Zadanie 1.

Podaj semantykę naturalną (semantykę operacyjną dużych kroków) dla języka o następującej gramatyce:

$$
\begin{array}{lcl}
\mathit{Num} \ni n & ::= & \ldots \mid \mathtt{-1} \mid \mathtt{0} \mid \mathtt{1} \mid \ldots \\
\mathsf{Var} \ni x & ::= & \mathtt{x} \mid \mathtt{y} \mid \ldots \mid \ldots \\
\mathit{Expr} \ni E & ::= & n \mid x \mid E_1 + E_2 \mid E_1 - E_2 \\
\mathit{Instr} \ni I & ::= & x{:=}E \mid\ I_1;I_2 \mid \mathtt{skip} \mid \mathtt{if}\ E = 0\ \mathtt{then}\ I_1\ \mathtt{else}\ I_2 \mid \\
 & & \mathtt{step}\ x\ \mathtt{by}\ E\ \mathtt{check} \mid \mathtt{for\ var}\ x := E_1\ \mathtt{to}\ E_2\ \mathtt{do}\ I\ \mathtt{end}
\end{array}
$$

Można przyjąć, że semantyka wyrażeń dana jest funkcją $\mathcal{E} \colon \mathit{Expr} \to (\mathsf{State} \to \mathbb{Z})$, gdzie jak zwykle, $\mathbb{Z}$ to zbiór liczb całkowitych, a $\mathsf{State} = \mathsf{Var} \to \mathbb{Z}$ jest zbiorem stanów, przypisujących wartości wszystkim zmiennym.

Instrukcje postaci $x{:=}E$, $I_1;I_2$, `skip` oraz `if` $E = 0$ `then` I_1 `else` I_2 wykonuje sie standardowo (w rozwiązaniach zadania należy jednak opisać ich semantykę).

Wykonanie pętli `for var` $x := E_1$ `to` E_2 `do` I `end` polega na obliczeniu wartości wyrażeń E_1 i E_2, przypisaniu wartości wyrażenia E_1 *zmiennej sterującej* x i wykonaniu *ciała pętli* I; obliczona tu wartość wyrażenia E_2 jest *wartością graniczną* zmiennej sterującej tej pętli.

Wykonanie instrukcji postaci `step` x `by` E `check` poza ciałem pętli ze zmienną sterująca x może być definiowane przez semantykę dowolnie. Niech zatem `for var` $x := E_1$ `to` E_2 `do` I `end` będzie najbardziej wewnętrzną pętlą ze zmienną sterująca x, w której ciele wystepuje dana instrukcja `step` x `by` E `check`. Wówczas wykonanie tej instrukcji polega na zwiększeniu wartości zmiennej x o wartość wyrażenia E, porównaniu tak otrzymanej wartości n zmiennej x z wartością graniczną m obliczoną w przy wejściu do tej petli, a dalej na:

- kontynuacji bieżącego obliczenia ciała pętli jeśli $n = m$, albo
- ponownym wykonaniu ciała petli I jeśli $n < m$, albo
- zakończeniu wykonania tej pętli jeśli $n > m$.

Przykład:

```
 0: z := 5;
 1: for var x := 1 to 4 do
 2:   for var y := 1 to z - 2 do
 3:     for var x := 1 to y + 2 do
 4:       z := z - 1; y := y + 1;
 5:       if z - 4 = 0
 6:       then step x by 1 check
 7:       else step y by 0 check;
 8:       x := x + 1
        end;
 9:     if y - 2 = 0 then skip else step x by 2 check
      end;
10:  z := 33
    end
```

