λ -notacja

Idea: $W[x]$ – wyrażenie zawierające x.
Jeśli podstawimy za x obiekt $d \in D$ to $W[d] \in D'$.

Napis $\lambda x.W[x]$ jest to tzw. λ-wyrażenie, gdzie x jest jego zmienną związaną, a treść wyrażenia rozciąga się jak najdalej na prawo. Np.

$$\begin{aligned} \lambda x.x + 1 &= \lambda x.(x + 1) \\ &\neq (\lambda x.x) + 1 \end{aligned}$$

Wyrażenie to oznacza funkcję

$$f : D \to D'$$

taką, że dla każdego $d \in D \quad fd = W[d]$

Przykłady

$\lambda x.x + 1(suc)$: $N \to N$

$\lambda x.x * x$: $N \to N$

$\lambda x.$**if** $x = 0$ **then** prawda **else** fałsz : $N \to B$

$\lambda(x, y).$**if** $x = y$ **then** prawda **else** fałsz : $[N \times N] \to B$

Funkcje wyższego rzędu

Są to funkcje, których argumenty lub wyniki są funkcjami. Np.

$$
\begin{aligned}
dwakroć &: [N \to N] \to [N \to N] \\
dwakroć &= \lambda f.\lambda x.f(f(x)) \\
dwakroć(suc) &= \lambda f.\lambda x.f(f(x))suc \\
&= \lambda x.suc(suc(x)) \\
&= \lambda x.\lambda y.(y+1)((\lambda y.y+1)x) \\
&= \lambda x.\lambda y.(y+1)(x+1) \\
&= \lambda x.x+1+1 \\
&= \lambda x.x+2
\end{aligned}
$$

Iloczyny kartezjańskie

Np.

$$\begin{aligned} plus &: [N \times N] \to N \\ plus &= \lambda(n,m)n+m \\ plusc &: N \to [N \to N] \\ plusc &= \lambda n.\lambda m.n+m \\ plusc(1) &= \lambda m.1+m \end{aligned}$$

Ogólnie:

$[D_1 \times D_2 \times \ldots \times D_n] \to D \approx D_1 \to \ldots \to D_n \to D$

Definiujemy równania semantyczne:

Aexp

$$\mathcal{A}[[n]] = \lambda\sigma \in \Sigma.n$$

$$\mathcal{A}[[v]] = \lambda\sigma \in \Sigma.\sigma(v)$$

$$\mathcal{A}[[e_0 + e_1]] = \lambda\sigma \in \Sigma.(\mathcal{A}[[e_0]]\sigma + \mathcal{A}[[e_1]]\sigma)$$

$$\mathcal{A}[[e_0 - e_1]] = \lambda\sigma \in \Sigma.(\mathcal{A}[[e_0]]\sigma - \mathcal{A}[[e_1]]\sigma)$$

$$\mathcal{A}[[e_0 * e_1]] = \lambda\sigma \in \Sigma.(\mathcal{A}[[e_0]]\sigma * \mathcal{A}[[e_1]]\sigma)$$

Bexp

$\mathcal{B}[[tt]] = \lambda\sigma \in \Sigma.tt$

$\mathcal{B}[[ff]] = \lambda\sigma \in \Sigma.ff$

$\mathcal{B}[[b_0 \vee b_1]] = \lambda\sigma \in \Sigma.$**if** $\mathcal{B}[[b_0]]\sigma = tt$ **then** tt **else**
if $\mathcal{B}[[b_1]]\sigma = tt$ **then** tt **else** ff

$\mathcal{B}[[e_0 = e_1]] = \lambda\sigma \in \Sigma.$**if** $\mathcal{A}[[e_0]]\sigma = \mathcal{A}[[e_1]]\sigma$ **then** tt
else ff

$\mathcal{B}[[\neg b]] = \lambda\sigma \in \Sigma.$**if** $\mathcal{B}[[b]]\sigma = tt$ **then** ff **else** tt.

