Formalizacja podstawowych pojęć λ-rachunku

Antoni Kościelski

Spis treści

1 Termy rachunku lambda

1.1 Zmienne

O zmiennych myślimy jak o znakach służących do zapisywania wyrażeń λ-rachunku. 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 je od innych rzeczy. Zmiennymi nie są trzy inne symbole występujące w wyrażeniach λ-rachunku, czyli litera lambda λ oraz nawiasy okrągłe (i), a także kropka . występująca w skróconych termach oraz symbol mnożenia $\cdot$ wykorzystywany czasami dla oznaczenia operacji aplikacji.

Dla oznaczenia zmiennych będziemy zwykle używać małych liter, w razie potrzeby z indeksami.

1.2 Wyrażenia (termy) λ-rachunku

Wyrażenia λ-rachunku często nazywa się termami λ-rachunku lub krótko, λ-termami, a nawet termami[1]. Zbiór wyrażeń rachunku lambda będziemy oznaczać symbolem Λ. Zbiór ten można definiować na wiele sposobów. Trudno zdecydować się na konkretną definicję, każda ma wady i zalety.

1.3 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 (λ-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 i odwrotnie. Co więcej, mając aplikację w sposób jednoznaczny powinniśmy ustalić,

[1] Słowo *term* jest używane przez logików na oznaczenie fragmentu formuły logicznej, zwłaszcza sformalizowanej, oznaczającego jakąś rzecz, coś, czego własności zostały wyrażone w formule. W językach programowania analogiczne fragmenty nazywa się często wyrażeniami. W tym tekście słowa *term* i *wyrażenie* będą używane zamiennie.

co aplikujemy i do czego, czyli żądamy, aby warunek $M_1N_1 = M_2N_2$ implikował, 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 $M_1 = M_2$.

1.4 Wyrażenia λ-rachunku jako drzewa

Wyrażenia lambda rachunku można uważać za drzewa binarne z węzłami etykietowanymi zmiennymi, 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.

1.5 Wyrażenia λ-rachunku jako słowa

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

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

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 $(\lambda x M) \in \Lambda$.

Zaletą tej definicji jest to, że tak rozumiane wyrażenia rachunku lambda dają się łatwo reprezentować, także pisemnie, wadą – duża liczba wymaganych nawiasów. Zwykle więc korzysta się z tej definicji w połączeniu z zasadami upraszczania napisów i 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
$$(\lambda x(M)) = (\lambda x M),$$

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 Kilka pojęć dotyczących termów

2.1 Wystąpienia znaków i podsłów

Przypuśćmy, że mamy dane pewne słowo w. Wystąpieniem znaku x w tym słowie nazywamy jeden konkretny znak x w tym słowie. Formalizując to pojęcie możemy powiedzieć, że jest to znak x wraz z informacją identyfikującą ten znak spośród być może wielu znaków x występujących w tym słowie. Taką informacją może być na przykład numer znaku (lub jego indeks), prefiks danego słowa poprzedzający wystąpienie znaku lub przynajmniej długość tego prefiksu, liczba znaków x występujących wcześniej i wiele innych.

Podobnie definiujemy wystąpienie podslowa. Jest to podsłowo wraz z informacją pozwalającą jednoznacznie umiejscowić to podsłowo w danym słowie, na przykład z długością prefiksu poprzedzającego wystąpienie.

2.2 Podtermy

Dla danego termu M symbolem $Sub(M)$ będziemy oznaczać zbiór właściwych podtermów termu M. Wobec tego elementy tego zbioru będziemy nazywać właściwymi podtermami termu M. Oprócz podtermów właściwych każdy term ma też podterm niewłaściwy. Niewłaściwym podtermem termu M jest term M.

Zbiory $Sub(M)$ definiujemy indukcyjnie następującymi wzorami

1) jeżeli term M jest zmienną, to $Sub(M) = \emptyset$ (zmienne nie mają właściwych podtermów),

2) jeżeli term M jest aplikacją AB, to $Sub(M) = Sub(A) \cup Sub(B) \cup \{A, B\}$ (właściwymi podtermami AB są termy A i B oraz ich właściwe podtermy),

3) jeżeli M jest abstrakcją λxA, to $Sub(M) = Sub(A) \cup \{A\}$ (właściwymi podtermami abstrakcji $\lambda x\ A$ są termy A i jego właściwe podtermy).

Mówiąc podterm najczęściej będziemy mieć na myśli konkretne wystąpienie podtermu. Zakładam, że odpowiednie uzupełnienie przytoczonej definicji nie sprawi Czytelnikom żadnego kłopotu.

