# Homework 7 (10 points)

## *Semantic analysis*

Symbols in Mini-Pascal have a nested scope. If a variable is not declared in the current scope, it is searched in the outer enclosing scope using the most closely nested rule. Similarly, a variable can be overloaded in the current scope and becomes again active when the scope is exited.

Expressions of type integer and real can be combined. The type of the resulting expression is real, if at least one operand is real, otherwise it is integer. If the variable in an assignment statement is real (integer) and the right-hand side expression is integer (real), the latter is implicitly converted to real (integer).

Extend your YACC-based Mini-Pascal parser with semantic analysis that performs the following semantic checks and outputs corresponding error messages (including line numbers).

1. Variable not declared; **(1 point)**
2. Variable already declared; **(1 point)**
3. Invalid array declaration; **(1 point)**
   Examples:
   - Wrong index type: **`a[1.0..9.9];`**
   - Wrong array bounds: **`a[4..1];`**
4. Invalid array subscript; **(1 point)**
   Examples:
   - Wrong index type: **`a[true]`**, **`a[3.14]`** (also with expressions/variables);
   - Wrong index value: **`a[0]`** if **a** is declared **`a[1..10];`**
   - Index range mismatch: **`a[1..10] = b[1..5]`** (number of array elements must be the same);
5. Type mismatch in assignment; **(1 point)**
   Examples:
   - **`b: real; b:=false;`**
   - Assume a is an array and b is a scalar
     - **`b:=`** **a** is not allowed
     - **`a:=`** **b** is allowed (assigns b to every array element);
6. Illegal operand type (does not match operator); **(1 point)**
   Example:
   - (**`3.14`** and **`2.0`**);
7. Condition not boolean (in **`if`** and **`while`** statements); **(1 point)**
   Example:
   - **`if (4–3) then …`**
8. Function not declared; **(1 point)**
9. Function already declared; **(1 point)**
10. Invalid function call (formal/actual parameter list mismatch). **(1 point)**