

# Mini-I Grammar

## PROGRAM:

Program -> Program function | empty

## FUNCTION:

Function -> "function" "identifier" " ," "beginparams" **dec\_loop** "endparams" "beginlocals"

**dec\_loop** "endlocals" "beginbody" **statement\_loop** "endbody"

**dec\_loop** -> **dec\_loop** **Declaration** " ," | empty

**Statement\_loop** -> **statement** " ," | **statement\_loop** **statement** " ,"

## DECLARATION:

**Declaration** -> **id\_loop** " ." **assignment**

**Id\_loop** -> "identifier" | **id\_loop** "identifier"

**assignment** -> "integer" | "array" "[ "number" "]" **assignment** " of " "integer"

**Assignment** -> empty | "[ "number" "]"

## STATEMENT:

**Statement** -> A|B|C|D|E|F|G|H|I

**A** -> **Var** " :=" **Expression**

**B** -> "if" **Bool-Exp** "then" **Statement\_loop** "endif" | "if" **Bool-Exp** "then" **Statement\_loop** "else"

**Statement\_loop** "endif"

**C** -> "while" **Bool-Exp** "beginloop" **Statement\_loop** "endloop"

**D** -> "do" "beginloop" **Statement\_loop** "endloop" "while" **Bool-Exp**

**E** -> "for" **Var** " :=" "number" " ," **Bool-Exp** " ," **Var** " :=" **Expression** "beginloop" **Statement\_loop** "endloop"

**F** -> "read" **Var** **Var\_Loop**

**Var\_Loop** -> " ," **Var** **Var\_Loop** | Empty

**G** -> "write" **Var** **Var\_Loop**

**H** -> "continue"

**I** -> "return" **Expression**

## Bool-Expr:

**Bool-Expr** -> **Relation-And-Expr** **B'**

**B'** -> "or" **Relation-And-Expr** **B'** | Empty

## Relation-And-Expr:

**Relation-And-Expr** -> **Relation-Expr** **RAE'**

**RAE'** -> "and" **Relation-Expr** **RAE'** | Empty

## Relation-Expr:

**Relation-Expr** -> "not" **R'** | **R'**

**R'** -> **Expression** **Comp** **Expression** | "true" | "false" | "(" **Bool-Expr** ")"

**COMP:**

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

**EXPRESSION:**

Expression -> Multiplicative-Expr Expression'

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

**Multiplicative-Expr:**

Multiplicative-Expr -> Term Multi'

Multi' -> "\*" Term Multi' | "/" Term Multi' | "%" Term Multi' | empty

**Term:**

Term -> "-" Term' | Term' | "identifier" "(" Expression\_loop ")"

Term' -> Var | "number" | "(" Expression ")"

Expression\_loop -> Expression "," Expression\_loop | Expression | empty

**Var:**

Var -> "identifier" | "identifier" "[" Expression "]" Var'

Var' -> empty | "[" Expression "]"