1 Składnia języka IMP

1.1 Dane zbiory

- **N** – zbiór liczb całkowitych (utożsamiamy liczby i ich przedstawienia),
- **Loc** – zbiór lokacji (elementy tego zbioru odgrywają rolę nazw zmiennych i samych zmiennych lub adresów zmiennych),

1.2 Konwencje notacyjne

Symbolem Σ oznaczamy zbiór $\{\sigma : \ \sigma : \mathbf{Loc} \to \mathbf{N}\}$.

Będziemy zakładać, że elementy poszczególnych zbiorów będą oznaczane odpowiednimi literami zgodnie z niżej podanymi ustaleniami:

- $n, m, p \in \mathbf{N}$,
- $X, Y \in \mathbf{Loc}$,
- $a, a_0, a_1, a' \in \mathbf{Aexp}$,
- $b, b_0, b_1 \in \mathbf{Bexp}$,
- $c, c_0, c_1 \in \mathbf{Com}$
- $t, t_0, t_1 \in \{\mathbf{true}, \mathbf{false}\}$
- $\sigma, \sigma', \sigma" \in \Sigma$

1.3 Reguły definiujące składnię

- Reguła definiująca **Aexp**:

$$a ::= n \mid X \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 * a_1$$

- Reguła definiująca **Bexp**:

$$b ::= \mathbf{true} \mid \mathbf{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \neg b \mid b_0 \wedge b_1 \mid b_0 \vee b_1$$

- Reguła definiująca **Com**:

$$c ::= \mathbf{skip} \mid X := a \mid c_0; c_1 \mid \ \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1 \mid \ \mathbf{while}\ b\ \mathbf{do}\ c$$

2 Semantyka operacyjna języka IMP

2.1 Reguły definiujące wartość wyrażenia arytmetycznego

$$\frac{}{(n, \sigma) \to n}$$

$$\frac{}{(X, \sigma) \to \sigma(X)}$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 + a_1, \sigma) \to n} \quad \text{gdzie } n \text{ jest sumą } n_0 \text{ i } n_1,$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 - a_1, \sigma) \to n} \quad \text{gdzie } n \text{ jest różnicą } n_0 \text{ i } n_1,$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 * a_1, \sigma) \to n} \quad \text{gdzie } n \text{ jest iloczynem } n_0 \text{ i } n_1.$$

2.2 Reguły definiujące wartość wyrażenia boolowskiego

$$\frac{}{(\mathbf{true}, \sigma) \to \mathbf{true}}$$

$$\frac{}{(\mathbf{false}, \sigma) \to \mathbf{false}}$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 = a_1, \sigma) \to \mathbf{true}} \quad \text{w przypadku, gdy liczby } n_0 \text{ i } n_1 \text{ są równe,}$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 = a_1, \sigma) \to \mathbf{false}} \quad \text{w przypadku, gdy liczby } n_0 \text{ i } n_1 \text{ są różne,}$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 \leq a_1, \sigma) \to \mathbf{true}} \quad \text{w przypadku, gdy liczba } n_0 \text{ jest mniejsza lub równa } n_1,$$

$$\frac{(a_0, \sigma) \to n_0,\ (a_1, \sigma) \to n_1}{(a_0 \le a_1, \sigma) \to \mathbf{false}}$$ w przypadku, gdy liczba n_0 jest większa od n_1,

$$\frac{(b, \sigma) \to \mathbf{true}}{(\neg b, \sigma) \to \mathbf{false}}$$

$$\frac{(b, \sigma) \to \mathbf{false}}{(\neg b, \sigma) \to \mathbf{true}}$$

$$\frac{(b_0, \sigma) \to t_0,\ (b_1, \sigma) \to t_1}{(b_0 \wedge b_1, \sigma) \to \mathbf{true}}$$ w przypadku, gdy t_0 i t_1 są równe **true**,

$$\frac{(b_0, \sigma) \to t_0,\ (b_1, \sigma) \to t_1}{(b_0 \wedge b_1, \sigma) \to \mathbf{false}}$$ w przypadku, gdy t_0 lub t_1 jest równa **false**,

$$\frac{(b_0, \sigma) \to t_0,\ (b_1, \sigma) \to t_1}{(b_0 \vee b_1, \sigma) \to \mathbf{true}}$$ w przypadku, gdy t_0 lub t_1 jest równa **true**,

$$\frac{(b_0, \sigma) \to t_0,\ (b_1, \sigma) \to t_1}{(b_0 \vee b_1, \sigma) \to \mathbf{false}}$$ w przypadku, gdy t_0 i t_1 są równe **false**

