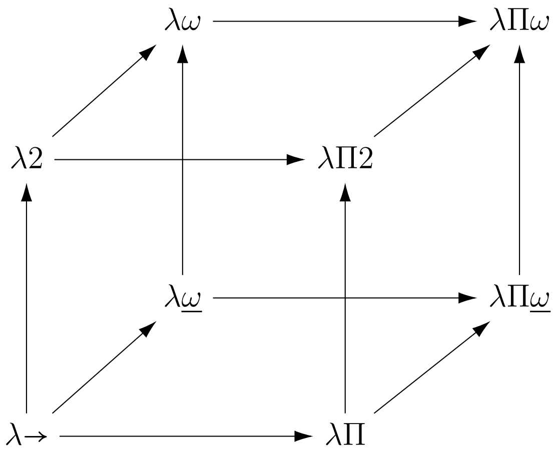
# 1 Rachunek $\lambda$ z typami prostymi

Przedstawimy system rachunku $\lambda$ z typami prostymi w stylu de Bruijna [BDS13; SU06]. Zgodnie z [HS08] odpowiada on najsłabszemu z *czystych systemów typów* (ang. *pure type systems*, PTS) w myśl klasyfikacji zaproponowanej przez Henka Barendregta w [Bar91]. Klasyfikację tę obrazuje Rysunek 1; kierunek krawędzi oznacza na nim zawieranie w sensie możliwości wyrażenia słabszego systemu przez system wzbogacony o nowe typy.




Rysunek 1: Kostka lambda H. Barendregta

## 1.1 Typy proste

Niech U będzie przeliczalnie nieskończonym zbiorem zmiennych przedmiotowych $p$, $q$, ... (być może indeksowanych liczbami naturalnymi), które będziemy nazywali *zmiennymi typowymi*.

**Definicja 1.** (Typy proste)
*Typami prostymi* będziemy określali najmniejszy w sensie mnogościowym zbiór wyrażeń taki, że:

T1. Jeśli $p$ jest zmienną typową, to $p$ jest typem prostym.

T2. Jeśli $\tau$ i $\sigma$ są typami prostymi, to $(\tau \rightarrow \sigma)$ jest typem prostym.

Typy proste zbudowane tylko wedle reguły T1 nazywamy typami *atomowymi*, zaś wyrażenia zbudowe wedle reguły T2 – typami *funkcyjnymi*. Zbiór typów prostych określony w myśl powyższej definicji będziemy oznaczali przez $\mathbf{T}_{\rightarrow}$.

Późniejsze litery alfabetu greckiego, tj. $\sigma$, $\tau$, $\rho$, ... będą służyły nam za zmienne metasyntaktyczne do oznaczania typów prostych. Dla lepszej czytelności będziemy pomijali najbardziej zewnętrzne nawiasy. Konstruktor typu $\to$ jest prawostronnie łączny, co oznacza, że typy $\tau \to \sigma \to \theta$ oraz $\tau \to (\sigma \to \theta)$ będziemy uznawali za tożsame.

Typy proste ujęte Definicją 1 mają strukturę drzewa binarnego. Wysokość takiego drzewa będziemy nazywali *stopniem* typu. Precyzyjnie ujmuje to pojęcie poniższa definicja.

**Definicja 2.** (Stopień typu)
Stopniem typu nazywamy funkcję $\delta : \mathbf{T}_{\to} \longrightarrow \mathbb{N}$ taką, że

$$\begin{aligned} \delta(p) &= 0, \text{ gdzie } p \text{ jest typem atomowym,} \\ \delta(\tau \to \sigma) &= 1 + \max\left(\delta(\tau),\ \delta(\sigma)\right). \end{aligned}$$

## 1.2 Pseudotermy

Niech V będzie przeliczalnie nieskończonym zbiorem zmiennych przedmiotowych $x$, $y$, ... (indeksowanych być może liczbami naturalnymi). Elementy takiego zbioru będziemy nazywali $\lambda$-*zmiennymi*.

