\infty \to \mathcal{D}_\infty] \to [\mathcal{D}_\infty \to \mathcal{D}_\infty]]$), więc elementy $\mathcal{D}_\infty$ można utożsamiać z funkcjami ciągłymi z $\mathcal{D}_\infty$ do $\mathcal{D}_\infty$, a nawet z $[\mathcal{D}_\infty \to \mathcal{D}_\infty]$ do $[\mathcal{D}_\infty \to \mathcal{D}_\infty]$. Powoduje to schizofreniczną dwuznaczność notacyjną, na szczęście nie prowadzącą do żadnej konfuzji. Jeśli bowiem funkcję $f : \mathcal{D}_\infty \to \mathcal{D}_\infty$ utożsamimy z elementem $f' = G(f)$, to zachodzą równości $f(a) = F(G(f))(a) = F(f')(a) = f' \cdot a$. A więc wartość funkcji f na argumencie a jest tym samym co wynik działania $f' \cdot a$ w modelu (który zwykle zapiszemy po prostu jako $f \cdot a$). Podobnie, jeśli φ jest funkcją ciągłą z $[\mathcal{D}_\infty \to \mathcal{D}_\infty]$ do $[\mathcal{D}_\infty \to \mathcal{D}_\infty]$, a f jest jak poprzednio, to możemy utworzyć aż trzy różne aplikacje φ do f. Jedna to $\varphi(f) \in [\mathcal{D}_\infty \to \mathcal{D}_\infty]$, druga to $\varphi'(f') \in \mathcal{D}_\infty$, gdzie $\varphi' = G \circ \varphi \circ F$, i wreszcie trzecia to $G(\varphi') \cdot f'$. I znowu wszystkie trzy oznaczają (z dokładnością do odpowiednich utożsamień) ten sam element modelu, bo np. $\varphi'(f') = G(\varphi(F(G(f)))) = G(\varphi(f))$.

D.1 Kombinator Y w modelu $\mathcal{D}_\infty$

Wartością termu $\mathbf{B}' = \lambda xyz.y(xz)$ w modelu $\mathcal{D}_\infty$ jest element $\mathbf{b}' = [\![\mathbf{B}']\!]$, spełniający dla dowolnych $a, b, c \in \mathcal{D}_\infty$ warunek $\mathbf{b}' \cdot a \cdot b \cdot c = b \cdot (a \cdot c)$. Element ten, zgodnie z ogólną schizofrenią, utożsamiamy z funkcją $\mathbf{b}' : \mathcal{D}_\infty \to \mathcal{D}_\infty$ o własności $\mathbf{b}'(a) \cdot b \cdot c = b \cdot (a \cdot c)$. Niech teraz $\mathrm{id} = [\![\mathbf{I}]\!]$. Wtedy $\mathrm{id}_0 = \bot$, oraz $\mathrm{id}_{n+1} = \mathrm{id}_{D_n}$, dla $n \in \mathbb{N}$.

Lemat D.1 *Element* $\mathrm{id} \in \mathcal{D}_\infty$ *jest jedynym punktem stałym funkcji* $\mathbf{b}'$.

Dowód: Nietrudno zauważyć, że id jest punktem stałym $\mathbf{b}'$, pozostaje pokazać, że $\mathbf{b}'$ ma tylko jeden punkt stały. Załóżmy więc, że ε jest punktem stałym, tj. że $\varepsilon \cdot b \cdot c = b \cdot (\varepsilon \cdot c)$, dla dowolnych b i c. Przypuśćmy, że $x \in D_{n+1}$ i $y \in D_n$ i policzmy co to jest $(\varepsilon \cdot x \cdot y)_n$. Otóż z lematu 11.6(3) wynika, że $(\varepsilon \cdot x \cdot y)_n = (\varepsilon \cdot x)_{n+1}(y) = \varepsilon_{n+2}(x)(y)$. Z drugiej strony mamy $(\varepsilon \cdot x \cdot y)_n = (x \cdot (\varepsilon \cdot y))_n$, bo to punkt stały, a ponieważ z lematu 11.6(1) wynika $x \cdot (\varepsilon \cdot y) = x((\varepsilon \cdot y)_n)$, więc wyrażenie $(x \cdot (\varepsilon \cdot y))_n$ można zapisać jako $(x((\varepsilon \cdot y)_n))_n$. To się dalej upraszcza do $x((\varepsilon \cdot y)_n)$, bo $x((\varepsilon \cdot y)_n) \in D_n$. Ale $(\varepsilon \cdot y)_n = \varepsilon_{n+1}(y)$, bo $y \in D_n$, więc w końcu wychodzi równość $(\varepsilon \cdot x \cdot y)_n = x(\varepsilon_{n+1}(y))$. Mamy więc dla wszystkich $n \in \mathbb{N}$:

$$\varepsilon_{n+2}(x)(y) = x(\varepsilon_{n+1}(y)). \qquad (*)$$

Teraz przez indukcję ze względu na n pokażemy, że $\varepsilon_n = \mathrm{id}_n$. Dla $n = 0$ i dowolnego $d \in D_0$, podstawiając $x = \lambda\!\!\lambda z.d$ i $y = \bot$ do (*), dostajemy równość $\varepsilon_2(\lambda\!\!\lambda z.d)(\bot) = d$, której lewa strona to nie co innego niż $\psi_1(\varepsilon_2)(d) = \varepsilon_1(d)$. A więc po prostu $\varepsilon_1(d) = d$ czyli $\varepsilon_1 = \mathrm{id}_1$. Dalej mamy już łatwą indukcję. Jeśli $\varepsilon_{n+1} = \mathrm{id}_{n+1}$ to $\varepsilon_{n+2}(x)(y) = x(y)$, czyli $\varepsilon_{n+2}(x) = x$. Pozostaje jeszcze zauważyć, że $\varepsilon_0 = \varepsilon_1(\bot) = \mathrm{id}_1(\bot) = \bot$. ∎

Niech teraz $f \in [\mathcal{D}_\infty \to \mathcal{D}_\infty]$ i niech $\rho(y) = f$ (pamiętamy o utożsamieniu $f = G(f) \in \mathcal{D}_\infty$). Przyjmijmy oznaczenie $\Delta_f = [\![\lambda x.y(xx)]\!]_\rho$. Wtedy $\Delta_f \cdot a = f(a \cdot a)$, dla każdego a. Przez $\mathrm{lfp}(f)$ oznaczymy najmniejszy punkt stały funkcji f.

Lemat D.2 *Jeśli* $e \leq \mathrm{id}$ *oraz* $e \cdot \Delta_f \cdot \Delta_f \leq \mathrm{lfp}(f)$, *to też* $\mathbf{b}'(e) \leq \mathrm{id}$ *oraz* $\mathbf{b}'(e) \cdot \Delta_f \cdot \Delta_f \leq \mathrm{lfp}(f)$.

Dowód: Liczymy najpierw tak: $\mathbf{b}'(e)(x)(y) = x \cdot (e \cdot y) \leq x \cdot y = x(y)$, a potem tak: $\mathbf{b}'(e) \cdot \Delta_f \cdot \Delta_f = \Delta_f \cdot (e \cdot \Delta_f) = f(e \cdot \Delta_f \cdot (e \cdot \Delta_f)) \leq f(e \cdot \Delta_f \cdot \Delta_f) \leq f(\mathrm{lfp}(f)) = \mathrm{lfp}(f)$. ∎

Wniosek D.3 *W modelu $\mathcal{D}_\infty$ znaczeniem kombinatora punktu stałego $\mathbf{Y}$ jest operator najmniejszego punktu stałego $\lambda\!\!\lambda f.\mathrm{lfp}(f)$. Ściślej, $[\![\mathbf{Y}]\!] \cdot a = G(\mathrm{lfp}(F(a)))$, dla $a \in \mathcal{D}_\infty$.*

Dowód: Niech $e^0 = \bot$ i $e^{n+1} = \mathbf{b}'(e^n)$, dla $n \in \mathbb{N}$. Na mocy lematu D.2 mamy dla wszystkich n nierówność $e^n \cdot \Delta_f \cdot \Delta_f \leq \mathrm{lfp}(f)$, skąd na mocy ciągłości aplikacji wynika, że także $\mathrm{id} = \mathrm{lfp}(\mathbf{b}')$ spełnia warunek $\mathrm{id} \cdot \Delta_f \cdot \Delta_f \leq \mathrm{lfp}(f)$. Inaczej mówiąc, $[\![\mathbf{Y}]\!] \cdot f = \Delta_f \cdot \Delta_f \leq \mathrm{lfp}(f)$. Oczywiście $[\![\mathbf{Y}]\!] \cdot f$ jest punktem stałym, bo jesteśmy w lambda-modelu. ∎

