///

GROUP - 50

COMPILER CONSTRUCTION

GRAMMAR (FIRST & FOLLOW SETS)

→ TEAM MEMBERS

1) Harshit Gupta (2019B2A70898P)
2) Hash Pandey (2019B3A70489P)
3) Aryan Pawar (2019B3A70470P)
4) Tejas Deshpande (2019B3A70562P)
5) Krish Vora (2019B3A70819P)

MODIFIED GRAMMAR

*(■) - Means that the rule has either been modified or added.

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> <leftRec1>
9. <leftREC1> → COMMA ID COLON <dataType> <leftRec1> | ε
10. <output_plist> → ID COLON <type> <leftRec2>
11. <leftRec2> → COMMA ID COLON <type> <leftRec2> | ε
12. <dataType> → ARRAY SQBO <arr_range> SQBC OF <type>
13. <arr_range> → <index> RANGEOP <index>
14. <type> → INTEGER | REAL | BOOLEAN
15. <moduleDef> → START <statements> END
16. <statements> → <statement> <statements> | ε
17. <statement> → <ioStmt> | <simpleStmt> | <declareStmt> | <conditionalStmt> | <iterativeStmt>
18. <ioStmt> → GET_VALUE BO ID BC SEMICOL | PRINT BO <var> BC SEMICOL
19. <var> → <var_num> | <boolConst>
20. <var_num> → ID <whichId> | NUM | RNUM
21. <boolConst> → TRUE | FALSE
22. <whichId> → SQBO <index> SQBC | ε
23. <simpleStmt> → <assignmentStmt> | <moduleReuseStmt>
24. <assignmentStmt> → ID <whichStmt>
25. <whichStmt> → <lvalueIDStmt> | <lvalueARRStmt>
26. <lvalueIDStmt> → ASSIGNOP <expression> SEMICOL

<value ARR Stmt> → SQBO <index> SQBC ASSIGNOP <expression> SEMICOL

<index> → NUM | ID

<moduleReuseStmt> → <optional> USE MODULE ID WITH PARAMETERS <idList> SEMICOL

30. <optional> → SQBO <idList> SQBC ASSIGNOP | ε

31. <idList> → ID <leftRec3>

32. <leftRec3> → COMMA ID <leftRec3> | ε

33. <expression> → <arithBoolExpr> | <unaryExpr>

34. <unaryExpr> → <unaryOP> <termAhead>

35. <termAhead> → BO <arithmeticExpr> BC | <var_num>

36. <unaryOP> → PLUS | MINUS

37. <arithBoolExpr> → <someTerm> <leftRec4>

38. <leftRec4> → <logicalOp> <someTerm> <leftRec4> | ε

39. <someTerm> → <arithmeticExpr> <leftRec7> | <booleanExpr> <leftRec5>

40. <leftRec5> → <relationalOp> <arithmeticExpr> <leftRec5> | ε

41. <arithmeticExpr> → <term> <leftRec6>

42. <leftRec6> → <op_lower> <term> <leftRec6> | ε

43. <term> → <factor> <leftRec7>

44. <leftRec7> → <op_higher> <factor> <leftRec7> | ε

45. <factor> → BO <arithmeticExpr> BC | <var_num>

46. <opLower> → PLUS | MINUS

47. <opHigher> → MUL | DIV

48. <logicalOp> → AND | OR

49. <relationalOp> → LT | LE | GT | GE | EQ | NE

50. <declareStmt> → DECLARE <idList> COLON <datatype> SEMICOL

51. <conditionalStmt> → SWITCH BO ID BC START <caseStmt> <default> END

52. <caseStmt> → CASE <value> COLON <statements> BREAK SEMICOL <leftRec8>

53. <leftRec8> → CASE <value> COLON <statements> BREAK SEMICOL <leftRec8> | ε

<value> → NUM | TRUE | FALSE

. <default> → DEAD DEFAULT COLON <statements> BREAK SEMICOL | ε

6. <iterativestmt> → FOR BO ID IN <range> BC START <statements>
END | WHILE BO <arithBoolExpr> BC START <statements>
END