Program po przypisaniu zmiennej z wartości 5 wejdzie do pętli w wierszu 1., przypisze tam zmiennej x wartość 1 oraz ustali wartość graniczną tej pętli, 4. Następnie wejdzie do pętli w wierszu 2., przypisze tam zmiennej y wartość 1 i ustali wartość graniczną tej pętli, 3. Dalej wejdzie do pętli w wierszu 3., przypisze tam zmiennej x wartość 1 oraz ustali wartość graniczną tej pętli, 3. Wykonanie ciała tej pętli rozpocznie od przypisań w wierszu 4., ustalając wartości zmiennych $z = 4$ i $y = 2$. Po sprawdzeniu warunku z wiersza 5. wykonana będzie instrukcja `step` z wiersza 6., która zwiększy wartość zmiennej x do $x = 2$ oraz porówna tę wartość z wartością graniczną 3 dla pętli z wiersza 3., co zdecyduje o ponownym wykonywaniu ciała tej pętli. Teraz przypisania z wiersza 4. ustalą wartość zmiennych $z = 3$ i $y = 3$, co po sprawdzeniu warunku z wiersza 5. prowadzi do instrukcji `step` z wiersza 7. Zwiększenie wartości zmiennej y o 0 zachowuje jej wartość 3, która jest równa wartości granicznej 3 dla pętli z wiersza 2. Zatem wykonanie ciała tej pętli będzie kontynuowane. Przypisanie z wiersza 8. nada zmiennej x wartość $x = 3$ i nastąpi zakończenie wykonywania ciała i całej pętli z wiersza 3. Następnie sprawdzony zostanie warunek z wiersza 9., co doprowadzi do wykonania instrukcji `step` z tego wiersza. Spowoduje ona zwiększenie wartości zmiennej x do $x = 5$ i porównanie tej wartości z wartością graniczną 4 pętli z wiersza 1. Ponieważ wartość ta została przekroczona, nastąpi zakończenie wykonywania pętli z wiersza 1. (bez kontynuacji wykonania ciała i całej pętli wewnętrznej z wiersza 2. oraz bez wykonania przypisania z wiersza 10.), a zatem także całego programu. W końcowym stanie zmienne mają wartości $z = 3, y = 3, x = 5$.

Rozwiązanie I:

Semantyka naturalna (duże kroki):
"Dobre" konfiguracje końcowe: $\mathsf{T} = \mathsf{State}$.
Konfiguracje: $\Gamma = \mathit{Instr} \times \mathsf{Env} \times \mathsf{State} \cup (\mathsf{Var} \times \mathit{Instr}) \times \mathsf{Env} \times \mathsf{State} \cup \mathsf{Var} \times \mathsf{State} \cup \mathsf{T}$.

- $\langle x, s\rangle \in \mathsf{Var} \times \mathsf{State}$ zapisuję jako $\langle \texttt{check}\ x, s\rangle$.
- $\langle x, I\rangle \in \mathsf{Var} \times \mathit{Instr}$ zapisuję jako $\texttt{do}\ I\ \texttt{checking}\ x$.
- Środowiska $\rho \in \mathsf{Env} = \mathsf{Var} \rightharpoonup \mathbb{Z}$ to funkcje częściowe, które zmiennym sterującym przypisują wartości graniczne dla odpowiadających im pętli.

Relacja "dojścia" ("dużych kroków"):

$$\rightsquigarrow\ \subseteq (\mathit{Instr} \times \mathsf{Env} \times \mathsf{State} \cup (\mathsf{Var} \times \mathit{Instr}) \times \mathsf{Env} \times \mathsf{State}) \times (\mathsf{Var} \times \mathsf{State} \cup \mathsf{T})$$

Reguły:
Standardowe reguły dla zwykłych obliczeń:

$$\frac{}{\langle x{:=}E, \rho, s\rangle \rightsquigarrow s[x{\mapsto}\mathcal{E}[\![E]\!]\,s]} \qquad \frac{}{\langle \texttt{skip}, \rho, s\rangle \rightsquigarrow s} \qquad \frac{\langle I_1, \rho, s\rangle \rightsquigarrow s' \quad \langle I_2, \rho, s'\rangle \rightsquigarrow s''}{\langle I_1;I_2, \rho, s\rangle \rightsquigarrow s''}$$

$$\frac{\langle I_1, \rho, s\rangle \rightsquigarrow s'}{\langle \texttt{if}\ E = 0\ \texttt{then}\ I_1\ \texttt{else}\ I_2, \rho, s\rangle \rightsquigarrow s'}\ \text{if } \mathcal{E}[\![E]\!]\,s = 0$$

$$\frac{\langle I_2, \rho, s\rangle \rightsquigarrow s'}{\langle \texttt{if}\ E = 0\ \texttt{then}\ I_1\ \texttt{else}\ I_2, \rho, s\rangle \rightsquigarrow s'}\ \text{if } \mathcal{E}[\![E]\!]\,s \neq 0$$