**Definicja 3.** (Pseudo-pretermy)
*Pseudo-pretermami* będziemy nazywali najmniejszy (w sensie mnogościowym) zbiór $\tilde{\mathbf{\Lambda}}_{\mathrm{T}}$ taki, że:

P1. Jeśli $x \in \mathrm{V}$, to $x \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$.

P2. Jeśli $M \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$ i $N \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$, to $(MN) \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$.

P3. Dla dowolnych $x \in \mathrm{V}$, $\sigma \in \mathbf{T}_{\to}$, $M \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$ mamy, że $(\lambda x^{\sigma}. M) \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}$.

Wyrażenia postaci P2 nazywamy *aplikacjami* $M$ do $N$, zaś wyrażenia postaci P3 – $\lambda$-*abstrakcjami*, gdzie o wszystkich podtermach termu $M$ mówi się, że są w *zasięgu* $\lambda$-abstraktora, zaś o $\lambda$-zmiennej $x$ mówi się, że jest nim *związana*.

Za zmienne metasyntaktyczne obieramy duże litery alfabetu łacińskiego $M$, $N$, ... Podobnie jak w podrozdziale 1.1 stosujemy konwencję o opuszczaniu najbardziej zewnętrznych nawiasów. Aplikacja termów jest łączna lewostronnie, co oznacza, że będziemy utożsamiali wyrażenia $MNP$ oraz $(MN)P$.

**Definicja 4.** (Zmienne wolne)

Dla pseudo-pretermu $M$ określamy zbiór *termów wolnych* FV w nastepujący sposób:

$$\begin{aligned} \mathrm{FV}(x) &= \{x\} \\ \mathrm{FV}(\lambda x^{\sigma}.\, P) &= \mathrm{FV}(P) \setminus \{x\} \\ \mathrm{FV}(PQ) &= \mathrm{FV}(P) \cup \mathrm{FV}(Q) \end{aligned}$$

Jesli $\mathrm{FV}(M) = \varnothing$, to mówimy, że $M$ jest *zamknięty.*

**Definicja 5.** (Podstawienie)
*Podstawieniem* $[x/N]$ pseudo-pretermu $N$ za $\lambda$-zmienną $x$ w $M$ nazwamy zdefiniowane następująco przekształcenie:

$$\begin{aligned} x[x/N] &= N, && \\ y[x/N] &= y, && \text{o ile } x \neq y, \\ (PQ)[x/N] &= P[x/N]\, Q[x/N], && \\ (\lambda y^{\sigma}.\, P)[x/N] &= \lambda y^{\sigma}.\, P[x/N], && \text{gdzie } x \neq y \text{ i } y \notin \mathrm{FV}(N). \end{aligned}$$

Zachodzą następujące fakty:

**Fakt 1.** *(a) Jeśli $x \notin \mathrm{FV}(M)$, to $M[x/N]$ jest poprawnym podstawieniem i $M[x/N] = M$.*

*(b) Jeśli $M[x/N]$ jest poprawnym podstawieniem, to $y \in \mathrm{FV}(M[x/N]$ wtw, gdy albo $y \in \mathrm{FV}(M)$ i $x \neq y$, albo $y \in \mathrm{FV}(N)$ i $x \in \mathrm{FV}(M)$.*

*(c) Podstawienie $M[x/x]$ jest poprawne i $M[x/x] = M$.*

*(d) Jeśli $M[x/y]$ jest poprawnym podstawieniem, to $M[x/y]$ ma tę samą długość, co $M$.*

**Fakt 2.** *Powiedzmy, że $M[x/N]$ jest poprawnym podstawieniem i $N[y/L]$ i $M[x/N][y/L]$ są poprawnymi podstawieniami, gdzie $x \neq y$. Jeśli $x \notin \mathrm{FV}(L)$ lub $y \notin \mathrm{FV}(M)$, to $M[y/L]$ i $M[y/L][x/N[y/L]]$ jest poprawnym podstawieniem oraz*

$$M[x/N][y/L] = M[y/L][x/N[y/L]].$$

**Fakt 3.** *Jesli $M[x/y]$ jest poprawnym postawieniem i $y \notin \mathrm{FV}(M)$, to $M[x/y][y/x]$ jest poprawnym podstawieniem oraz $M[x/y][y/x] = M$.*

**Definicja 6.** ($\alpha$-konwersja)
$\alpha$-konwersją nazywamy najmniejszą (w sensie mnogościowym) zwrotną i przechodnią relację binarną $=_{\alpha}$ określoną na zbiorze pseudotermów $\tilde{\mathbf{\Lambda}}_{\mathrm{T}}$ spełniającą poniższe warunki:

(a) Jeśli $y \notin \mathrm{FV}(M)$ i $M[x/y]$ jest poprawnym podstawieniem, to $\lambda x.\, M =_\alpha \lambda y.\, M[x/y]$.

