Steven Truong (struo006) SSID:861161106
Kevin Kim (kkim068) SSID:861168574

_ = non terminal
‘ ’ = terminals

Program -> func

func -> Function func | ‘epsilon’

Function -> ‘function’ ‘identifier’ ‘;’ ‘beginparams’ Dec ‘endparams’ ‘beginlocals’ Dec ‘endlocals’
‘beginbody’ state2help1 ‘endbody’

Dec -> Declaration ; Dec | ‘epsilon’

Declaration -> dec1 ‘:’ dec2
dec1 -> ‘identifier’ ‘,’ dec1 | ‘identifier’
dec2 -> ‘array’ ‘[‘ ‘number’ ‘]’ ‘of’ ‘integer’ | ‘integer’

Statement -> statement1 | statement2 | statement3 | statement4 | statement5 | statement6 | statement7 | statement8 | statement9

statement1 -> Var ‘:=’ Expression

statement2 -> ‘if’ Bool_Exp ‘then’ state2help1 state2help2 ‘endif’

state2help1 -> Statement ‘;’ state2help1 | Statement ‘;’
state2help2 -> ‘else’ state2help1 | ‘epsilon’

statement3 -> ‘while’ Bool_Exp ‘beginloop’ state2help1 ‘endloop’

statement4 -> ‘do’ ‘beginloop’ state2help1 ‘endloop’ ‘while’ Bool_Expr

statement5 -> ‘foreach’ ‘identifier’ ‘in’ ‘identifier’ ‘beginloop’ state2help1 ‘endloop’

statement6 -> ‘read’ Var state6help

state6help -> ‘,’ Var state6help | ‘epsilon’

statement7 -> ‘write’ Var state6help1

statement8 -> ‘continue’
statement9 -> ‘return’ Expression

Bool_Expr -> Relation_And_Expr orHelper

orHelper -> ‘or’ Relation_And_Expr orHelper | ‘epsilon’

Relation_And_Expr -> Relation_Expr andHelper

andHelper -> ‘and’ Relation_Expr andHelper | ‘epsilon’

Relation_Expr -> relationExprHelper | not relationExprHelper

relationExprHelper -> Expression Comp Expression | ‘true’ | ‘false | ‘(‘ Bool_Expr ‘)’

Comp -> ‘==’ | ‘<>’ | ‘<’ | ‘>’ | ‘<=’ | ‘>=’

Expression -> Multiplicative_Expr multExprHelper

multExprHelper -> ‘+’ Multiplicative_Expr multExprHelper |
‘-’ Multiplicative_Expr multExprHelper | ‘epsilon’

Multiplicative_Expr -> Term termHelper

termHelper -> ‘*’ Term termHelper | ‘/’ Term termHelper | ‘%’ Term termHelper |
‘epsilon’

Term -> identifierTerm | varTerm | ‘-’ varTerm

identifierTerm -> ‘identifier’ ‘(‘ identifierHelp ‘)’

identifierHelp -> Expression ‘,’ identifierHelp | Expression

varTerm -> Var | ‘number’ | ‘(‘ Expression ‘)’

Var **-**> ‘identifier’ | ‘identifier’ ‘[‘ <u>Expression</u> ‘]’