Formalizacja podstawowych pojęć rachunku lambda

Antoni Kościelski

1 Zmienne

Zbiór zmiennych będziemy oznaczać literą V. Zakładamy, że jest to zbiór nieskończony (przeliczalny). Chcemy mieć do dyspozycji każdą skończoną liczbę zmiennych. To, czym są zmienne nie ma większego znaczenia. Ważne jest to, że odróżniamy zmienne od innych rzeczy. Możemy myśleć, że są znaki lub identyfikatory. Dla oznaczenia zmiennych będziemy zwykle używać małych liter, w razie potrzeby z jakimiś indeksami.

2 Wyrażenia lambda rachunku

Wyrażenia lambda rachunku często nazywa się termami lambda rachunku lub krótko, λ-termami, a nawet termami. Zbiór wyrażeń lambda rachunku będziemy oznaczać symbolem Λ. Zbiór ten można definiować na wiele sposobów. Trudno zdecydować się na konkretną definicję, każda ma wady i zalety. Oprócz zmiennych w definicji będziemy używać

$$\lambda, \quad \cdot, \quad ., \quad (\ \text{oraz}\ ).$$

Są to różne znaki, które nie są zmiennymi.

2.1 Abstrakcyjna definicja wyrażeń

Ta definicja jest najprostsza, ale nie wyjaśnia, czym są wyrażenia. Zgodnie z tą definicją Λ jest najmniejszym zbiorem takim, że

1) zmienne są wyrażeniami (lambda rachunku),

2) jeżeli M i N są wyrażeniami, to MN jest wyrażeniem,

3) jeżeli x jest zmienną i M jest wyrażeniem, to λxM jest wyrażeniem.

Albo inaczej: Λ jest najmniejszym zbiorem takim, że

1) $x \in V \Rightarrow x \in \Lambda$,

2) $M, N \in \Lambda \Rightarrow MN \in \Lambda$ (ewentualnie $M, N \in \Lambda \Rightarrow M \cdot N \in \Lambda$),

3) $x \in V \wedge M \in \Lambda \Rightarrow \lambda xM \in \Lambda$.

Definicja ta powinna gwarantować, że są trzy, rozróżnialne (rozłączne) rodzaje wyrażeń: zmienne, aplikacje (wyrażenia postaci MN) i abstrakcje (wyrażenia postaci λxM). W szczególności, żadna aplikacja nie może być jednocześnie abstrakcją. Co więcej, mając aplikację w sposób jednoznaczny powinniśmy ustalić, co aplikujemy i do czego, czyli żądamy, że warunek $M_1N_1 = M_2N_2$ implikuje, że $M_1 = M_2$ i $N_1 = N_2$. Analogiczna własność powinna przysługiwać abstrakcji, a więc jeżeli $\lambda x_1 M_1 = \lambda x_2 M_2$, to $x_1 = x_2$ oraz $M_1 = M_2$.

Poza tym własności wyrażeń lambda rachunku powinno dać się dowodzić przez indukcję, zgodnie z następującym twierdzeniem

Twierdzenie 2.1 *Przypuśćmy, że φ jest własnością, która przysługuje lub nie poszczególnym wyrażeniom lambda rachunku. Jeżeli jednak*

1) *własność φ przysługuje wszystkim zmiennym z V,*

2) *fakt, że wyrażenia M i N mają własność φ pociąga za sobą, że MN też ma własność φ oraz*

3) *z tego, że M ma własność φ wynika, że dla dowolnej zmiennej x własność φ ma także wyrażenie λxM,*

to wszystkie wyrażenia lambda rachunku mają własność φ. □

2.2 Wyrażenia lambda rachunku jako drzewa

Wyrażenia lambda rachunku można uważać za drzewa binarne z węzłami etykietowanymi zmiennymi i symbolami λ oraz $\cdot$. W tym przypadku Λ jest najmniejszym zbiorem drzew binarnych spełniającym

1) do Λ należą jednoelementowe drzewa binarne z węzłem z etykietą, która jest zmienną,

2) jeżeli drzewa M i N należą do Λ, to do Λ należy także drzewo z korzeniem z etykietą $\cdot$, z lewym poddrzewem M i prawym poddrzewem N,

3) jeżeli $M \in \Lambda$, to do Λ należy także dowolne drzewo z korzeniem z etykietą λ, którego lewe poddrzewo ma jeden węzeł z etykietą będącą zmienną, a prawym poddrzewem jest M.

Łatwo przekonać się, że tak zdefiniowane wyrażenia mają własności wymienione w poprzednim rozdziale.

2.3 Wyrażenia lambda rachunku jako słowa

$$\hat{x}M = {\wedge}xM = {\wedge}xM = \lambda xM$$

Wyrażenia lambda rachunku można uważać też za słowa tworzone z liter alfabetu zawierającego zmienne z V oraz znaki

$$\lambda, \quad (\ \text{oraz}\ ).$$

Zbiór tak rozumianych wyrażeń jest najmniejszym zbiorem słów Λ takim, że

1) $V \subseteq \Lambda$,

2) jeżeli $M, N \in \Lambda$, to $(MN) \in \Lambda$,

3) jeżeli $M \in \Lambda$ i $x \in V$, to $(\lambda xM) \in \Lambda$.

Zaletą tej definicji jest to, że tak rozumiane wyrażenia rachunku lambda dają się łatwo reprezentować, także pisemnie, wadą – duża liczba wymaganych nawiasów. Zwykle korzysta się z tej definicji w połączeniu z zasadami opuszczania nawiasów.

Przyjmuje się, że

1) mamy prawo opuścić wewnętrzne nawiasy w wyrażeniu $((KM)N)$, a więc przyjmujemy, że
$$((KM)N) = (KMN),$$

2) mamy prawo opuścić wewnętrzne nawiasy w wyrażeniu $(\lambda x(M))$, a więc przyjmujemy, że
$$(\lambda x(M)) = (\lambda xM),$$

3) Mamy prawo pominąć najbardziej zewnętrzne nawiasy wyrażenia, czyli
$$(M) = M$$
(ale w tym przypadku dotyczy to szczególnych nawiasów, a nie jest to ogólna zasada),

4) zamiast λxM możemy napisać $\lambda x.M$, a wyrażenie $\lambda x_1 \ldots x_n.\lambda xM$ możemy skrócić do postaci $\lambda x_1 \ldots x_n x.M$, to znaczy przyjmujemy, że
$$\lambda x_1 \ldots x_n.M = \lambda x_1 \ldots \lambda x_n M$$

2.4 Gramatyka definiująca lambda wyrażenia

Niżej jest przedstawiona próba zdefiniowania gramatyki generującej wyrażenia rachunku lambda zgodne z zasadami opisanymi w poprzednim rozdziale.

1) ⟨λ-wyrażenie⟩ ::= ⟨uogólniona aplikacja⟩ | ⟨abstrakcja⟩

2) ⟨wyrażenie proste⟩ ::= ⟨zmienna⟩ | (⟨aplikacja⟩) | (⟨abstrakcja⟩)

3) ⟨uogólniona aplikacja⟩ ::= ⟨zmienna⟩ | ⟨aplikacja⟩

4) ⟨aplikacja⟩ ::= ⟨zmienna⟩⟨wyrażenie proste⟩ |
(⟨abstrakcja⟩)⟨wyrażenie proste⟩ | ⟨aplikacja⟩⟨wyrażenie proste⟩

5) ⟨abstrakcja⟩ ::= λ ⟨zmienne⟩ . ⟨uogólniona aplikacja⟩

6) ⟨zmienne⟩ ::= ⟨zmienna⟩ | ⟨zmienna⟩⟨zmienne⟩

7) ⟨zmienna⟩ ::= ⟨mała litera, ewentualnie z indeksami⟩

3 Podstawianie i podstawialność

3.1 Definicje

Dla dwóch λ-wyrażeń M i N i zmiennej x rekurencyjnie definiujemy podstawienie $M[x := N]$ wyrażenia N za zmienną x w wyrażeniu M. Przyjmujemy, że

1) jeżeli $M = x$, to $M[x := N] = x[x := N] = N$,

2) jeżeli M jest zmienną $y \neq x$, to $M[x := N] = y[x := N] = y$,

3) jeżeli M jest aplikacją M_1M_2, to

$$M[x := N] = (M_1M_2)[x := N] = (M_1[x := N])(M_2[x := n]),$$

4) jeżeli M jest abstrakcją $\lambda x.M_1$, to $M[x := N] = (\lambda x.M_1)[x := N] = \lambda x.M_1$,

5) jeżeli M jest abstrakcją $\lambda y.M_1$ i $y \neq x$, to

$$M[x := N] = (\lambda y.M_1)[x := N] = \lambda x.(M_1[y := N]).$$

Term N jest podstawialny za zmienną x w termie M, jeżeli w termie M żadne wolne wystąpienie zmiennej x nie znajduje się w zasięgu operatora abstrakcji λ wiążącego zmienną wolną termu N.

3.2 Najprostsze własności

Lemat 3.1 *Zawsze zachodzi wzór* $M[x := x] = M$. □

Lemat 3.2 *Jeżeli zmienna* x *nie jest wolna w termie* M, *to* $M[x := N] = M$ *dla dowolnego* N. □

Ważną własność podstawiania wyraża

Lemat 3.3 *Jeżeli* $x \neq y$ *i* $x \notin FV(L)$, *to*

$$M[x := N][y := L] = M[y := L][x := N[y := L]].$$ □

4 Wyrażenia de Bruijna

4.1 Definicja wyrażeń de Bruijna

Znaczenie programistyczne ma sposób przedstawiania wyrażeń lambda rachunku wymyślony przez Nicolasa de Bruijna. Może zostać opisany poprzez następującą gramatykę.

1) ⟨w. de Bruijna⟩ ::= ⟨zmienna⟩ | ⟨aplikacja⟩ | ⟨abstrakcja⟩

2) ⟨wyrażenie proste⟩ ::= ⟨zmienna⟩ | (⟨aplikacja⟩) | (⟨abstrakcja⟩)

3) ⟨aplikacja⟩ ::= ⟨zmienna⟩⟨wyrażenie proste⟩ |
(⟨abstrakcja⟩)⟨wyrażenie proste⟩ | ⟨aplikacja⟩⟨wyrażenie proste⟩

4) ⟨abstrakcja⟩ ::= λ ⟨w. de Bruijna⟩

5) ⟨zmienna⟩ ::= ⟨liczba naturalna⟩

W wyrażeniach tego rodzaju po symbolu λ nie piszemy zmiennej, a na pozostałych pozycjach, w których zwykłe są zmienne, umieszczamy liczby naturalne. Liczb naturalnych jednak nie można utożsamiać ze zmiennymi: w w wyrażeniach de Bruijna tylko przekazują informacje o zmiennych. Co więcej, w ustalonym wyrażeniu ta sama liczba przekazuje informacje zależne od jej umiejscowienia i w różnych miejscach może opisywać różne zmienne.

Wyrażenia de Bruijna można też uważać za drzewa. Mogą to być drzewa binarne, które mają trzy rodzaje węzłów: liście, które odpowiadają zmiennym i mają etykiety będące liczbami naturalnymi oraz wskaźniki do dwóch pustych drzew, węzły odpowiadające abstrakcjom, które zamiast lewego poddrzewa mają wskaźnik pusty, i w końcu węzły odpowiadające aplikacjom, które mają dwa wskaźniki do dwóch niepustych poddrzew.

Aby przekształcać wyrażenia w wyrażenia de Bruijna i odwrotnie potrzebny jest tzw. kontekst. Kontekst może być rozumiany jako ciąg wszystkich zmiennych bez powtórzeń. Przyjmijmy, że jeżeli Γ jest kontekstem, to Γ_n oznacza zmienną znajdującą się w kontekście Γ na n-tym miejscu, a $\Gamma(x)$ oznacza numer zmiennej x w kontekście Γ.

4.2 Podstawianie w wyrażeniach de Bruijna

Dla wyrażenia de Bruijna M i liczb naturalnych a i b zdefiniujemy teraz wyrażenie $M[a \leftarrow b]$. Operacja ta ma być odpowiednikiem operacji podstawiania $M[\Gamma_a := \Gamma_b]$ w zwykłym termie M za zmienną Γ_a zmiennej Γ_b.

Przyjmujemy, że

1) $a[a \leftarrow b] = b$ oraz $c[a \leftarrow b] = c$ dla liczb $c \neq a$,

2) $(MN)[a \leftarrow b] = M[a \leftarrow b]N[a \leftarrow b]$,

3) $(\lambda M)[a \leftarrow b] = \lambda M[a+1 \leftarrow b+1]$.

Definicja tej operacji pozwala częściowo odtworzyć sposób tworzenia wyrażeń de Bruijna. Przekształcając zwykły lambda term (rozumiany jako drzewo) w wyrażenie de Bruijna, konkretne wolne wystąpienie zmiennej x zastępujemy numerem zmiennej x powiększonym o liczbę operatorów lambda na ścieżce od tego wystąpienia do korzenia.

4.3 Przekształcanie wyrażeń w wyrażenia de Bruijna

Zdefiniujemy teraz funkcję *Usun_nazwy*, która dane lambda wyrażenie przekształca w odpowiadające mu wyrażenie de Bruijna. Algorytm definiujący tę funkcję będzie rekurencyjny i będzie korzystać z pomocniczej zmiennej h o wartościach naturalnych. Definicję funkcji *Usun_nazwy* można przedstawić w następujący sposób:

1) Dane: lambda wyrażenie W i kontekst Γ.

2) $Usun\_nazwy(W) = Usun\_nazwy(W, 0)$.

3) Jeżeli x jest zmienną, to $Usun\_nazwy(x, h) = \Gamma(x) + h$.

4) $Usun\_nazwy(MN, h) = Usun\_nazwy(M, h)Usun\_nazwy(N, h)$.

5) $Usun\_nazwy(\lambda x M, h) = \lambda Usun\_nazwy(M, h+1)[\Gamma(x) + h + 1 \leftarrow 0]$.

Teraz można pokusić się o wyjaśnienie, co znaczy liczba n w wyrażeniu de Bruijna. Wystąpienie liczby 0 oznacza albo zmienną związaną pierwszym (licząc od wystąpienia liczby) operatorem λ znajdującym się na ścieżce od tego wystąpienia do korzenia, albo zmienną Γ_0, jeżeli na tej ścieżce nie ma żadnego operatora. Wystąpienie liczby 1 oznacza albo zmienną związaną drugim operatorem λ znajdującym się na ścieżce od tego wystąpienia do korzenia, albo zmienną Γ_0, jeżeli na tej ścieżce jest jeden operator λ, albo też zmienną Γ_1, jeżeli na rozważanej ścieżce nie ma żadnego operatora. Dla większych n sytuacja jest analogiczna.