(b) Jeśli $M =_\alpha N$, to dla każdej $\lambda$-zmiennej $x$ mamy $\lambda x.\, M =_\alpha \lambda x.\, N$.

(c) Jeśli $M =_\alpha N$, to $MZ =_\alpha NZ$.

(d) Jeśli $M =_\alpha N$, to $ZM =_\alpha ZN$.

**Fakt 4.** *Relacja* $=_\alpha$ *jest symetryczna.*

**Fakt 5.** $=_\alpha$ *jest relacją równoważności.*

**Fakt 6.** *Jeśli* $M =_\alpha N$, *to* $\mathrm{FV}(M) = \mathrm{FV}(N)$.

Dysponując powyższymi rozstrzygnięciami otrzymujemy wygodne utożsamienie pseudo-pretermów, które różnią się między sobą tylko zmiennymi związanymi.

**Definicja 7.** (Pseudotermy)
Klasy abstrakcji relacji $\alpha$-konwersji nazywamy *pseudotermami*. Zbiór wszystkich pseudotermów oznaczamy następująco:

$$\mathbf{\Lambda}_{\mathrm{T}} = \left\{[M]_\alpha \mid M \in \tilde{\mathbf{\Lambda}}_{\mathrm{T}}\right\}$$

Nadużywając notacji będziemy odnosili się do pseudotermów tylko przez ich reprezentantów, tj. zamiast $[\lambda x^\sigma.M]_\alpha$ będziemy pisali krótko: $\lambda x^\sigma.M$.

## 1.3 Typowalność

**Definicja 8.** (Kontekst)
*Kontekstem* nazywamy skończoną funkcję częściową $\Gamma : \mathrm{V} \longrightarrow \mathbf{T}_\to$, czyli zbiór par postaci $\Gamma = \{x_1^{\tau_1}, \ldots, x_n^{\tau_n}\}$, gdzie $(x_i^{\tau_i}) = (x_i, \tau_i)$ oraz $x_i \neq x_j$ dla $i \neq j$. Zbiór

$$\mathrm{dom}(\Gamma) = \{x \in \mathrm{V} \mid \exists \tau (x^\tau \in \Gamma)\}$$

nazywamy *dziedziną* kontekstu $\Gamma$, zaś

$$\mathrm{rg}(\Gamma) = \{\tau \in \mathbf{T}_\to \mid \exists x (x^\tau \in \Gamma)\}$$

– *zakresem* kontekstu $\Gamma$. Piszemy:
– $x_1^{\tau_1}, x_2^{\tau_2}$ zamiast $\{x_1^{\tau_1}, x_2^{\tau_2}\}$, o ile $x_1^{\tau_1}$ i $x_2^{\tau_2}$ są różne,
– $\Gamma, x^\tau$ zamiast $\Gamma \cup \{x^\tau\}$, o ile $x^\tau \notin \Gamma$,
– $\Gamma, \Delta$ zamiast $\Gamma \cup \Delta$, o ile $\Gamma \cap \Delta = \varnothing$.

Okreslimy teraz system przypisywania typów do pseudotermów w stylu dedukcji naturalnej. *Sekwentami* w tym systemie będziemy nazywali wyrażenia postaci $\Gamma \vdash M^\sigma$, gdzie $M \in \mathbf{\Lambda}_{\mathrm{T}}$, $\sigma \in \mathbf{T}_{\to}$, zaś $\Gamma$ jest pewnym kontekstem.