Com

$$\mathcal{C}[[\underline{\text{skip}}]] = \lambda\sigma \in \Sigma.\sigma$$

$$\mathcal{C}[[v := e]] = \lambda\sigma \in \Sigma.\sigma[\mathcal{A}[[e]]\sigma/v]$$

$$\mathcal{C}[[c_1; c_2]] = \lambda\sigma \in \Sigma.\mathcal{C}[[c_2]](\mathcal{C}[[c_1]]\sigma)$$

$$\mathcal{C}[[\underline{\text{if}}\ b\ \underline{\text{then}}\ c_1\ \underline{\text{else}}\ c_2]] = \lambda\sigma \in \Sigma.\textbf{if}\ \mathcal{B}[[b]]\sigma = tt\ \textbf{then}\ \mathcal{C}[[c_1]]\sigma \\ \textbf{else}\ \mathcal{C}[[c_2]]\sigma$$

$$\mathcal{C}[[\underline{\text{while}}\ b\ \underline{\text{do}}\ c]] = ?$$

Korzystamy z równoważności instrukcji (Fakt 6):

$$\underbrace{\underline{\text{while}}\ b\ \underline{\text{do}}\ c}_{w} = \underline{\text{if}}\ b\ \underline{\text{then}}\ c; \underbrace{\underline{\text{while}}\ b\ \underline{\text{do}}\ c}_{w}\ \underline{\text{else}}\ \underline{\text{skip}}$$

Równanie semantyczne ma postać:

$$w = \lambda\sigma \in \Sigma.\textbf{if}\ \mathcal{B}[[b]]\sigma = tt\ \textbf{then}\ w(\mathcal{C}[[c]]\sigma)\ \textbf{else}\ \sigma \qquad (1)$$

Tak więc
$\mathcal{C}[[\underline{\text{while}}\ b\ \underline{\text{do}}\ c]]$ = najmniejsze rozwiązanie w równania (1).

Prawa strona równania (1), dla ustalonych $\mathcal{B}[[b]]$, $\mathcal{C}[[c]]$, – funkcja: $[\Sigma \to \Sigma] \to [\Sigma \to \Sigma]$:

$\mathcal{F}(\mathcal{B}[[b]], \mathcal{C}[[c]]) = \lambda w \in [\Sigma \to \Sigma].\lambda\sigma \in \Sigma.$**if** $\mathcal{B}[[b]]\sigma = tt$ **then**
$w(\mathcal{C}[[c]]\sigma)$ **else** σ

$w = \mathcal{F}(\mathcal{B}[[b]], \mathcal{C}[[c]])(w)$

$\mathcal{C}[[\underline{\text{while}}\ b\ \underline{\text{do}}\ c]] = fix(\mathcal{F}(\mathcal{B}[[b]], \mathcal{C}[[c]]))$

Podstawowe pojęcia matematyczne

Porządki zupełne i funkcje ciągłe

Definicja 7
Relację dwuargumentową $R \subseteq X^2$ nazywamy *porządkiem* w zbiorze X, jeżeli jest

1. zwrotna: $\forall x \in X.xRx$
2. antysymetryczna: $\forall x, y \in X.xRy \wedge yRx \Rightarrow x = y$
3. przechodnia: $\forall x, y, z \in X.xRy \wedge yRz \Rightarrow xRz$

Parę $(X, \sqsubseteq)$, gdzie $\sqsubseteq$ jest porządkiem w X nazywamy zbiorem uporządkowanym. ∎

Definicja 8

1. Element $x \in X$ nazywamy *minimalnym (maksymalnym)*, jeżeli nie istnieje $y \in X$ różny od x taki, że $y \sqsubseteq x (x \sqsubseteq y)$.

2. $x \in X$ jest elementem *najmniejszym (największym)*, jeżeli dla każdego $y \in X$ $x \sqsubseteq y$ $(y \sqsubseteq x)$. Element najmniejszy będziemy oznaczać $\bot$.

3. Niech $Y \subseteq X$ i $(X, \sqsubseteq)$. $x \in X$ jest *ograniczeniem górnym (dolnym)* zbioru Y, jeżeli dla każdego $y \in Y$ zachodzi $y \sqsubseteq x (x \sqsubseteq y)$.

