

## SEMANTYKA I WERYFIKACJA - Zadanie domowe nr 2

Napisz semantykę denotacyjną instrukcji i deklaracji języka o następującej gramatyce:

$$\begin{aligned} Num \ni n &::= 0 \mid 1 \mid -1 \mid 2 \mid -2 \mid \dots \\ Var \ni x &::= x \mid y \mid \dots \\ PName \ni p &::= p \mid q \mid \dots \\ Expr \ni e &::= n \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\ BExpr \ni b &::= \text{true} \mid \text{false} \mid e_1 < e_2 \mid e_1 = e_2 \mid b_1 \wedge b_2 \mid \neg b \\ Decl \ni D &::= \text{var } x = e \mid \text{proc } p(x \text{ default } e) I \mid D_1; D_2 \\ Instr \ni I &::= \text{skip} \mid x := e \mid I_1; I_2 \mid \text{if } b \text{ then } I_1 \text{ else } I_2 \mid \\ &\quad \text{while } b \text{ do } I \mid \text{begin } D; I \text{ end} \mid \text{call } p(e) \mid \text{call } p \end{aligned}$$

Jest to język z rekurencyjnymi procedurami z jednym parametrem przekazywanym przez wartość i z przypisaną wartością domyślną, zadeklarowaną w postaci wyrażenia arytmetycznego. Podczas wywołania procedury zadeklarowanej przez:

$$\text{proc } p(x \text{ default } e) I$$

za pomocą instrukcji `call p`, najpierw obliczana jest aktualna wartość wyrażenia  $e$ , a następnie ta wartość jest przekazywana do ciała procedury  $p$  tak jak zwykły argument. Przy wywołaniu instrukcją `call p(e)`, domyślna wartość argumentu jest ignorowana. Widoczność wszystkich identyfikatorów jest statyczna.

Przykładowo, po wykonaniu fragmentu programu:

```
var x = 0;
var y = 0;
proc p(z default x+y) begin
  x := x+1;
  y := y+z
end;
call p(3);
call p;
```

zmienna  $x$  przyjmuje wartość 2, a zmienna  $y$  – wartość 7.

W rozwiązaniu można pominąć te równania, które nie różnią się od standardowej semantyki języka z procedurami z parametrami przekazywanymi przez wartość. Należy jednak podać typy wszystkich funkcji semantycznych wraz z użytymi dziedzinami pomocniczymi.