2.3 Konteksty

Konteksty są właściwie termami, w których wybrana zmienna występuje w dokładnie jednym miejscu i nie bezpośrednio za operatorem λ. Zwykle jest to specjalna zmienna, nie należąca do V. W tym opracowaniu będzie oznaczana symbolem [ ].

Konteksty definiujemy indukcyjnie w następujący sposób:

1) [ ] jest kontekstem,

2) jeżeli C jest kontekstem, a M jest λ-termem, to CM i MC są kontekstami,

3) jeżeli C jest kontekstem, a x jest zmienną, to $\lambda x\ C$ też jest kontekstem.

Oczywiście, jest to definicja w stylu abstrakcyjnej definicji wyrażeń. Konteksty pozwalają analizować wystąpienia podtermów. Czasem jednak będziemy analizować jednocześnie wystąpienia kilku podtermów pewnego termu. Wtedy być może uogólnimy definicję kontekstu.

2.4 Podstawianie w kontekstach

W kontekstach możemy podstawiać albo umieszczać termy. Dla danego kontekstu C i termu N symbolem $C[N]$ będziemy oznaczać term N umieszczony w kontekscie C (lub wynik podstawienia termu N za zmienną [ ] w kontekscie C). Będzie to term określony w następujący sposób:

1) $[N] = N$ ($[\,][N] = N$?, umieszczając N w kontekscie [ ] dostajemy N),

2) dla dowolnego kontekstu C i dowolnego termu M mamy $(CM)[N] = C[N]M$ oraz $(MC)[N] = MC[N]$,

3) dla dowolnego kontekstu C i dowolnej zmiennej x mamy $(\lambda x\ C)[N] = \lambda x\, C[N]$.

Łatwo sprawdzić, że po umieszczeniu termu N w kontekscie C otrzymujemy term.

2.5 Konteksty a podtermy

Lemat 2.1 *Jeżeli C jest kontekstem, a M – termem, to M jest podtermem termu $C[M]$.* □

Lemat 2.2 *Jeżeli N jest podtermem termu M, to $M = C[N]$ dla pewnego kontekstu C. Co więcej, dla określonego wystąpienia termu N kontekst C jest jednoznacznie wyznaczony.* □

2.6 Operatory λ, wystąpienia i zasięg

Dla każdego (wystąpienia) operatora λ w termie M definiujemy jego zasięg.

Jeżeli term M jest aplikacją M_1M_2, to operatory λ w termie M występują albo w M_1, albo w M_2 i zasięg wystąpienia operatora λ w M jest taki, jak jego zasięg w M_1 lub odpowiednio w M_2 w zależności od tego, w którym z podtermów operator występuje.

Zasięgiem operatora λ rozpoczynającego abstrakcję $M = \lambda x\, A$ (w M) jest podterm A (czasem do zasięgu zaliczymy także zmienną x zapisaną tuż przed A). Zasięg termie M pozostałych operatorów λ jest taki, jak ich zasięg w termie A.

Nietrudno zauważyć, że każde wystąpienie w termie M operatora λ wyznacza lub rozpoczyna pewien podterm $\lambda x\, A$ termu M złożony z operatora, pewnej zmiennej x i zasięgu A.

2.7 Wystąpienia zmiennych, wolne i związane

W rachunku λ zwrot „wystąpienie zmiennej" ma kilka nieco różniących się znaczeń. Wynika to stąd, że same zmienne pełnią w rachunku λ dwie, a może nawet trzy role.

Zmienna x jest jednym z symboli, którymi mamy prawo posługiwać się zapisując λ-termy. Może więc mówić o wystąpieniu x tak, jak o wystąpieniach każdego innego symbolu, abstrahując od wszelkich szczegółów.

Zmienne mogą „dookreślać" operator λ lub wskazywać określone miejsce w termie. W abstrakcji $\lambda x\, A$ symbol x (drugi znak tej abstrakcji) jest zmienną tylko formalnie, nie pełni on roli zmiennej, na przykład nic nie będziemy podstawiać za to x. Ten symbol nazywa się zmienną wiązaną operatorem λ (tym znajdującym się bezpośrednio przed x). Ze względów językowych będziemy go również nazywać zmienną określającą poprzedzający go operator. Jego rolą jest tylko przekazywanie informacji, że poprzedzający go operator λ dotyczy wystąpień zmiennych x czyli wiąże te zmienne.

