SEMANTYKA I WERYFIKACJA - Zadanie poprawkowe nr 1

Napisz semantykę operacyjną dużych kroków języka o gramatyce:

$$
\begin{aligned}
Num \ni n &::= \texttt{0} \mid \texttt{1} \mid -\texttt{1} \mid \texttt{2} \mid -\texttt{2} \mid \cdots \\
Var \ni x &::= \texttt{x} \mid \texttt{y} \mid \cdots \\
IVar \ni i &::= \texttt{i} \mid \texttt{j} \mid \cdots \\
PName \ni p &::= \texttt{p} \mid \texttt{q} \mid \cdots \\
Expr \ni e &::= n \mid x \mid e_1 + e_2 \mid e_1 * e_2 \mid e_1 - e_2 \\
BExpr \ni b &::= \texttt{true} \mid \texttt{false} \mid e_1 < e_2 \mid e_1 = e_2 \mid b_1 \wedge b_2 \mid \texttt{not } b \\
Decl \ni d &::= \texttt{var } x = e \mid \texttt{proc } p(i)\ I \mid d_1; d_2 \\
Instr \ni I &::= x := e \mid I_1; I_2 \mid \texttt{if } b \texttt{ then } I \mid \texttt{while } b \texttt{ do } I \mid \texttt{begin } d; I \texttt{ end} \mid \texttt{call } p(I) \mid i
\end{aligned}
$$

Jest to język z rekurencyjnymi procedurami, przyjmującymi instrukcje jako argumenty. Deklaracja `proc` $p(i)$ I wprowadza procedurę o nazwie p i ciele I, w którym można wykonać parametr i tak samo jak zwykłą instrukcję. Instrukcja `call` $p(I)$ wywołuje procedurę p, przekazując jej instrukcję I jako argument.

Odbywa się to zgodnie zasadami statycznej widoczności i przesłaniania identyfikatorów. Na przykład na końcu wykonania bloku

```
begin
    var x=0; var y=0;
    proc p(i)
        x := x+1; i;
    begin
       var x=0;
       call p(x := x+2);
       y := x
    end
end
```

zmienna `x` zadeklarowana w drugim wierszu przyjmuje wartość 1, a zmienna `y` – wartość 2.

Z kolei na końcu wykonania bloku

```
begin
    var x=100;
    proc p(i)
        x := x*2; i;
    call p(call p(x:=x+3));
end
```

zmienna `x` przyjmuje wartość 403.

Semantyka pozostałych konstrukcji jest standardowa.

W rozwiązaniu należy podać co najmniej reguły wyprowadzania dla instrukcji `call` oraz i, a także dla deklaracji procedur. Pozostałe reguły można pominąć jeżeli są standardowe lub zupełnie analogiczne do standardowych. Należy natomiast podać definicje wszystkich dziedzin pomocniczych i ogólną postać konfiguracji i relacji ewaluacji dla każdej kategorii syntaktycznej.

Można używać semantycznej funkcji *alloc*, która dla danego stanu pamięci zwraca pewną nieużywaną w tym stanie komórkę pamięci. Można też założyć, że programy nie korzystają z niewprowadzonych zmiennych, procedur ani identyfikatorów instrukcji i.