

Semantyka i weryfikacja programów

semestr zimowy 2004/2005

Zadanie 1. Napisać semantykę naturalną (duże kroki) języka o następującej gramatyce (TINY z dodanymi dwiema instrukcjami):

$$\begin{aligned} N \in \mathbf{Num} &::= 0 \mid 1 \mid 2 \mid \dots \\ x \in \mathbf{Var} &::= x_1 \mid x_2 \mid \dots \\ e \in \mathbf{Exp} &::= N \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\ b \in \mathbf{BExp} &::= \mathbf{true} \mid \mathbf{false} \mid e_1 \leq e_2 \mid \neg b \mid b_1 \wedge b_2 \\ S \in \mathbf{Stmnt} &::= x := e \mid \mathbf{skip} \mid S_1; S_2 \mid \mathbf{if } b \mathbf{ then } S_1 \mathbf{ else } S_2 \mid \\ &\quad \mid \mathbf{while } b \mathbf{ do } S \mid \mathbf{for } x = e_1 \mathbf{ to } e_2 \mathbf{ try } S \mathbf{ else } S' \mid \mathbf{fail}. \end{aligned}$$

Znaczenie powyższych konstrukcji (poza $\mathbf{for } x = e_1 \mathbf{ to } e_2 \mathbf{ try } S \mathbf{ else } S' \mathbf{ i fail}$) jest standardowe (takie jak na wykładzie). Znaczenie instrukcji $\mathbf{for } x = e_1 \mathbf{ to } e_2 \mathbf{ try } S \mathbf{ else } S'$ jest następujące: najpierw jest obliczana wartość wyrażenia e_1 i podstawiana na zmienną x . Następnie wykonywana jest instrukcja S (z odpowiednio podstawioną wartością na zmienną x , oczywiście instrukcję S wykonujemy tylko wtedy, gdy $x \leq e_2$). Jeśli w S nie napotkana zostanie instrukcja \mathbf{fail} to przywracamy starą wartość zmiennej x (tzn. sprzed wykonania instrukcji \mathbf{for}) i przechodzimy do dalszej części programu. W przeciwnym wypadku przywracamy stan sprzed wykonania instrukcji \mathbf{for} (zachowując oczywiście wartość zmiennej x), zwiększamy wartość zmiennej x o 1 i powtarzamy procedurę opisaną powyżej tak długo aż nie napotkamy instrukcji \mathbf{fail} lub $x > e_2$ (wartość wyrażenia e_2 wyliczamy raz, w stanie sprzed wykonania instrukcji \mathbf{for}). Jeśli $x > e_2$ (zdarza się to wtedy, gdy napotkana została instrukcja \mathbf{fail} w instrukcji S dla $x = e_2$) to wykonujemy S' w stanie sprzed instrukcji \mathbf{for} (w szczególności wartość zmiennej x na początku wykonywania S' powinna być taka jaka była przed wykonaniem instrukcji \mathbf{for}). Instrukcja \mathbf{fail} poza pętlą \mathbf{for} powinna spowodować zakończenie programu w stanie "porażki".

Przykład:

```
x := 0; y := 1;
for x = 1 to 5 try
  y := y + 1;
  if x ≤ 4 then fail else z := x
else skip
```

Wartością zmiennej x po wykonaniu programu powinno być $x = 0$, $y = 2$, natomiast zmiennej $z = 5$.