Pozostałe wystąpienia zmiennych, te które nie określają operatorów, dzielą się na dwa rodzaje: wystąpienia wolne i wystąpienia związane. Wystąpienie zmiennej x jest związane, jeżeli znajduje się w zasięgu operatora λ wiążącego zmienną x (pewnego, może być ich kilka, najważniejszy jest operator „najbliższy"). Te wystąpienia zmiennej, które ani nie określają operatorów, ani nie są związane, nazywamy wolnymi.

Aby zorientować się w sytuacji można przeanalizować następujący term:

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

Jest to sztuczny term, będziemy unikać posługiwania się takimi, zamiast niego lepiej posłużyć się termem $x(\lambda a\, a(\lambda b\, b(\lambda c\, y)))$.

2.8 Zbiór zmiennych wolnych

Symbolem $FV(M)$ będziemy oznaczać zbiór zmiennych, które występują jako wolne w termie M. Zbiory $FV(M)$ definiujemy indukcyjnie następującymi wzorami

1) jeżeli term M jest zmienną x, to $FV(M) = \{x\}$,

2) jeżeli term M jest aplikacją AB, to $FV(M) = FV(A) \cup FV(B)$

3) jeżeli M jest abstrakcją λxA, to $FV(M) = FV(A) \setminus \{x\}$.

2.9 Zbiór zmiennych związanych

Symbolem $BV(M)$ będziemy oznaczać zbiór zmiennych, które występują w termie M jako zmienne związane (pewnym operatorem λ). Zbiory $BV(M)$ definiujemy indukcyjnie następującymi wzorami

1) jeżeli term M jest zmienną, to $BV(M) = \emptyset$,

2) jeżeli term M jest aplikacją AB, to $BV(M) = BV(A) \cup BV(B)$

3) jeżeli M jest abstrakcją λxA, to $BV(M) = BV(A) \cup (\{x\} \cap FV(A))$.

3 Podstawianie i podstawialność

3.1 Podstawianie

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

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

3.2 Niezgodność definicji podstawiania z intuicjami

W rachunku lambda, podobnie jak w logice, podstawianie powoduje przeróżne komplikacje i musi być ostrożnie stosowane. Świadczą o tym następujące rozważania.

Weźmy term $F = \lambda xy.\, yx$. Zgodnie z powszechnymi wyobrażeniami wyrażenie to oznacza operację, która polega na aplikowaniu drugiego argumentu do pierwszego. Należałoby się więc spodziewać, że dla dowolnych termów M i N zachodzi równość $FMN = NM$. W szczególności powinna zachodzić równość $Fyx = xy$.

Jeżeli funkcja jest definiowana pewnym wzorem, to zwykle wartość tej funkcji znajdujemy podstawiając do tego wzoru argument. Tak też chcielibyśmy postępować w rachunku lambda. Wobec tego, zgodnie z podaną definicją

$$Fy = (\lambda xy.\, yx)y = (\lambda y\, yx)[x := y] = \lambda y\, yy$$

oraz

$$Fyx = (\lambda y.\, yy)x = (yy)[y := x] = xx.$$

Jeżeli jednak zgodzimy się, że w rachunku lambda zachodzi równość $xy = xx$, to powinniśmy zgodzić się także z równością $\lambda xy.\, xy = \lambda xy.\, xx$. Z tej ostatniej równości wynika równość każdej pary termów. Weźmy dowolny term M oraz $I = \lambda z.\, z$. Dla tych termów mamy

$$M = IM = (\lambda xy.\, xy)IM = (\lambda xy.\, xx)IM = (\lambda y\, II)M = II = I.$$

Każdy term jest więc równy termowi I. Trudno pasjonować się rachunkiem, w którym każde dwa wyrażenia są równe.

3.3 Podstawialność

Zmienna y jest podstawialna 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 y.

Ta definicja wyrażona bardziej formalnie stwierdza, że zmienna y jest podstawialna w termie M za zmienną x, jeżeli

$$\forall A, B \in Sub(M)\ (A = \lambda yB \Rightarrow x \notin FV(B)).$$

Term N jest podstawialny za zmienną x w termie M, jeżeli każda jego zmienna wolna jest podstawialna w termie M za x.

Zmienne wolne w termie podstawianym są uważane za absolutnie wolne, nie mogą stać się związane w wyniku podstawiania. Podstawialność gwarantuje, że pozostaną wolne po podstawieniu.

