/* Notational Conventions:

- F(T1,...Tn) denotes a tree with the root labeled F, and subtrees T1 to Tn.
- If any of Ti is of the form 'X*' or 'X+' then

  X* denotes a (possibly empty) list of children each with root labeled X

  X+ denotes a non-empty list of children each with root labeled X
- All Identifiers start with "ID_"
- X = Y | Z specifies that X is Y or Z

*/

Program(Funct*, MainFun)

MainFun(Declaration*, Statement+)

Funct(Function_Name, Return_Type, Parameters, Declaration*, Statement)

Declaration=Var

Var(id)

ResultType=TK_KEY_NONE | TK_KEY_VAR

Parameters = TK_KEY_NONE | FormalParamList

FormalParamList = ID_formalParam+

```
Statement = AssignmentStmt | ConditionalStmt | RepetitiveStmt |
ReturnStmt | IO_Stmt | FunctionCallStmt | EmptyStmt

EmptyStmt(NULL)
```

AssignmentStmt(ID_left, Expr)

Expr= ID_Name | Number | PLUS_Expr | MINUS_Expr | MUL_Expr | DIV_Expr | FunctionCall

PLUS_Expr (Expr,Expr)

MINUS_Expr (Expr,Expr)

MUL_Expr (Expr,Expr)

MOD_Expr (Expr,TK_NUM)

DIV_Expr (Expr,Expr)

FunctionCall (ID_FunctName, ActualParams)

ActualParams = TK_KEY_NONE | ParamsList

ParamsList = ID_ParamName+

ConditionalStmt(BoolExp, Statement*)

BoolExp = EqualityExp | NotEqualityExp | LT_Exp | LE_Exp | GT_Exp | GE_Exp | ID_Exp

ID_Exp(ID_Left)

EqualityExp(ID_Left, ID_right)

NotEqualityExp(ID_Left, ID_right)

LT_Exp(ID_Left, ID_right)

GT_Exp(ID_Left, ID_right)

LE_Exp(ID_Left, ID_right)

GE_Exp(ID_Left, ID_right)

RepetitiveStmt(BoolExp, Statement*)

ReturnStmt = ID_return

FunctionCallStmt = FunctionCall

IO_Stmt = INStmt | OUTStmt

INStmt(ID_input)

```
OUTStmt(ID_output)
```