4.4 Przekształcanie wyrażeń de Bruijna w λ-termy

Teraz zdefiniujemy funkcję *Dodaj_nazwy*, która dane wyrażenie de Bruijna zamienia na wyrażenie rachunku lambda. Funkcja ta będzie korzystać z pewnego parametru $c \in N$, od którego będą zależeć wybierane nazwy zmiennych związanych. W ogólnym przypadku jest potrzebna jakaś zasada wyboru nazw tych zmiennych. Będziemy zakładać, że liczba c jest większa od wszystkich liczb występujących w danym jako argument wyrażeniu de Bruijna i jako zmiennych związanych będziemy używać zmiennych o numerach większych od c.

można przedstawić w następujący sposób:

1) Dane: lambda wyrażenie de Bruijna W i kontekst Γ.

2) $Dodaj\_nazwy(W) = Dodaj\_nazwy(W, 0)$.

3) Jeżeli x jest liczbą, to $Dodaj\_nazwy(x, h) = \Gamma_{x-h}$.

4) $Dodaj\_nazwy(MN, h) = Dodaj\_nazwy(M, h) Dodaj\_nazwy(N, h)$.

5) $Dodaj\_nazwy(\lambda M, h) = \lambda\Gamma_c Dodaj\_nazwy(M[0 \leftarrow c + h + 1], h + 1)$ i dodatkowo wykorzystanie nazwy (zmiennej) o numerze c powinno spowodować zwiększenie c o 1.

5 Rodzaje relacji

5.1 Relacje zgodne

Relacja R w zbiorze λ-termów jest zgodna (z operacjami λ-rachunku) jeżeli dla wszystkich $M, N, Z \in \Lambda$

1) warunek $M\ R\ N$ pociąga za sobą $(ZM)\ R\ (ZN)$ oraz $(MZ)\ R\ (NZ)$,

2) warunek $M\ R\ N$ implikuje, że $(\lambda x.M)\ R\ (\lambda x.N)$.

5.2 Kongruencje

Kongruencjami nazywamy zgodne relacje równoważności. Najprostszym przykładem kongruencji jest relacja równości.

5.3 Redukcje

Mamy dwa rodzaje redukcji: w jednym i w wielu krokach. Redukcja w jednym kroku najczęściej jest definiowana jako najmniejsza relacja zgodna rozszerzające pewne proste przekształcenie. Redukcja w jednym kroku wyznacza redukcję w wielu krokach, która jest krótko nazywana redukcją.

Redukcją zwykle nazywamy zgodną relację zwrotną i przechodną.

Mając redukcję w jednym kroku $\rightarrow$ definiujemy relację $\twoheadrightarrow$ przyjmując, że jest to najmniejsza relacja spełniająca dla wszystkich $L, M, N \in \Lambda$ warunki

1) jeżeli $M = N$, to $M \twoheadrightarrow N$,

2) jeżeli $M \rightarrow N$, to $M \twoheadrightarrow N$,

3) jeżeli $M \twoheadrightarrow L$ i $L \twoheadrightarrow N$, to $M \twoheadrightarrow N$.

Lemat 5.1 *Jeżeli relacja redukcji w jednym kroku $\rightarrow$ jest zgodna, to relacja $\twoheadrightarrow$ jest redukcją.* □

Bywa, że musimy rozważać bardziej ogólne redukcje, rozumiane jako zgodne i przechodnie rozszerzenia pewnej kongruencji $\cong$.

Wtedy, mając redukcję w jednym kroku $\rightarrow$, definiujemy relację $\twoheadrightarrow$ przyjmując, że jest to najmniejsza relacja spełniająca dla wszystkich $L, M, N \in \Lambda$ warunki

1) jeżeli $M \cong N$, to $M \twoheadrightarrow N$,

2) jeżeli $M \rightarrow N$, to $M \twoheadrightarrow N$,

3) jeżeli $M \twoheadrightarrow L$ i $L \twoheadrightarrow N$, to $M \twoheadrightarrow N$.

Lemat 5.2 *Jeżeli $\cong$ jest kongruencją i relacja redukcji w jednym kroku $\rightarrow$ jest zgodna, to relacja $\twoheadrightarrow$ jest redukcją.* □

5.4 Konwersje

Mając relację redukcji $\twoheadrightarrow$ definiujemy związaną z nią relację konwersji $\equiv$ przyjmując, że jest to najmniejsza relacja spełniająca dla wszystkich $L, M, N \in \Lambda$ warunki

1) jeżeli $M \twoheadrightarrow N$, to $M \equiv N$,

2) jeżeli $M \equiv N$, to $N \equiv M$,

3) jeżeli $M \equiv L$ i $L \equiv N$, to $M \equiv N$.

Lemat 5.3 *Jeżeli relacja $\twoheadrightarrow$ jest redukcją, to relacja $\cong$ jest kongruencją.* □

6 α-konwersja

Relację α-redukcji w jednym kroku, czyli relację $\rightarrow_\alpha$, definiujemy jako najmniejszą relację zgodną z operacjami rachunku lambda zawierającą wszystkie pary

$$\lambda x.M \rightarrow_\alpha \lambda y.M[x := y],$$

gdzie y jest zmienną nie będącą wolną w termie M ($y \notin FV(M)$) i podstawialną w M za zmienną x.

Relacja α-redukcji w jednym kroku wyznacza tak, jak to zostało wyżej opisane, relację α-redukcji $\twoheadrightarrow_\alpha$ i relację α-konwersji $\equiv_\alpha$. Relację α-konwersji będziemy najczęściej oznaczać symbolem $\equiv$, a czasem może być ona oznacza także symbolem $=_\alpha$, a nawet $=$.

Oczywiście, α-konwersja jest kongruencją.

Lemat 6.1 *1) Jeżeli $y \notin FV(M)$, to $x \notin FV(M[x := y])$.*

2) Jeżeli $y \notin FV(M)$, to y nie występuje w termie $M[x := y]$ jako zmienna wolna w zasięgu kwantyfikatora wiążącego x.

3) Jeżeli $y \notin FV(M)$, to zmienna x jest podstawialna za y w termie $M[x := y]$.

4) Jeżeli $y \notin FV(M)$, to $\lambda y.M[x := y] \to_\alpha M[x := y][y := x] = M$.

5) Relacja $\to_\alpha$ jest symetryczna.

6) Relacja $\twoheadrightarrow_\alpha$ jest symetryczna. □

Wniosek 6.2 *Relacja α-redukcji $\twoheadrightarrow_\alpha$ jest relacją α-konwersji $\equiv_\alpha$.* □

7 α-konwersja a wyrażenia de Bruijna

7.1 Zmienne w wyrażeniach de Bruijna

Rolę zmiennych w termach de Bruijna pełnią liczby naturalne. Będziemy analizować wystąpienia zmiennych w takich wyrażeniach.

Jeżeli wyrażenie de Bruijna uważamy za drzewo, to wystąpieniem zmiennej w tym wyrażeniu będziemy nazywać dowolny liść tego drzewa. Jeżeli liściem tym (w tym liściu) jest liczba x, to będziemy mówić, że jest to wystąpienie liczby x, a nawet to wystąpienie – nie do końca poprawnie – będziemy utożsamiać z liczbą x.

Jeżeli wyrażenie de Bruijna uważamy za ciąg znaków i liczb, to wystąpienie zmiennej x to pozycja w tym ciągu, na której znajduje się liczba x.

Dla każdego wystąpienia zmiennej x w wyrażeniu de Bruijna M definiujemy indukcyjnie zagnieżdżenie $z_M(x)$ tego wystąpienia. Jeżeli M jest zmienną (czyli zmienną x), to $z_M(x) = 0$. Jeżeli $M = M_1M_2$ i wystąpienie x znajduje się w termie M_i, to $z_M(x) = z_{M_i}(x)$. W końcu, jeżeli $M = \lambda M'$, to $z_M(x) = z_{M'}(x) + 1$.

Wystąpienia w termie M zmiennej x nazywamy związanym, jeżeli $x < z_M(x)$. Pozostałe wystąpienia zmiennych nazywamy wolnymi.

Lemat 7.1 *Wykonywanie podstawienia $M[a \leftarrow b]$ polega na zamianie wystąpień liczb x takich, że $x = a + z_M(x)$ liczbami $b + z_M(x)$.* □

Lemat 7.2 *Jeżeli w wyrażeniu de Bruijna M występuje liczba x taka, że $z_M(x) \leqslant x < z_M(x) + h$, to podczas wykonywania procedury $Dodaj\_nazwy(M, h)$ występuje błąd polegający na próbie ustalenia nazwy Γ_c dla pewnego $c < 0$. W przeciwnym razie, jeżeli dla wszystkich wystąpień liczb x w termie M mamy albo $x < z_M(x)$, albo $z_M(x) + h \leqslant x$, to procedura $Dodaj\_nazwy(x, h)$ jest wykonywana poprawnie.* □

Wniosek 7.3 *Procedura $Dodaj\_nazwy(M)$, czyli $Dodaj\_nazwy(M, 0)$ jest wykonywana poprawnie dla dowolnego wyrażenia de Bruijna M.* □

7.2 Występowanie zmiennych w termach de Bruijna

Zdefiniujemy teraz pojęcie, które ma odpowiadać w przypadku termów de Bruijna warunkowi $x \in FV(M)$. Pojęcie to zostanie zdefiniowane przez indukcję ze względu na budowę termu de Bruijna M. Pamiętajmy, że rolę zmiennych w termach de Bruijna pełnią liczby naturalne.

Jeżeli term de Bruijna M jest liczbą naturalną, to liczba x występuje w M wtedy i tylko wtedy, gdy jest równa M. Jeżeli term M jest aplikacją M_1M_2, to liczba x występuje w M wtedy i tylko wtedy, gdy x występuje w M_1 lub w M_2. Jeżeli term M jest abstrakcją λN, to liczba x występuje w M wtedy i tylko wtedy, gdy liczba $x+1$ występuje w termie N.

Lemat 7.4 *Dla dowolnego termu de Bruijna M i dla dowolnych liczb x, y i $a \neq x$, jeżeli x nie występuje w M, to nie występuje też w termie $M[y \leftarrow a]$.*

Dowód. Lemat ten dowodzimy przez indukcję ze względu na M. Sprawdzimy go jedynie w przypadku abstrakcji $M = \lambda N$.

Załóżmy, że x nie występuje w termie λN. Oznacza to, że $x+1$ nie występuje w termie N. Term $(\lambda N)[y \leftarrow a]$ jest równy $\lambda N[y+1 \leftarrow a+1]$. Aby sprawdzić, że x nie występuje w $\lambda N[y+1 \leftarrow a+1]$, badamy, czy $x+1$ nie występuje w $N[y+1 \leftarrow a+1]$. Tak jest na mocy założenia indukcyjnego dla termu N. □

Lemat 7.5 *Jeżeli $x \neq a$, to x nie występuje w termie $M[x \leftarrow a]$.*

Dowód. Przez indukcję ze względu na M dowodzimy, że teza lematu zachodzi dla wszystkich liczb x i a. □

Lemat 7.6 *Jeżeli liczba x nie występuje w termie de Bruijna M, to $M[x \leftarrow a] = M$.*

Dowód. Lemat ten ma oczywisty dowód przez indukcję ze wględu na M. □

7.3 Własności podstawiania

Lemat 7.7 *Jeeli liczby a, a', b, b' speniają warunki $a \neq a'$, $a \neq b'$ oraz $b \neq a'$, to dla wszystkich termów de Bruijna M mamy*

$$M[a \leftarrow b][a' \leftarrow b'] = M[a' \leftarrow b'][a \leftarrow b].$$

Dowód. Lemat dość oczywisty, dowodzony przez indukcję ze względu na budowę termu M, dla wszystkich możliwych parametrów. Najważniejsze, że przechodzi dla elementarnych termów, czyli liczb. „Drugie" kroki są łatwe do wykazania. □

Lemat 7.8 *Jeżeli $a \neq b$, to dla wszystkich termów de Bruijna M mamy*

$$M[a \leftarrow b][a \leftarrow b'] = M[a \leftarrow b].$$

Dowód. Taki jak poprzedni lub z lematów 7.5 i 7.6. □

Lemat 7.9 *Dla wszystkich termów de Bruijna M i liczb b nie występujących w M zachodzi równość*

$$M[a \leftarrow b][b \leftarrow c] = M[a \leftarrow c].$$

Dowód. Taki jak poprzedni. □

7.4 Własności usuwania

Lemat 7.10 *Jeżeli zmienna x nie jest wolna w (zwykłym) termie M, to $\Gamma(x)+h$ nie występuje w termie $Usun\_nazwy(M,h)$.*

Dowód. Również ten lemat łatwo dowodzi się przez indukcję ze względu na budowę termu M. □

Lemat 7.11 *Jeżeli M jest λ-termem i podstawialna za x w M zmienna y nie należy do $FV(M)$, to dla wszystkich liczb h*

$$Usun\_nazwy(M,h)[\Gamma(x)+h \leftarrow \Gamma(y)+h)] = Usun\_nazwy(M[x := y], h).$$

Dowód. Przez indukcję ze względu na M.

Przypadek 1: $M = x$. Wtedy

$$Usun\_nazwy(x,h)[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \Gamma(y)+h.$$

Podobnie przekształcamy drugą stronę wzoru do tego samego rezultatu.

Przypadek 2: zmienną M jest $z \neq x$. Wtedy także $z \neq y$ oraz

$$Usun\_nazwy(z,h)[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \Gamma(z)+h,$$

gdyż dla różnych x i z mamy $\Gamma(x) \neq \Gamma(z)$ (z własności kontekstów). Tak samo przekształcamy drugą stronę wzoru.

Przypadek 3: M jest aplikacją. Teza wynika stąd, że wszystkie rozważane operacje, a więc oba podstawiania $\cdot[\cdot := \cdot]$ oraz $\cdot[\cdot \leftarrow \cdot]$, a także $Usun\_nazwy$, można przestawiać z operacją aplikacji.

Przypadek 4: M jest abstrakcją $\lambda x M'$. Wtedy

$$\begin{aligned}
&Usun_nazwy(\lambda x M', h)[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \\
&= (\lambda Usun_nazwy(M', h+1)[\Gamma(x)]+h+1 \leftarrow 0])[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \\
&= \lambda Usun_nazwy(M', h+1)[\Gamma(x)]+h+1 \leftarrow 0][\Gamma(x)+h+1 \leftarrow \Gamma(y)+h+1] = \\
&= \lambda Usun_nazwy(M', h+1)[\Gamma(x)]+h+1 \leftarrow 0] = \\
&= Usun_nazwy(\lambda x M', h) = Usun_nazwy((\lambda x M')[x := y], h),
\end{aligned}$$

na mocy lematu 7.8.