Wprowadzamy następujące reguły dowodzenia:

$$\frac{}{\Gamma, x^\tau \vdash x^\tau}\ (\mathrm{Var}),\qquad \frac{\Gamma, x^\varphi \vdash M^\psi}{\Gamma \vdash (\lambda\, x^\varphi.\, M)^{\varphi\to\psi}}\ (\mathrm{Abs}),\qquad \frac{\Gamma \vdash M^{\varphi\to\psi} \qquad \Gamma \vdash N^\varphi}{\Gamma \vdash (MN)^\psi}\ (\mathrm{App}).$$

**Definicja 9.** (Typowalność)
Mówimy, że pseudoterm $M$ jest typu $\sigma$ w kontekście $\Gamma$ (jest *typowalny*), jeśli istnieje skończone drzewo sekwentów spełniające poniższe warunki:

1. W korzeniu drzewa znajduje się sekwent $\Gamma \vdash M^\sigma$.
2. Liście są *aksjomatami*, tj. sekwentami postaci $\Gamma, x^\sigma \vdash x^\sigma$.
3. Każdego rodzica można otrzymać z jego dzieci przez zastosowanie którejś z reguł wyprowadzania nowych sekwentów.

Tak określone drzewo będziemy nazywali *wyprowadzeniem* typu i pisali $\Gamma \vdash M^\sigma$. W danym kontekście każdy $\lambda$-term ma jednoznacznie przypisany typ, co stwierdza następujące twierdzenie:

**Fakt 7.** *Jesli* $\Gamma \vdash M^\sigma$ *oraz* $\Gamma \vdash M^\tau$*, to* $\sigma = \tau$*.*

Otrzymujemy ostateczne określenie $\lambda$-termów w omawianym systemie.

**Definicja 10.** ($\lambda$-termy)
Wszystkie typowalne pseudotermy w pewnym kontekście $\Gamma$ nazywamy *$\lambda$-termami* (z typami prostymi w kontekście $\Gamma$).

*Uwaga.* $\lambda$-term w kontekście $\Gamma_1$ może nie być typowalny w innym kontekście $\Gamma_2$.

**Przykład 1.** Niech $\Gamma = \{x^\sigma,\, y^\tau\}$. Pokażemy, że $\mathrm{K} = \lambda x^\sigma y^\tau.\, x$ ma typ $\sigma \to \tau \to \sigma$. Istotnie,

$$\cfrac{\cfrac{x^\sigma, y^\tau \vdash x^\sigma}{x^\sigma \vdash (\lambda y^\tau.\, x)^{\tau\to\sigma}}\ (\mathrm{Abs})}{\vdash (\lambda x^\sigma \lambda y^\tau.\, x)^{\sigma\to\tau\to\sigma}}\ (\mathrm{Abs})$$

**Przykład 2.** Niech $\Gamma = \{x^{\tau\to\rho},\, y^{\sigma\to\tau},\, z^\sigma\}$. Wówczas

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma \vdash z^\sigma \qquad \Gamma \vdash y^{\sigma\to\tau}}{\Gamma \vdash yz^\tau}\ (\mathrm{App}) \qquad \Gamma \vdash x^{\tau\to\rho}}{\Gamma \vdash x(yz)^\rho}\ (\mathrm{App})}{x^{\tau\to\sigma}, y^{\sigma\to\rho} \vdash (\lambda z^\sigma.\, x(yz))^{\sigma\to\rho}}\ (\mathrm{Abs})}{x^{\tau\to\rho} \vdash (\lambda y^{\sigma\to\tau} \lambda z^\sigma.\, x(yz))^{(\sigma\to\tau)\to\sigma\to\rho}}\ (\mathrm{Abs})}{\vdash (\lambda x^{\tau\to\rho} \lambda y^{\sigma\to\tau} \lambda z^\sigma.\, x(yz))^{(\tau\to\rho)\to(\sigma\to\tau)\to\sigma\to\rho}}\ (\mathrm{Abs})$$

Term $\lambda x^{\tau\to\rho} y^{\sigma\to\tau} z^{\sigma}.x(yz)$ odpowiada operacji złożenia funkcji. Co więcej, można wykazać że typ $(\tau \to \rho) \to (\sigma \to \tau) \to \sigma \to \rho$ może mieć tylko i wyłącznie operacja złożenia [Wad89].