57. <range> → NUM RANGEOP NUM

*( ) denotes rules which were modified or added

	Non Terminals	FIRST	FOLLOW
	program	{DECLARE}	{$}
	moduleDeclarations	{DECLARE}	{DEF, DRIVERDEF, $}
	moduleDeclaration	{DECLARE}	{DECLARE, DEF, DRIVERDEF, $}
4.	otherModules	{DEF, ε}	{DRIVERDEF, $}
5.	driverModule	{DRIVERDEF}	{DEF, $}
6.	module	{DEF}	{DEF, $}
7.	ret	{RETURNS}	{START}
8.	leftRec1	{COMMA}	{SQBC}
9.	output_plist	{ID}	{SQBC}
10.	leftRec2	{COMMA, ε}	{SQBC}
11.	dataType	{ARRAY}	{SQBC}
12.	input_plist	{ID}	{SQBC}
13.	arr_range	{NUM, ID}	{SQBC}
14.	type	{INTEGER, REAL, BOOLEAN START}	{SQBC}
15.	moduleDef	{DEF, $}	
16.	statements	{GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE, ε}	{BREAK, END}
17.	statement	{GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE}	{GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE}
18.	ioStmt	{GET_VALUE, PRINT}	{GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE}
19.	var	{ID, NUM, RNUM, TRUE, FALSE}	{BC}
20.	var_num	{ID, NUM, RNUM}	{BC, SEMICOLON}
21.	booleanConst	{TRUE, FALSE}	{BC}
22.	whichId	{SQBO, ε}	{BC, SEMICOLON}
23.	simpleStmt	{ID, SQBO, USE}	{BREAK, END}
24.	assignmentStmt	{ID}	{BREAK, END}
25.	whichStmt	{ASSIGNOP, SQBO}	{BREAK, END}
26.	lvalueIDStmt	{ASSIGNOP}	{BREAK, END}
27.	lvalueARRStmt	{SQBO}	{BREAK, END}
28.	index	{NUM, ID}	{SQBC, RANGEOP}
29.	moduleReuseStmt	{SQBO, USE}	{BREAK, END}

Page 4

Non Terminals	FIRST	FOLLOW
optional	{SQBO, ε}	{USE}
idList	{ID}	{SEMICOL, SQBC, COLON}
leftRec3	{COMMA, ε}	{SEMICOL, SQBC, COLON}
33. expression	{PLUS, MINUS, BO, ID, NUM, RNUM, TRUE, FALSE}	{SEMICOL}
34. unaryExpr	{PLUS, MINUS}	{SEMICOL}
35. termAhead	{BO, ID, NUM, RNUM}	{SEMICOL}
36. unaryOp	{PLUS, MINUS}	{SEMICOL}
37. arithBoolExpr	{BO, ID, NUM, RNUM, TRUE, FALSE}	{SEMICOL, BC}
38. leftRec4	{AND, OR, ε}	{SEMICOL, BC}
39. someTerm	{BO, ID, NUM, RNUM, TRUE, FALSE}	{SEMICOL, BC}
40. leftRec5	{LT, LE, GT, GE, EQ, NE, ε}	{SEMICOL, BC}
41. arithmeticExpr	{BO, ID, NUM, RNUM}	{BC, SEMICOL, LT, LE, GT, GE, NQ, NE}
42. term	{BO, ID, NUM, RNUM}	{PLUS, MINUS, BC, SEMICOL, LT, LE, GT, GE, NQ, NE}
43. leftRec6	{PLUS, MINUS, ε}	{BC, SEMICOL, LT, LE, GT, GE, NQ, NE}
44. leftRec7	{MUL, DIV, ε}	{PLUS, MIUS, BC, SEMICOL, LT, LE, GT, GE, NQ, NE}
45. factor	{BO, ID, NUM, RNUM}	{MUL, DIV, PLUS, MINUS, BC, SEMICOL, LT, LE, GT, GE, NQ, NE}
46. opLow	{PLUS, MINUS}	{BO, ID, NUM, RNUM}
47. opHigh	{MUL, DIV}	{BO, ID, NUM, RNUM}
48. logicalOp	{AND, OR}	{BO, ID, NUM, RNUM, TRUE, FALSE}
49. relationalOp	{LT, LE, GT, GE, NQ, NE}	{BO, ID, NUM, RNUM}
50. declareStmt	{DECLARE}	{GET-VALUE, PRINT, ID, SQBO, DECLARE, SWITCH FOR, WHILE}

	Non-Terminals	FIRST	FOLLOW
	conditionalStmt	{SWITCH}	{GET-VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE}
	caseStmt	{CASE}	{DEFAULT, AND}
53.	leftRec8	{CASE, ε}	{DEFAULT, AND}
54.	value	{NUM, TRUE, FALSE}	{COLON}
55.	default	{DEFAULT, ε}	{END}
56.	iterativeStmt	{FOR, WHILE}	{GET-VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE}
57.	range	{NUM}	{BC}