Semantyka `for` *przez semantykę* `do`*:*

$$\frac{\langle \texttt{do}\ I\ \texttt{checking}\ x, \rho[x{\mapsto}\mathcal{E}[\![E_2]\!]\,s], s[x{\mapsto}\mathcal{E}[\![E_1]\!]\,s]\rangle \rightsquigarrow s'}{\langle \texttt{for var}\ x := E_1\ \texttt{to}\ E_2\ \texttt{do}\ I\ \texttt{end}, s\rangle \rightsquigarrow s'}$$

$$\frac{\langle \texttt{do}\ I\ \texttt{checking}\ x, \rho[x{\mapsto}\mathcal{E}[\![E_2]\!]\,s], s[x{\mapsto}\mathcal{E}[\![E_1]\!]\,s]\rangle \rightsquigarrow \langle \texttt{check}\ y, s'\rangle}{\langle \texttt{for var}\ x := E_1\ \texttt{to}\ E_2\ \texttt{do}\ I\ \texttt{end}, s\rangle \rightsquigarrow \langle \texttt{check}\ y, s'\rangle}$$

Zgłoszenie sytuacji wyjątkowej:

$$\frac{}{\langle \texttt{step}\ x\ \texttt{by}\ E\ \texttt{check}, \rho, s\rangle \rightsquigarrow s[x{\mapsto}(s\,x + \mathcal{E}[\![E]\!]\,s)]}\ \text{if } \rho\,x = (s\,x + \mathcal{E}[\![E]\!]\,s)$$

$$\frac{}{\langle \texttt{step}\ x\ \texttt{by}\ E\ \texttt{check}, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s[x{\mapsto}(s\,x + \mathcal{E}[\![E]\!]\,s)]\rangle}\ \text{if } \rho\,x \neq (s\,x + \mathcal{E}[\![E]\!]\,s)$$

Propagowanie sytuacji wyjątkowej:

$$\frac{\langle I_1, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}{\langle I_1;I_2, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle} \qquad \frac{\langle I_1, \rho, s\rangle \rightsquigarrow s' \quad \langle I_2, \rho, s'\rangle \rightsquigarrow \langle \texttt{check}\ x, s''\rangle}{\langle I_1;I_2, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s''\rangle}$$

$$\frac{\langle I_1, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}{\langle \texttt{if}\ E = 0\ \texttt{then}\ I_1\ \texttt{else}\ I_2, \rho, s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}\ \text{if } \mathcal{E}[\![E]\!]\,s = 0$$

$$\frac{\langle I_2,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}{\langle \texttt{if}\ E=0\ \texttt{then}\ I_1\ \texttt{else}\ I_2,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}\ \text{if}\ \mathcal{E}[\![E]\!]\,s \neq 0$$

$$\frac{\langle I,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ y, s'\rangle}{\langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ y, s'\rangle}\ \text{if}\ x \neq y$$

Obsługa sytuacji wyjątkowej:

$$\frac{\langle I,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle}{\langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s\rangle \rightsquigarrow s'}\ \text{if}\ s'\,x > \rho\,x$$

$$\frac{\langle I,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle \quad \langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s'\rangle \rightsquigarrow s''}{\langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s\rangle \rightsquigarrow s''}\ \text{if}\ s'\,x < \rho\,x$$

$$\frac{\langle I,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle \quad \langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s'\rangle \rightsquigarrow \langle \texttt{check}\ y, s''\rangle}{\langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ y, s''\rangle}\ \text{if}\ s'\,x < \rho\,x,\ \ x \neq y$$

Celowo nie ma tu klauzuli dla przypadku $\langle I,\rho,s\rangle \rightsquigarrow \langle \texttt{check}\ x, s'\rangle$ i $\langle \texttt{do}\ I\ \texttt{checking}\ x,\rho,s'\rangle \rightsquigarrow \langle \texttt{check}\ x, s''\rangle$ przy $s'\,x < \rho\,x$: z instrukcji `do`/`for` ze zmienną sterującą x nigdy nie wychodzimy szukając zmiennej sterującej x "wyżej".