Lemat 3.1 *Jeżeli pewna zmienna w termie nie jest wiązana żadnym operatorem λ, to jest podstawialna za dowolną zmienną.* □

Lemat 3.2 *Term N jest podstawialny za zmienną z w aplikacji M_1M_2 wtedy i tylko wtedy, gdy jest podstawialny za tę zmienną zawrówno w M_1, jak i w M_2.* □

Lemat 3.3 *Interesuje nas podstawialność za zmienną z w abstrakcji $\lambda x\ M$. W tej sytuacji*

1) *term z wolnym wystąpieniem x jest podstawialny wtedy i tylko wtedy, gdy $z \notin FV(M)$,*

2) *jeżeli $z \neq x$, to term bez wolnych wystąpień x jest podstawialny wtedy i tylko wtedy, gdy jest podstawialny w M,*

3) *jeżeli $z = x$, to wszystkie termy są podstawialne (ale nie zostaną podstawione).* □

3.4 Własności podstawiania

Lemat 3.4 *Zawsze zachodzi wzór $M[x := x] = M$.* □

Lemat 3.5 *Jeżeli zmienna x nie jest wolna w termie M, to $M[x := N] = M$.*

Dowód. Przez indukcję ze względu na budową termu M. □

Lemat 3.6 *Zachodzi zawieranie*

$$FV(M[x := N]) \subseteq (FV(M) \setminus \{x\}) \cup FV(N).$$

Dowód. Przez indukcję ze względu na budową termu M. □

Lemat 3.7 *Jeżeli $x \neq y$ i $x, y \notin FV(L)$, to*

$$M[x := N][y := L] = M[y := L][x := N[y := L]].\ \square$$

Dowód. Przez indukcję ze względu na budową termu M.

Przypadek 1: M jest zmienną $z \in V$.

Przypadek 1.1: $z = y$. Na mocy definicji podstawiania, lewa strona wzoru jest równa L. Prawa także jest równa L na mocy lematu 3.5.

Przypadek 1.2: $z \neq y$. Obie strony wzoru są równe z definicji podstawiania.

Przypadek 2: M jest aplikacją. Równość jest oczywista na mocy założenia indukcyjnego.

Przypadek 3: $M = \lambda z M'$.

Przypadek 3.1: $z = x$. Lewa strona wzoru jest równa $\lambda x(M'[y := L])$, a prawa – $(\lambda x(M'[y := L]))[y := L]$. Na mocy lematu 3.6 zmienna y nie jest wolna w $\lambda x(M'[y := L])$. W tej sytuacji teza wynika z lematu 3.5.

Przypadek 3.2: $z = y$. Na mocy definicji podstawiania obie strony równości są równe $\lambda y(M'[x := N])$.

Przypadek 3.3: $z \neq x$ i $z \neq y$. Dowodzona równość jest oczywistą konsekwencją założenia indukcyjnego. □

Lemat 3.8 *Jeżeli* $x \neq y$, $y \notin FV(N)$ *i* $x \notin FV(L)$, *to*

$$M[x := N][y := L] = M[y := L][x := N]. \ \square$$

Dowód. Podobny do poprzedniego i prostszy. W miejscach, gdzie teza nie wynika w oczywisty sposób z definicji lub założenia indukcyjnego, korzystamy z lematu 3.5. □

Lemat 3.9 *Jeżeli* $x \neq y$, $x \notin FV(L)$ *i* N *jest podstawialny za* x *w termie* M, *to*

$$M[x := N][y := L] = M[y := L][x := N[y := L]]. \ \square$$

Dowód. Przez indukcję ze względu na budową termu M.

Przypadek 1: M jest zmienną $z \in V$.

Przypadek 1.1: $z = y$. Na mocy definicji podstawiania, lewa strona wzoru jest równa L. Prawa także jest równa L na mocy lematu 3.5.

Przypadek 1.2: $z \neq y$. Obie strony wzoru są równe z definicji podstawiania.

Przypadek 2: M jest aplikacją. Równość jest oczywista na mocy założenia indukcyjnego i definicji podstawialności.

Przypadek 3: $M = \lambda z M'$.

Przypadek 3.1: $z = x$. Obie strony są równe, ponieważ nie wykonujemy różniących się podstawień za zmienną x.

Przypadek 3.2: $z = y \notin FV(N)$. Na mocy definicji podstawiania i lematu 3.5 obie strony równości są równe $\lambda y(M'[x := N])$.

Przypadek 3.3: $z = y \in FV(N)$. W tym przypadku term N nie jest podstawialny za x w abstrakcji $\lambda y M'$. Tak więc przypadek ten nie może zajść na mocy ostatniego założenia.