Przypadek 5: M jest abstrakcją $\lambda z M'$ dla $z \neq x$ oraz $x \notin FV(M')$. Wtedy

$$\begin{aligned}
&Usun_nazwy(\lambda z M', h)[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \\
&= (\lambda Usun_nazwy(M', h+1)[\Gamma(z)]+h+1 \leftarrow 0])[\Gamma(x)+h \leftarrow \Gamma(y)+h] = \\
&= \lambda Usun_nazwy(M', h+1)[\Gamma(z)]+h+1 \leftarrow 0][\Gamma(x)+h+1 \leftarrow \Gamma(y)+h+1] = \\
&= \lambda Usun_nazwy(M', h+1)[\Gamma(x)]+h+1 \leftarrow 0] = \\
&= Usun_nazwy(\lambda z M', h) = Usun_nazwy((\lambda z M')[x := y], h)
\end{aligned}$$

na mocy lematów z rozdziału 7.2.

Przypadek 6: M jest abstrakcją $\lambda z M'$ dla $z \neq x$ oraz $x \in FV(M')$. Tym razem podstawialność y implikuje, że $y \neq z$. Na mocy lematu 7.7

$$
\begin{aligned}
&Usun\_nazwy(\lambda z M', h)[\Gamma(x) + h \leftarrow \Gamma(y) + h] = \\
&= (\lambda Usun\_nazwy(M', h+1)[\Gamma(z)] + h + 1 \leftarrow 0])[\Gamma(x) + h \leftarrow \Gamma(y) + h] = \\
&= \lambda Usun\_nazwy(M', h+1)[\Gamma(z)] + h + 1 \leftarrow 0][\Gamma(x) + h + 1 \leftarrow \Gamma(y) + h + 1] = \\
&= \lambda Usun\_nazwy(M', h+1)[\Gamma(x) + h + 1 \leftarrow \Gamma(y) + h + 1][\Gamma(z)] + h + 1 \leftarrow 0] = \\
&= \lambda Usun\_nazwy(M'[x := y], h+1)[\Gamma(z) + h + 1 \leftarrow 0] = \\
&= Usun\_nazwy((\lambda z M')[x := y], h). \ \square
\end{aligned}
$$

Wniosek 7.12 *Przypuśćmy, że mamy dane lambda term M i zmienną y. która nie jest wolna w M i jest podstawialna w M za zmienną x. Wtedy dla dowolnego naturalnego h zachodzi równość*

$$Usun\_nazwy(\lambda x.M, h) = Usun\_nazwy(\lambda y.M[x := y], h).$$

Dowód. Zauważmy, że

$$
\begin{aligned}
&Usun\_nazwy(\lambda y.M[x := y], h) = \\
&= \lambda Usun\_nazwy(M[x := y], h+1)[\Gamma(y) + h + 1 \leftarrow 0] = \\
&= \lambda Usun\_nazwy(M, h+1)[\Gamma(x) + h + 1 \leftarrow \Gamma(y) + h + 1][\Gamma(y) + h + 1 \leftarrow 0] = \\
&= \lambda Usun\_nazwy(M, h+1)[\Gamma(x) + h + 1 \leftarrow 0] = Usun\_nazwy(\lambda y.M[x := y], h).
\end{aligned}
$$

Poszczególne równości otrzymujemy z lematów 7.11, 7.10 oraz 7.9. $\square$

7.5 Termy h-poprawne

Zaczynamy od pomocniczego pojęcia związanego z operacją *Dodaj_nazwy*. Wykonanie tej operacji może zakończyć się błędem polegającym na otrzymaniu ujemnego argumentu kontekstu. Gwarancją poprawności wykonania operacji $Dodaj\_nazwy(M, h)$ jest h poprawność termu M.

Term de Bruijna M nazywamy h-poprawnym, jeżeli jest on liczbą i $M \geqslant h$, albo jest on aplikacją $M_1 M_2$ i oba jej człony M_1 i M_2 są h poprawne, albo też jest on abstrakcją $\lambda M'$ i term $M'[0 \leftarrow c]$ jest $h+1$-poprawny dla $c \geqslant h+1$ i większego od innych liczb w termie M' (np. dla najmniejszej liczby c o podanych własnościach).

Lemat 7.13 *Jeżeli M jest h-poprawny, to także h-poprawnym jest dowolny term $M[a \leftarrow b + h]$.*

Dowód. Dowodzimy to przez indukcję ze względu na M i dowód jest prosty. $\square$

Z powyższych lematów wynika

Wniosek 7.14 *Jeżeli M jest h-poprawny, to operacja $Dodaj\_nazwy(M, h)$ jest wykonywana poprawnie (nie powoduje błędu).* $\square$

7.6 Usuwanie i dodawanie razem

Lemat 7.15 *Dla dowolnego h i dowolnego termu h-poprawnego termu de Bruijna M mamy*

$$Usun\_nazwy(Dodaj\_nazwy(M,h),h) = M.$$

W szczególności, operacja Usun_nazwy przyjmuje jako wartości wszystkie termy de Bruijna.

Dowód. Przez indukcję ze względu na M.

Jeżeli M jest liczbą, to

$$Usun\_nazwy(Dodaj\_nazwy(M,h),h) = Usun\_nazwy(\Gamma_{M-h},h) = (M-h)+h = M.$$

Jeżeli M jest aplikacją, to teza lematu wynika w oczywisty sposób z założeń indukcyjnych.

Przypuśćmy, że M jest abstrakcją $\lambda M'$. Wtedy

$$\begin{aligned}
&Usun_nazwy(Dodaj_nazwy(M,h),h) = \\
&= Usun_nazwy(\lambda\Gamma_c Dodaj_nazwy(M'[0 \leftarrow c+h+1], h+1), h) = \\
&= \lambda Usun_nazwy(Dodaj_nazwy(M'[0 \leftarrow c+h+1], h+1), h+1) \\
&\qquad\qquad [\Gamma(\Gamma_c)+h+1 \leftarrow 0] = \\
&= \lambda M'[0 \leftarrow c+h+1][\Gamma(\Gamma_c)+h+1 \leftarrow 0] = \\
&= \lambda M'[0 \leftarrow c+h+1][c+h+1 \leftarrow 0] = \lambda M' = M. \ \square
\end{aligned}$$

Lemat 7.16 *Dla dowolnego h i dowolnego termu h-poprawnego termu de Bruijna M mamy*

$$Dodaj\_nazwy(Usun\_nazwy(M,h),h) \equiv_\alpha M.$$

Dowód. Przez indukcję ze względu na budowę termu M. Jest to oczywiste dla aplikacji. Dla zmiennej M mamy

$$\begin{aligned}
&Dodaj_nazwy(Usun_nazwy(M,h),h) = Dodaj_nazwy(\Gamma(M)+h,h) = \\
&= \Gamma_{\Gamma(M)+h-h} = \Gamma_{\Gamma(M)} = M \equiv_\alpha M.
\end{aligned}$$

W końcu, dla abstrakcji $\lambda x M$ mamy

$$\begin{aligned}
&Dodaj_nazwy(Usun_nazwy(\lambda x M,h),h) = \\
&= Dodaj_nazwy(\lambda Usun_nazwy(M,h+1)[\Gamma(x)+h+1 \leftarrow 0], h) = \\
&= \lambda\Gamma_c Dodaj_nazwy(\\
&\qquad Usun_nazwy(M,h+1)[\Gamma(x)+h+1 \leftarrow 0][0 \leftarrow c+h+1], h+1) \\
&= \lambda\Gamma_c Dodaj_nazwy(Usun_nazwy(M,h+1)[\Gamma(x)+h+1 \leftarrow c+h+1], h+1) \\
&= \lambda\Gamma_c Dodaj_nazwy(Usun_nazwy(M,h+1),h+1)[x := \Gamma_c] \\
&= \lambda\Gamma_c M[x := \Gamma_c] \equiv_\alpha M. \ \square
\end{aligned}$$

Jako wniosek z prowadzonych rozważań otrzymujemy

Twierdzenie 7.17 *Dla każdej pary wyrażeń rachunku lambda M i N, warunek $M \equiv_\alpha N$ jest równoważny równości $Usun\_nazwy(M) = Usun\_nazwy(N)$.* $\square$

8 β-konwersja

8.1 β-redukcja w jednym kroku

Relację β-redukcji w jednym kroku, czyli relację $\rightarrow_\beta$, definiujemy jako najmniejszą relację zgodną z operacjami rachunku lambda zawierającą pary

$$(\lambda x.M)N \rightarrow_\beta M'[x := N],$$

dla pewnego termu M', dla którego $M \twoheadrightarrow_\alpha M'$ i term N spełnia warunek podstawialności w M' za zmienną x.

Przypomnijmy, że term N spełnia warunek podstawialności w termie M' za zmienną x, jeżeli żadne wolne wystąpienie zmiennej x w termie M' nie znajduje się w zasięgu operatora λ wiążącego zmienne wolne z termu N.

Lemat 8.1 *Jeżeli $M \rightarrow_\beta M'$ i $N \rightarrow_\beta N'$ dla pewnych lambda termów M i N takich, że $M \equiv_\alpha N$, to także $M' \equiv_\alpha N'$.*

8.2 β-redukcja

Relacja β-redukcji w jednym kroku wyznacza (podobnie, jak to zostało wyżej opisane) relację β-redukcji $\twoheadrightarrow_\beta$. Przyjmujemy, że

1) jeżeli $M \equiv N$, to $M \twoheadrightarrow_\beta N$,

2) jeżeli $M \rightarrow_\beta N$, to $M \twoheadrightarrow_\beta N$,

3) jeżeli $M \twoheadrightarrow_\beta L$ i $L \twoheadrightarrow_\beta N$, to $M \twoheadrightarrow_\beta N$.

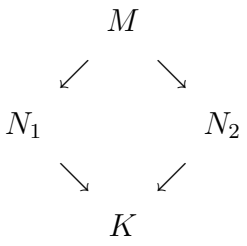
Relacja β-redukcji $\twoheadrightarrow_\beta$ wyznacza relację β-konwersji $=_\beta$ tak, jak to zostało wyżej opisane. Symbole $\rightarrow_\beta$, $\twoheadrightarrow_\beta$ i $=_\beta$ będziemy najczęściej zastępować symbolami $\rightarrow$, $\twoheadrightarrow$ i $=$.

Oczywiście, relacja β-konwersji jest kongruencją.

9 Abstrakcyjne twierdzenie Churcha-Rossera

9.1 Opis sytuacji

Przypuśćmy że mamy określoną w jakimś zbiorze relację $\rightarrow$ taką, że dla dowolnych M, N_1 i N_2 takich, że $M \rightarrow N_1$ i $M \rightarrow N_2$ istnieje K spełniający $N_1 \rightarrow K$ oraz $N_2 \rightarrow K$. Sytuację tę można przedstawić na rysunku



O takiej relacji mówimy, że ma własność $\diamond$.

Relację $\rightarrow$ uważamy za redukcję w jednym kroku i rozszerzamy do relacji redukcji $\twoheadrightarrow$ przyjmując, że jest to najmniejsza relacja spełniająca warunki

1) jeżeli $M = N$, to $M \twoheadrightarrow N$,

2) jeżeli $M \to N$, to $M \twoheadrightarrow N$,

3) jeżeli $M \twoheadrightarrow L$ i $L \twoheadrightarrow N$, to $M \twoheadrightarrow N$.

Relację $\twoheadrightarrow$ można też definiować zgodnie z następującym lematem:

Lemat 9.1 *Relacja $\twoheadrightarrow$ jest najmniejszą relacją spełniającą następujące warunki:*

1) jeżeli $M = N$, to $M \twoheadrightarrow N$,

2) jeżeli $M \to L$ i $L \twoheadrightarrow N$, to $M \twoheadrightarrow N$.

Dowód. Przyjmijmy, że $\twoheadrightarrow$ oznacza zwykła relację redukcji, zdefiniowaną jako najmniejszą relację o własnościach podanych przed sformułowaniem lematu, a $\twoheadrightarrow'$ – analogicznie zdefiniowaną relację o własnościach podanych w treści lematu.

Oczywiście, relacja $\twoheadrightarrow$ ma własności wymagane od relacji $\twoheadrightarrow'$. Wobec tego, mamy zawieranie $\twoheadrightarrow' \subseteq \twoheadrightarrow$.

Jest też oczywiste, że relacja $\twoheadrightarrow'$ ma dwie pierwsze własności wymagane od relacji $\twoheadrightarrow$. Pokażemy, że ma też trzecią własność. Jeżeli uda się nam to zrobić, to relacja $\twoheadrightarrow$, jako najmniejsza o tych trzech własnościach, okaże się zawarta w $\twoheadrightarrow'$. Oba zawierania $\twoheadrightarrow' \subseteq \twoheadrightarrow$ i $\twoheadrightarrow \subseteq \twoheadrightarrow'$ z kolei implikują równość $\twoheadrightarrow' = \twoheadrightarrow$, która jest tezą lematu.

Mamy więc wykazać, że jeżeli $M \twoheadrightarrow' L$ i $L \twoheadrightarrow' N$, to $M \twoheadrightarrow' N$. W tym celu rozważmy relację

$$\mathcal{R} = \{\langle M, L\rangle : \forall\, N\ (L \twoheadrightarrow' N \Rightarrow M \twoheadrightarrow' N)\}.$$

Powinniśmy dowieść, że relacja $\mathcal{R}$ ma obie własności wymagane w sformułowaniu lematu od relacji $\twoheadrightarrow'$. Nie jest to trudne do sprawdzenia. Z tego faktu wynika, że relacja $\twoheadrightarrow'$ zawiera się w $\mathcal{R}$, a to oznacza, że relacja $\twoheadrightarrow'$ ma trzecią własność wymaganą od relacji $\twoheadrightarrow$. Wyżej już zauważyliśmy, że to kończy dowód. □

Z kolei relacja $\twoheadrightarrow$ wyznacza konwersję $\equiv$, czyli najmniejszą relację taką, że

1) jeżeli $M \twoheadrightarrow N$, to $M \equiv N$,

2) jeżeli $M \equiv N$, to $N \equiv M$,

3) jeżeli $M \equiv L$ i $L \equiv N$, to $M \equiv N$.