Rozwiązanie II:

Semantyka małych kroków:

Rozszerzamy klasę instrukcji o nową konstrukcję $\texttt{do}\ I$, tj., rozszerzamy gramatykę dla instrukcji jak następuje:

$$\begin{array}{rcl} \mathit{Instr}^+ \ni I & ::= & x{:=}E \mid I_1;I_2 \mid \texttt{skip} \mid \texttt{if}\ E=0\ \texttt{then}\ I_1\ \texttt{else}\ I_2 \mid \\ & & \texttt{step}\ x\ \texttt{by}\ E\ \texttt{check} \mid \texttt{for var}\ x := E_1\ \texttt{to}\ E_2\ \texttt{do}\ I\ \texttt{end} \mid \texttt{do}\ I \end{array}$$

"Dobre" konfiguracje końcowe: $\mathsf{T} = \mathsf{State}$.
Konfiguracje: $\Gamma = \mathit{Instr}^+ \times (\mathsf{Var} \times \mathbb{Z} \times \mathit{Instr})^* \times \mathsf{State} \cup$
$\mathsf{Var} \times (\mathsf{Var} \times \mathbb{Z} \times \mathit{Instr})^* \times \mathsf{State} \cup$
T.

- $\langle x, k, I\rangle \in \mathsf{Var} \times \mathbb{Z} \times \mathit{Instr}$ zapisuję jako $\texttt{do}\ I\ \texttt{checking}\ x \leq k$,
- $\langle x, \rho, s\rangle \in \mathsf{Var} \times (\mathsf{Var} \times \mathbb{Z} \times \mathit{Instr})^* \times \mathsf{State}$ zapisuję jako $\langle \texttt{check}\ x, \rho, s\rangle$.
- dla ciągów $\rho \in (\mathsf{Var} \times \mathbb{Z} \times \mathit{Instr})^*$ używać będę standardowo zdefiniowanych operacji (cokolwiek potrzebne: *head*, *tail*, *cons*, *append*, też pusty ciąg $\langle\rangle$).
- Wprowadzam pomocniczą funkcję częściową $lv\colon \mathsf{Var}\times(\mathsf{Var}\times\mathbb{Z}\times\mathit{Instr})^* \rightharpoonup \mathbb{Z}$ zdefiniowaną jako najmniejsza funkcja spełniająca $lv(x, cons(\texttt{do}\ I\ \texttt{checking}\ x \leq k, \rho)) = k$ oraz $lv(x, cons(\texttt{do}\ I\ \texttt{checking}\ y \leq k, \rho)) = lv(x, \rho)$ dla $x \neq y$ ($lv(x, \rho)$ nie jest zdefiniowane, gdy na liście ρ nie ma trójki postaci $\langle x, k, I\rangle$, a gdy takie trójki są, to $lv(x, \rho)$ wybiera k z pierwszej z nich).

Relacja "przejścia" ("małych kroków"):

$$\Rightarrow\ \subseteq \big(\mathit{Instr}^+ \times (\mathsf{Var} \times \mathbb{Z} \times \mathit{Instr})^* \times \mathsf{State}\big) \times (\Gamma)$$

