Program -> Function Program | epsilon

Functions → Function Functions | epsilon

Function -> "function" Ident ";" "beginparams" Declarations "endparams" "beginlocals" Declatarions "endlocals" "beginbody" Statements "endbody"

Declarations -> Declaration ";" Declarations | epsilon

Declaration -> Identifiers “:” “integer” | Identifiers “:” “array” “[” “number” “]” “of” “integer” | Identifiers “:” “enum” “(” Identifiers “)”

Statements -> Statement ";" Statements |epsilon

Statement -> A|B|C|D|E|F|G|H
- A -> Var ":=" Expression
- B -> "if" Bool_Expr "then" Statement "endif" | "if" Bool_Expr "then" Statement "else" Statement "endif"
- C -> "while" Bool_Expr "beginloop" Statement "endloop"
- D -> "do" "beginloop" Statement "endloop" "while" Bool_Expr
- E -> "read" Vars
- F -> "write" Vars
- G -> "continue"
- H -> "return" Expression

Identifiers -> Ident | Ident “,” Identifiers

Ident -> “identifiers”

Vars -> Var | Var “,” Vars

Relation_And_Expr -> Relation_Expr | Relation_And_Expr "and" Relation_Expr

Relation_Expr -> "not" exp1| exp1

exp1 -> Expression Comp Expression | "true" | "false" | "(" Bool_Expr ")"

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

Expression -> Multiplicative-Expr | Expression "-" Multiplicative-Expr | Expression "+" Multiplicative-Expr

Multiplicative-Expr -> Term | Multiplicative-Expr "%" Term | Multiplicative-Expr "/" Term | Multiplicative-Expr "*" Term

Term -> "-" Var | Var | “-” “number” | “number” | “-” "(" Expression ")" | "(" Expression ")" | | Ident "(" Preloop ")"

Preloop -> Expression "," Preloop| Expression

Var -> Ident | Ident "[" Expression "]"