9.2 Lematy pomocnicze

Lemat 9.2 *Jeżeli relacja $\to$ ma własność $\diamond$, to dla dowolnych M, M_1 i M_2 takich, że $M \to M_1$ i $M \twoheadrightarrow M_2$ istnieje N spełniający $M_1 \twoheadrightarrow N$ oraz $M_2 \to N$.*

Dowód. Zamiast korzystać bezpośrednio z definicji relacji $\twoheadrightarrow$ posługujemy się charakteryzacją tej relacji z lematu 9.1. □

Lemat 9.3 *Jeżeli relacja $\to$ ma własność $\diamond$, to dla dowolnych M, M_1 i M_2 takich, że $M \twoheadrightarrow M_1$ i $M \twoheadrightarrow M_2$ istnieje N spełniający $M_1 \twoheadrightarrow N$ oraz $M_2 \twoheadrightarrow N$.*

Dowód. Lemat ten wynika z poprzedniego i z charakteryzacji relacji $\twoheadrightarrow$ z lematu 9.1. □

9.3 Abstrakcyjne twierdzenie Churcha-Rossera

Twierdzenie 9.4 *Jeżeli relacja $\rightarrow$ spełnia warunek $\diamond$, to dla dowolnych M_1 i M_2 takich, że $M_1 \equiv M_2$ istnieje N spełniający $M_1 \twoheadrightarrow N$ oraz $M_2 \twoheadrightarrow N$.*

Dowód. Znowu definiujemy pomocnicze relacje

$$\mathcal{R} = \{\langle M_1, M_2\rangle : \exists\, N\ M_1 \twoheadrightarrow N \wedge M_2 \twoheadrightarrow N\}.$$

□

10 Wnioski z twierdzenia Churcha-Rossera

10.1 Relacja równoległej β-redukcji

Relacja równoległej β-redukcji (w jednym kroku) $\rightarrow_{r\beta}$ jest najmniejszą relacją w zbiorze lambda termów spełniającą

1) $M \rightarrow_{r\beta} M$,

2) jeżeli $M \rightarrow_{r\beta} M_1$, $M_1 \rightarrow_\alpha M_1'$, $N \rightarrow_{r\beta} N_1$ i N_1 jest podstawialny za zmienną x w termie M_1', to $(\lambda x M)N \rightarrow_{r\beta} M_1'[x := N_1]$,

3) jeżeli $M \rightarrow_{r\beta} M_1$ oraz $N \rightarrow_{r\beta} N_1$, to $MN \rightarrow_{r\beta} M_1N_1$,

4) jeżeli $M \rightarrow_{r\beta} M_1$, to $\lambda x M \rightarrow_{r\beta} \lambda x M_1$.

Relacja równoległej redukcji odpowiada sytuacji, w której mając dany term lambada rachunku wybieramy w nim kilka redeksów (także 0 redeksów), a następnie redukujemy jednocześnie wszystkie wybrane redeksy.

Lemat 10.1 *Relacja $\rightarrow_\beta$ jest częścią równoległej β-redukcji $\rightarrow_{r\beta}$, a więc $\rightarrow_\beta \subseteq \rightarrow_{r\beta}$.* □

Lemat 10.2 *Relacja równoległej β-redukcji $\rightarrow_{r\beta}$ jest częścią relacji β-redukcji $\rightarrow \rightarrow_\beta$, a więc $\rightarrow_{r\beta} \subseteq \twoheadrightarrow_\beta$.*

Dowód. Relacja $\twoheadrightarrow_\beta$ ma własności wymagane od relacji $\rightarrow_{r\beta}$. Na przykład, ma własność 2) relacji $\rightarrow_{r\beta}$, czyli

2) jeżeli $M \twoheadrightarrow_\beta M_1$, $M_1 \rightarrow_\alpha M_1'$, $N \twoheadrightarrow_\beta N_1$ i N_1 jest podstawialny za zmienną x w termie M_1', to $(\lambda x M)N \twoheadrightarrow_\beta M_1'[x := N_1]$.

Dowód tej własności nie nastręcza większych trudności. Jeżeli $M \twoheadrightarrow_\beta M_1$, to ze zgodności $\twoheadrightarrow_\beta$ także mamy $\lambda x M \twoheadrightarrow_\beta \lambda x M_1$ oraz $(\lambda x M)N \twoheadrightarrow_\beta (\lambda x M_1)N$. Podobnie, $(\lambda x M_1)N \twoheadrightarrow_\beta (\lambda x M_1)N_1$. W końcu, $(\lambda x M_1)N_1 \rightarrow_\beta M_1'[x := N_1]$. Korzystając z przechodniości relacji $\twoheadrightarrow_\beta$ otrzymujemy, że $(\lambda x M)N \twoheadrightarrow_\beta M_1'[x := N_1]$, a to kończy dowód rozważanej własności.

Pozostałe, wymagane własności relacji $\twoheadrightarrow_\beta$ dowodzimy analogicznie. Relacja $\rightarrow_{r\beta}$, jako najmniejsza relacja o tych własnościach, spełnia zawieranie $\rightarrow_{r\beta} \subseteq \twoheadrightarrow_\beta$. □

Z udowodnionych lematów jako oczywisty wniosek otrzymujemy

Twierdzenie 10.3 *Relacje $\twoheadrightarrow_\beta$ oraz $\twoheadrightarrow_{r\beta}$ są identyczne.* □

10.2 Termy z kolorowymi redeksami

Będziemy rozważać termy z kilkoma rodzajami operatorów lambda. W tym kontekscie będziemy mówić o kolorowych operatorach lambda. Jeżeli będzie to konieczne, będziemy je oznaczać symbolem $\underline{\lambda}$, ale może być ich kilka rodzajów. Można by je rzeczywiście zapisywać używając kolorowego druku. Oprócz kolorowych operatorów będą też niepokolorowane, możnaby je określać jako bezbarwne, albo wręcz przeciwnie, jako czarne, zapisane w zwykły sposób. Będą oznaczane zwykłym symbolem λ. Będziemy dodatkowo zakładać, że kolorowe mogą być tylko pierwsze operatory lambda w redeksach.

Pojęcie termu z kolorowymi redeksami definiujemy (w zwykłej konwencji) jak następuje:

1) zmienne są termami (z kolorowymi redeksami),

2) jeżeli M i N są termami, to MN jest termem,

3) jeżeli x jest zmienną i M jest termem, to λxM jest termem,

4) jeżeli x jest zmienną oraz M i N są termami, to $(\underline{\lambda}xM)N$ jest termem.

Pojęcia α-konwersji, podstawiania i podstawialności dla termów z kolorowymi redeksami definiujemy tak, jak dla zwykłych termów, nie zwracając uwagi na kolory operatorów λ. W formalnych definicjach takie podejście wymaga dopisania własności kolorowych operatorów lambda analogicznych do własności zwykłych operatorów.

Relację β-redukcji w jednym kroku dla termów z kolorowymi redeksami, czyli relację $\rightarrow_{k\beta}$, definiujemy jako najmniejszą relację zgodną z operacjami rachunku lambda zawierającą pary

$$(\underline{\lambda}x.M)N \rightarrow_{k\beta} M'[x := N],$$

dla pewnego termu M', dla którego $M \twoheadrightarrow_{\alpha} M'$ i term N spełnia warunek podstawialności w M' za zmienną x. Oznacza to, że nie redukujemy w tym sensie redeksów z bezbarwnymi operatorami lambda.

Kolorowanie redeksów wprowadza do lambda rachunku istotne ograniczenie. W zwykłym lambda rachunku redukcja redeksu może zwiększać liczbę redeksów na dwa sposoby: poprzez kopiowania i poprzez tworzenie nowych redeksów. Na przykład, redukując term $(\lambda x.xxx)M$ do MMM, trzykrotnemu skopiowaniu ulegają redeksy występujące w termie M. Jednocześnie może powstać nowy redeks: wystarczy, aby term M był abstrakcją: powstaje wtedy redeks MM. Jeżeli redukcję ograniczamy do kolorowych redeksów, to zwiększenie liczby kolorowych redeksów powodowane jest wyłącznie kopiowaniem i nie powstają nowe kolorowe redeksy. Nowy redeks co prawda powstaje, ale nie może być on kolorowy i nie można go zredukować stosując tylko redukcje kolorowych redeksów.

Definiujemy także równoległą redukcję kolorowych redeksów: relacja równoległej β-redukcji w jednym kroku kolorowych redeksów $\rightarrow_{rk\beta}$ jest najmniejszą relacją w zbiorze lambda termów z kolorowymi redeksami spełniającą warunki

1) $M \rightarrow_{rk\beta} M$,

2) jeżeli $M \rightarrow_{rk\beta} M_1$, $M_1 \rightarrow_{\alpha} M_1'$, $N \rightarrow_{rk\beta} N_1$ i N_1 jest podstawialny za zmienną x w termie M_1', to $(\underline{\lambda}xM)N \rightarrow_{rk\beta} M_1'[x := N_1]$,

3) jeżeli $M \rightarrow_{rk\beta} M_1$ oraz $N \rightarrow_{rk\beta} N_1$, to $MN \rightarrow_{rk\beta} M_1N_1$,

4) jeżeli $M \to_{rk\beta} M_1$, to $\lambda x M \to_{rk\beta} \lambda x M_1$ oraz $\underline{\lambda} x M \to_{rk\beta} \underline{\lambda} x M_1$.

Obie relacje redukcji kolorowych redeksów w jednym kroku rozszerzemy w standardowy sposób do redukcji $\twoheadrightarrow_{k\beta}$ oraz $\twoheadrightarrow_{rk\beta}$. Tak samo, jak analogiczny fakt w poprzednim rozdziale, dowodzimy

Twierdzenie 10.4 *Relacje $\twoheadrightarrow_{k\beta}$ oraz $\twoheadrightarrow_{rk\beta}$ są identyczne.* □

10.3 Własność $\diamond$ redukcji równoległej

Relacje redukcji równoległych $\to_{r\beta}$ i $\to_{rk\beta}$ mają własność $\diamond$. W obu przypadkach dowodzimy to w bardzo podobny sposób.

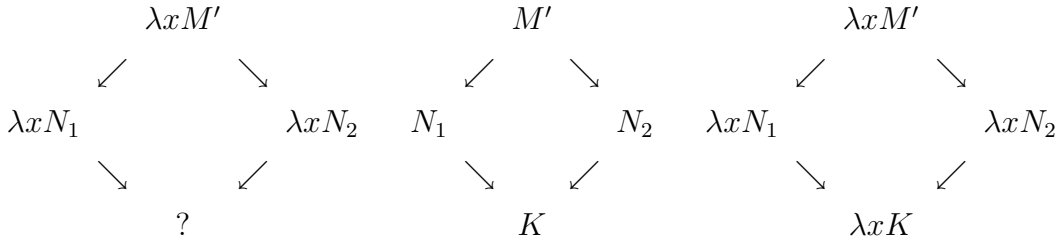
Lemat 10.5 *Relacja redukcji równoległej $\to_{r\beta}$ ma własność $\diamond$, a dokładniej.*

Dowód. Niech M oznacza term, który przekształcamy na dwa sposoby: $M \to_{r\beta} N_1$ i $M \to_{r\beta} N_2$. Lemat dowodzimy przez indukcję ze względu na budowę termu M. Dla każdego rodzaju termów będziemy rozważać wiele przypadków odpowiadających różnym dopuszczalnym sposobom przekształcania.

Przypadek 1: $N_2 = M$. Zawsze możemy korzystać ze zwrotności $\to_{r\beta}$, czyli przekształcać nic nie robiąc. Wtedy N_1 przekształcamy w $K = N_1$ oraz $N_2 = M$ też przekształcamy w $N_1 = K$. Dalej zakładamy, że wykonujemy istotne przekształcenia M.

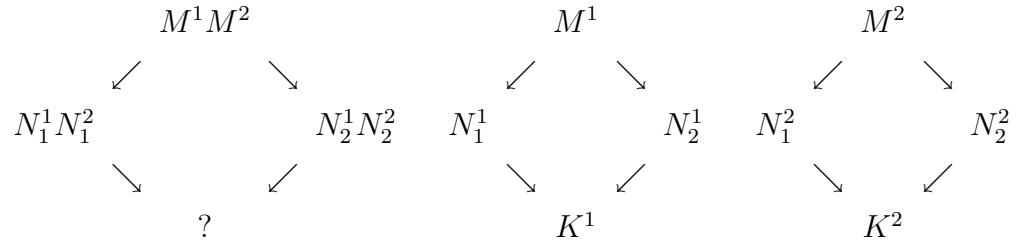
Przypadek 2: M jest zmienną. Zmiennej nie możemy redukować w istotny sposób.

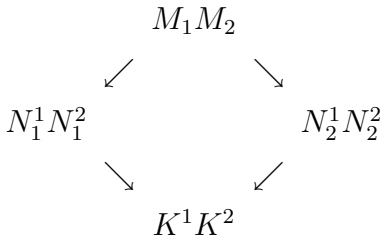
Przypadek 3: $M = \lambda x M'$. Jedyny sposób redukowania abstrakcji $\lambda x M'$ to przekształcanie termu M'. Po przekształceniu na dwa sposoby dostajemy termy $\lambda x N_1$ i $\lambda x N_2$ takie, jak na pierwszej części poniższego rysunku.



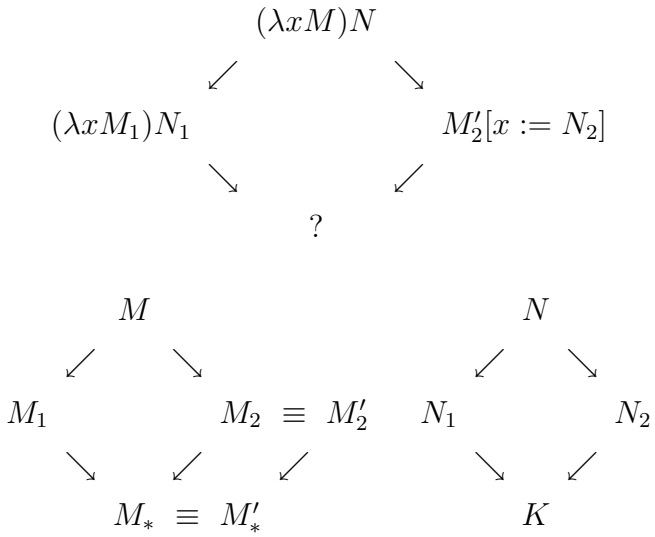
Termu N_1 i N_2 są takie, jak w środkowej części rysunku. Dla nich znajdujemy term K korzystając z założenia indukcyjnego. Term ten ma własności pokazane na ostatniej części rysunku.

Przypadek 4: M jest aplikacją, która nie jest redeksem, lub w żadnym przypadku nie jest redukowana jak redeks. Dowód jest analogiczny do dowodu z poprzedniego przypadku.





Przypadek 5: M jest redeksem redukowanym w mieszany sposób, jako redeks i jako zwykła aplikacja.



11 Podwyrażenia (podtermy)

11.1 Definicja

Niech M będzie wyrażeniem lambda rachunku. Wyrażenie N jest podwyrażeniem M, jeżeli

1) $N = M$, jest to tzw. niewłaściwe podwyrażenie M w przeciwieństwie do pozostałych, czyli właściwych, podwyrażeń,

2) N jest (właściwym lub nie) podwyrażeniem M_1 lub podwyrażeniem M_2 w przypadku, gdy $M = M_1M_2$,

3) N jest (właściwym lub nie) podwyrażeniem M_1 w przypadku, gdy $M = \lambda xM_1$.

Zauważmy, że zgodnie z przytoczoną definicją zmiennej x znajdującej się w abstrakcji λxM bezpośrednio za symbolem λ nie uważamy za podwyrażenie.