Konfiguracje $\langle \texttt{check}\ x, \rho, s\rangle$ są pomocnicze, "wewnętrzne" dla ciał pętli, dla opisu przejść konfiguracji z pętlami — poza ciałem pętli powodują blokadę.
Reguły:
Standardowe reguły dla zwykłych kroków:

$$\frac{}{\langle x{:=}E, \rho, s\rangle \Rightarrow s[x{\mapsto}\mathcal{E}[\![E]\!]\, s]} \qquad \frac{}{\langle \texttt{skip}, \rho, s\rangle \Rightarrow s}$$

$$\frac{\langle I_1, \rho, s\rangle \Rightarrow s'}{\langle I_1;I_2, \rho, s\rangle \Rightarrow \langle I_2, \rho, s'\rangle} \qquad \frac{\langle I_1, \rho, s\rangle \Rightarrow \langle I_1', \rho', s'\rangle}{\langle I_1;I_2, \rho, s\rangle \Rightarrow \langle I_1';I_2, \rho', s'\rangle}$$

$$\frac{}{\langle \texttt{if}\ E=0\ \texttt{then}\ I_1\ \texttt{else}\ I_2, \rho, s\rangle \Rightarrow \langle I_1, \rho, s\rangle}\ \text{if } \mathcal{E}[\![E]\!]\, s = 0$$

$$\frac{}{\langle \texttt{if}\ E=0\ \texttt{then}\ I_1\ \texttt{else}\ I_2, \rho, s\rangle \Rightarrow \langle I_2, \rho, s\rangle}\ \text{if } \mathcal{E}[\![E]\!]\, s \neq 0$$

Semantyka `for` *przez* `do`*:*

$$\frac{}{\begin{array}{l}\langle \texttt{for var}\ x := E_1\ \texttt{to}\ E_2\ \texttt{do}\ I\ \texttt{end}, \rho, s\rangle \Rightarrow \langle \texttt{do}\ I, \\ \qquad\qquad cons(\texttt{do}\ I\ \texttt{checking}\ x \leq \mathcal{E}[\![E_2]\!]\, s, \rho), \\ \qquad\qquad s[x{\mapsto}(s\,x + \mathcal{E}[\![E]\!]\, s)]\rangle\end{array}}$$

Semantyka `step` *– przejście do sytuacji wyjątkowej:*

$$\frac{}{\langle \texttt{step}\ x\ \texttt{by}\ E\ \texttt{check}, \rho, s\rangle \Rightarrow s[x \mapsto s\,x + (\mathcal{E}[\![E]\!]\, s)]} \text{ if } lv(x,\rho) = s\,x + (\mathcal{E}[\![E]\!]\, s)$$

$$\frac{}{\langle \texttt{step}\ x\ \texttt{by}\ E\ \texttt{check}, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, \rho, s[x \mapsto s\,x + (\mathcal{E}[\![E]\!]\, s)]\rangle} \text{ if } lv(x,\rho) \neq s\,x + (\mathcal{E}[\![E]\!]\, s)$$

Propagowanie sytuacji wyjątkowej:

$$\frac{\langle I_1, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, \rho', s'\rangle}{\langle I_1; I_2, s\rangle \Rightarrow \langle \texttt{check}\ x, \rho', s'\rangle}$$

Semantyka `do`*:*

$$\frac{\langle I, \rho, s\rangle \Rightarrow \langle I', \rho', s'\rangle}{\langle \texttt{do}\ I, \rho, s\rangle \Rightarrow \langle \texttt{do}\ I', \rho', s'\rangle}$$

$$\frac{\langle I, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, cons(\texttt{do}\ I'\ \texttt{checking}\ y \leq k, \rho'), s'\rangle}{\langle \texttt{do}\ I, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, \rho', s'\rangle} \text{ if } x \neq y$$

$$\frac{\langle I, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, cons(\texttt{do}\ I'\ \texttt{checking}\ x \leq k, \rho'), s'\rangle}{\langle \texttt{do}\ I, \rho, s\rangle \Rightarrow s'} \text{ if } s'\,x > k$$

$$\frac{\langle I, \rho, s\rangle \Rightarrow \langle \texttt{check}\ x, cons(\texttt{do}\ I'\ \texttt{checking}\ x \leq k, \rho'), s'\rangle}{\langle \texttt{do}\ I, \rho, s\rangle \Rightarrow \langle \texttt{do}\ I', cons(\texttt{do}\ I'\ \texttt{checking}\ x \leq k, \rho'), s'\rangle} \text{ if } s'\,x < k$$