*Uwaga* 1. Widzimy, że wyprowadzanie typu odpowiada w gruncie rzeczy konstrukcji $\lambda$-termu. Ponieważ każde wyprowadzenie musi być skończone, typowalne są tylko pseudotermy skończonej długości.

*Uwaga* 2. Mówiąc o $\lambda$-termach i nie podając żadnego związanego z nimi kontekstu $\Gamma$ będziemy implicite zakładali, że istnieje pewien kontekst w którym są one typowalne.

Typ dowolnego $\lambda$-termu będziemy w ramach konwencji notowali używając adnotacji typowej umieszczonej w górnym indeksie. Dla przykładu, pisząc $(M^{\sigma\to\tau} N^{\sigma})^{\tau}$ będziemy mieli na myśli, że ten $\lambda$-term jest w pewnym kontekście $\Gamma$ typu $\tau$.

**Przykład 3.** Nie wszystkie pseudotermy są typowalne w rachunku $\lambda$ z typami prostymi. Istotnie, przypuśćmy że $\omega = (\lambda x^{\sigma\to\sigma}.\, xx)$ jest typowalny. Wówczas dla kontekstu $\Gamma$ mamy $x^{\sigma\to\sigma} \in \Gamma$. Ponieważ $\omega$ zawiera w sobie podterm $(xx)$, to w wyprowadzeniu musiał on zostać otrzymany przez zastosowanie reguły (App). Wówczas $x^{\sigma} \in \Gamma$ i $x^{\sigma\to\sigma} \in \Gamma$, co nie jest możliwe, bo $\Gamma$ musiałby nie być prawidłowo określonym kontekstem.

Przykład 3 wskazuje, że rachunek $\lambda$ z typami prostymi nie ma wystarczających środków wyrazu, aby definiować w nim funkcje rekurencyjne. W rachunku $\lambda$ bez typów definicje takie są osiągane za pomocą *kombinatorów punktu stałego*, które w rachunku $\lambda$ z typami prostymi nie są typowalne właśnie ze względu na fakt, że nie jest możliwe wyprowadzenie typu dla aplikacji termu do samego siebie.

## 1.4 Redukcja

**Definicja 11.** (Zgodność)
Relację R na zbiorze termów $\mathbf{\Lambda}_{\mathrm{T}}$ nazywamy *zgodną*, jeśli dla $M$, $N$, $Z \in \mathbf{\Lambda}_{\mathrm{T}}$ spełnia ona następujące warunki:

i) Jeśli $M \mathrm{R} N$, to $(\lambda x^{\sigma}.\, M)\, \mathrm{R}\, (\lambda x^{\sigma}.\, N)$ dla dowolnych $x \in \mathrm{V}$ i $\sigma \in \mathbf{T}_{\to}$.

ii) Jeśli $M \mathrm{R} N$, to $(MZ)\, \mathrm{R}\, (NZ)$.

iii) Jeśli $M \mathrm{R} N$, to $(ZM)\, \mathrm{R}\, (ZN)$.

Przy powyższych ustaleniach *kongruencją* będziemy nazywali każdą zgodną relację równowazności na $\mathbf{\Lambda}_{\mathrm{T}}$, zaś *redukcją* – każdą zgodną, zwrotną i przechodnią relację na $\mathbf{\Lambda}_{\mathrm{T}}$.

**Definicja 12.** ($\beta$-redukcja)
$\beta$-redukcją nazywamy najmniejsza w sensie mnogościowym *zgodną* relację binarną $\longrightarrow_\beta$ określoną zbiorze na pseudotermów $\mathbf{\Lambda}_\mathrm{T}$ za pomocą podstawienia

$$(\lambda x^\sigma .\, P)Q \longrightarrow_\beta P[x/Q].$$

*$\beta$-redeksami* bedziemy nazywali wyrażenia postaci $(\lambda x^\sigma .\, M)N$, zaś rezultat ich $\beta$-redukcji w postaci termu $M[x/N]$ – *$\beta$-reduktem.*

Nadużywając notacji, z każdym $\beta$-redeksem $\Delta$ postaci $(\lambda x^\tau .\, P^\rho)R$ będziemy wiązali jego *stopień* i pisali $\delta((\lambda x^\tau .\, P^\rho)R) = \delta(\tau \to \rho)$, gdzie występująca po prawej stronie równości $\delta$ jest określona w myśl Definicji 2. Dysponując tą konwencją możemy określić pojęcie stopnia dowolnego $\lambda$-termu.