11.2 Porządek podwyrażeń

W zbiorze podwyrażeń termu M definiujemy porządek. Mówimy, że podterm N_1 leży na lewo od podtermu N_2 (lub podterm N_2 leży na prawo od podtermu N_1), jeżeli zachodzi jedna z następujących możliwości:

1) $N_1 = M$ i N_2 jest właściwym podtermem M,

2) $M = M_1 M_2$, N_1 jest podwyrażeniem M_1 i N_2 jest podwyrażeniem M_2,

3) $M = M_1 M_2$, N_1 i N_2 są podwyrażeniami M_i (dla $i = 1$ lub $i = 2$) oraz jako podwyrażenia M_i podwyrażenie N_1 leży na lewo od N_2,

4) $M = \lambda x M_1$, N_1 i N_2 są podwyrażeniami M_1 oraz jako podwyrażenia M_1 podwyrażenie N_1 leży na lewo od N_2.

11.3 Prefiks i sufiks podwyrażenia

Jeżeli termy rachunku lambda definiujemy jako słowa i term N jest podtermem termu M, to słowo N jest podsłowem słowa M, a więc wyznacza pewien prefiks $\mathit{pref}_M(N)$ słowa M i pewien sufiks $\mathit{suff}_M(N)$ tego słowa, takie że M jest konkatenacją $\mathit{pref}_M(N)\ N\ \mathit{suff}_M(N)$.

Za pomocą prefiksów podwyrażeń możemy wyrazić porządek podwyrażeń.

Lemat 11.1 *W termie M podwyrażenie N_1 leży na lewo od podwyrażenia N_2 wtedy i tylko wtedy, gdy słowo $\mathit{pref}_M(N_1)$ jest właściwym prefiksem słowa $\mathit{pref}_M(N_2)$ lub słowo $\mathit{pref}_M(N_1)N_1$ jest właściwym prefiksem słowa $\mathit{pref}_M(N_2)N_2$.* □

Jeżeli nie będzie to prowadzić do nieporozumień, zamiast $\mathit{pref}_M(N)$ będziemy pisać $\mathit{pref}(N)$.

11.4 Dokładniej o podtermach

Aby dokładniej mówić o podtermach wprowadzimy specjalną notację. Będziemy rozważać wyrażenia rachunku lambda ze specjalnymi zmiennymi. Wszystkie te zmienne będziemy przedstawiać tym samym symbolem [ ]. Każda może wystąpić w termie dokładnie jeden raz, a więc różne wystąpienia symbolu [ ] w termie oznaczają różne zmienne tego rodzaju. Symbol $M[\,]$ oznacza term, w którym występuje jedna (ewentualnie przynajmniej jedna) taka zmienna. Zapis $M[N]$ oznacza wynik zastępowania zmiennej [ ] przez N, czyli term $M[[\,] := N]$.

Fakt 11.2 Term N jest podtermem wyrażenia M wtedy i tylko wtedy, gdy $M = M'[N]$ dla pewnego wyrażenia M' z jednym wystąpieniem zmiennej [ ]. □

Umawiamy się, że różne wystąpienia zmiennych [ ] w wyrażeniu podajemy w kolejności od najbardziej lewego do prawego. Tak więc $M[\,][N]$ oznacza wyrażenie z podtermem N. W tym wyrażeniu na lewo od podtermu N znajduje się wystąpienie zmiennej [ ].

Lemat 11.3 *Jeżeli w wyrażeniu M podterm N_1 znajduje się na lewo od podtermu N_2, to albo $M = M'[N_1][N_2]$ dla pewnego M', albo też $M = M'[N_1'[N_2]]$ dla pewnych M' i N_1'. W pierwszym przypadku $\mathit{pref}_M(N_1)\ N_1$ jest prefiksem $\mathit{pref}_M(N_2)$, w drugim – $\mathit{pref}_M(N_2)\ N_2$ jest prefiksem $\mathit{pref}_M(N_1)\ N_1$.*

Lemat 11.4 *Przypuśćmy, że K jest podtermem wyrażenia $M[x := N] = Q[K]$. Wtedy istnieje $M'[\,]$ takie, że albo $M = M'[K']$, $Q[\,] = M'[x := N][\,]$ i $K = K'[x := N]$ dla pewnego $K' \neq x$, albo $M = M'[x]$, $Q[\,] = M'[x := N][N'[\,]]$ i $N = N'[K]$ dla pewnego $N'[\,]$. Co więcej, takie, jak wyżej, termy M', K' i N' są wyznaczone jednoznacznie.* □

11.5 Przodek podtermu

Teraz interesują nas podtermy termu otrzymanego w wyniku β-redukcji.

Przypuśćmy, że wyrażenie S otrzymaliśmy w wyniku pojedyńczej β redukcji z wyrażenia R, a więc $R \to_\beta S$. Wtedy

Fakt 11.5 Istnieją termy R', M i N oraz zmienna x takie, że

$$R = R'[(\lambda x M)N] \to_\beta R'[M[x := N]] = S. \ \square$$

Niech K będzie podtermem wyrażenia $S = R'[M[x := N]] = Q[K]$. Z lematu 11.4 wynika, że są możliwe następujące przypadki:

1) K jest na lewo od termu podstawianego w $R'[\,]$, a więc istnieje R'' takie, że $R'[\,] = R''[K][\,]$ oraz $S = R'[M[x := N]] = R''[K][M[x := N]]$, i podobnie, gdy K jest na prawo od termu podstawianego w $R'[\,]$,

2) K istotnie obejmuje redukt $M[x := N]$, a więc $S = Q[K'[M[x := N]]]$, $R'[\,] = Q[K'[\,]]$ oraz $K = K'[M[x := N]]$ dla $K' \neq [\,]$,

3) K jest częścią termu podstawianego w $R'[\,]$ i powstaje z części M, a więc istnieją M' i K' takie, że $K = K'[x := N]$ i $M = M'[K']$, oraz

$$S = R'[M[x := N]] = R'[(M'[K'])[x := N]] = R'[M'[x := N][K'[x := N]] = \\ = R'[M'[x := N][K]],$$

4) K jest częścią termu podstawianego w $R'[\,]$ zawartą w pewnej kopii N, a więc istnieją M' i N' takie, że $M = M'[x := N][N]$ i $N = N'[K]$, oraz

$$S = R'[M[x := N]] = R'[M'[x := N][N]] = R'[M'[x := N][N'[K]]].$$

W każdym z powyższych przypadków dla podtermu K definiujemy podterm $p_{R,S}(K)$ termu R. Tak więc

1) jeżeli $S = R''[K][M[x := N]]$ oraz $R'[\,] = R''[K][\,]$, to $p_{R,S}(K) = K$ i jest podtermem takim, że $R = R''[p_{R,S}(K)][(\lambda x M)N]$, przyjmujemy też analogiczną definicję, gdy K jest po drugiej stronie reduktu,

2) jeżeli $S = Q[K'[M[x := N]]]$, $R'[\,] = Q[K'[\,]]$ i $K = K'[M[x := N]]$ dla $K' \neq [\,]$, to $p_{R,S}(K) = K'[(\lambda x M)N]$ i jest podtermem takim, że $R = Q[p_{R,S}(K)]$,

3) jeżeli $S = R'[M'[x := N][K]]$ dla termów M' i K' takich, że $K = K'[x := N]$, $M = M'[K']$ oraz $K' \neq x$, to $p_{R,S}(K) = K'$ i jest podtermem takim, że $R = R'[(\lambda x M'[p_{R,S}(K)])N]$,

4) jeżeli $S = R'[M'[x := N][N'[K]]]$, to $p_{R,S}(K) = K$ oraz $R = R'[(\lambda x M)N'[p_{R,S}(K)]]$.

Zwróćmy uwagę, że w przypadku $R = x(y((\lambda z M)N))$ i $S = x(y(M[z := N]))$ mamy $p_{R,S}(M[z := N]) = M$ oraz $p_{R,S}(y(M[z := N])) = y((\lambda z M)N)$.

Lemat 11.6 *Przypuśćmy, że $R \to_\beta S$ i K jest podtermem S takim, że $p_{R,S}(K)$ w termie R leży na lewo od redukowanego redeksu. Wtedy $pref_S(K) = pref_R(p_{R,S}(K))$.*

Dowód. Dowód jest łatwy, ale może być trudno się przez niego przegryść. □

Lemat 11.7 *Przypuśćmy, że $R \to_\beta S$ i L jest podtermem R, leżącym na lewo od redeksu redukowanego podczas przekształcania R w S. Wtedy $L = p_{R,S}(K)$ dla pewnego podtermu K.*

Lemat 11.8 *Przypuśćmy, że $R \to_\beta S$ i K jest podtermem S, który jest abstrakcją. Wtedy $p_{R,S}(K)$ też jest abstrakcją.*

Lemat 11.9 *Przypuśćmy, że $R \to_\beta S$ i K jest podtermem S taki, że $p_{R,S}(K)$ jest redeksem (lub abstrakcją). Wtedy K też jest redeksem (odpowiednio: abstrakcją).*

Dowód. Wyrażenie $p_{R,S}(K)$ jest definiowane na jeden z czterech sposobów. W pierwszym i ostatnim przypadku z definicji $p_{R,S}$ lemat jest oczywisty. W trzecim przypadku także, ponieważ podstawiając cokolwiek w redeksie otrzymujemy redeks. Nieco bardziej skomplikowaną sytuację mamy w przypadku drugim.

W tym przypadku $p_{R,S}(K) = K'[(\lambda x M)N]$ dla pewnego $K' \neq [\,]$. Jeżeli K' jest jakąś inną zmienną, to $p_{R,S}(K)$ też jest zmienną, a nie redeksem. Podobnie pokazujemy, że K' nie może być abstrakcją. Musi więc być aplikacją. Pierwszy człon tej aplikacji powinien być abstrakcją. Nie może być aplikacją, ani zwykłą zmienną. Nie może być też zmienną $[\,]$, gdyż w tym przypadku, po podstawieniu otrzymujemy aplikację. A jeżeli K' jest redeksem, to $K = K'[M[x := N]]$ też jest redeksem. □

Lemat 11.10 *Przypuśćmy, że $R \to_\beta S$ i K jest podtermem S, który jest redeksem. Wtedy K jest zielonym redeksem wtedy i tylko wtedy, gdy $p_{R,S}(K)$ też jest zielonym redeksem (i analogicznie dla pozostałych kolorów).* □

12 Twierdzenie o normalizacji

12.1 Rodzaje redukcji

Przypuśćmy, że mamy ciąg wyrażeń M_0, M_1, M_2, $M_3 \ldots$ lambda rachunku (skończony lub nie), taki że

$$M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta M_3 \ldots \tag{1}$$

Taki ciąg nazywamy redukcją normalną, jeżeli dla wszystkich i z wyjątkiem ostatniego, przekształcając M_i w M_{i+1}, redukujemy pierwszy redeks w termie M_i (a więc leżący najbardziej na lewo).

Ciąg (1) nazywamy redukcją standardową, jeżeli dla wszystkich $i > 0$ z wyjątkiem ostatniego, redeks R_{i-1} leży w termie M_{i-1} na lewo od termu $p_{M_{i-1},M_i}(R_i)$. Tutaj R_j oznacza redeks redukowany podczas przeksztacania M_j w M_{j+1}.

Ciąg (1) jest redukcją quasi-normalną, jeżeli dowolnie daleko w tym ciągu są redukowane pierwsze redeksy.

Lemat 12.1 *Jeżeli*

$$M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta M_3 \ldots \to_\beta M_m$$

jest redukcją standardową, a wyrażenie M_m jest w postaci normalnej, to ta redukcja jest normalna.

Dowód. Przez indukcję ze względu na m, że jeżeli pierwszy redeks L termu M_0 nie został zredukowany, to po wykonaniu m kroków redukcji standardowej otrzymujemy term M_m, który nie jest w postaci normalnej.

Załóżmy, że redeks L nie został zredukowany podczas redukcji termu M_0 do M_1. Zgodnie z lematem 11.7 istnieje podterm K termu M_1 taki, że $p_{M_0,M_1}(K) = L$. Z lematu 11.9 otrzymujemy, że K jest redeksem. Gdyby K został zredukowany podczas przekształcania M_1 w M_2, to na mocy lematu 11.6 redukcja nie byłaby standardowa. Jeżeli nie został zredukowany, to z założenia indukcyjnego otrzymujemy, że term M_m nie ma postaci normalnej. □

12.2 Redukcja normalna

Twierdzenie 12.2 (o normalizacji) *Jeżeli wyrażenie M_0 ma postać normalną, to redukcja normalna*

$$M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta M_3 \dots$$

jest skończona, a jeżeli jest też dostatecznie długa, to jej ostatni wyraz jest postacią normalną M_0.

Dowód. Po pierwsze, redukcja normalna jest jednoznacznie wyznaczona. Po drugie, na mocy twierdzenia o standaryzacji 12.7 i lematu 12.1, stosując redukcję normalną term M_0 można sprowadzić do postaci normalnej. Jednoznaczność redukcji normalnej gwarantuje, że dana redukcja jest częścią redukcji normalnej prowadzącej do postaci normalnej. □

Wniosek 12.3 *Jeżeli istnieje nieskończona redukcja normalna, zaczynająca się termem M_0, to term ten nie ma postaci normalnej.* □

Lemat 12.4 *Przyjmijmy, że mamy cztery wyrażenia M_0, M_1, N_0, N_1, które redukują się, jak na rysunku*

$$\begin{array}{ccc} M_0 & \xrightarrow{S}_\beta & M_1 \\ {\scriptstyle R_0}\downarrow_\beta & & {\scriptstyle R_1}\downarrow_\beta \\ N_0 & & N_1 \end{array}$$

na którym obok symboli redukcji są podane przekształcane redeksy. Jeżeli $R_0 = p_{M_0,M_1}(R_1)$ i R_0 leży w wyrażeniu M_0 na lewo od S (a więc redukcja $M_0 \to_\beta M_1 \to_\beta N_1$ nie jest standardowa), to $N_0 \twoheadrightarrow_\beta N_1$ (a więc można ją zastąpić bardziej standardową redukcją $M_0 \to_\beta N_0 \twoheadrightarrow_\beta N_1$). Ponadto w tej sytuacji, jeżeli R_1 jest pierwszym redeksem w termie M_1, to R_0 jest pierwszym redeksem w termie M_0.

