While-programy

Składnia[1]

Wyrażenia logiczne (B) : $\mathit{false},\ \mathit{true},\ E_1 = E_2,\ \neg B,\ B_1 \wedge B_2,\ B_1 \vee B_2.$

Wyrażenia arytmetyczne (E) : $\underline{n},\ x,\ E_1 + E_2,\ E_1 * E_2,\ -E.$

Programy (P) : $\mathit{skip},\ \mathit{loop},\ x := E,\ \texttt{begin}\ P_1; P_2\ \texttt{end},\ \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2,\ \texttt{while}\ B\ \texttt{do}\ P_1.$

Semantyka operacyjna małych kroków

Stan to funkcja s, która zmiennym przypisuje wartości całkowite. Definiujemy trzy relacje:

$$\langle B, s\rangle \to \langle B', s\rangle, \qquad \langle E, s\rangle \to \langle E', s\rangle, \qquad \langle P, s\rangle \to \langle P', s'\rangle.$$

Postaci normalne: $\langle \mathit{skip}, s\rangle$, $\langle \underline{n}, s\rangle$, $\langle \mathit{false}, s\rangle$, $\langle \mathit{true}, s\rangle$.

Aksjomaty i reguły dla wyrażeń:[2]

$$\langle \underline{n} + \underline{m}, s\rangle \to \langle \underline{n+m}, s\rangle, \quad \langle \underline{n} * \underline{m}, s\rangle \to \langle \underline{n \cdot m}, s\rangle, \quad \langle -\underline{n}, s\rangle \to \langle \underline{-n}, s\rangle, \quad \langle x, s\rangle \to \langle \underline{s(x)}, s\rangle,$$

$$\langle \underline{n} = \underline{n}, s\rangle \to \langle \mathit{true}, s\rangle, \quad \langle \underline{m} = \underline{n}, s\rangle \to \langle \mathit{false}, s\rangle, \text{ gdy } m \neq n.$$

$$\langle \mathit{true} \wedge \mathit{true}, s\rangle \to \langle \mathit{true}, s\rangle, \quad \langle \mathit{true} \wedge \mathit{false}, s\rangle \to \langle \mathit{false}, s\rangle, \text{ etc.}^3$$

$$\frac{\langle B_1, s\rangle \to \langle B_1', s\rangle}{\langle B_1 \wedge B_2, s\rangle \to \langle B_1' \wedge B_2, s\rangle} \qquad \frac{\langle B_2, s\rangle \to \langle B_2', s\rangle}{\langle \mathit{true} \wedge B_2, s\rangle \to \langle \mathit{true} \wedge B_2', s\rangle} \qquad \frac{\langle B_2, s\rangle \to \langle B_2', s\rangle}{\langle \mathit{false} \wedge B_2, s\rangle \to \langle \mathit{false} \wedge B_2', s\rangle}$$

$$\frac{\langle E_1, s\rangle \to \langle E_1', s\rangle}{\langle E_1 = E_2, s\rangle \to \langle E_1' = E_2, s\rangle} \qquad \frac{\langle E_2, s\rangle \to \langle E_2', s\rangle}{\langle \underline{n} = E_2, s\rangle \to \langle \underline{n} = E_2', s\rangle}$$

$$\frac{\langle E_1, s\rangle \to \langle E_1', s\rangle}{\langle E_1 + E_2, s\rangle \to \langle E_1' + E_2, s\rangle} \qquad \frac{\langle E_2, s\rangle \to \langle E_2', s\rangle}{\langle \underline{n} + E_2, s\rangle \to \langle \underline{n} + E_2', s\rangle}$$

Aksjomaty i reguły dla programów:

$$\langle \mathit{loop}, s\rangle \to \langle \mathit{loop}, s\rangle \qquad \langle x := \underline{n}, s\rangle \to \langle \mathit{skip}, s[x \mapsto n]\rangle \qquad \frac{\langle E, s\rangle \to \langle E', s\rangle}{\langle x := E, s\rangle \to \langle x := E', s\rangle}$$

$$\frac{\langle P_1, s\rangle \to \langle P_1', s'\rangle}{\langle \texttt{begin}\ P_1; P_2\ \texttt{end}, s\rangle \to \langle \texttt{begin}\ P_1'; P_2\ \texttt{end}, s'\rangle} \qquad \langle \texttt{begin}\ \mathit{skip}; P\ \texttt{end}, s\rangle \to \langle P, s\rangle$$

$$\frac{\langle B, s\rangle \to \langle B', s\rangle}{\langle \texttt{if}\ B\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle \to \langle \texttt{if}\ B'\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle}$$

$$\langle \texttt{if}\ \mathit{true}\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle \to \langle P_1, s\rangle \qquad \langle \texttt{if}\ \mathit{false}\ \texttt{then}\ P_1\ \texttt{else}\ P_2, s\rangle \to \langle P_2, s\rangle$$

$$\langle \texttt{while}\ B\ \texttt{do}\ P, s\rangle \to \langle \texttt{if}\ B\ \texttt{then}\ \texttt{begin}\ P; \texttt{while}\ B\ \texttt{do}\ P\ \texttt{end}\ \texttt{else}\ \mathit{skip}, s\rangle$$

[1] Nawiasowanie jest domyślne.

