CS152 Parser Language Rules

Stanley Cohen (scohe001)
Wyatt Sullivan (wsull001)
Preston Giorgianni (pgior002)

Program ⇒ `Function Program` | ϵ

Function ⇒ `FUNCTION IDENTIFIER SEMICOLON BEGIN_PARAMS Declaration_blk END_PARAMS BEGIN_LOCALS Declaration END_LOCALS BEGIN_BODY Statement_blk END_BODY` | ϵ

Declaration_blk ⇒ `Declaration SEMICOLON Declaration_blk` | ϵ

Declaration ⇒ `IDENTIFIER Identifier_blk COLON Array_declaration INTEGER`

Identifier_blk ⇒ `COMMA IDENTIFIER Identifier_blk` | ϵ

Array_declaration ⇒ `ARRAY L_SQUARE_BRACKET NUMBER R_SQUARE_BRACKET OF` | ϵ

Statement_blk ⇒ `Statement SEMICOLON Statement_blk` | ϵ

Statement ⇒ `Var SEMICOLON EQ Expression` | `IF Bool_exp BEGINLOOP Statement SEMICOLON Statement_blk ELSE` | `WHILE Bool_exp BEGINLOOP Statement SEMICOLON Statement_blk ENDLOOP` | `DO BEGINLOOP Statement SEMICOLON Statement_blk ENDLOOP WHILE Bool_exp` | `READ Var Var_blk` | `WRITE Var Var_blk` | `CONTINUE` | `RETURN Expression`

Else_blk ⇒ `ELSE Statement SEMICOLON Statement_blk` | ϵ

Bool_exp ⇒ `Relation_and_exp Or`

Or ⇒ `OR Relation_and_exp Or` | ϵ

Relation_and_exp ⇒ `Relation_exp And`

And ⇒ `AND Relation_exp And` | ϵ

Relation_exp ⇒ `Not Expression Comp Expression` | `Not TRUE` | `Not FALSE` | `Not L_PAREN Bool_exp R_PAREN`

Not ⇒ `NOT` | ϵ

Comp ⇒ `EQ` | `LT` | `GT` | `NEQ` | `LTE` | `GTE`

Expression ⇒ `Multiplicative_exp Multiplicative_exp_blk`

Multiplicative_exp_blk ⇒ `Multiplicative_exp_add Multiplicative_exp_blk` | `Multiplicative_exp_sub Multiplicative_exp_blk` | ϵ

Multiplicative_exp_add ⇒ `ADD Multiplicative_exp`

Multiplicative_exp_sub ⇒ `SUB Multiplicative_exp`

Multiplicative_exp ⇒ `Term Term_blk` | `Term`

Term_blk ⇒ `MULT Term Term_blk` | `DIV Term Term_blk` | `MOD Term Term_blk` | ϵ

Var ⇒ `IDENTIFIER` | `IDENTIFIER L_SQUARE_BRACKET Expression R_SQUARE_BRACKET`

Var_blk ⇒ `COMMA Var Var_blk` | ϵ

Term $\Rightarrow$ `SUB Var` | `Var` | `SUB NUMBER` | `NUMBER` | `SUB L_PAREN Expression R_PAREN` | `SUB L_PAREN Expression R_PAREN` | `IDENTIFIER L_PAREN Expression Expression_blk R_PAREN` | `IDENTIFIER L_PAREN R_PAREN`

Expression_blk $\Rightarrow$ `COMMA Expression Expression_blk` | ϵ