Dowód. Możemy pokolorować oba przekształcane redeksy (R_0 i S) w termie M_0. Dla redukcji kolorowych redeksów, redukcja $M_0 \to_\beta M_1 \to_\beta N_1$ przekształca M_0 w postać normalną N_1 (jest to redukcja od końca). Term M_0 możemy też przekształcić do N_0, a następnie N_0 do postaci normalnej, redukując kolorowe redeksy (S lub jego kopie). Ponieważ jest to zawsze wykonalne, a dla redukcji kolorowych redeksów postać normalna jest jednoznacznie wyznaczona, term N_0 też można przekształcić do N_1. Dalej korzystamy z lematów 11.7, 11.9 i 11.6. □

Wniosek 12.5 *Przyjmijmy, że $M_0 \twoheadrightarrow_\beta N_1$ oraz $M_0 \to_\beta N_0$ i w każdej z tych redukcji (chociaż raz) redukujemy pierwsze redeksy. Wtedy $M_0 \twoheadrightarrow_\beta N_1$.* □

Wniosek 12.6 *Jeżeli istnieje nieskończona redukcja quasi-normalna, zaczynająca się termem M_0, to term ten nie ma postaci normalnej.* □

12.3 Standaryzacja

Twierdzenie 12.7 (o standaryzacji) *Jeżeli $M \twoheadrightarrow_\beta N$, to istnieje redukcja standardowa M do N.*

Dowód. Twierdzenie to dowodzimy przez indukcję korzystając z lematu 12.10. □

Najpierw pokażemy lematy pomocnicze.

Lemat 12.8 *Przypuśćmy, że w każdym kroku redukcji*

$$C_0 \to_\beta C_1 \to_\beta C_2 \to_\beta \ldots \to_\beta C_n$$

jest redukowany najbardziej lewy pokolorowany redeks. Niech M_0 będzie podtermem C_0, który jest abstrakcją i leży na lewo od wszystkich kolorowych redeksów. Wtedy istnieje podterm N wyrażenia C_n, który jest abstrakcją, taki że

$$p^*_{C_0,C_n}(N) = M_0 \quad oraz \quad pref_{C_0}(M_0) = pref_{C_n}(N).$$

Dowód. Z lematu 11.7 wynika, że istnieje podterm M_1 wyrażenia C_1 taki, że $p_{C_0,C_1}(M_1) = M_0$. Z lematu 11.6 otrzymujemy, że $pref_{C_0}(M_0) = pref_{C_1}(M_1)$ a z lematu 11.9 – że M_1 jest abstrakcją. Jeżeli pokażemy, że M_1 leży w termie C_1 na lewo od wszystkich pokolorowanych redeksów, to tezę otrzymamy z zasady indukcji.

Przypuśćmy, że redukcja $C_0 \to_\beta C_1$ polegała na zastąpieniu redeksu X reduktem Y. Wtedy $pref_{C_0}(X) = pref_{C_1}(Y)$. Ponieważ X jest pierwszym kolorowym redeksem, więc prefiks $pref_{C_0}(X)$ nie zawiera kolorowych opreratorów abstrakcji. Zawiera za to prefiks $pref_{C_0}(M_0)$. Wobec tego także prefiks $pref_{C_1}(M_1)$ nie zawiera żadnych kolorowych operatorów i leży na lewo od wszystkich kolorowych redeksów. □

Lemat 12.9 *Przypuśćmy, że w każdym kroku redukcji*

$$C_0 \to_\beta C_1 \to_\beta C_2 \to_\beta \ldots \to_\beta C_n$$

jest redukowany najbardziej lewy pokolorowany redeks. Niech M będzie podtermem C_n, który jest abstrakcją i $p^*_{C_0,C_n}(M)$ leży w termie C_0 na lewo od wszystkich kolorowych redeksów. Wtedy

$$pref_{C_0}(p^*_{C_0,C_n}(M)) = pref_{C_n}(M).$$

Dowód. Podobny do dowodu poprzedniego lematu. □

Lemat 12.10 *Przypuśćmy, że $A_0 \to_\beta B_0 \twoheadrightarrow_\beta B$ i redukcja B_0 do B jest standardowa. Wtedy A_0 też można standardowo zredukować do B.*

Dowód. Będziemy tworzyć i stopniowo rozbudowywać następujący diagram

$$
\begin{array}{ccccccccccc}
A_0 & & & & \xrightarrow{S}_\beta & & & & & & B_0 \\
{\scriptstyle R_0}\downarrow_\beta & & & & & & & & & & {\scriptstyle Q_0}\downarrow_\beta \\
A_1 & \twoheadrightarrow_\beta & & & \dots & & & & & \twoheadrightarrow_\beta & B_1 \\
{\scriptstyle R_1}\downarrow_\beta & & & & & & & & & & {\scriptstyle Q_1}\downarrow_\beta \\
\vdots & & & & & & & & & & \vdots \\
A_{i-1} & \to_\beta & C'_1 & \to_\beta & C'_2 & \to_\beta & \dots & \to_\beta & C'_n & \to_\beta & B_{i-1} \\
{\scriptstyle R_{i-1}}\downarrow_\beta & & & & & & & & & & {\scriptstyle Q_{i-1}}\downarrow_\beta \\
A_i & \xrightarrow{Z_0}_\beta & C_1 & \xrightarrow{Z_1}_\beta & C_2 & \xrightarrow{Z_2}_\beta & \dots & \xrightarrow{Z_{m-1}}_\beta & C_m & \xrightarrow{Z_m}_\beta & B_i \\
 & & & & {\scriptstyle R_i}\downarrow_\beta & & & & & & {\scriptstyle Q_i}\downarrow_\beta \\
 & & & & A_{i+1} & \twoheadrightarrow_\beta & & & & \twoheadrightarrow_\beta & B_{i+1} \\
 & & & & & & & & & & {\scriptstyle Q_{i+1}}\downarrow_\beta
\end{array}
$$

Na tym diagramie obok symboli redukcji są podane nazwy przekształcanych redeksów. Jest przedstawiony fragment danej redukcji standardowej przekształcającej $B_0 \to_\beta B_1 \to_\beta$ aż do B i redukcja A_0 do B_0.

Będziemy korzystać z kolorowych redeksów. Na początku koloruje np. na zielono redeks S. Przekształcenie A_i w B_i (to uwidocznione na rysunku) będzie polegać tylko na redukcji zielonych redeksów, wszystkich możliwych. Konstrukcję będziemy tak prowadzić, aby redukcja $A_0 \to_\beta A_1 \to_\beta \dots \to_\beta A_i$ i dalej $A_i = C_0 \to_\beta C_1 \twoheadrightarrow_\beta B_i$ była standardowa, a jej ostatni fragment od A_i do B_i był redukcją normalną dla kolorowych (zielonych) redeksów prowadzącą do termu bez kolorowych redeksów. Pierwszy krok konstrukcji zostanie wykonany zgodnie z lematem 12.4. Załóżmy, że fragment konstrukcji zawierający definicję redukcji od $A_i = C_0$ do B_i jest już wykonany.

Przypadek 1: ostatnim wyrazem redukcji $A_0 \twoheadrightarrow_\beta A_i \twoheadrightarrow_\beta B_i$ jest ostatni wyraz redukcji $B_0 \twoheadrightarrow_\beta B$, czyli $B_i = B$. W tym przypadku twierdzenie jest oczywiste i wynika z założonego niezmiennika konstrukcji.

Przypadek 2: $B_i \neq B$, czyli nie dotarliśmy jeszcze do końca redukcji. Powinniśmy teraz wskazać redeks R_i. Najpierw musimy zbadać przodka $p_{C_m,B_i}(Q_i)$ w termie C_m.

Przypadek 2.1: $p_{C_m,B_i}(Q_i)$ leży na prawo od Z_m. Oznacza to, że redukcja od A_0 do A_i, dalej od A_i do B_i i jeszcze od B_i do B jest standardowa i twierdzenie zostało dowiedzione.

Przypadek 2.2: $p_{C_m,B_i}(Q_i)$ leży na lewo od Z_m. Znajdujemy najmniejszą liczbę k taką, że dla wszystkich l speniających $k \leqslant l \leqslant m$ term $p^*_{C_l,B_i}(Q_i)$ leży na lewo do Z_l ($p^*_{C_l,B_i}$ odpowiednie złożenie funkcji p, na przykład $p^*_{C_{m-1},B_i}(Q_i) = p_{C_{m-1},C_m}(p_{C_m,B_i}(Q_i))$).

Przyjmijmy, że $R_i = p^*_{C_k,B_i}(Q_i)$. W termie C_k redeks Z_k jest zielonym redeksem położonym spośród zielonych najbardziej na lewo i sam ma po lewej stronie redeks R_i. Tak więc redeks R_i ma po prawej stronie wszystkie zielone redeksy w termie C_k. Można więc zredukować R_i, a następnie jakikolwiek zielony redeks zachowując standardowość redukcji.

Na chwilę pomalujmy na czerwono redeks R_i. Powoduje to, że wszystkie termy z redukcji od C_k do B_i zawierają po jednym czerwonym redeksie. W termie B_i czerwonym redeksem jest Q_i. Redukując go otrzymujemy B_{i+1}, term bez zielonych oraz bez czerwonych redeksów. Term B_{i+1} jest więc postacią normalną termu C_k dla redukcji kolorowych redeksów. Do postaci normalnej dochodzimy zawsze, redukując kolorowe redeksy w dowolnej kolejności. Możemy więc najpierw w termie C_k zredukować redeks czerwony (otrzymujemy A_{i+1}), a następnie redukować redeksy zielone zawsze biorąc pierwszy redeks od lewej. Redukując w ten sposób też otrzymamy postać normalną. W szczególności, istnieje normalna dla zielonych redeksów redukcja termu A_{i+1} do B_{i+1} i standardowa od C_k do B_{i+1}.

Znowu są możliwe dwa przypadki:

Przypadek 2.2.1: $k > 0$.

W tym przypadku redukcja prowadząca od A_0 do A_i, dalej od A_i (oznaczanego też C_0) do C_k i na koniec do A_{i+1} otrzymanego z termu C_k przez redukcję redeksu R_i jest standardowa. Jej standardowość wynika z definicji k.

Dodając do niej standardową redukcję prowadzącą od C_k do B_{i+1} otrzymujemy standardową redukcję od A_0 do B_{i+1}.

Przypadek 2.2.2: $k = 0$. Teraz mamy dwie redukcje od A_0 do A_i oraz od A_i do A_{i+1} i dalej do B_{i+1}, obie standardowe, ale nie jest jasne, czy łącząc je otrzymamy redukcję standardową. Sytuacja została przedstawiona na poniższym rysunku:

$$\begin{array}{ccccccccccc}
C'_0 & \to_\beta & C'_1 & \to_\beta & C'_2 & \to_\beta & \dots & \to_\beta & C'_n & \to_\beta & B_{i-1} \\
{\scriptstyle R_{i-1}}\downarrow_\beta & & & & & & & & & & {\scriptstyle Q_{i-1}}\downarrow_\beta \\
A_i & \xrightarrow{Z_0}_\beta & C_1 & \xrightarrow{Z_1}_\beta & C_2 & \xrightarrow{Z_2}_\beta & \dots & \xrightarrow{Z_{m-1}}_\beta & C_m & \xrightarrow{Z_m}_\beta & B_i \\
{\scriptstyle R_i}\downarrow_\beta & & & & & & & & & & {\scriptstyle Q_i}\downarrow_\beta \\
A_{i+1} & \twoheadrightarrow_\beta & & & & \dots & & & & \twoheadrightarrow_\beta & B_{i+1}
\end{array}$$

Redeks R_{i-1} (zgodnie z oznaczeniami z rysunku) jest równy $p^*_{C'_0,B_{i-1}}(Q_{i-1})$, a redeks $R_i = p^*_{A_i,B_i}(Q_i)$. Konstrukcja tych redeksów gwarantuje również, że leżą one na lewo od zielonych redeksów w odpowiednich termach.

Aby wspomniane redukcje po połączeniu dały redukcję standardową, redeks R_{i-1} powinien być po lewej stronie redeksu $p_{C'_0,A_i}(R_i)$. Są to różne redeksy. Dla dowodu nie wprost załóżmy, że R_{i-1} jest po prawej stronie $p_{C'_0,A_i}(R_i)$, a więc słowo $pref_{C'_0}(p_{C'_0,A_i}(R_i))$ samo jest prefiksem $pref_{C'_0}(R_{i-1})$.

Na mocy lematu 12.9 zachodzi równość

$$pref_{A_i}(R_i) = pref_{A_i}(p^*_{A_i,B_i}(Q_i)) = pref_{B_i}(Q_i).$$

Z tego samego powodu zachodzi też równość

$$pref_{C'_0}(R_{i-1}) = pref_{C'_0}(p^*_{C'_0,B_{i-1}}(Q_{i-1})) = pref_{B_{i-1}}(Q_{i-1}).$$

Z z założenia dowodu nie wprost i lematu 11.6 otrzymujemy, że

$$pref_{C'_0}(p_{C'_0,A_i}(R_i)) = pref_{A_i}(R_i).$$

Teraz do redukcji od C'_0 do B_{i-1} i $p_{C'_0,A_i}(R_i)$ zastosujmy lemat 12.8. Wynika z niego, że dla pewnego podtermu N wyrażenia B_{i-1} mamy

$$pref_{C'_0}(p_{C'_0,A_i}(R_i)) = pref_{B_{i-1}}(N).$$

Z dowiedzionych równości prefiksów otrzymujemy, że

$$pref_{B_i}(Q_i) = pref_{B_{i-1}}(N)$$

i dodatkowo, że słowo $pref_{B_{i-1}}(N)$ jest prefiksem $pref_{B_{i-1}}(Q_{i-1})$. Przyjrzyjmy się więc redukcji $B_{i-1} \to_\beta B_i$.

Z lematów 11.7 i 11.6 znajdujemy podterm K wyrażenia B_i taki, że $N = p_{B_{i-1},B_i}(K)$ oraz

$$pref_{B_i}(K) = pref_{B_{i-1}}(p_{B_{i-1},B_i}(K)) = pref_{B_{i-1}}(N) = pref_{B_i}(Q_i).$$

Z kolei z lematów 11.8 i 11.9 wynika, że podterm K jest abstrakcją. Dla abstrakcji operacja prefiksu jest różnowartościowa. Wobec tego $K = Q_i$ i $p_{B_{i-1},B_i}(Q_i)$, czyli N, leży na lewo od Q_{i-1}. To jednak przeczy założeniu, że redukcja $B_0, B_1, B_2, \ldots$ jest standardowa. □

13 Lambda definiowalność

13.1 Minimum historii

Alonzo Church chyba w pierwszej chwili uważał rachunek lambda za system formalny, który mógłby być podstawą sformalizowanej – a więc być może pozbawionej sprzeczności – matematyki. Wcześniej próby zbudowania podobnego, formalnego systemu prowadził Emil Post, który rozważał bardzo skomplikowane gramatyki. Jego prace jednak nie odziaływały istotnie na ówczesne badania naukowe.

