/\* 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)*