[2] Pominięto reguły dla alternatywy, negacji, odejmowania i mnożenia.

[3] Możliwy jest też wybór aksjomatów i reguł *non-strict* np. $\langle \underline{0} * E, s\rangle \to \langle \underline{0}, s\rangle$, $\langle \mathit{false} \wedge B, s\rangle \to \langle \mathit{false}, s\rangle$, etc.

Semantyka naturalna (dużych kroków)

Definiujemy relacje $\langle B, s\rangle \Downarrow b$, $\langle E, s\rangle \Downarrow \underline{n}$ $\langle P, s\rangle \Downarrow s'$, gdzie $b \in \{\mathit{true}, \mathit{false}\}$ a s jest stanem.

Wyrażenia: $\langle \underline{n}, s\rangle \Downarrow \underline{n}$ $\quad \langle \mathit{true}, s\rangle \Downarrow \mathit{true}$ $\quad \langle \mathit{false}, s\rangle \Downarrow \mathit{false}$ $\quad \langle x, s\rangle \Downarrow \underline{s(x)}$

$$\langle \underline{n} = \underline{n}, s\rangle \Downarrow \mathit{true} \qquad \langle \underline{m} = \underline{n}, s\rangle \Downarrow \mathit{false}$$

$$\frac{\langle E_1, s\rangle \Downarrow \underline{n} \quad \langle E_2, s\rangle \Downarrow \underline{m}}{\langle E_1 + E_2, s\rangle \Downarrow \underline{n+m}} \qquad \frac{\langle E_1, s\rangle \Downarrow \underline{n} \quad \langle E_2, s\rangle \Downarrow \underline{m}}{\langle E_1 \cdot E_2, s\rangle \Downarrow \underline{n \cdot m}} \qquad \frac{\langle E_1, s\rangle \Downarrow \underline{n}}{\langle -E_1, s\rangle \Downarrow \underline{-n}}$$

$$\frac{\langle B_1, s\rangle \Downarrow \mathit{true} \quad \langle B_2, s\rangle \Downarrow \mathit{true}}{\langle B_1 \wedge B_2, s\rangle \Downarrow \mathit{true}} \qquad \frac{\langle B_1, s\rangle \Downarrow \mathit{true} \quad \langle B_2, s\rangle \Downarrow \mathit{false}}{\langle B_1 \wedge B_2, s\rangle \Downarrow \mathit{false}} \qquad \text{i tak dalej}$$

Programy:

$$\langle \mathit{skip}, s\rangle \Downarrow s \qquad \frac{\langle E, s\rangle \Downarrow \underline{n}}{\langle x := E, s\rangle \Downarrow s[x \mapsto n]} \qquad \frac{\langle P_1, s\rangle \Downarrow s' \quad \langle P_2, s'\rangle \Downarrow s''}{\langle \mathtt{begin}\ P_1; P_2\ \mathtt{end}, s\rangle \Downarrow s''}$$

$$\frac{\langle B, s\rangle \Downarrow \mathit{true} \quad \langle P_1, s\rangle \Downarrow s'}{\langle \mathtt{if}\ B\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2, s\rangle \Downarrow s'} \qquad \frac{\langle B, s\rangle \Downarrow \mathit{false} \quad \langle P_2, s\rangle \Downarrow s'}{\langle \mathtt{if}\ B\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2, s\rangle \Downarrow s'}$$

$$\frac{\langle B, s\rangle \Downarrow \mathit{true} \quad \langle P, s\rangle \Downarrow s' \quad \langle \mathtt{while}\ B\ \mathtt{do}\ P, s'\rangle \Downarrow s''}{\langle \mathtt{while}\ B\ \mathtt{do}\ P, s\rangle \Downarrow s''} \qquad \frac{\langle B, s\rangle \Downarrow \mathit{false}}{\langle \mathtt{while}\ B\ \mathtt{do}\ P, s\rangle \Downarrow s}$$

