Krishaan Patel

CS152

Grammar PDF (Phase 2)

Program -> **functions**

functions -> empty | **Function functions**

Function -> “function” “identifier” “;” “beginparams” **Dec** “endparams” “beginlocals” **Dec** “endlocals” “beginbody” **Statements** “endbody”

Dec -> **Declaration** “;” **Dec** | empty

Declaration -> **identifiers** “COLON” “INTEGER” | **identifiers** “COLON” “ARRAY” “L_SQUARE_BRACKET” “NUMBER” “R_SQUARE_BRACKET” “OF” “INTEGER” | **identifiers** “COLON” “ENUM” “L_PAREN” **identifiers** “R_PAREN”

identifiers -> **ident** | **ident** “COMMA” **identifiers**

ident -> IDENT

Statements-> **Statement** “SEMICOLON” | **Statement** “SEMICOLON” **Statements**

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

A -> **Var** “:=” **Expression**

B -> “if” **Bool-Exp** “then” **Statements** “endif” | “if” **Bool-Exp** “then” **Statements** “else” **Statements** “endif”

C -> “while” **Bool-Expr** “beginloop” **Statements** “endloop”

D -> “do” “beginloop” **Statements** “endloop” “while” **Bool-Expr**

E - > “read” **Var E’**

E’ -> “,” **Var E’** | empty

F -> “write” **Var E’**

G -> “continue”

H -> “return” **Expression**

Bool-Expr **-> Relation-And-Expr | Relation-And-Expr** “OR” **Bool-Expr**

Relation-And-Expr -> **Relation-Expr** | **Relation-Expr** “AND” **Relation-And-Expr**

Relation-Expr -> **Expression Comp Expression** | “NOT” **Expression Comp Expression** | “TRUE” | “NOT” “TRUE” | “FALSE” | “NOT” “FALSE” | “L_PAREN” **Bool-Expr** “R_PAREN” | “NOT” “L_PAREN” **Bool-Expr** “R_PAREN”

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

Expression -> **Multiplicative-Expr** | **Multiplicative-Expr** “ADD” **Expression** | **Multiplicative-Expr** “SUB” **Expression**

Multiplicative-Expr -> **Term** | **Term** “MULT” **Multiplicative-Expr** | **Term** “DIV”

Multiplicative-Expr | **Term** “MOD” **Multiplicative-Expr**

Term -> **Term_A | Term_B**

Term_A -> **Var |** “SUB” **Var** | “NUMBER” | **“**SUB” “NUMBER” | “L_PAREN” **Expression**

“R_PAREN” | “SUB” “L_PAREN” **Expression** “R_PAREN”

Term_B -> **ident** “L_PAREN” **Expr** “R_PAREN”

Expr -> **Expression** | **Expression** “COMMA” **Expr** | empty

Var -> **ident** | **ident** “L_SQUARE_BRACKET” **Expression** “R_SQUARE_BRACKET”