COMPILER CONSTRUCTION TERM PROJECT :

MODIFIED GRAMMAR AND FIRST & FOLLOW SETS

PRANJAL GUPTA 2013B4A7470P
TANAYA JHA 2013B3A7304P

BATCH NO : 82

GRAMMAR

1 <program> ⟶ <moduleDeclarations> <otherModules> <driverModule> <otherModules>

2 <moduleDeclarations> ⟶ <moduleDeclaration> <moduleDeclarations> | ε

3 <moduleDeclaration> ⟶ **DECLARE MODULE ID SEMICOL**

4 <otherModules> ⟶ <module> <otherModules> | ε

5 <driverModule> ⟶ **DRIVERDEF DRIVER PROGRAM DRIVERENDDEF** <moduleDef>

6 <module> ⟶ **DEF MODULE ID ENDDEF TAKES INPUT SQBO** <input_plist> **SQBC SEMICOL** <ret> <moduleDef>

7 <ret> ⟶ **RETURNS SQBO** <output_plist> **SQBC SEMICOL** | ε

8 <input_plist> ⟶ **ID COLON** <dataType> <input_plistRec>

9 <input_plistRec> ⟶ **COMMA ID COLON** <dataType> <input_plistRec> | ε

10 <output_plist> ⟶ **ID COLON** <type> <output_plistRec>

11 <output_plistRec> ⟶ **COMMA ID COLON** <type> <output_plistRec> | ε

12 <type> ⟶ **INTEGER** | **REAL** | **BOOLEAN**

13 <dataType> ⟶ <type> | **ARRAY SQBO** <range> **SQBC** OF <type>

14 <moduleDef> ⟶ **START** <statements> **END**

15 <statements> ⟶ <statement> <statements> | ε

16 <statement> ⟶ <ioStmt> | <simpleStmt> | <declareStmt> | <conditionalStmt> | <iterativeStmt> | **SEMICOL**

17	<ioStmt>	⟶	**GET_VALUE BO ID** <whichId> **BC SEMICOL** | **PRINT BO** <var> **BC SEMICOL**
18	<whichId>	⟶	**SQBO** <index> **SQBC** | ε
19	<index>	⟶	**NUM** | **ID**
20	<simpleStmt>	⟶	<assignmentStmt> | <moduleReuseStmt>
21	<assignmentStmt>	⟶	**ID** <whichId> **ASSIGNOP** <expression> **SEMICOL**
22	<moduleReuseStmt>	⟶	<optional> **USE MODULE ID WITH PARAMETERS** <idList> **SEMICOL**
23	<optional>	⟶	**SQBO** <idList> **SQBC ASSIGNOP** | ε
24	<idList>	⟶	**ID** <idListRec>
25	<idListRec>	⟶	**COMMA ID** <idListRec> | ε
26	<expression>	⟶	<arithOrBoolExpr> | **MINUS BO** <arithmeticExpr> **BC**
27	<arithOrBoolExpr>	⟶	<anyTerm> <arithOrBoolExprRec>
28	<arithOrBoolExprRec>	⟶	<logicalOp> <anyTerm> <arithOrBoolExprRec> | ε
29	<anyTerm>	⟶	<arithmeticExpr> <anyTermRec>
30	<anyTermRec>	⟶	<relationalOp> <arithmeticExpr> <anyTermRec> | ε
31	<arithmeticExpr>	⟶	<term> <arithmeticExprRec>
32	<arithmeticExprRec>	⟶	<pm> <term> <arithmeticExprRec> | ε
33	<term>	⟶	<factor> <termRec>
34	<termRec>	⟶	<md> <factor> <termRec> | ε
35	<factor>	⟶	**BO** <arithmeticExpr> **BC** | <var>
36	<var>	⟶	**ID** <whichId> | **NUM** | **RNUM** | **TRUE** | **FALSE**
37	<pm>	⟶	**PLUS** | **MINUS**
38	<md>	⟶	**MUL** | **DIV**
39	<logicalOp>	⟶	**AND** | **OR**
40	<relationalOp>	⟶	**LT** | **LE** | **GT** | **GE** | **EQ** | **NE**
41	<declareStmt>	⟶	**DECLARE** <idList> **COLON** <dataType> **SEMICOL**

42	<conditionalStmt>	⟶	**SWITCH BO ID BC START** <caseStmts> <default> **END**
43	<caseStmts>	⟶	**CASE** <value> **COLON** <statements> **BREAK SEMICOL** <caseStmtsRec>
	<caseStmtsRec>	⟶	**CASE** <value> **COLON** <statements> **BREAK SEMICOL** <caseStmtsRec> | ε
44	<value>	⟶	**NUM** | **TRUE** | **FALSE**
45	<default>	⟶	**DEFAULT COLON** <statements> **BREAK SEMICOL** | ε
46	<iterativeStmt>	⟶	**FOR BO ID IN** <range> **BC START** <statements> **END** | **WHILE BO** <arithOrBoolExpr> **BC START** <statements> **END**
47	<range>	⟶	**NUM RANGEOP NUM**

ASSUMPTIONS

1. There can be ‘Empty Statements’ in the language. This statement consist simply of a semicolon and performs no action **[Rule 16]**

2. GET_VALUE statement can take ID as well as an element of the array of type INTEGER, REAL, BOOLEAN **[Rule 17]**

3. PRINT statement can be used to print boolean constants TRUE and FALSE as well **[Rule 18]**