Maszyna abstrakcyjna SMC

Konfiguracja maszyny to stan, lub trójka $\langle r, s, c\rangle$ składająca się ze *stosu* r, stanu s i *kodu* c. Stos jest listą zmiennych, programów, wartości liczbowych i logicznych. Kod jest listą wyrażeń, programów i operatorów ze zbioru $\{+, *, -, \wedge, \vee, \neg, \text{asg}, \text{if}, \text{while}\}$. Konfiguracja początkowa ma postać $\langle \mathit{nil}, s, P\rangle$. Definiujemy relację przejścia $\to$ pomiędzy konfiguracjami.

$$\langle r,\ s,\ \mathit{true}\!:\!c\rangle \to \langle \mathit{true}\!:\!r,\ s,\ c\rangle \qquad \langle r,\ s,\ \mathit{false}\!:\!c\rangle \to \langle \mathit{false}\!:\!r,\ s,\ c\rangle$$

$$\langle r,\ s,\ \underline{n}\!:\!c\rangle \to \langle \underline{n}\!:\!s,\ r,\ c\rangle \qquad \langle r,\ s,\ x\!:\!c\rangle \to \langle s(x)\!:\!r,\ s,\ c\rangle$$

$$\langle r,\ s,\ E_1 + E_2\!:\!c\rangle \to \langle r,\ s,\ E_1\!:\!E_2\!:\ +\ :\!c\rangle \qquad \langle r,\ s,\ B_1 \wedge B_2\!:\!c\rangle \to \langle r,\ s,\ B_1\!:\!B_2\!:\ \wedge\ :\!c\rangle \quad \text{etc.}$$

$$\langle \underline{n} : \underline{m} : r,\ s,\ + : c\rangle \to \langle m + n : r,\ s,\ c\rangle \qquad \langle \mathit{true} : \mathit{false} : r,\ s,\ \wedge : c\rangle \to \langle \mathit{false} : r,\ s,\ c\rangle \quad \text{etc.}$$

$$\langle r,\ s,\ \mathit{loop}\!:\!c\rangle \to \langle r,\ s,\ \mathit{loop}\!:\!c\rangle \qquad \langle r,\ s,\ \mathit{skip}\!:\!c\rangle \to \langle r,\ s,\ c\rangle$$

$$\langle r,\ s,\ x\!:=\!E\!:\!c\rangle \to \langle x\!:\!r,\ s,\ E\!:\!\text{asg}\!:\!c\rangle \qquad \langle r,\ s,\ \mathtt{begin}\ P_1; P_2\ \mathtt{end}\!:\!c\rangle \to \langle r,\ s,\ P_1\!:\!P_2\!:\!c\rangle$$

$$\langle r,\ s,\ \mathtt{if}\ B\ \mathtt{then}\ P_1\ \mathtt{else}\ P_2\!:\!c\rangle \to \langle P_1\!:\!P_2\!:\!r,\ s,\ B\!:\!\text{if}\!:\!c\rangle$$

$$\langle r,\ s,\ \mathtt{while}\ B\ \mathtt{do}\ P\!:\!c\rangle \to \langle B\!:\!P\!:\!r,\ s,\ B\!:\!\text{while}\!:\!c\rangle$$

$$\langle \underline{n}\!:\!x\!:\!r,\ s,\ \text{asg}\!:\!c\rangle \to \langle r,\ s[x \mapsto n],\ c\rangle$$

$$\langle \mathit{true}\!:\!P_1\!:\!P_2\!:\!r,\ s,\ \text{if}\!:\!c\rangle \to \langle r,\ s,\ P_1\!:\!c\rangle \qquad \langle \mathit{false}\!:\!P_1\!:\!P_2\!:\!r,\ s,\ \text{if}\!:\!c\rangle \to \langle r,\ s,\ P_2\!:\!c\rangle$$

$$\langle \mathit{true}\!:\!B\!:\!P\!:\!r,\ s,\ \text{while}\!:\!c\rangle \to \langle r,\ s,\ P\!:\!\mathtt{while}\ B\ \mathtt{do}\ P\!:\!c\rangle$$

$$\langle \mathit{false}\!:\!B\!:\!P\!:\!r,\ s,\ \text{while}\!:\!c\rangle \to \langle r,\ s,\ c\rangle$$

$$\langle r,\ s,\ \mathit{nil}\rangle \to s$$

Semantyka denotacyjna

Niech $\mathcal{S}$ będzie zbiorem wszystkich możliwych stanów. Określamy znaczenie wyrażeń logicznych $[\![B]\!] : \mathcal{S} \to \{0,1\}$ i wyrażeń arytmetycznych $[\![E]\!] : \mathcal{S} \to \mathbb{Z}$, oraz znaczenie programów $[\![P]\!] : \mathcal{S} \multimap\!\!\to \mathcal{S}$, gdzie symbol $\multimap\!\!\to$ oznacza funkcję częściową. Zamiast $[\![X]\!](s)$ piszemy $[\![X]\!]_s$.