Przypadek 3.4: $z \neq x$ i $z \neq y$. Dowodzona równość jest konsekwencją założenia indukcyjnego i definicji podstawialności. □

4 Rodzaje relacji

4.1 Relacje zgodne

Relacja $\succ$ w zbiorze λ-termów jest zgodna z operacjami λ-rachunku jeżeli dla wszystkich $M, N, Z \in \Lambda$ i $x \in V$

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

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

Lemat 4.1 *Niech* $\succ$ *będzie relacją zgodną z operacjami* λ*-rachunku. Wtedy dla dowolnego kontekstu* C *i dla dowolnych termów* $M, N \in \Lambda$ *takich, że* $M \succ N$ *zachodzi także relacja* $C[M] \succ C[N]$. □

4.2 Kongruencje

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

4.3 Redukcje

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

Redukcją zwykle nazywamy zgodną relację, która jest zwrotna i przechodnia.

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 $M \twoheadrightarrow L$ i $L \rightarrow N$, to $M \twoheadrightarrow N$.

Zgodnie z równoważną definicją redukcję $\twoheadrightarrow$ w wielu krokach definiujemy jako najmniejszą relację spełniającą 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 $M \twoheadrightarrow N$.

Czasem w definicji redukcji może być uzasadniona rezygnacja z warunku zwrotności.

Lemat 4.2 *Jeżeli relacja redukcji w jednym kroku $\rightarrow$ jest zgodna z operacjami rachunku λ, to relacja $\twoheadrightarrow$ jest redukcją zgodną z operacjami rachunku λ.* $\square$

Bywa również, że musimy rozważać bardziej ogólne redukcje, rozumiane jako zgodne z operacjami 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 4.3 *Jeżeli $\cong$ jest kongruencją i relacja redukcji w jednym kroku $\rightarrow$ jest zgodna z operacjami rachunku λ, to relacja $\twoheadrightarrow$ jest redukcją (zgodną z operacjami λ-rachunku).* $\square$

4.4 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 4.4 *Jeżeli relacja $\twoheadrightarrow$ jest redukcją, to relacja $\cong$ jest kongruencją.* $\square$

5 α-konwersja

5.1 α-redukcja

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

Relacja α-redukcji w jednym kroku wyznacza relację $\twoheadrightarrow_\alpha$ α-redukcji tak, jak to zostało wyżej opisane. Relacja $\twoheadrightarrow_\alpha$ jest więc zwrotnym i przechodnim domknięciem relacji $\rightarrow_\alpha$, czyli najmniejszą relacją taką, że

1) $M \twoheadrightarrow_\alpha M$,

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

3) jeżeli $M \twoheadrightarrow_\alpha N$ i $N \rightarrow_\alpha P$, to $M \twoheadrightarrow_\alpha P$.

Relacja α-redukcji, w przeciwieństwie do α-konwersji, nie ma większego znaczenia. W tym tekście odgrywa głównie rolę techniczną. α-konwersją nazywamy najmniejszą symetryczną i przechodnią relację zawierającą relację α-redukcji. Relację α-konwersji będziemy oznaczać symbolem $=_\alpha$.

5.2 Najprostsze własności α-redukcji

Termy będące w relacji α-redukcji są bardzo podobne, mają taką samę strukturę, różnią jedynie zmiennymi występującymi jako zmienne związane. Można sformułować wiele lematów stwierdzających ten fakt, w tym następujące.

Lemat 5.1 *Jeżeli $M \twoheadrightarrow_\alpha N$, to albo jednocześnie termy M i N są aplikacjami, albo są abstrakcjami, albo są identycznymi zmiennymi.*

Dowód. Rozważmy relację

$$\mathcal{R} = \{(M, N) \in \Lambda^2 : (M \text{ i } N \text{ są aplikacjami}) \vee (M \text{ i } N \text{ są abstrakcjami}) \vee M = N\}.$$

Relacja ta ma własności wymagane od relacji α-redukcji w jednym kroku. Ponieważ α-redukcja w jednym kroku jest najmniejszą taką relacją, więc $\rightarrow_\alpha \subseteq \mathcal{R}$.

Tak więc relacja $\mathcal{R}$ ma własność 2) z definicji relacji $\twoheadrightarrow_\alpha$. Łatwo wykazać, że ma też pozostałe własności wymienione w tej definicji. Stąd otrzymujemy tezę lematu, gdyż $\twoheadrightarrow_\alpha$ jest najmniejszą taką relacją. □

Lemat 5.2 *Jeżeli M i N są różnymi termami takimi, że $M \twoheadrightarrow_\alpha N$, to pierwszy znak różniący termy M i N jest zmienną wiązaną pewnym operatorem λ.*

Dowód. Analogiczny do poprzedniego. Tym razem rozważamy relację złożoną z par termów: dwóch termów identycznych, bądź różniących się na pierwszym miejscu zmienną wiązaną. □

5.3 α-redukcja naprawdę jest α-konwersją

Przypomnijmy, że α-konwersją nazywamy najmniejszą relację symetryczną i przechodnią, która zawiera relację α-redukcji. Oznaczamy ją symbolem $=_\alpha$.

