

Zadanie 1

Rozważmy następujący język programowania:

$$\begin{aligned}\mathbf{Number} &\ni n ::= \dots \mid -1 \mid 0 \mid 1 \mid \dots \\ \mathbf{Var} &\ni x ::= x_1 \mid x_2 \mid \dots \\ \mathbf{Expr} &\ni e ::= n \mid x \mid e_1 + e_2 \mid x := e \mid e_1; e_2 \mid \\ &\quad \mathbf{if} \ e \geq 0 \ \mathbf{then} \ e_1 \ \mathbf{else} \ e_2 \mid \mathbf{fix} \ x = e\end{aligned}$$

Język ten oparty jest wyłącznie o wyrażenia, wyliczające się do wartości liczbowej typu **Number**. Wyrażenia postaci $n, x, e_1 + e_2$ mają standardowe znaczenie, wyliczane są lewostronnie (najpierw lewy składnik sumy, później prawy). Zakładamy, że wszystkie zmienne mają określone początkowe wartości (zadane przez uruchamiającego dane wyrażenie).

Język ten zawiera wyrażenia o charakterze imperatywnym:

- Wyliczenie wyrażenia $x := e$ wymaga najpierw wyliczenia wartości liczbowej n wyrażenia e , następnie przypisania tej wartości zmiennej x . Wartość takiego wyrażenia to n .
- Wyliczenie wyrażenia $e_1; e_2$ wymaga najpierw wyliczenia wartości wyrażenia e_1 , a następnie wyliczenia wartości wyrażenia e_2 , którego wartość staje się końcową wartością całego wyrażenia $e_1; e_2$.
- Wyliczenie wyrażenia **if** $e \geq 0$ **then** e_1 **else** e_2 zaczyna się od wyliczenia wartości liczbowej n wyrażenia e . W zależności od tego czy jego wartość liczbową jest nieujemna, czy ujemna, zostaje wyliczone odpowiednio wyrażenie e_1 lub e_2 . Drugie z wyrażeń nie jest wyliczane. Wartość liczbową całego wyrażenia e_1 lub e_2 staje się wartością całego wyrażenia **if**.
- Wyliczenie wyrażenia **fix** $x = e$ wymaga iteracyjnego wyliczania wartości wyrażenia e i przypisywania tej wartości zmiennej x . Takie iteracyjne wyliczanie zatrzymuje się w pierwszym momencie, gdy nowo przypisywana na x wyliczona wartość wyrażenia e jest równa aktualnej wartości zmiennej x . Jeżeli wyliczanie wartości **fix** $x = e$ się zakończy, to wartość tego wyrażenia jest równa finalnej wartości zmiennej x .

Wyrażenia postaci **fix** mogą być dowolnie zagnieżdżone. W szczególności, wewnątrz podwyrażenia e wyrażenia **fix** $x = e$ może wystąpić kolejne wyrażenie **fix** $x = e'$ z tą samą zmienną x . Wyliczanie takiego wyrażenia powinno się odbywać zgodnie z powyższym opisem, poprzez podwójną iterację: iteracyjnie wyliczane jest zewnętrzne wyrażenie e , a każda taka iteracja może prowadzić do iteracyjnego wyliczania wartości wyrażenia e' .

Polecenie

Napisać semantykę operacyjną w wybranym przez siebie stylu: małych kroków lub dużych kroków, opisującą wyliczanie wartości wyrażeń powyższego języka.

verte!

Przykład

Rozważmy następujące wyrażenie:

```
x := 1;
y := 0;
z := 0;
fix x = (y := 1);
fix z = (
  if z + (-4) >= 0 then
    4
  else
    ((y := y + z); z + 1)
);
y+z
```

Pierwsze trzy linijki tego wyrażenia doprowadzą do ustalenia wartości zmiennych x , y i z na odpowiednio 1, 0 i 0. Wyrażenie to najpierw ustali wartości zmiennych x , y i z . Następnie wyliczone zostanie wyrażenie $y := 1$. Przypisze ono zmiennej y wartość 1 i całe przyjmie wartość 1. Wartość ta jest równa aktualnej wartości zmiennej x , więc wyliczenie tego wyrażenia **fix** $x = (y := 1)$ się zakończy (z wartością 1).

Następnie wyliczane będzie drugie wyrażenie **fix**. W tym celu sprawdzany będzie warunek $z + (-4) \geq 0$. Dla wartości zmiennej $z = 0, 1, 2, 3$ warunek jest fałszywy. W związku z tym wyliczone zostanie podwyrażenie $((y := y + z); z + 1)$, które ustali wartość y na odpowiednio 1, 2, 4, 7 i przyjmie wartość odpowiednio 1, 2, 3, 4. Po ustawieniu wartości zmiennej z na wartość 4, warunek $z + (-4) \geq 0$ stanie się prawdziwy i całe wyrażenie **if** przyjmie wartość 4, równą aktualnej wartości z . Zakończy to wyliczenie tego wyrażenia **fix**.

