

E = EPSILON

Program start

prog_start -> Function prog_start | E

Function -> FUNCTION Ident SEMICOLON BEGIN_PARAMS Declaration_loop SEMICOLON
END_PARAMS BEGIN_LOCALS Declaration_loop SEMICOLON END_LOCALS BEGIN_BODY
Statement_loop SEMICOLON END_BODY

Declaration -> Ident_loop COLON INTEGER
 | Ident_loop COLON ARRAY L_SQUARE_BRACKET NUMBER %d
R_SQUARE_BRACKET OF INTEGER

Ident_loop -> ident | Ident COMMA Ident_loop

Statement -> Statement1 | Statement2 | Statement3 | Statement4 | Statement5 | Statement6 |
Statement7 | Statement8 | Statement9

Statement1 -> Var ASSIGN Expression
Statement2 -> IF Bool-Expr THEN Statement_loop ElseStatement ENDIF
Statement3 -> WHILE Bool-Expr BEGINLOOP Statement_loop SEMICOLON ENDLOOP
Statement4 -> DO BEGINLOOP Statement_loop SEMICOLON ENDLOOP WHILE Bool-Exp
Statement5 -> FOR Var ASSIGN NUMBER SEMICOLON Bool-Expr SEMICOLON Var ASSIGN
Expression BEGINLOOP Statement_loop SEMICOLON ENDLOOP
Statement6 -> READ Var
Statement7 -> WRITE Var
Statement8 -> CONTINUE
Statement9 -> RETURN Expression

Statement_loop -> Statement SEMICOLON Statement_loop | Statement SEMICOLON

ElseStatement ELSE Statement_loop | E

Bool-Expr -> Relation-And-Expr

Relation-And-Expr -> Relation-Expr | Relation-Expr AND Relation-And-Expr

Relation-Expr -> NOT Relation-Expr_loop | Relation-Expr_loop

Relation-Expr_loop -> Expression Comp Expression
 | TRUE
 | FALSE
 L_PAREN Bool-Exp R_PAREN

Comp -> EQ | NEQ | LT | GT | LTE | GTE

Expression -> Multiplicative-Expr | Multiplicative-Expr SUB Expression | Multiplicative-Expr
ADD Expression

Expression_loop -> Expression COMMA Expression_loop | Expression | E

Multiplicative-Expr -> Term | MOD Term Multiplicative-Expr | DIV Term Multiplicative-Expr |
MULT Term Multiplicative-Expr

Term -> Var | NUMBER | L_PAREN Expression R_PAREN | SUB Var | SUB NUMBER
| SUB L_PAREN Expression R_PAREN | Ident L_PAREN Expression_loop R_PAREN

Var -> Ident | Ident L_SQUARE_BRACKET Expression R_SQUARE_BRACKET

Var_loop -> Var | Var COMMA Var_loop