4. Jeżeli w zbiorze wszystkich ograniczeń górnych (dolnych) zbioru Y istnieje element najmniejszy (największy), to nazywamy go *kresem górnym Y (kresem dolnym Y)*. Oznaczamy go $\sqcup Y (\sqcap Y)$. ∎

Definicja 9

Niech $(X, \sqsubseteq)$ będzie zbiorem uporządkowanym. Porządek $(X, \sqsubseteq)$ nazywamy *zupełnym* (w skrócie **cpo**), jeżeli każdy ciąg wstępujący $x_1 \sqsubseteq_x x_2 \sqsubseteq_x \ldots$ elementów X posiada kres górny w X. Piszemy wtedy $\bigsqcup_{i=1}^{\infty} x_i$. Mówimy, że $(X, \sqsubseteq)$ jest porządkiem zupełnym z "pinezką" ($\perp$), jeśli cpo ma element najmniejszy $\perp_x$). ∎

Definicja 10

1. Funkcję $f : X \to Y$ nazywamy monotoniczną, jeżeli dla każdej pary $x_1, x_2 \in X$ warunek $x_1 \sqsubseteq_X x_2$ pociąga za sobą $f(x_1) \sqsubseteq_Y f(x_2)$.

2. Jeżeli $(X, \sqsubseteq_X)$ i $(Y, \sqsubseteq_Y)$ są porządkami zupełnymi, to funkcję f nazywamy ciągłą, jeżeli dla każdego ciągu wstępującego $x_1 \sqsubseteq_X x_2 \sqsubseteq_X \ldots$ elementów zbioru X

$$f(\bigsqcup_{i=1}^{\infty} x_i) = \bigsqcup_{i=1}^{\infty} f(x_i)$$

■

Definicja 11

Niech $H : D \to D$. $a \in D$ jest najmniejszym rozwiązaniem równania $x = H(x)$ (lub *najmniejszym punktem stałym* funkcji H), jeżeli $a = H(a)$ oraz dla każdego rozwiązania b, $a \sqsubseteq b$. Rozwiązanie to, jeżeli istnieje, będziemy oznaczać $fix(H)$. ∎

Twierdzenie 12

Jeżeli $(D, \sqsubseteq)$ jest porządkiem zupełnym z "pinezką", a H jest funkcją ciągłą, to $fix(H)$ istnieje oraz

$$fix(H) = \bigsqcup_{i=1}^{\infty} H^i(\bot)$$

∎

Wracamy do definicji semantyki. Dokonujemy następujących modyfikacji:

1. Funkcje częściowe na stanach zastępujemy całkowitymi funkcjami ciągłymi rozszerzając Σ o nowy element $\perp$. Otrzymujemy $\Sigma_\perp$ uporządkowany przez relację:

$$\forall \sigma . \perp \sqsubseteq \sigma,$$

gdzie $\perp$ – stan niezdefiniowany, tzn. wszystkie identyfikatory mają nieokreślone wartości. Zbiór $[\Sigma \to \Sigma]$ funkcji częściowych jest izomorficzny z $[\Sigma \to \Sigma_\perp]$ całkowitych funkcji ciągłych.

2. W konsekwencji rozszerzamy: N do $N_\perp$ oraz B do $B_\perp$, gdzie $\perp$ oznacza "brak wartości" .

3. Zmodyfikowane dziedziny semantyczne:

$$\mathcal{D}_{Aexp} = [\Sigma_\perp \to N_\perp]$$

$$\mathcal{D}_{Bexp} = [\Sigma_\perp \to B_\perp]$$

$$\mathcal{D}_{Com} = [\Sigma_\perp \to \Sigma_\perp]$$

Dla większej jasności symbol $\perp$ będziemy indeksować symbolem dziedziny, do której się odnosi: $\perp_N$, $\perp_B$, $\perp_\Sigma$.
$\perp_\Sigma = \lambda x \in Ide.\perp_N$.

Zastąpienie funkcji częściowych całkowitymi powoduje drobną modyfikację definicji.

