

Context Rules

- 1) Any Var ID referred should be uniquely defined within the current scope.
- 2) Any fun ID referred in the current scope should be defined uniquely above the current scope.
- 3) In Funct with resultType = VAR, there must be a return statement
- 4) In main function, return statement is not allowed
- 5) Function overloading is not allowed
- 6) Function call should refer to the function which is already defined
- 7) Reference to the same function is not allowed.

Type Rules

- 1) In AssignmentStmt(ID_left, Expr)
 - a. ID_Left.Type is always VAR
 - b. Expr.Type is always VAR
- 2) Type(X) and Type(Y) unifies, if both are of type VAR
 - a. X and Y can be ID, Number
 - b. If X is a function, X.ResultType =VAR
 - c. If Y is a function, Y.ResultType =VAR
- 3) In Expr= ID_Name | Number | PLUS_Expr | MINUS_Expr | TIMES_Expr | DIV_Expr | FunctionCall
 - a. ID_Name.Type is always VAR
 - b. Number is always integer
 - c. In PLUS_Expr(Expr,Expr), both Expr.Type's are INT

d. In MINUS_Expr (Expr,Expr) , both Expr.Type's are INT

e. In TIMES_Expr (Expr,Expr) , both Expr.Type's are INT

f. In MOD_Expr (Expr,Num), Expr.Type is INT

g. In DIV_Expr (Expr,Expr, , both Expr.Type's are INT

h. In FunctionCall, the ResultType must be INT

h.i. The Function call should match with the function signature

4) In FunctionCallStmt = FunctionCall, the resultType of function is NONE.

5) Type of BoolExp is INT

a. In EqualityExp(ID_Left, ID_right), ID_Left.Type and ID_Right.Type are INT

b. In NotEqualityExp(ID_Left, ID_right)), ID_Left.Type and ID_Right.Type are INT

c. In LT_Exp(ID_Left, ID_right)), ID_Left.Type and ID_Right.Type are INT

d. In GT_Exp(ID_Left, ID_right)), ID_Left.Type and ID_Right.Type are INT

e. In LE_Exp(ID_Left, ID_right)), ID_Left.Type and ID_Right.Type are INT

f. In GE_Exp(ID_Left, ID_right)), ID_Left.Type and ID_Right.Type are INT

#####