

SEMANTYKA I WERYFIKACJA - Zadanie egzaminacyjne nr 2

Napisz semantykę denotacyjną, w stylu kontynuacyjnym, instrukcji języka o gramatyce:

$$\begin{aligned} Num \ni n &::= 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\ Var \ni x &::= x \mid y \mid \dots \\ PName \ni p &::= p \mid q \mid \dots \\ Expr \ni e &::= n \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\ BExpr \ni b &::= \text{true} \mid \text{false} \mid e_1 < e_2 \mid e_1 = e_2 \mid b_1 \wedge b_2 \mid \text{not } b \\ VDecl \ni d_V &::= \text{var } x \\ PDecl \ni d_P &::= \text{proc } p \ (I) \\ Instr \ni I &::= x := e \mid I_1; I_2 \mid \text{if } b \text{ then } I \mid \text{while } b \text{ do } I \mid \text{begin } d_V; d_P; I \text{ end} \mid \\ &\quad \text{call } p \mid \text{break} \mid \text{continue} \mid \text{exit} \end{aligned}$$

Jest to język z rekurencyjnymi procedurami bezparametrowymi oraz z prostymi mechanizmami przerywania pętli i procedur.

Instrukcja **break** powoduje zakończenie wykonywania najbliższej aktualnie wykonywanej pętli i przejście do dalszego wykonywania programu. Dzieje się tak nawet wtedy, jeśli instrukcja **break** nie występuje w ciele tej pętli, ale w procedurze wywołanej (bezpośrednio lub poprzez inne procedury) z tej pętli. Na przykład, w programie:

```
begin
  var x;
  proc p (if x>3 then break);
  x := 1;
  while true do
    call(p);
    x := x+1
  end
```

wykonanie pętli kończy się, a zmienna **x** przyjmuje wartość 4.

Instrukcja **continue** powoduje zakończenie wykonywania aktualnej iteracji najbliższej aktualnie wykonywanej pętli i przejście do wykonywania następnej iteracji tej pętli, z uwagami jak dla instrukcji **break** powyżej. Na przykład, gdyby w powyższym przykładzie zastąpić instrukcję **break** instrukcją **continue**, to program zapętlą się w stanie pamięci, w którym zmienna **x** przyjmuje wartość 4.

Instrukcja **exit** powoduje zakończenie wykonywania najbliższej aktualnie wykonywanej procedury. Gdyby w powyższym przykładzie zastąpić instrukcję **break** instrukcją **exit**, to program zapętlą się, a zmienna **x** będzie przyjmować coraz większe wartości.

Semantyka pozostałych konstrukcji jest standardowa. Widoczność wszystkich identyfikatorów jest statyczna.

W rozwiązaniu należy podać co najmniej równania semantyczne dla instrukcji **break**, **continue**, **exit**, **while..do..** i **call**, a także dla deklaracji procedur. Pozostałe równania można pominąć, jeżeli są standardowe lub zupełnie analogiczne do standardowych. Można założyć, że dana jest funkcja semantyczna dla wyrażeń i wyrażeń logicznych, w wybranym przez siebie wariantcie – klasycznym denotacyjnym lub w stylu kontynuacyjnym. Należy natomiast zdefiniować wszystkie potrzebne dziedziny pomocnicze i typy wszystkich funkcji semantycznych.

Można założyć, że programy nie korzystają z niezadeklarowanych zmiennych ani procedur. Można też założyć, że instrukcja **exit** może wystąpić tylko w ciele jakiejś procedury, a instrukcje **break** i **continue** będą wykonywane tylko podczas wykonywania jakiejś pętli.