D.2 Pełna abstrakcyjność modelu $\mathcal{D}_\infty$

Naszkicujemy teraz schemat dowodu twierdzenia 11.11, pomijając co trudniejsze jego fragmenty. Posłużymy się drzewami Böhma, dowodząc w istocie „przy okazji" twierdzenia 8.9. Razem dostaniemy to:

Twierdzenie D.4 *Następujące warunki są równoważne:*

1. *Termy M i N są obserwacyjnie równoważne;*
2. *$BT(M) \approx_\eta BT(N)$;*
3. *$\mathcal{D}_\infty \models M = N$.*

Implikacja (1) ⇒ (2) to w istocie nieznaczne uogólnienie znanego nam twierdzenia Böhma 8.1. Implikację (3) ⇒ (1) nazywamy *adekwatnością* modelu, a implikacja (1) ⇒ (3) nosi nazwę *pełnej abstrakcyjności*. Adekwatność można uważać za „słabą pełność" (bo nie twierdzimy, że $M =_\beta N$ a tylko, że $M \equiv N$), natomiast pełna abstrakcyjność to „mocna poprawność", stwierdzająca, że termy nieodróżnialne obliczeniowo (nawet niekoniecznie równe) są interpretowane w ten sam sposób.

Dowód twierdzenia D.4 wymaga pewnych pojęć pomocniczych. Wprowadzimy dwie relacje dla drzew Böhma. Napis $B \sqsubseteq B'$ oznacza, że B' powstaje z B przez wstawienie jakichś poddrzew w miejsca, w których w B występuje Ω. Relacja $B \preceq_\eta B'$ zachodzi zaś, gdy istnieje (skończony lub nieskończony) ciąg eta-ekspansji $B = B_0 \ {}_\eta\!\leftarrow B_1 \ {}_\eta\!\leftarrow B_2 \ {}_\eta\!\leftarrow B_3 \ {}_\eta\!\leftarrow \cdots$ zbieżny do B'. Oczywiście $B \approx_\eta B'$ oznacza, że $B \preceq_\eta B'' \ {}_\eta\!\succeq B'$ dla pewnego B''. Notację $\preceq_\eta$ i $\approx_\eta$ stosujemy też dla termów; mamy wtedy na myśli ich drzewa Böhma.

Aproksymant to skończone drzewo Böhma (inaczej: term w postaci normalnej, w którym może występować stała Ω). Dla dowolnego termu M, mamy zbiór *aproksymantów termu M*:

$$A(M) = \{A \mid A \text{ jest aproksymantem oraz } A \sqsubseteq M\}$$

W analogiczny sposób definiujemy $A(T)$ dla dowolnego drzewa T. Przyjmując, że znaczeniem stałej Ω jest $\bot$, możemy przypisać każdemu aproksymantowi wartość w modelu $\mathcal{D}_\infty$. Dalej możemy przyjąć definicję $[\![T]\!]_\rho = \sup\{[\![A]\!]_\rho \mid A \in A(T)\}$.

Najważniejszy fakt, z którego wszystko wynika (w szczególności równość $[\![M]\!]_\rho = [\![BT(M)]\!]_\rho$), i którego dowód teraz opuścimy, to następujące twierdzenie o aproksymacji:

Twierdzenie D.5 (o aproksymacji) *Wartość dowolnego termu jest kresem górnym wartości jego aproksymantów, tj.* $[\![M]\!]_\rho = \sup\{[\![A]\!]_\rho \mid A \in A(M)\}$.

Lemat D.6 *Term M jest rozwiązalny wtedy i tylko wtedy, gdy $[\![M]\!]_\rho \neq \bot$ przy pewnym ρ.*

Dowód: Najpierw zauważmy, że jeśli A jest aproksymantem, to $[\![A]\!]_\rho \neq \bot$, dla pewnego ρ, gdy tylko $A \neq \Omega$. Jeśli bowiem $A = \lambda\vec{x}.y\vec{B}$, gdzie y jest zmienną wolną, to wystarczy wziąć $\rho(y) = \lambda\!\!\lambda\vec{a}.d$, gdzie $d \neq \bot$. A jeśli $A = \lambda\vec{x}.x_i\vec{B}$, to należy użyć $\lambda\!\!\lambda\vec{a}.d$ jako i-tego argumentu.