Aexp

$$\mathcal{A}[[n]] = \lambda\sigma \in \Sigma.n$$

$$\mathcal{A}[[v]] = \lambda\sigma \in \Sigma.\sigma(v)$$

$$\mathcal{A}[[e_0 + e_1]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{A}[[e_0]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else}$$
$$\quad \textbf{if } \mathcal{A}[[e_1]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else } (\mathcal{A}[[e_0]]\sigma + \mathcal{A}[[e_1]]\sigma)$$

$$\mathcal{A}[[e_0 - e_1]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{A}[[e_0]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else}$$
$$\quad \textbf{if } \mathcal{A}[[e_1]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else } (\mathcal{A}[[e_0]]\sigma - \mathcal{A}[[e_1]]\sigma)$$

$$\mathcal{A}[[e_0 * e_1]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{A}[[e_0]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else}$$
$$\quad \textbf{if } \mathcal{A}[[e_1]]\sigma = \perp_N \textbf{ then } \perp_N \textbf{ else } (\mathcal{A}[[e_0]]\sigma * \mathcal{A}[[e_1]]\sigma)$$

Bexp

$$\mathcal{B}[[tt]] = \lambda\sigma \in \Sigma.tt$$

$$\mathcal{B}[[ff]] = \lambda\sigma \in \Sigma.ff$$

$$\mathcal{B}[[b_0 \vee b_1]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b_0]]\sigma = \perp_B \textbf{ then } \perp_B \textbf{ else}$$
$$\textbf{if } \mathcal{B}[[b_1]]\sigma = \perp_B \textbf{ then } \perp_B \textbf{ else if } \mathcal{B}[[b_0]]\sigma = tt \textbf{ then } tt$$
$$\textbf{else if } \mathcal{B}[[b_1]]\sigma = tt \textbf{ then } tt \textbf{ else } ff$$

$$\mathcal{B}[[e_0 = e_1]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{A}[[e_0]]\sigma = \perp_N \textbf{ then } \perp_B \textbf{ else}$$
$$\textbf{if } \mathcal{A}[[e_1]]\sigma = \perp_N \textbf{ then } \perp_B$$
$$\textbf{else if } \mathcal{A}[[e_0]]\sigma = \mathcal{A}[[e_1]]\sigma \textbf{ then } tt \textbf{ else } ff$$

$$\mathcal{B}[[\neg b]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b_0]]\sigma = \perp_B \textbf{ then } \perp_B \textbf{ else}$$
$$\textbf{if } \mathcal{B}[[b]]\sigma = tt \textbf{ then } ff \textbf{ else } tt.$$

Com

$$\mathcal{C}[[\underline{\text{skip}}]] = \lambda\sigma \in \Sigma.\sigma$$

$$\mathcal{C}[[v := e]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{A}[[e]]\sigma = \perp_N \textbf{ then } \perp_\Sigma \textbf{ else } \sigma[\mathcal{A}[[e]]\sigma/v]$$

$$\mathcal{C}[[c_1; c_2]] = \lambda\sigma \in \Sigma.\mathcal{C}[[c_2]](\mathcal{C}[[c_1]]\sigma)$$

$$\mathcal{C}[[\underline{\text{if}}\ b\ \underline{\text{then}}\ c_1\ \underline{\text{else}}\ c_2]] = \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b]]\sigma = \perp_B \textbf{ then } \perp_\Sigma \textbf{ else if } \mathcal{B}[[b]]\sigma = tt \textbf{ then } \mathcal{C}[[c_1]]\sigma \textbf{ else } \mathcal{C}[[c_2]]\sigma$$

Wróćmy do instrukcji $\underline{\text{while}}\ b\ \underline{\text{do}}\ c$.

Zgodnie z twierdzeniem 12 mamy:

$$\mathcal{C}[[\underline{\text{while}}\ b\ \underline{\text{do}}\ c]] = \bigsqcup_{i=1}^{\infty} f^i(\bot),$$

gdzie $f = \mathcal{F}(\mathcal{B}[[b]], \mathcal{C}[[c]]) : \mathcal{D}_{Com} \to \mathcal{D}_{Com}$,
a $\mathcal{D}_{Com} = [\Sigma_\bot \to \Sigma_\bot]$

$\bot \in \mathcal{D}_{Com}$ jest zdefiniowana jako $\lambda\sigma \in \Sigma.\bot_\Sigma$.