**Definicja 13.** (stopień $\lambda$-termu)
Stopniem $\mathrm{d}(M)$ $\lambda$-termu $M$ nazywamy supremum zbioru stopni $\beta$-redeksów, które są zawarte w $M$, czyli

$$\mathrm{d}(M) = \sup\{\delta(N) \mid N \text{ jest } \beta\text{-redeksem w } M\}.$$

*Uwaga.* Z każdym $\beta$-redeksem $\Delta$ związane są więc dwa rodzaje stopni: stopień $\mathrm{d}(\Delta)$ jako $\lambda$-termu w myśl Defnicji 13 i stopień $\beta$-redeksu $\delta(\Delta)$.

Określamy następujące relacje:

B1. $\longrightarrow_\beta^+$ jest przechodnim domknięciem relacji $\longrightarrow_\beta$ w zbiorze pseudotermów $\mathbf{\Lambda}_\mathrm{T}$.

B2. $\longrightarrow_\beta^*$ jest domknięciem przechodnio-zwrotnim w $\mathbf{\Lambda}_\mathrm{T}$ relacji $\longrightarrow_\beta$, a zatem jest *redukcją.*

B3. $=_\beta$ jest najmniejszą relację równowazności zawierającą relację $\longrightarrow_\beta$, (czyli *kongruencją*).

**Definicja 14.** (Postać normalna)
Powiemy, że $\lambda$-term $M$ jest w *postaci normalnej*, jeśli żadna z jego podformuł nie jest $\beta$-redeksem. Przez $\mathrm{NF}_\beta$ będziemy oznaczali zbiór wszystkich $\lambda$-termów w postaci normalnej.

Definicję postaci normalnej można ująć w alternatywny sposób w postaci Faktu 8.

**Fakt 8.** $M$ ma postać normalną*, jeśli $M =_\beta N$ dla pewnego $N$, który jest w postaci normalnej.*

*Uwaga.* Fakt, że $=_\beta$ jest relacją równoważności może rodzić pokusę, aby utożsamić ze sobą wszystkie postacie normalne danego $\lambda$-termu. Z Twierdzenia 3 wynika jednak, że jest to zupełnie zbędne, bowiem o ile tylko $\lambda$-term jest normalizowalny, to wiemy, że posiada dokładnie jedną postać normalną i każde takie utożsamienie byłoby trywialne.

**Definicja 15.** ($\eta$-redukcja)
$\eta$-redukcją nazywamy najmniejszą (w sensie mnogościowym) *zgodną* relację w $\mathbf{\Lambda}_{\mathrm{T}}$ taką, że

$$\lambda x^\sigma .\, Mx \longrightarrow_\eta M,$$

o ile $x \notin \mathrm{FV}(M)$.

Zdefiniowane powyżej $\beta$- i $\eta$-redukcje zachowują typ, jak stwierdza Fakt 9. Własność ta pozwala sensownie mówić o redukcji $\lambda$-termów pomimo, że te relacje są zdefiniowane dla pseudotermów.

**Fakt 9.** *(O poprawności redukcji) Jeśli* $\Gamma \vdash M^\sigma$ *i* $M \longrightarrow^*_{\beta\eta} N$, *to* $\Gamma \vdash N^\sigma$.

## 1.5 Normalizacja

Powiemy, że $\lambda$-term $M$ ma własność:
- *(słabej) normalizacji* (symbolicznie: $M \in \mathrm{WN}_\beta$) wtedy i tylko wtedy, gdy istnieje ciąg $\beta$-redukcji rozpoczynający się od $M$ i kończący się termem w postaci normalnej $N$.
- *silnej normalizacji* (symbolicznie: $M \in \mathrm{SN}_\beta$), jeśli wszystkie ciągi $\beta$-redukcji rozpoczynające się od $M$ są skończone.