Ostatnie podwyrażenie to $y + z$, którego wartość to 11 i do tej właśnie wartości wyliczy się całe dane w przykładzie wyrażenie.

Przykładowe rozwiązanie — małe kroki

Podajemy najpierw semantykę w stylu małych kroków. Rozważmy zbiór **State** = (**Var** → **Number**). Rozszerzmy gramatykę wyrażeń **Expr** o dodatkowe wyrażenie **once** e **fix** $x = e'$. Niech wszystkie konfiguracje będą postaci $s, e \in \mathbf{State} \times \mathbf{Expr}$ natomiast konfiguracje końcowe postaci $s, n \in \mathbf{State} \times \mathbf{Number}$.

$$\begin{array}{c}
 \frac{}{s, x \rightarrow s, s(x)} \\
 \\
 \frac{s, e_1 \rightarrow s', e'_1}{s, e_1 + e_2 \rightarrow s', e'_1 + e_2} \quad \frac{s, e_2 \rightarrow s', e'_2}{s, n_1 + e_2 \rightarrow s', n_1 + e'_2} \quad \frac{}{s, n_1 + n_2 \rightarrow s, \underline{n_1 + n_2}} \\
 \\
 \frac{s, e \rightarrow s', e'}{s, x := e \rightarrow s', x := e'} \quad \frac{}{s, x := n \rightarrow s[x \mapsto n], n} \\
 \\
 \frac{s, e_1 \rightarrow s', e'_1}{s, e_1; e_2 \rightarrow s', e'_1; e_2} \quad \frac{}{s, n_1; e_2 \rightarrow s, e_2} \\
 \\
 \frac{s, e \rightarrow s', e'}{s, \text{ if } e \geq 0 \text{ then } e_1 \text{ else } e_2 \rightarrow s', \text{ if } e' \geq 0 \text{ then } e_1 \text{ else } e_2} \\
 \\
 \frac{n \geq 0}{s, \text{ if } n \geq 0 \text{ then } e_1 \text{ else } e_2 \rightarrow s, e_1} \quad \frac{n < 0}{s, \text{ if } n \geq 0 \text{ then } e_1 \text{ else } e_2 \rightarrow s, e_2} \\
 \\
 \frac{}{s, \text{ fix } x = e \rightarrow s, \text{ once } e \text{ fix } x = e} \\
 \\
 \frac{s, e \rightarrow s', e'}{s, \text{ once } e \text{ fix } x = e \rightarrow s', \text{ once } e' \text{ fix } x = e} \\
 \\
 \frac{s(x) = n}{s, \text{ once } n \text{ fix } x = e \rightarrow s, n} \quad \frac{s(x) \neq n}{s, \text{ once } n \text{ fix } x = e \rightarrow s[x \mapsto n], \text{ fix } x = e}
 \end{array}$$

Przykładowe rozwiązanie — duże kroki

Podajemy teraz semantykę w stylu dużych kroków. Rozważmy zbiór $\mathbf{State} = (\mathbf{Var} \rightarrow \mathbf{Number})$. Niech konfiguracje robocze będą postaci $s, e \in \mathbf{State} \times \mathbf{Expr}$ natomiast konfiguracje końcowe postaci $n, s \in \mathbf{Number} \times \mathbf{State}$.

$$\begin{array}{c}
 \frac{}{s, x \rightarrow s(x), s} \\
 \frac{s, e_1 \rightarrow n_1, s' \quad s', e_2 \rightarrow n_2, s''}{s, e_1 + e_2 \rightarrow \underline{n_1 + n_2}, s''} \\
 \frac{s, e \rightarrow n, s'}{s, x := e \rightarrow n, s'[x \mapsto n]} \\
 \frac{s, e_1 \rightarrow n_1, s' \quad s', e_2 \rightarrow n_2, s''}{s, e_1; e_2 \rightarrow n_2, s''} \\
 \frac{s, e \rightarrow n, s' \quad n \geq 0 \quad s', e_1 \rightarrow n_1, s''}{s, \text{ if } e \geq 0 \text{ then } e_1 \text{ else } e_2 \rightarrow n_1, s''} \quad \frac{s, e \rightarrow n, s' \quad n < 0 \quad s', e_2 \rightarrow n_2, s''}{s, \text{ if } e \geq 0 \text{ then } e_1 \text{ else } e_2 \rightarrow n_2, s''} \\
 \frac{s, e \rightarrow n, s' \quad s'(x) = n}{s, \mathbf{fix} \ x = e \rightarrow n, s'} \\
 \frac{s, e \rightarrow n, s' \quad s'(x) \neq n \quad s'[x \mapsto n], \mathbf{fix} \ x = e \rightarrow n', s''}{s, \mathbf{fix} \ x = e \rightarrow n', s''}
 \end{array}$$