2.3 Reguły definiujące znaczenie instrukcji

$$\frac{}{(\mathbf{skip}, \sigma) \to \sigma}$$

$$\frac{(a, \sigma) \to n}{(X := a, \sigma) \to \sigma[n/X]}$$

$$\frac{(c_0, \sigma_0) \to \sigma_1,\ (c_1, \sigma_1) \to \sigma_2}{(c_0; c_1, \sigma_0) \to \sigma_2}$$

$$\frac{(b, \sigma_0) \to \mathbf{true},\ (c_0, \sigma_0) \to \sigma_1}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \sigma_0) \to \sigma_1}$$

$$\frac{(b,\sigma_0)\to\mathbf{false},\ (c_1,\sigma_0)\to\sigma_1}{(\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1,\sigma_0)\to\sigma_1}$$

$$\frac{(b,\sigma_0)\to\mathbf{false}}{(\mathbf{while}\ b\ \mathbf{do}\ c,\sigma_0)\to\sigma_0}$$

$$\frac{(b,\sigma_0)\to\mathbf{true},\ (c,\sigma_0)\to\sigma_1,\ (\mathbf{while}\ b\ \mathbf{do}\ c,\sigma_1)\to\sigma_2}{(\mathbf{while}\ b\ \mathbf{do}\ c,\sigma_0)\to\sigma_2}$$

3 Semantyka denotacyjna języka IMP

Semantyka denotacyjna IMP składa się z trzech funkcji:

$\mathcal{A}:\mathbf{Aexp}\to(\Sigma\to\mathbf{N})$,

$\mathcal{B}:\mathbf{Bexp}\to(\Sigma\to\{\mathbf{true},\mathbf{false}\})$,

$\mathcal{C}:\mathbf{Com}\to(\Sigma\to\Sigma)$

3.1 Definicja funkcji $\mathcal{A}$

$$\mathcal{A}[\![a]\!]\sigma=\begin{cases} n & \text{jeżeli } a=n\in\mathbf{N},\\ \sigma(X) & \text{jeżeli } a=X\in\mathbf{Loc},\\ \mathcal{A}[\![a_0]\!]\sigma+\mathcal{A}[\![a_1]\!]\sigma & \text{jeżeli } a=a_0+a_1,\\ \mathcal{A}[\![a_0]\!]\sigma-\mathcal{A}[\![a_1]\!]\sigma & \text{jeżeli } a=a_0-a_1,\\ \mathcal{A}[\![a_0]\!]\sigma*\mathcal{A}[\![a_1]\!]\sigma & \text{jeżeli } a=a_0*a_1.\end{cases}$$

3.2 Definicja funkcji $\mathcal{B}$

$$
\mathcal{B}\llbracket b \rrbracket\sigma = \begin{cases}
\mathbf{true} & \text{jeżeli } b = \mathbf{true}, \\
\mathbf{false} & \text{jeżeli } b = \mathbf{false}, \\
\mathbf{true} & \text{jeżeli } b = a_0 = a_1 \text{ i } \mathcal{A}\llbracket a_0 \rrbracket\sigma = \mathcal{A}\llbracket a_1 \rrbracket\sigma, \\
\mathbf{false} & \text{jeżeli } b = a_0 = a_1 \text{ i } \mathcal{A}\llbracket a_0 \rrbracket\sigma \neq \mathcal{A}\llbracket a_1 \rrbracket\sigma, \\
\mathbf{true} & \text{jeżeli } b = a_{\leq} a_1 \text{ i } \mathcal{A}\llbracket a_0 \rrbracket\sigma \leq \mathcal{A}\llbracket a_1 \rrbracket\sigma, \\
\mathbf{false} & \text{jeżeli } b = a_0 \leq a_1 \text{ i } \mathcal{A}\llbracket a_0 \rrbracket\sigma > \mathcal{A}\llbracket a_1 \rrbracket\sigma, \\
\neg\mathcal{B}\llbracket b_0 \rrbracket\sigma & \text{jeżeli } b = \neg b_0, \\
\mathcal{B}\llbracket b_0 \rrbracket\sigma \wedge \mathcal{B}\llbracket b_1 \rrbracket\sigma & \text{jeżeli } b = b_0 \wedge b_1, \\
\mathcal{B}\llbracket b_0 \rrbracket\sigma \vee \mathcal{B}\llbracket b_1 \rrbracket\sigma & \text{jeżeli } b = b_0 \vee b_1,
\end{cases}
$$