Ponieważ $[\![M]\!]_\rho = \sup\{[\![A]\!]_\rho \mid A \in A(M)\}$, więc z powyższego wynika $[\![M]\!]_\rho \neq \bot$, jeśli tylko M ma choć jeden nietrywialny aproksymant, tj. gdy $BT(M) \neq \Omega$. Przypomnijmy zaś, że term jest rozwiązalny wtedy i tylko wtedy, gdy ma nietrywialne drzewo Böhma (istnieje czołowa postać normalna). ∎

Wniosek D.7 (adekwatność) *Jeśli $[\![M]\!]_\rho = [\![N]\!]_\rho$ dla dowolnego ρ, to $M \equiv N$.*

Dowód: Jeśli $[\![M]\!]_\rho = [\![N]\!]_\rho$ to także $[\![C[M]]\!]_\rho = [\![C[N]]\!]_\rho$ dla dowolnego $C[\ ]$. Jeśli więc $[\![C[M]]\!]_\rho \neq \bot$ to i $[\![C[N]]\!]_\rho \neq \bot$, czyli z rozwiązalności $C[M]$ wynika rozwiązalność $C[N]$. ∎

Teraz udowodnimy pełną abstrakcyjność modelu $\mathcal{D}_\infty$. Na początek taki prosty lemat:

Lemat D.8 *Niech $T_1 \preceq_\eta T_2$. Wówczas:*

1. *Dla dowolnego $A \in A(T_1)$ istnieje takie $B \in A(T_2)$, że $B \twoheadrightarrow_\eta A$.*
2. *Dla dowolnego $B \in A(T_2)$ istnieje takie $A \in A(T_1)$, że $B \twoheadrightarrow_\eta A$.*

Dowód: (1) W nieskończonym ciągu eta-ekspansji od T_1 do T_2, tylko skończenie wiele kroków dotyczy wierzchołków należących do aproksymanta A. Te eta-ekspansje przekształcają A w pewnego aproksymanta drzewa T_2.

(2) Analogicznie, tylko skończenie wiele eta-ekspansji pozostawia swój ślad w drzewie B. Odpowiadają one pewnej eta-redukcji z B do jakiegoś aproksymanta drzewa T_1. ∎

Lemat D.9 *Jeśli $T_1 \preceq_\eta T_2$ to $[\![T_1]\!]_\rho = [\![T_2]\!]_\rho$, dla każdego ρ.*

Dowód: Przypomnijmy, że $[\![T_1]\!]_\rho = \sup\{[\![A]\!]_\rho \mid A \in A(T_1)\}$ i podobnie dla T_2. Ponieważ eta-równe termy są równe w modelu, więc z lematu D.8 wynika od razu, że odpowiednie kresy są równe. ∎

Stąd natychmiast otrzymujemy:

Wniosek D.10 (pełna abstrakcyjność) *Jeśli $M \equiv N$, to $[\![M]\!]_\rho = [\![N]\!]_\rho$ dla dowolnego ρ.*

Dowód: Skoro $M \equiv N$, to $BT(M) \approx_\eta BT(N)$, czyli $BT(M) \preceq_\eta T \,_\eta\!\succeq BT(N)$. Z lematu D.9 wynika od razu, że $[\![BT(M)]\!]_\rho = [\![BT(N)]\!]_\rho$. A z twierdzenia o aproksymacji otrzymujemy, że także wartości $[\![M]\!]_\rho$ i $[\![N]\!]_\rho$ muszą być równe. ∎

Podziękowania

Dziękuję Pani Małgorzacie Maciejewskiej i Panom Tomaszowi Domańskiemu, Stanisławowi Findeisenowi, Krzysztofowi Gerasowi, Danielowi Hansowi, Szczepanowi Hummelowi, Wojciechowi Jaworskiemu, Szymonowi Kitowskiemu, Michałowi Kotowskiemu, Radosławowi Kujawie, Michałowi Misiakowi, Filipowi Murlakowi, Łukaszowi Osipiukowi, Pawłowi Parysowi, Jakubowi Pochrybniakowi, Michałowi Rutkowskiemu, Aleksemu Schubertowi, Adamowi Slaskiemu i Mateuszowi Zakrzewskiemu za wykrycie rozmaitych błędów we wcześniejszych wersjach tych notatek.