

Jesse Garcia Phase 2

CAPITAL = terminals

Non Capital = non-terminals

E = EPSILON

### — Program start —

prog\_start -> Function prog\_start | E

### — Function Grammar —

Function -> FUNCTION Ident SEMICOLON BEGIN\_PARAMS Declaration\_loop END\_PARAMS  
BEGIN\_LOCALS Declaration\_loop END\_LOCALS BEGIN\_BODY Statement\_loop END\_BODY

### — Declaration Grammar with a loop —

Declaration -> Ident\_loop COLON INTEGER  
                  | Ident\_loop COLON ARRAY L\_SQUARE\_BRACKET NUMBER  
R\_SQUARE\_BRACKET OF INTEGER

Declaration\_loop -> Declaration SEMICOLON Declaration\_loop | E

### — Identifier loop —

Ident\_loop -> ident | Ident COMMA Ident\_loop

### — Statement Grammar Broken up into multiple statements —

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\_loop

Statement7 -> WRITE Var\_loop

Statement8 -> CONTINUE

Statement9 -> RETURN Expression

Statement\_loop -> Statement SEMICOLON Statement\_loop | Statement SEMICOLON

### — Else statement Grammar for if and else —

ElseStatement ELSE Statement\_loop | E

### — Different Expression Grammar —

Bool\_Expr -> Relation\_And\_Expr | Relation\_Expr OR Bool\_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

Multiplicative\_Expr -> Term | MOD Term Multiplicative\_Expr | DIV Term Multiplicative\_Expr |  
MULT Term Multiplicative\_Expr

### — Comp Grammer —

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

### — Regular Expression with an Expression loop Grammar —

Expression -> Multiplicative\_Expr | Multiplicative\_Expr SUB Expression | Multiplicative\_Expr  
ADD Expression

Expression\_loop -> Expression COMMA Expression\_loop | Expression | E

### — Term and Var with a Var loop Grammar —

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