3.3 Definicja funkcji $\mathcal{C}$

Funkcję $\mathcal{C}$ definiujemy jako funkcję

$$\mathcal{C} : \mathbf{Com} \to 2^{\Sigma \times \Sigma}$$

(a więc wartości $\mathcal{C}$ nie są funkcjami, ale ich wykresami).

$$
\mathcal{C}\llbracket c \rrbracket = \begin{cases}
\{(\sigma, \sigma) : \sigma \in \Sigma\} & \text{jeżeli } c = \mathbf{skip}, \\
\{(\sigma, \sigma[\mathcal{A}\llbracket a \rrbracket\sigma / X]) : \sigma \in \Sigma\} & \text{jeżeli } c = X := a, \\
\mathcal{C}\llbracket c_1 \rrbracket \circ \mathcal{C}\llbracket c_0 \rrbracket & \text{jeżeli } c = c_0; c_1, \\
\{(\sigma, \sigma') \in \mathcal{C}\llbracket c_0 \rrbracket : \mathcal{B}\llbracket b \rrbracket\sigma = \mathbf{true}\} & \\
\quad \cup \{(\sigma, \sigma') \in \mathcal{C}\llbracket c_1 \rrbracket : \mathcal{B}\llbracket b \rrbracket\sigma = \mathbf{false}\} & \text{jeżeli } c = \ \mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1, \\
\mathbf{fix}(\Gamma) & \text{jeżeli } c = \ \mathbf{while}\ b\ \mathbf{do}\ c.
\end{cases}
$$

gdzie

$$\Gamma : 2^{\Sigma \times \Sigma} \to 2^{\Sigma \times \Sigma}$$

oznacza operację, która relacji R w zbiorze Σ przyporządkowuje inną relację w tym zbiorze, zdefiniowaną wzorem

$$\Gamma(R) = \{(\sigma,\sigma) : \mathcal{B}[\![b]\!]\sigma = \mathbf{false}\} \cup \{(\sigma,\sigma') \in R \circ \mathcal{C}[\![c]\!] : \mathcal{B}[\![b]\!]\sigma = \mathbf{true}\}.$$

Operacja Γ jest monotoniczna (niemalejąca) i ciągła. Istnieje więc najmniejszy punkt stały Γ. Ten punkt stały jest oznaczany symbolem $\mathbf{fix}(\Gamma)$ i może zostać zdefiniowany wzorem

$$\mathbf{fix}(\Gamma) = \bigcup_{n=0}^{\infty} \Gamma^n(\emptyset).$$

Może też zostać zdefiniowany jako najmniejszy podzbiór $\Sigma \times \Sigma$ zamknięty ze względu na następujące reguły:

$$\frac{}{(\sigma,\sigma)}$$

dla $\sigma \in \Sigma$ takich, że $\mathcal{B}[\![b]\!]\sigma = \mathbf{false}$ oraz

$$\frac{(\sigma,\sigma") \in \mathcal{C}[\![c]\!],\ (\sigma",\sigma')}{(\sigma,\sigma')}$$

dla $\sigma \in \Sigma$ takich, że $\mathcal{B}[\![b]\!]\sigma = \mathbf{true}$

4 Semantyka aksjomatyczna

4.1 Rozszerzenie składni

Bierzemy zbiór **Intvar**. Przyjmujemy, że

- $i, j \in$ **Intvar**.

Definiujemy nowe zbiory wyrażeń: **Aexpv** (zbiór wyrażeń ze zmiennymi, elementy tego zbioru oznaczamy takimi samymi symbolami jak elementy **Aexp**) i **Assn** (zbiór asercji). Przyjmujemy, że

- $A, A_0, A_1, B \in$ **Assn**.

$$a ::= n \mid X \mid i \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 * a_1$$

$$A ::= \mathbf{true} \mid \mathbf{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid \neg A \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \exists i.A \mid \forall i.A$$

4.2 Rozszerzenie semantyki denotacyjnej

Zdefiniujemy dodatkową funkcję

$\mathcal{A}v : \mathbf{Aexpv} \to ((\mathbf{Intvar} \to \mathbf{N}) \to (\Sigma \to \mathbf{N})),$

$$\mathcal{A}v[\![a]\!]I\sigma = \begin{cases} n & \text{jeżeli } a = n \in \mathbf{N}, \\ \sigma(X) & \text{jeżeli } a = X \in \mathbf{Loc}, \\ I(i) & \text{jeżeli } a = i \in \mathbf{Intvar}, \\ \mathcal{A}v[\![a_0]\!]\sigma + \mathcal{A}v[\![a_1]\!]\sigma & \text{jeżeli } a = a_0 + a_1, \\ \mathcal{A}v[\![a_0]\!]\sigma - \mathcal{A}v[\![a_1]\!]\sigma & \text{jeżeli } a = a_0 - a_1, \\ \mathcal{A}v[\![a_0]\!]\sigma * \mathcal{A}v[\![a_1]\!]\sigma & \text{jeżeli } a = a_0 * a_1. \end{cases}$$