Lemat 5.3 *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 $\lambda y.M[x := y] \ \to_\alpha \ \lambda x.M[x := y][y := x] \ = \ \lambda x.M$.

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

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

Wniosek 5.4 *Relacja α-redukcji $\twoheadrightarrow_\alpha$ jest kongruencją i – w konsekwencji – jest relacją α-konwersji $=_\alpha$.* $\square$

5.4 Coś, co przypomina postać normalną

Przypuśćmy, że mamy term M_0, w którym występuje n operatorów lambda, oraz n zmiennych $z_1, z_2, \ldots, z_n$. Term M_0 przekształcamy w term $M_0[z_1, z_2, \ldots, z_n]$ wykonując następujący algorytm:

$M = M_0$; $i = 1$;
`while` $i \leqslant n$ `do`

`(a) w termie` M `znajdź` i`-ty symbol` λ;
`(b) znajdź podterm` $\lambda x\, A$ `termu` M `zaczynający się tym` λ;
`(c) znajdź kontekst` C `taki, że` $M = C[\lambda x\, A]$;
`(d) zastąp` M `przez` $C[\lambda z_i\, A[x := z_i]]$;
`(e)` $i = i + 1$;

`return` M.

Lemat 5.5 *Jeżeli w termie M_0 występuje n operatorów λ, a parami różne zmienne $z_1, z_2, \ldots, z_n$ nie występują w M_0 (ani jako zmienne wolne, ani jako wiązane lub związane), to niezmiennikami pętli while w wyżej podanym programie są następujące stwierdzenia:*

1) zmienne $z_i, z_{i+1}, \ldots, z_n$ nie występują w termie M,

2) k-ty operator λ w termie M wiąże zmienną z_k dla wszystkich $k < i$,

3) $M_0 \twoheadrightarrow_\alpha M$,

4) zmienna w jest podstawialna za y w termie M (o ile jest to zmienna spoza ciągu $z_1, z_2, \ldots, z_n$),

5.5 α-redukcja a podstawianie

Lemat 5.6 *Jeżeli $M \to_\alpha M'$ i term N jest podstawialny za zmienną z w termach M i M', to $M[z := N] \to_\alpha M'[z := N]$.*

Dowód. Dla danej zmiennej z i danego termu N przyjmijmy, że $\Phi(M, M')$ oznacza stwierdzenie

$$(N \text{ jest podstawialny za } z \text{ w } M \text{ i } M') \;\Rightarrow\; M[z := N] \to_\alpha M'[z := N]$$

i rozważamy zbiór

$$\mathcal{R} = \{(M, M') \in \to_\alpha :\ \Phi(M, M')\} = \{(M, M') \in \Lambda^2 :\ M \to_\alpha M' \wedge \Phi(M, M')\}.$$

Pokażemy, że zbiór ten ma własności wymagane od relacji $\to_\alpha$ i jako taki, zawiera tę relację (a właściwie jest jej równy). Oczywiście, to zawieranie pociąga za sobą stwierdzenie

$$\forall M, M' \in \Lambda \;\; M \to_\alpha M' \;\Rightarrow\; \Phi(M, M'),$$

czyli tezę lematu.

Najpierw wykażemy zgodność relacji $\mathcal{R}$ z działaniami. Weźmy więc dwa termy M i M' takie, że $M \to_\alpha M'$ i $\Phi(M, M')$.

Zgodność z aplikowaniem. Do zbioru $\mathcal{R}$ należą para (KM, KM'). Jest to właściwie oczywiste. Z definicji $\to_\alpha$ otrzymujemy, że $KM \to_\alpha KM'$, a jeżeli term N jest podstawialny za z w KM i KM', to także jest podstawialny w M i M', z założenia $\Phi(M, M')$ wynika, że $M[z := N] \to_\alpha M'[z := N]$, a także $K[z := N]M[z := N] \to_\alpha K[z := N]M'[z := N]$. W przypadku drugiem pary postępujemy analogicznie.