Church dość szybko (ok. 1932 roku) zorientował się, że rachunek lambda pozwala na sformalizowanie pojęcia obliczalności i może doprowadzić do rozwiązania problemu znanego jako Entscheidungsproblem, postawionego przez Dawida Hilberta, polegającego na skonstruowaniu procedury pozwalającej na rozstrzyganie hipotez matematycznych. W 1936 rozwiązał ten problem negatywnie proponując jednocześnie utożsamianie pojęcia obliczalności z lambda definiowalnością. Stwierdzenie, że te pojęcia są tożsame, można nazwać tezą Churcha, a właściwie jest jedną z wersji tej tezy.

Church o słuszności swojej tezy przekonywał przytaczając twierdzenia o równoważność różnych definicji obliczalności, opartych o różne intuicje. W pierwszym rzędzie powoływał się równoważność lambda definowalności i definicji związanych z pracami Kurta Gödla, w tym przytoczonej dalej definicji klasy funkcji rekurencyjnych udoskonalonej przez Stephena Kleene'ego.

Podobne rozważania prowadził Alan Turing, a wyniki swoich badań ujawnił w 1936 roku, prawie równocześnie z Churchem. W przeciwieństwie do Churcha, Turing uzasadniał swoją definicję obliczalności argumentami filozoficznym opartymi o analizę pracy mózgu i obliczeń dokonywanych przez ludzi. Przekonywał, że maszyny Turinga działają podobnie, jak ludzie wykonujący obliczenia, i z tego powodu są w stanie zrealizować wszelkie obliczenia możliwe do wykonania przez ludzi.

13.2 Numerały Churcha

Aby mówić w lambda rachunku o obliczalności należy w pierwszym rzędzie w tym rachunku reprezentować liczby naturalne. Pierwszy sposób reprezentowania zaproponował Church. Jego termy reprezentujące liczby naturalne nazywamy numerałami Churcha.

Przyjmijmy, że M i N są termami rachunku lambda. Wtedy $M^k(N)$ oznacza term zdefiniowany rekurencyjnie wzorami

$$M^0(N) = N \text{ oraz } M^{k+1}(N) = M(M^k(N)).$$

Liczbę naturalną k będziemy w lambda rachunku reprezentować jako tzw. numerał Churcha c_n dany wzorem

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

Numerały Churcha są oczywiście termami w postaci normalnej.

Z twierdzenia Churcha-Rossera wynika, że różne numerały Churcha są termami, o których w lambda rachunku nie można dowieść, że są równe.

Znane są też inne sposoby reprezentowania liczb naturalnych.

Przyjrzyjmy się jeszcze numerałom Churcha. Ich intuicyjna interpretacja jest następująca:

1) $c_n fx = f^n(x)$ to n-krotne aplikowanie funkcji f do x,

2) $c_n f = \lambda x.f^n(x)$ to funkcja, która jest n-krotnym złożeniem funkcji f,

3) $c_n = \lambda fx.f^n(x)$ to abstrakcyjna operacja n-krotnego składania.

Mamy wzór $c_n f(fx) = c_{n+1}fx$. Stąd term $S = \lambda afx.af(fx)$ spełnia

$$Sc_n = \lambda fx.c_n f(fx) = \lambda fx.f^{n+1}(x) = c_{n+1}.$$

Tak więc term S definiuje operację następnika (przyporządkowującą liczbie naturalnej n liczbę $n+1$). Mamy także

$$(\lambda afx.f(afx))c_n = \lambda fx.f(c_n fx) = \lambda fx.f(f^n(x)) = c_{n+1}.$$

Podobnie, z wzoru $c_n f(c_m fx) = f^{n+m}(x)$ wynika, że termy

$$\lambda abfx.af(bfx) \text{ oraz } \lambda abfx.bf(afx)$$

definiują dodawanie liczb naturalnych.

Natomiast z wzoru $c_m(c_n f)x = f^{mn}(x)$ otrzymujemy, że termy

$$\lambda abfx.a(bf)x \text{ oraz } \lambda abfx.b(af)x$$

definiują mnożenie liczb naturalnych. Powyższy wzór, także przytoczona wyżej interpretacja numerałow Churcha sugerują także wzór: $c_m(c_n f) = f^{mn}$. Zauważmy, że także

$$(\lambda abf.a(bf))c_m c_n = c_{mn}.$$

13.3 Lambda definiowalność wg Churcha

Będziemy rozważać częściowe funkcje wielu zmiennych naturalnych przyjmujące wartości naturalne. Funkcje częściowe n zmiennych to takie, które niekoniecznie są określone dla wszystkich możliwych układów n liczb naturalnych. Funkcje określone dla wszystkich możliwych układów argumentów nazywamy całkowitymi. Tak więc funkcje całkowite są szczególnym przypadkiem funkcji częściowych.

Funkcja częściowa dwóch zmiennych $f : N^2 \to N$ jest lambda definiowalna, jeżeli dla pewnego termu $F \in \Lambda$ są spełnione dla dowolnych $m, n \in N$ następujące warunki:

1) jeżeli wartość $f(m,n)$ jest określona, to w lambda rachunku daje się dowieść, że $Fc_mc_n = c_{f(m,n)}$ (lub równoważnie $Fc_mc_n \twoheadrightarrow_\beta c_{f(m,n)}$),

2) jeżeli wartość $f(m,n)$ nie jest określona, to wyrażenie Fc_mc_n nie ma postaci normalnej.

O termie F spełniającym powyższe warunki mówimy, że reprezentuje funkcję f. Bez trudu tę definicję uogólniamy na przypadek funkcji częściowych innej liczby zmiennych.

Przytoczona definicja została podana przez Churcha i jest dość naturalna. Mimo to sprawia trochę kłopotów. Świadczy o tym następujący przykład. Funkcja stale równa zero f może być reprezentowana przez $F = Kc_0$, a funkcja nigdzie nie określona g – przez term $G = K\Omega$. Złożenie f i g jest funkcją nigdzie nie określoną, a jego naturalną reprezentacją powinien być term $\lambda x.F(Gx)$. Ten term reprezentuje jednak funkcją stale równą zero.

13.4 Funkcje pierwotnie rekurencyjne

Klasa funkcji pierwotnie rekurencyjnych jest najmniejszą klasą naturalnych funkcji całkowitych

1) zawierającą funkcje Z, S i $U_{n,k}$ spełniające dla wszystkich możliwych naturalnych argumentów równości

$$Z(m) = 0, \qquad S(m) = m + 1 \quad \text{oraz} \quad U_{n,k}(m_1, m_2, \ldots, m_n) = m_k$$

2) i zamkniętą ze względu na złożenie

$$h(m_1, m_2, \ldots, m_n) = f(g_1(m_1, m_2, \ldots, m_n), \ldots, g_k(m_1, m_2, \ldots, m_n))$$

(czyli h jest pierwotnie rekurencyjne dla wszystkich pierwotnie rekurencyjnych funkcji f i $g_1, \ldots, g_k$)

3) oraz definiowanie przez rekursję prostą

$$f(0, m_1, m_2, \ldots, m_n) = g(m_1, m_2, \ldots, m_n) \quad \text{oraz}$$

$$f(k+1, m_1, m_2, \ldots, m_n) = h(f(k, m_1, m_2, \ldots, m_n), k, m_1, m_2, \ldots, m_n)$$

(czyli f jest pierwotnie rekurencyjne dla wszystkich pierwotnie rekurencyjnych funkcji g i h).

Definiowanie przez rekursję prostą obejmuje także przypadek $n = 0$. W tym przypadku definicja przyjmuje postać

$$f(0) = g \quad \text{oraz} \quad f(k+1) = h(f(k), k)$$

dla dowolnie ustalonej liczby g.

Zdecydowana większość używanych funkcji naturalnych to funkcje pierwotnie rekurencyjne. Wielu matematyków nie poda przykładu funkcji naturalnej, która nie jest pierwotnie rekurencyjna.

13.5 λ-definiowalność funkcji pierwotnie rekurencyjnych

Twierdzenie 13.1 *Klasa całkowitych funkcji lambda definiowalnych jest zamknięta ze względu na złożenie.*

Dowód. Ten dowód ma jasną ideę i może zostać przedstawiony na przykładzie. Przypuśćmy, że funkcja h jest złożeniem całkowitych funkcji f, g_1 i g_2 takim, że

$$h(m,n) = f(g_1(m,n), g_2(m,n)).$$

Niech F, G_1 i G_2 będą termami definiującymi odpowiednio funkcje f, g_1 i g_2. Wtedy term

$$\lambda ab.F(G_1ab)(G_2ab)$$

definiuje funkcję h. Ponieważ zajmujem się funkcjami całkowitymi, więc wystarczy sprawdzić tylko warunek 1) z definicji funkcji lambda definiowalnej (patrz str. 28). Sprawdzenie tego warunku nie nastręcza żadnych trudności. □

Twierdzenie 13.2 *Klasa całkowitych funkcji lambda definiowalnych jest zamknięta ze względu na definiowanie przez rekursję prostą.*

Dowód. Najpierw zajmiemy się przykładem, z którego powinno wynikać, że rekursja ma coś wspólnego z iteracją, zwłaszcza w przypadku, gdy kolejną wartość definiujemy bez parametrów.

Pokażemy lambda definiowalność funkcji takiej, że $f(0) = 1$ i $f(n+1) = 2 \cdot f(n)$. Oczywiście, $f(n) = 2^n$). Nietrudno zauważyć, że $f(n) = h^n(1)$, gdzie $h(n) = 2 \cdot n$. Jeżeli znajdziemy term H definiujący h, to korzystając z interpretacji numerałów Churcha możemy uznać za definiujący f term $\lambda a.aHc_1$. Term H możemy analogicznie zdefiniować jako $\lambda a.aLc_0$, gdzie L jest termem definiujący funkcję powiększającą argument o 2. Możemy przyjąć, że $L = \lambda afx.af(f(fx))$.

Teraz musimy rozbudować aparat. Przyjmijmy, że

$$[M,N] = \lambda x.xMN, \quad K = \mathbf{true} = \lambda xy.x \quad \text{oraz} \quad K* = \mathbf{false} = \lambda xy.y.$$

Oczywiście, mamy $[M,N]K = M$ oraz $[M,N]\mathbf{false} = N$, o ile zmienna x nie występuje w termach M i N.

Pokażemy (znowu na przykładzie) lambda definiowalność funkcji f definiowanej wzorami

$$f(0,m) = g(m) \ \text{ oraz } \ f(k+1,m) = h(f(k,m),k,m).$$

Przyjmijmy, że G i H są termami definiującymi odpowiednio g i h.

Nawiasy kwadratowe oznaczają rodzaj funkcji pary. Niech q oznacza funkcję taką, że

$$q(m,[a,b]) = [h(a,b,m), b+1].$$

W rachunku lambda taką funkcję reprezentuje term

$$Q = \lambda uvw.w(H(vK)(vK^*)u)(S(vK^*).$$

Weźmy

$$F = \lambda ab.a(Qb)[Gb, c_0]K,$$

Pokażemy najpierw, że

$$Qc_m[c_f, c_k] = [c_{h(f,k,m)}, c_{k+1}].$$

Oczywiście,

$$Qc_m[c_f, c_k] = \lambda w.w(H([c_f, c_k]K)([c_f, c_k]K^*)c_m)(S([c_f, c_k]K^*)) =$$

$$= \lambda w.w(Hc_fc_kc_m)(Sc_k) = [Hc_fc_kc_m, c_{k+1}] = [c_{h(f,k,m)}, c_{k+1}].$$

Stąd przez indukcję otrzymujemy, że

$$(Qc_m)^k([c_{g(m)}, c_0]) = [c_{f(k,m)}, c_k]$$

Zauważmy, że $k = 0$ równość ta jest oczywista. Ponadto

$$(Qc_m)^{k+1}([c_{g(m)}, c_0]) = Qc_m[c_{f(k,m)}, c_k] = [c_{h(f(k,m),k,m)}, c_{k+1}] = [c_{f(k+1,m)}, c_{k+1}].$$

Stąd, jako wniosek, otrzymujemy

$$Fc_kc_m = c_k(Qc_m)[Gc_m, c_0]K = (Qc_m)^k([c_{g(m)}, c_0])K = [c_{f(k,m)}, c_k]K = c_{f(k,m)}.$$

Równość ta świadczy o tym, że F definiuje funkcję f. □

Wniosek 13.3 *Funkcja poprzednik $p : N \to N$ taka, że $p(0) = 0$ oraz $p(n+1) = n$ jest definiowalna w rachunku lambda.*

Dowód. Jest to wniosek ważny z historycznego punktu widzenia. Znalezienie dowodu tego faktu sprawiało trochę kłopotów.

Funkcję p można zdefiniować przez rekursję prostą („pustą", nie odwołującą się do wartości poprzedniej) przyjmując, że

$$p(0) = 0 \ \text{ oraz } \ p(n+1) = U_{2,2}(p(n), n).$$

Wystarczy więc powtórzyć wyżej przedstawioną konstrukcję w tym przypadku. □

Twierdzenie 13.2 można dowieść także jeszcze inną, ważną metodą, korzystając z twierdzenia o punkcie stałym.

Dowód. Dowodzimy raz jeszcze twierdzenie 13.2 korzystając z twierdzenia o punkcie stałym i oznaczeń w wprowadzonych w dowodzie poprzednim.

Przyjmijmy, że

$$M = \lambda fax.\mathbf{Zero}\ a(Gx)(H(f(Pa)x)(Pa)x).$$

W tym wzorze P oznacza term reprezentujący funkcję poprzednik, a **Zero** jest równe $\lambda x.x(K\mathbf{false})\mathbf{true}$ i zaaplikowane do c_n przyjmuje jako wartość albo $K = \mathbf{true}$, albo $K^* = \mathbf{false}$ zależnie od tego, czy n jest równe 0, czy nie.

Korzystając z twierdzenia o punkcie stałym znajdujemy term F taki, że $MF = F$. O tym termie dowodzi się, że

$$Fc_0c_m = c_{g(m)} \ \text{ oraz } \ Fc_{k+1}c_m = H(Fc_kc_m)c_kc_m.$$

Stąd już łatwo, przez indukcję, wyprowadzić tezę twierdzenia. □

Wniosek 13.4 *Funkcje pierwotnie rekurencyjne są lambda definiowalne.* □

13.6 Operacja minimum

Operacją minimum nazywamy przyporządkowanie funkcji naturalnej $g : N^{n+1} \to N$ funkcji $f : N^n \to N$ oznaczanej wzorem

$$f(k_1, \ldots, k_n) = \mu m(g(k_1, \ldots, k_n, m) = 0)$$

przyjmującą jako wartość $f(k_1, \ldots, k_n)$ najmniejszą liczbę m spełniającą warunek $g(k_1, \ldots, k_n, m) = 0$.