$$[\![x]\!]_s = s(x) \qquad [\![\underline{n}]\!]_s = n \qquad [\![\mathit{true}]\!]_s = 1 \qquad [\![\mathit{false}]\!]_s = 0$$

$$[\![E_1 + E_2]\!]_s = [\![E_1]\!]_s + [\![E_2]\!]_s \qquad [\![B_1 \wedge B_2]\!]_s = \min\{[\![B_1]\!]_s, [\![B_2]\!]_s\} \qquad \text{i tak dalej.}$$

$$[\![\mathit{skip}]\!] = \textsc{id} \qquad [\![\mathit{loop}]\!] = \lambda s.\,\bot \qquad [\![x := E]\!] = \lambda s.\, s[x \mapsto [\![E]\!]_s]$$

$$[\![\texttt{begin } P_1; P_2 \texttt{ end}]\!] = [\![P_2]\!] \circ [\![P_1]\!] \qquad [\![\texttt{if } B \texttt{ then } P_1 \texttt{ else } P_2]\!] = \textsc{cond}([\![B]\!], [\![P_1]\!], [\![P_2]\!])$$

$$[\![\texttt{while } B \texttt{ do } P]\!] = \textsc{lfp}(\mathcal{F}), \text{ gdzie } \mathcal{F}(\varphi) = \textsc{cond}([\![B]\!], \varphi \circ [\![P]\!], \textsc{id}).$$

Definicje: Jeśli φ i ψ są funkcjami częściowymi, to $\varphi \sqsubseteq \psi$ oznacza, że $\mathrm{Dom}(\varphi) \subseteq \mathrm{Dom}(\psi)$ oraz $\varphi(x) = \psi(x)$ dla wszystkich $x \in \mathrm{Dom}(\varphi)$. Napis $\varphi(x) = \bot$ oznacza, że $x \notin \mathrm{Dom}(\varphi)$, tj. $\varphi(x)$ jest nieokreślone. Złożenie funkcji częściowych jest „ścisłe":

$$g \circ f(x) = \begin{cases} g(f(x)), & \text{jeśli } f(x) \text{ jest określone;} \\ \text{nieokreślone}, & \text{w przeciwnym przypadku.} \end{cases}$$

Symbol $\textsc{id}$ oznacza funkcję identycznościową, a symbol $\textsc{cond}$ oznacza operator warunkowy:

$$\textsc{cond}(f, g, h)(x) = \begin{cases} g(x), & \text{jeśli } f(x) = 1; \\ h(x), & \text{jeśli } f(x) = 0. \end{cases}$$

Jeśli F jest operatorem działającym na funkcjach częściowych, to symbol $\textsc{lfp}(F)$ oznacza najmniejszą (ze względu na $\sqsubseteq$) funkcję częściową φ o własności $F(\varphi) = \varphi$. A zatem:

$$F(\textsc{lfp}(F)) = \textsc{lfp}(F) \qquad \text{oraz} \qquad \textit{jeśli } F(\varphi) = \varphi \textit{ to } \textsc{lfp}(F) \sqsubseteq \varphi.$$

Semantyka aksjomatyczna: reguły Hoare'a

$$\{\varphi\}\, \mathit{skip}\, \{\varphi\} \qquad \{\varphi[E/x]\}\, x := E\, \{\varphi\} \qquad \{\mathit{true}\}\, \mathit{loop}\, \{\mathit{false}\}$$

$$\frac{\{\varphi\}\, P_1\, \{\psi\} \quad \{\psi\}\, P_2\, \{\vartheta\}}{\{\varphi\}\, \texttt{begin } P_1; P_2 \texttt{ end}\, \{\vartheta\}}$$

$$\frac{\{\varphi \wedge B\}\, P_1\, \{\psi\} \quad \{\varphi \wedge \neg B\}\, P_1\, \{\psi\}}{\{\varphi\}\, \texttt{if } B \texttt{ then } P_1 \texttt{ else } P_2\, \{\psi\}}$$

$$\frac{\{\varphi \wedge B\}\, P\, \{\varphi\}}{\{\varphi\}\, \texttt{while } B \texttt{ do } P\, \{\varphi \wedge \neg B\}}$$

$$\frac{\varphi' \Rightarrow \varphi \quad \{\varphi\}\, P\, \{\psi\} \quad \psi \Rightarrow \psi'}{\{\varphi'\}\, P\, \{\psi'\}}$$