Obliczymy 2 kolejne aproksymacje $\mathcal{C}[[\underline{\text{while}}\ b\ \underline{\text{do}}\ c]]$.

$$
\begin{aligned}
f(\bot) &= \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b]]\sigma = \bot_B \textbf{ then } \bot_\Sigma \textbf{ else} \\
&\quad \textbf{if } \mathcal{B}[[b]]\sigma = ff \textbf{ then } \sigma \textbf{ else } \bot(\mathcal{C}[[c]]\sigma) \\
&= \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b]]\sigma = \bot_B \textbf{ then } \bot_\Sigma \textbf{ else} \\
&\quad \textbf{if } \mathcal{B}[[b]]\sigma = ff \textbf{ then } \sigma \textbf{ else } \bot_\Sigma.
\end{aligned}
$$

$$
\begin{aligned}
f^2(\bot) &= \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b]]\sigma = \bot_B \textbf{ then } \bot_\Sigma \textbf{ else} \\
&\quad \textbf{if } \mathcal{B}[[b]]\sigma = ff \textbf{ then } \sigma \textbf{ else } f(\bot)(\mathcal{C}[[c]]\sigma) \\
&= \lambda\sigma \in \Sigma.\textbf{if } \mathcal{B}[[b]]\sigma = \bot_B \textbf{ then } \bot_\Sigma \textbf{ else} \\
&\quad \textbf{if } \mathcal{B}[[b]]\sigma = ff \textbf{ then } \sigma \textbf{ else} \\
&\quad \textbf{if } \mathcal{B}[[b]](\mathcal{C}[[c]]\sigma) = ff \textbf{ then } \mathcal{C}[[c]]\sigma \textbf{ else } \bot_\Sigma.
\end{aligned}
$$

Równoważność semantyki operacyjnej i denotacyjnej dla PJP

Twierdzenie 13

1. Dla dowolnego wyrażenia $e \in Aexp$

$$\mathcal{A}[[e]] = \{(\sigma, n) \mid \langle e, \sigma \rangle \rightsquigarrow^* \langle n, \sigma \rangle\}$$

2. Dla dowolnego $b \in Bexp$

$$\mathcal{B}[[b]] = \{(\sigma, t) \mid \langle b, \sigma \rangle \rightsquigarrow^* \langle t, \sigma \rangle\}$$

3. Dla każdej instrukcji $c \in Com$

$$\mathcal{C}[[c]] = \{(\sigma, \sigma') \mid \langle c, \sigma \rangle \rightsquigarrow^* \langle \epsilon, \sigma' \rangle\}$$

■

Przykład 3

Zdefiniować prosty język funkcyjny (bez rekursji).

Składnia

- $n ::=$ stała
- $v ::=$ identyfikator zmiennych
- $f ::=$ identyfikator funkcji
- $e ::= n \mid v \mid e_1 + e_2 \mid f\,e \mid let\ f = \lambda v.e\ in\ e_1$

Dziedziny syntaktyczne

- $Ide = VIde + FIde$
- $Aexp$

Dziedziny semantyczne

- $N = \{\ldots, -1, 0, 1, 2, \ldots\}$
- $\Sigma = [VIde \to N]$
- $F\Sigma = [FIde \to [N \to N]]$

Funkcja semantyczna

- $\mathcal{A} : Aexp \to [F\Sigma \to [\Sigma \to N]]$

Równania semantyczne

- $\mathcal{A}[[n]] = \lambda\tau \in F\Sigma.\lambda\sigma \in \Sigma.n$
- $\mathcal{A}[[v]] = \lambda\tau \in F\Sigma.\lambda\sigma \in \Sigma.\sigma(v)$
- $\mathcal{A}[[e_1 + e_2]] = \lambda\tau \in F\Sigma.\lambda\sigma \in \Sigma.\mathcal{A}[[e_1]]\tau\sigma + \mathcal{A}[[e_2]]\tau\sigma$
- $\mathcal{A}[[f\,e]] = \lambda\tau \in F\Sigma.\lambda\sigma \in \Sigma.\tau(f)(\mathcal{A}[[e]]\tau\sigma)$
- $\mathcal{A}[[let\ f = \lambda v.e\ in\ e_1]] = \lambda\tau \in F\Sigma.\lambda\sigma \in \Sigma.$
  $\mathcal{A}[[e_1]](\tau[\lambda x \in N.\mathcal{A}[[e]]\tau(\sigma[x/v])/f])\sigma$ ∎