Definicja ta z kilku powodów może być niejasna. Możemy ją stosować dla różnych rodzajów funkcji g. W najbardziej ogólnym przypadku, g może być dowolną naturalną funkcją częściową. Wtedy pojawiają się niejasności związane z definicją wartości funkcji f. Jest jednak naturalny algorytm pozwalający na obliczanie wartości f. Algorytm ten przeszukuje kolejno liczby naturalne dopóty, dopóki nie znajdzie wartości m spełniającej warunek $g(k_1, \ldots, k_n, m) = 0$. Operacja minimum powinna być tak rozumiana, aby ten algorytm pozwalał na wyliczanie wartości funkcji definiowanych za pomocą tej operacji. Nietrudno zauważyć, że ten algorytm zawodzi w dwóch sytuacjach: jeżeli nie jest w stanie obliczyć potrzebnej wartości funkcji g, (na przykład zawsze oblicza $g(k_1, \ldots, k_n, 0)$), lub gdy funkcja g dla odpowiednich parametrów nie przyjmuje wartości 0. W tych dwóch przypadkach funkcja definiowana za pomocą operacji minimum nie jest określona.

O operacji minimum mówimy, że jest efektywna, jeżeli stosujemy ją wyłącznie do całkowitych funkcji g takich, że

$$\forall k_1, \ldots, k_n \exists m \;\; g(k_1, \ldots, k_n, m) = 0.$$

Będziemy też rozważać operację minimum ograniczoną do całkowitych funkcji g.

13.7 Funkcje rekurencyjne

Klasa całkowitych funkcji rekurencyjnych jest najmniejszą klasą funkcji zawierającą Z, S oraz $U_{n,k}$ i zamkniętą ze na złożenie, rekursję prostą i efektywną operację minimum.

Klasa (częściowych) funkcji rekurencyjnych jest najmniejszą klasą funkcji zawierającą Z, S oraz $U_{n,k}$ i zamkniętą ze na złożenie, rekursję prostą i operację minimum.

Definicja klasy funkcji rekurencyjnych wymaga wyjaśnienia, co to jest złożenie funkcji częściowych i jak definiujemy przez rekursję prostą w przypadku takich funkcji. Te pojęcia można wyjaśnić podobnie, jak operację minimum, wskazując naturalne algorytmy, które powinny obliczać odpowiednie funkcje i żądając zgodności definicji i obliczeń za pomocą odpowiedniego algorytmu.

Zgodnie z tezą Churcha, klasa częściowych funkcji rekurencyjnych jest klasą funkcji naturalnych, które są obliczalne w jakimkolwiek, intuicyjnym sensie.

Klasę funkcji rekurencyjnych można definiować na wiele sposobów. Na przykład, można w definicji tej klasy nie wspominać o rekursji prostej. Rekursja prosta jest potrzebna do zdefiniowania trzech funkcji: dodawania, mnożenia i funkcji charakterystycznej relacji nierówności. Mając te trzy funkcje, możemy kodować ciągi liczb naturalnych za pomocą liczb naturalnych, a to z kolei pozwala rekursję prostą zastąpić operacją minimum.

Można też ograniczać rolę operacji minimum. Można stosować ją wyłącznie do funkcji całkowitych. Bardzo silny rezultat tego typu wyraża twierdzenie o postaci normalnej.

Twierdzenie 13.5 (o postaci normalnej Kleene'ego) *Jeżeli f jest częściową funkcją rekurencyjną, to istnieją dwie pierwotnie rekurencyjne (a więc całkowite) funkcje g i h takie, że*

$$f(\vec{k}) = h(\mu m(g(\vec{k}, m) = 0)).$$

Dowód. O h można nawet założyć, że jest określoną, bardzo prostą funkcją. Szkic dowodu twierdzenia jest następujący: zmieniamy trochę i rozbudowujemy algorytm obliczający wartości funkcji f, dodajemy licznik kroków algorytmu i dodatkowo o 1 zwiększamy ewentualny wynik obliczeń. Niech g oznacza funkcję

$$f'(\vec{k}, t) = \begin{cases} 0 & \text{jeżeli obliczenie wartości } f(\vec{k}) \text{ wymaga} > t \text{ kroków,} \\ f(\vec{k}) + 1 & \text{jeżeli obliczenie wartości } f(\vec{k}) \text{ wymaga} \leqslant t \text{ kroków.} \end{cases}$$

Można przekonać się, że funkcja f' jest pierwotnie rekurencyjna.

W teorii rekursji przyjmuje się, że 0 koduje prawdę. Funkcję g definiujemy jako funkcję boolowską, taką że

$$g(\vec{k}, w) = 0 \equiv f'(\vec{k}, (w)_1) = (w)_0 \wedge (w)_0 > 0,$$

gdzie w jest liczbą, o ktrórej myślimy, że koduje ciąg (ewentualnie parę) liczb o pierwszej współrzędnej $(w)_0$ i drugiej $(w)_1$. Niech $h(x) = (x)_0 - 1$ będzie zdefiniowana jako zmniejszona o 1 pierwsza współrzędna pary kodowanej przez x. Dla tak zdefiniowanej funkcji h zachodzi wzór z tezy twierdzenia. □

Wniosek 13.6 *Klasa częściowych funkcji rekurencyjnych jest najmniejszą klasą funkcji zawierającą funkcje pierwotnie rekurencyjne i zamkniętą ze względu na złożenie i operację minimum stosowaną do funkcji całkowitych.* □

13.8 Operacja minimum i λ-definiowalność

Twierdzenie 13.7 *Niech $g : N^2 \to N$ będzie całkowitą funkcją definiowaną w lambda rachunku za pomocą termu G. Wtedy funkcja $f : N \to N$ taka, że $f(m) = \mu n(g(m, n) = 0)$ jest lambda definiowalna.*

Dowód. Zdefiniujmy pomocniczą funkcję $h : N^2 \to N$ przyjmując, że

$$h(k, m) = \begin{cases} k & \text{jeżeli } g(m, k) = 0 \\ h(k + 1, m) & \text{w przeciwnym przypadku} \end{cases}$$

Funkcja h jest definiowana przez specyficzną indukcję. Powinna być definiowana za pomocą termu H takiego, że

$$H = \lambda xy.\mathbf{Zero}(Gyx)x(H(Sx)y),$$

gdzie **Zero** jest termem zdefiniowanym na stronie 30.

Zauważmy, że term H o wyżej podanej własności spełnia też warunki

1) jeżeli $g(m, k) = 0$, to $Hc_kc_m \twoheadrightarrow_\beta c_k$,

2) jeżeli $g(m, k) \neq 0$, to $Hc_kc_m \twoheadrightarrow_\beta Hc_{k+1}c_m$, Przy czym wykonując podaną redukcję przynajmniej raz redukujemy pierwszy o lewej strony redeks.

Pokażemy, że term $F = Hc_0$ definiuje funkcję f. Nietrudno zauważyć, że jeżeli wartość $f(m)$ jest określona, to dla pewnego k mamy

$$g(m,l) \neq 0 \ \text{ dla } \ l < k \ \text{ oraz } \ g(m,k) = 0.$$

Wobec tego mamy następującą redukcję

$$Fc_m = Hc_0c_m \twoheadrightarrow_\beta Hc_1c_m \twoheadrightarrow_\beta \ldots Hc_{k-1}c_m \twoheadrightarrow_\beta Hc_kc_m \twoheadrightarrow_\beta c_k = c_{f(m)}.$$

Natomiast jeżeli $f(m)$ nie jest określona i $g(m,k) \neq 0$ dla wszystkich k, to mamy nieskończoną redukcję

$$Fc_m = Hc_0c_m \twoheadrightarrow_\beta Hc_1c_m \twoheadrightarrow_\beta Hc_2c_m \twoheadrightarrow_\beta \ldots$$

która jest quasi-normalna. Z wniosku 12.6 otrzymujemy, że term Fc_m nie ma postaci normalnej.

Pozostaje podać konstrukcję termu H. Wymaga to posłużenia się operatorem punktu stałego. Musi to być szczególny operator, gwarantujący wymagane własności H. Takim może być tzw. operator Turinga. Niech więc

1) $A = \lambda xy.y(xxy)$,

2) $\Theta = AA$,

3) $M = \lambda hxy.\mathbf{Zero}(Gyx)x(h(Sx)y)$.

Wtedy

$$H = \Theta M = AAM \rightarrow_\beta M(AAM) = MH \rightarrow_\beta \lambda xy.\mathbf{Zero}(Gyx)x(H(Sx)y)$$

i zachodzą wymienione wcześniej własności H. □

13.9 Nowsze rozumienie definiowalności

Funkcje definiowalne według Churcha mają może nawet bardzo intuicyjną definicję, ale jest ona zbyt restrykcyjna i trudno się nią posugiwać. Często inaczej formułuje się warunek nieokreśloności funkcji.

Term M nazywamy rozwiązalnym (ang. solvable), jeżeli istnieją termy $N_1, \ldots, N_k$ takie, że

$$(\lambda x_1 \ldots x_n.M)N_1 \ldots N_k = I,$$

gdzie zmienne $x_1, \ldots, x_n$ są wszystkimi zmiennymi wolnymi w termie M.

Nietrudno zauważyć, że numerały Churcha są rozwiązalne. Mamy bowiem $c_nII = I$. Także term $xy(\omega\omega)$ (gdzie $\omega = \lambda z.zz$) jest rozwiązałny, ponieważ $(\lambda xy.xy(\omega\omega))KI = I$, jednak nie ma on postaci normalnej.

Mówimy, że term jest w głowowej postaci normalnej, jeżeli jest on postaci

$$\lambda x_1 \ldots x_n.xN_1 \ldots N_m,$$

gdzie m i n są dowolnymi liczbami naturalnymi, także mogą być równe 0, x oraz $x_1, \ldots, x_n$ są dowolnymi zmiennymi, a $N_1 \ldots N_m$ są dowolnymi termami.

Term $\omega\omega$ nie jest w głowowej postaci normalnej i nie ma głowowej postaci normalnej.

Lemat 13.8 *Zachodzą następujące fakty:*

1) Term w postaci normalnej jest w głowowej postaci normalnej.

2) Jeżeli term ma postać normalną, to ma też głowową postać normalną.

3) Jeżeli term ma głowową postać normalną, to jest rozwiązalny. □

Podany lemat jest łatwy do wykazania, ale prawdziwa jest też trudniejsza do udowodnienia implikacja odwrotna do ostatniej z wymienionych, a więc mamy

Twierdzenie 13.9 *Term ma głowową postać normalną wtedy i tylko wtedy, gdy jest rozwiązalny.* □

Możemy teraz zmodyfikować definicję definiowalności (znaną ze strony 28). Funkcja częściowa dwóch zmiennych $f : N^2 \to N$ jest lambda definiowalna, jeżeli dla pewnego termu $F \in \Lambda$ są spełnione dla dowolnych $m, n \in N$ następujące warunki:

1) jeżeli wartość $f(m, n)$ jest określona, to w lambda rachunku daje się dowieść, że $Fc_mc_n = c_{f(m,n)}$ (lub równoważnie $Fc_mc_n \twoheadrightarrow_\beta c_{f(m,n)}$),

2) jeżeli wartość $f(m, n)$ nie jest określona, to wyrażenie Fc_mc_n nie ma głowowej postaci normalnej a więc nie jest rozwiązalne.

Twierdzenie 13.10 *Złożenie częściowych funkcji lambda definiowalnych jest lambda definiowalne (w sensie powyższej definicji).*

Dowód. Dowód znowu przedstawimy na przykładzie. Przypuśćmy, że funkcja h jest złożeniem całkowitych funkcji f, g_1 i g_2 takim, że

$$h(m, n) = f(g_1(m, n), g_2(m, n)).$$

Niech F, G_1 i G_2 będą termami definiującymi odpowiednio funkcje f, g_1 i g_2. Wtedy term

$$H = \lambda ab.G_1abII\ G_2abII\ F(G_1ab)(G_2ab)$$

definiuje złożenie h. Sprawdzenie warunku 1) z definicji funkcji lambda definiowalnej nie nastręcza żadnych trudności. Wystarczy zauważyć, że jeżeli wartość $g_i(m, n)$ jest określona, to $G_ic_mc_nII \twoheadrightarrow_\beta c_{g_i(m,n)}II \twoheadrightarrow_\beta I$.

Aby sprawdzić, że spełniony jest warunek 2), rozważamy kilka przypadków. Jeżeli wartość $g_1(m, n)$ nie jest określona, a mimo to term Hc_mc_n jest rozwiązalny, to

$$Hc_mc_nN_1 \dots N_k \twoheadrightarrow_\beta I$$

dla pewnych termów $N_1, \dots, N_k$. Ale także

$$Hc_mc_nN_1 \dots N_k \twoheadrightarrow_\beta G_1c_mc_nII\ G_2ac_mc_nII\ F(G_1c_mc_n)(G_2c_mc_n)N_1 \dots N_k,$$

więc z twierdzenia Churcha-Rossera mamy

$$G_1c_mc_nII\ G_2ac_mc_nII\ F(G_1c_mc_n)(G_2c_mc_n)N_1 \dots N_k \twoheadrightarrow_\beta I.$$

To przeczy jedna stwierdzeniom, że wartość $g_1(m, n)$ nie jest określona i term $G_1c_mc_n$ nie jest rozwiązalny.

Analogiczne rozumowanie jest słuszne w pozostałych przypadkach: gdy wartość $g_1(m, n)$ jest, a $g_2(m, n)$ nie jest określona, oraz gdy wartości $g_1(m, n)$ i $g_2(m, n)$ są określone, a wartość $f(g_1(m, n), g_2(m, n))$ nie jest określona. □

Dowód twierdzenia 13.7 można uzupełnić tak, aby pozostał prawdziwy po zmianie pojęcia definiowalności.

Twierdzenie 13.11 *Niech $g : N^2 \to N$ będzie całkowitą funkcją definiowaną w lambda rachunku za pomocą termu G. Wtedy funkcja $f : N \to N$ taka, że $f(m) = \mu n(g(m, n) = 0)$ jest lambda definiowalna.*

Dowód. Zostanie tylko uzupełniony przy zachowaniu oznaczeń.

Zmianie ulega jedynie fragment, w którym sprawdzamy własności F w przypadku, gdy $f(m)$ nie jest określona. Załóżmy więc, że tak jest i dodatkowo, że term $Fc_m = Hc_0c_m$ jest jednak rozwiązalny. Wtedy znajdujemy termy $N_1, \ldots, N_k$ takie, że

$$Hc_0c_mN_1 \ldots N_k \twoheadrightarrow_\beta I.$$

Jednak w dalszym ciągu powyższy term można redukować w następujący sposób:

$$Hc_0c_mN_1 \ldots N_k \twoheadrightarrow_\beta Hc_1c_mN_1 \ldots N_k \twoheadrightarrow_\beta Hc_2c_mN_1 \ldots N_k \twoheadrightarrow_\beta \ldots$$

i jest to redukcja quasi-normalna. Z wniosku 12.6 wynika, że term $Hc_0c_mN_1 \ldots N_k$ nie ma postaci normalnej i nie może zostać zredukowany do termu I, który jest w postaci normalnej. □