*Uwaga.* Z powyższego określenia widzimy, że własność $\mathrm{SN}_\beta$ pociąga za sobą własność $\mathrm{WN}_\beta$.

**Definicja 16.** (Strategia redukcji)
*Strategią redukcji* nazywamy odwzorowanie $F : \mathbf{\Lambda}_{\mathrm{T}} \longrightarrow \mathbf{\Lambda}_{\mathrm{T}}$ takie, że $F(M) = M$, gdy $M$ jest w postaci normalnej i $M \longrightarrow_\beta F(M)$ w przeciwnym wypadku. Mówimy, że strategia $F$ jest *normalizująca*, jeśli dla każdego $M \in \mathrm{WN}_\beta$ istnieje $i \in \mathbb{N}$ takie, że $F^i(M)$ jest w postaci normalnej.

**Twierdzenie 1.** *(*Własność $\mathrm{WN}_\beta$*) Wszystkie $\lambda$-termy mają postać normalną.*

*Dowód.* Pokażemy, że dla dowolnego $\lambda$-termu $M$ istnieje normalizująca strategia redukcji.

Oznaczmy przez $\mathrm{R}_\beta(M)$ zbiór $\beta$-redeksów znajdujących się w $M$. Jeśli $M \in \mathrm{NF}_\beta$, to $\mathrm{R}_\beta(M) = \varnothing$ i twierdzenie zachodzi w sposób trywialny. Jeśli $M \notin \mathrm{NF}_\beta$, to istnieje w $M$ przynajmniej jeden $\beta$-redeks. Z Uwagi 1 $M$ jest skończonej długości, więc $\mathrm{R}_\beta(M)$ jest skończony. Możemy więc wybrać z $M$ $\beta$-redeks znajdujący się (rozpoczynający się) w $M$ najbardziej na prawo. Oznaczmy taki $\beta$-redeks przez $\Delta$.

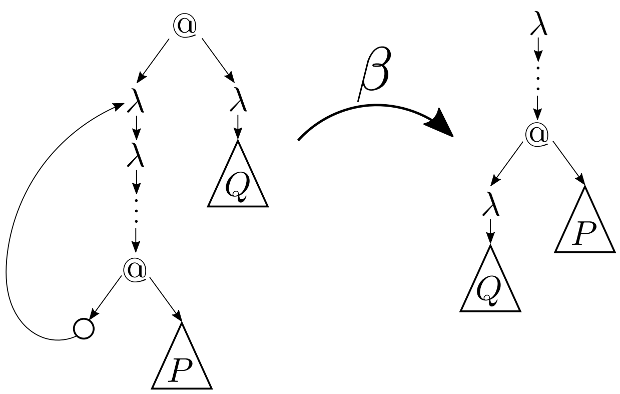
Niech $\delta_M$ będzie stopniem $M$ (Definicja 13). Ponieważ wiele redeksów w $M$ może mieć ten sam stopień $\delta_M$, przez $n_M$ oznaczmy liczbę wystąpień redeksów stopnia $\delta_M$ w $M$.

Niech $F$ będzie strategią redukcji polegającą na $\beta$-redukowaniu redeksu $\Delta$ wybranego jak wyżej i niech $M' = F(M)$. Zauważmy, że $n_M < n_{M'}$, gdyż strategia $F$ eliminuje $\Delta$ z $M'$ i może prowadzić do powstania redeksów tylko mniejszego stopnia. Istotnie, ilość redeksów w $M$ może zwiększyć się na skutek $\beta$-redukcji tylko w jeden z poniższych sposobów:

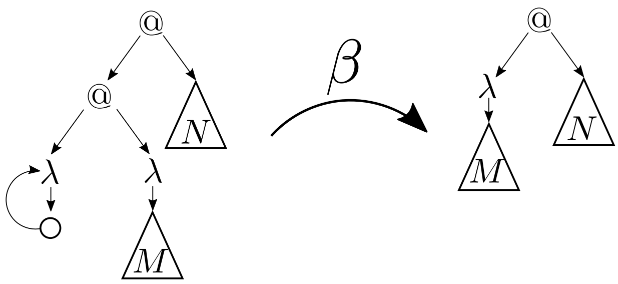
i) powstanie nie występujących wcześniej redeksów.

ii) powielenie już istniejących redeksów.

