Raymond Farias 861158225
John Herrera

Mini L Grammar

Program
Program -> FunctionS
FunctionS -> Function FunctionS | empty

Function
Function -> function identifier ; beginparams DeclarationS endparams beginlocals DeclarationS endlocals beginbody StatementNS endbody

DeclarationS -> Declaration ; DeclarationS | empty
StatementNS -> Statement ; StatementNS | Statement ;

Declaration
Declaration -> IdentifierNS : ArrayInt
IdentifierNS -> identifier , IdentifierNS | identifier ,
ArrayInt -> integer | array [number] of integer

Statement
Statement-> A | B | C | D | E | F | G | H | I
A -> Var := Expression
B -> if Bool-Exp then Statement ; endif | if Bool-Exp then else StatementS endif
C -> while Bool-Expr beginloop StatementS endloop
D-> do beginloop StatementNS endloop while Bool-Expr
E -> foreach identifier in identifier beginloop StatementNS endloop
F -> read VarNS
G -> write VarNS
H -> continue
I -> return Expression

VarNS -> Var , VarNS | Var ,

Bool-Expr
Bool-Expr -> Relation-And-Expr OrS
OrS -> or Relation-And-Expr OrS | empty

Relation-And-Expr
Relation-And-Expr -> Relation-Expr AndS
AndS-> and Relation-Expr AndS | empty

Relation-Expr

Relation-Expr -> not Rexpr | Rexpr
Rexpr -> Expression Comp Expression | true | false | (Bool-Expr)

Comp
Comp -> == | <> | < | > | <= | >=

Expression
Expression -> Multiplicative-Expr ExprSumS
ExprSumS -> + Multiplicative-Expr ExprSumS | - Multiplicative-Expr ExprSumS | empty

Multiplicative-Expr
Multiplicative-Expr -> Term TermS
TermS -> * Term TermS | / Term TermS | % Term TermS | empty

Term
Term -> UpTerm | - UpTerm | identifier TermIdentifier
UpTerm -> Var | number | (Expression)
TermIdentifier -> (TermExpression) | ( )
TermExpression -> Expression | Expression , TermExpression

Var
Var -> identifier | identifier [Expression]