Zgodność z abstrakcją. Teraz pokażemy, że także para $(\lambda x\, M, \lambda x\, M')$ należy do $\mathcal{R}$. Jest jasne, że $\lambda x\ M \to_\alpha \lambda x\ M'$. Dalej rozważamy kilka przypadków.

Jeżeli $z = x$ lub $z \notin FV(M)$, to sprawa jest prosta: podstawianie tak naprawdę niczego nie robi i teza wynika bezpośrednio z założenia, że $M \to_\alpha M'$ i definicji $\to_\alpha$.

W przeciwnym razie, podstawianie i dopisywanie operatora abstrakcji można wykonywać w dowolnej kolejności: $(\lambda x\ M)[z := N] = \lambda x\ M[z := N]$. Z założenia indukcyjnego mamy $M[z := N] \to_\alpha M'[z := N]$, możemy więc do obu stron relacji dopisać operator λx i wyciągnąć za niego operację podstawiania.

Pozostało jeszcze wykazać, że relacja $\mathcal{R}$ dopuszcza

Możliwość zamiany zmiennych. Przypuśćmy więc, że $M = \lambda x\ A$, $M' = \lambda y\ A[x := y]$ dla pewnej zmiennej $y \notin FV(M)$, podstawialnej za x w termie A. Naszym celem jest pokazanie, że $(M, M') \in \mathcal{R}$. W szczególności więc, $\mathbf{0}M \to_\alpha M'$. To jest oczywiste na mocy definicji $\to_\alpha$.

Aby wykazać drugą z wymaganych własności pary (M, M') załóżymy, że mamy term N podstawialny za z w tych obu termach i pokażemy, że $(\lambda x\ A)[z := N] \to \alpha(\lambda y\ A[x := y])[z := N]$. W tym celu rozważamy kilka przypadków.

Przypadek 1: $z = x$ i $z = y$. Teza zachodzi, ponieważ żadne z podstawień nie powoduje zmian oraz $M \to_\alpha M'$.

Przypadek 2: $z = x$ i $z \neq y$. Także w tym przypadku żadne z podstawień nie zmienia termów. Tak jest, gdyż w termie $\lambda y\ A[x := y]$ nie ma wolnych wystąpień zmiennej z, wszystkie wolne wystąpienia z zostały zamienione na wystąpienia y.

Przypadek 3: $z \neq x$ i $z = y$. Podobnie, jak w poprzednim przypadku. W termie $\lambda x\ A$ nie ma wolnych wystąpień zmiennej y, czyli zmiennej z.

Przypadek 4: $z \neq x$ i $z \neq y$. W tym przypadku pokażemy, że $A[x := y][z := N] = A[z := N][x := y]$, zmienna y nie jest wolna w $A[z := N]$ i jest podstawialna za x w $A[z := N]$. Warunki te pociągają za sobą należenie interesującej nas pary do relacji $\rightarrow_\alpha$ na mocy jej definicji.

Spełnianie powyższych warunków jest oczywiste, gdy $z \notin FV(A)$. W tym przypadku podstawienie N za z nie zmienia wymienionych termów.

Jeżeli $z \in FV(A)$, to założenia o podstawialności N implikują, że zmienne x i y nie są wolne w N. W tej sytuacji równość podstawień wynika z lematu 3.8. □

Lemat 5.7 *Przypuśćmy, że mamy term M_0, w którym występuje n operatorów λ, oraz term N podstawialny za zmienną y w termie M_0. Załóżmy też, że zmienne $z_1, z_2, \ldots, z_n$ nie występują w termie M_0 ani jako zmienne wolne, ani wiązane, a term N nie zawiera ich wolnych wystąpień. Wtedy po wykonaniu algorytmu z rozdziału 5.4 prawdziwe jest stwierdzenie*

$$M_0[y := N] \twoheadrightarrow_\alpha M_0[z_1, z_2, \ldots, z_n)[y := N].$$

Dowód. Aby dowieść ten lemat, wystarczy zauważyć, że niezmiennikiem algorytmu z rozdziału 5.4 jest stwierdzenie, że

$$M_0[y := N] \twoheadrightarrow_\alpha M[y := N].$$

Z lematu 5.5 i założenia o podstawialności N za zmienną y w M_0 wynika, że podczas działania algorytmu z rozdziału 5.4 term N jest podstawialny w M za zmienną y. Załóżmy też, że po wejściu do pętli w rozważanym algorytmie zachodzi warunek $M_0[y := N] \twoheadrightarrow_\alpha M[y := N]$. Podczas wykonania tej pętli term M jest przekształcany w term M' taki, że $M \rightarrow_\alpha M'$. Zgodnie z lematem 5.6 zachodzi także $M[y := N] \rightarrow_\alpha M'[y := N]$. Tym bardziej mamy $M_0[y := N] \twoheadrightarrow_\alpha M'[y := N]$, a ponieważ M' jest wartością zmiennej M po wykonaniu pętli, fakt ten świadczy o tym, że mamy do czynienia z niezmiennikiem. Stąd i z teorii niezmnienników wynika prawdziwość tezy dowodzonego lematu. □