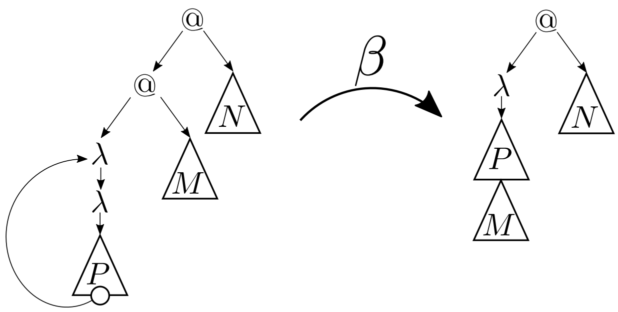
W przypadku i)




(a) Redukcja $(\lambda x^{\rho\to\mu}\overbrace{\dots\ .xP^\rho\dots}^{M^\tau})(\lambda y^\rho.\,Q^\mu)^{\rho\to\mu}$ do nowego redeksu $\underbrace{(\dots\ .(\lambda y^\rho.\,Q^\mu)P^\rho\dots)}_{M[x/Q]^\tau}$.




(b) Redukcja $(\lambda x^{\tau} \lambda y^{\rho}.\, P^{\sigma}) M^{\tau} N^{\rho}$ do nowego redeksu postaci $(\lambda y^{\rho}.\, P[x/M]^{\sigma}) N^{\rho}$.




(c) Redukcja $(\lambda x^{\tau \to \rho}.\, x)(\lambda y^{\tau}.\, M^{\rho}) N^{\tau}$ do nowego redeksu $(\lambda y^{\tau}.\, M^{\rho}) N^{\tau}$.

Rysunek 2: test

Powtarzając otrzymujemy więc $\lambda$-term w postaci normalnej. $\square$

- WCR: $\forall a, b, c \in A\,(a \longrightarrow b \wedge a \longrightarrow c) \to \exists d \in A\,(b \longrightarrow^{*} d \wedge c \longrightarrow^{*} d)$
- CR: $\forall a, b, c \in A\,(a \longrightarrow^{*} b \wedge a \longrightarrow^{*} c) \to \exists d \in A\,(b \longrightarrow^{*} d \wedge c \longrightarrow^{*} d)$

**Twierdzenie 2.** (Lemat Newmana) *Niech $\to$ bedzie relacją binarną spełniającą $SN$. Jeśli $\to$ spełnia WCR, to spełnia CR.*

*Dowód.*

**Twierdzenie 3.** *(Własność CR) Rachunek $\lambda$ z typami prostymi ma własność Churcha-Rossera.*

**Twierdzenie 4.** *(*Własność $\mathrm{SN}_\beta$*) Wszystkie $\lambda$-termy mają własność silnej normalizacji.*

*Dowód.*

# Literatura


[Bar91] Henk Barendregt. "Introduction to generalized type systems". In: *Journal of Functional Programming* 1.2 (1991), pp. 125–154. DOI: 10.1017/S0956796800020025.

[BDS13] Henk Barendregt, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in Logic. Cambridge University Press, 2013. DOI: 10.1017/CBO9781139032636.

[HS08] J. Roger Hindley and Jonathan P. Seldin. *Lambda-Calculus and Combinators: An Introduction*. 2nd ed. New York, NY, USA: Cambridge University Press, 2008. ISBN: 0521898854, 9780521898850.

[SU06] Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. New York, NY, USA: Elsevier Science Inc., 2006. ISBN: 0444520775.

[Wad89] Philip Wadler. "Theorems for Free!" In: *Proceedings of the Fourth International Conference on Functional Programming Languages and Computer Architecture*. FPCA '89. Imperial College, London, United Kingdom: ACM, 1989, pp. 347–359. ISBN: 0-89791-328-0. DOI: 10.1145/99370.99404. URL: http://doi.acm.org/10.1145/99370.99404.