Mini-L Rules from Syntax Diagrams

- program → function program
  | epsilon

- function → FUNCTION identifier SEMICOLON BEGIN_PARAMS declarationloop END_PARAMS BEGIN_LOCALS declarationloop END_LOCALS BEGIN_BODY statementloop END_BODY

- declaration → identifierloop COLON INTEGER
  | identifierloop COLON ARRAY L_SQUARE_BRACKET NUMBER R_SQUARE_BRACKET OF INTEGER

- declarationloop → declaration SEMICOLON declarationloop
  | epsilon

- statement → var ASSIGN expression
  | IF bool_exp THEN statementloop ENDIF
  | IF bool_exp THEN statementloop ELSE statementloop ENDIF
  | WHILE bool_exp BEGINLOOP statementloop ENDLOOP
  | DO BEGINLOOP statementloop ENDLOOP WHILE bool_exp
  | FOR var ASSIGN number SEMICOLON bool_exp SEMICOLON var ASSIGN expression BEGINLOOP statementloop ENDLOOP
  | READ varloop
  | WRITE varloop
  | CONTINUE
  | RETURN expression

- statementloop → statement SEMICOLON statementloop
  | statement SEMICOLON

- bool_exp → relation_and_exp OR bool_exp
  | relation_and_exp

- relation_and_exp → notloop AND relation_and_exp
  | notloop

relation_exp → expression comp expression
| TRUE
| FALSE
| L_PAREN bool_exp R_PAREN

notloop → NOT notloop
| relation_exp

comp → EQ
| NEQ
| LT
| GT
| LTE
| GTE

expression → multiplicative_expression
| multiplicative_expression ADD expression
| multiplicative_expression SUB expression

multiplicative_expression → term
| term MULT multiplicative_expression
| term DIV multiplicative_expression
| term MOD multiplicative_expression

term → SUB var
| SUB number
| SUB L_PAREN expression R_PAREN
| var
| number
| L_PAREN expression R_PAREN
| identifier L_PAREN expressionloop R_PAREN

varloop → var
| var COMMA varloop

> var → identifier L_SQUARE_BRACKET expression R_SQUARE_BRACKET
| identifier

> number → NUMBER

> expressionloop → expression COMMA expressionloop
| expression
| epsilon

> identifier → IDENT

> identifierloop → identifier COMMA identifierloop
| identifier