5.6 Dwie bardzo ważne własności α-konwersji

Twierdzenie 5.8 *Dla każdych dwóch λ-termów M_0 i N oraz dla dowolnej zmiennej $y \in V$ istnieje term M taki, że $M_0 \twoheadrightarrow_\alpha M$, w którym term N jest podstawialny za zmienną y.*

Dowód. Weźmy termy M_0 z n operatorami λ i N, i wybierzmy zmienne $z_1, z_2, \ldots z_n$ nie występujące ani w termie M_0, ani w termie N. Term M_0 przekształcamy do termu $M_0[z_1, z_2, \ldots, z_n]$ zgodnie z algorytmem z rozdziału 5.4. W ten sposób otrzymaliśmy pewien term $M = M_0[z_1, z_2, \ldots, z_n]$. Pokażemy, że spełnia on tezę dowodzonego twierdzenia.

To, że mamy $M_0 \twoheadrightarrow_\alpha M$ wynika z teorii niezmienników i lematu 5.5. Stwierdzenie to zachodzi przed wykonaniem pętli z algorytmu wykorzystanego w konstrukcji M, i jako niezmiennik jest więc prawdziwe po wykonaniu pętli i całego algorytmu.

Posługując się niezmiennikami z lematu 5.5 pokazujemy również, że zmiennymi wiązanymi operatorami λ w termie M są jedynie $z_1, z_2, \ldots, z_n$. Zmienne te nie występują w termie N, a więc zmienne z N nie są wiązane w M żadnym operatorem λ. W tej sytuacji term N jest podstawialny w M na mocy lematu 3.1. □

Twierdzenie 5.9 *Przypuśćmy, że mamy dwa termy M_1 i M_2, które są w relacji α-konwersji (czyli takie, że $M_1 =_\alpha M_2$) i term N, który jest w nich podstawialny za zmienną y. Wtedy termy $M_1[y := N]$ i $M_2[y := N]$ też są w relacji α-konwersji (czyli $M_1[y := N] =_\alpha M_2[y := N]$).*

Dowód. Mając termy M_1 i M_2, oraz term N, spełniające założenia, wybieramy zmienne $z_1, z_2, \ldots, z_n$, które nie występują ani w M_1, ani w M_2, ani nie są wolne w N. Wybieramy tyle zmiennych, ile operatorów λ znajduje się w termie M_1 (a także w M_2, termy α-konwertowalne mają tyle samo operatorów λ).

Teraz oba termy M_1 i M_2 przekształcamy w termy $M_1(z_1, z_2, \ldots, z_n)$ i $M_2(z_1, z_2, \ldots, z_n)$ stosując algorytm z rozdziału 5.4. Otrzymane w ten sposób termy są identyczne. Z lematu 5.5 otrzymujemy bowiem dwa fakty: że w każdym z tych termów k-ty operator λ wiąże zmienną z_k dla wszystkich $k < n$, jak również, że termy te można otrzymać w wyniku α-redukcji z α-konwertowalnych termów M_1 i M_2. W tej sytuacji, identyczność wspomnianych termów wynika z wniosku 5.4 i lematu 5.2.

Na mocy lematu 5.7, termy $M_1[y := N]$ i $M_2[y := N]$ można α-zredukować do $M_1(z_1, z_2, \ldots, z_n)[y := N]$ i $M_2(z_1, z_2, \ldots, z_n)[y := N]$ odpowiednio. Tak więc termy $M_1[y := N]$ i $M_2[y := N]$ można α-zredukować do tego samego termu $M_1(z_1, z_2, \ldots, z_n)[y := N] = M_2(z_1, z_2, \ldots, z_n)[y := N]$. Stąd teza. □

5.7 Raz jeszcze o λ-termach

6 Dodatki

6.1 Ewolucja symbolu abstrakcji

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

6.2 Gramatyka definiująca λ-wyrażenia

Niżej jest przedstawiona próba zdefiniowania gramatyki generującej wyrażenia rachunku lambda zgodne z zasadami opisanymi w rozdziale 1.5, także dotyczącymi opuszczania nawiasów.

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⟩