4.3 Zmiana definicji funkcji $\mathcal{C}$

$$\Sigma_\perp = \Sigma \cup \{\perp\}$$

Przyjmujemy, że

$$\mathcal{C} : \mathbf{Com} \to (\Sigma_\perp \to \Sigma_\perp).$$

Umawiamy się, że

$$\neg\exists\sigma'\ (\sigma, \sigma') \in \mathcal{C}[\![c]\!] \text{ zastępujemy przez } \mathcal{C}[\![c]\!]\sigma = \perp .$$

Przyjmujemy też, że $\mathcal{C}[\![c]\!] \perp = \perp$.

4.4 Spełnianie asercji

Definiujemy relację $\sigma \models^I A$ dla $\sigma \in \Sigma_\perp$, $I : \mathbf{Intvar} \to \mathbf{N}$ i $A \in \mathbf{Assn}$. Relację tę definiujemy przez indukcję ze względu na budowę A.

Do tej relacji należą wszystkie trójki postaci $\perp \models^I A$. Dalej zakładamy, że $\sigma \in \Sigma$. Trójki $\sigma \models^I \mathbf{true}$ są w definiowanej relacji, a trójki $\sigma \models^I \mathbf{false}$ do tej relacji nie należą. Poza tym

$$
\begin{aligned}
\sigma \models^I a_0 = a_1 &\iff \mathcal{A}v[\![a_0]\!]I\sigma = \mathcal{A}v[\![a_1]\!]I\sigma \\
\sigma \models^I a_0 \leq a_1 &\iff \mathcal{A}v[\![a_0]\!]I\sigma \leq \mathcal{A}v[\![a_1]\!]I\sigma \\
\sigma \models^I \neg A &\iff \text{nieprawda, że } \sigma \models^I A \\
\sigma \models^I A_0 \wedge A_1 &\iff \sigma \models^I A_0 \text{ i } \sigma \models^I A_1 \\
\sigma \models^I A_0 \vee A_1 &\iff \sigma \models^I A_0 \text{ lub } \sigma \models^I A_1 \\
\sigma \models^I \exists i.A &\iff \sigma \models^{I[n/i]} A \text{ dla pewnej liczby } n \in \mathbf{N} \\
\sigma \models^I \forall i.A &\iff \sigma \models^{I[n/i]} A \text{ dla pewnej liczby } n \in \mathbf{N}
\end{aligned}
$$

Asercje częściowej poprawności (stwierdzające częściową poprawność programu) to napisy postaci

$$\{A\}c\{B\}.$$

Przyjmujemy następujące definicje:

$$
\begin{aligned}
\sigma \models^I \{A\}c\{B\} &\iff \sigma \models^I A \Rightarrow \mathcal{C}[\![c]\!]\sigma \models^I B \\
\models^I \{A\}c\{B\} &\iff \forall \sigma \in \Sigma_\perp \ \sigma \models^I \{A\}c\{B\} \\
\models \{A\}c\{B\} &\iff \forall I : \mathbf{Intvar} \to \mathbf{N} \ \models^I \{A\}c\{B\}
\end{aligned}
$$

4.5 Definicja relacji $\vdash$

Relacji $\vdash$ jest najmniejszą relacją w zbiorze asercji częściowej poprawności, zamkniętą ze względu na następujące reguły:

$$\frac{}{\{A\}\mathbf{skip}\{A\}}$$

$$\frac{}{\{A[a/X]\}X := a\{A\}}$$

$$\frac{\{A_0\}c_0\{B\}, \ \{B\}c_1\{A_1\}}{\{A_0\}c_0; c_1\{A_1\}}$$

$$\frac{\{A \wedge b\}c_0\{B\},\ \{A \wedge \neg b\}c_1\{B\}}{\{A\}\textbf{if } b \textbf{ then } c_0 \textbf{ else } c_1\{B\}}$$

$$\frac{\{A \wedge b\}c\{A\}}{\{A\}\textbf{while } b \textbf{ do } c\{A \wedge \neg b\}}$$

$$\frac{\models A \Rightarrow A',\ \{A'\}c\{B'\},\ \models B' \Rightarrow B}{\{A\}c\{B\}}$$