• Vatsal Pattni — 2019B5A70697P
• Aviral Omar — 2019B3A70411P
• Aman Agarwal — 2019B2A70966P
• Chandra Sekhar Reddy E — 2019B4A70634P
Group-51

Grammar

<program> → <moduleDeclarations> <otherModules> <driverModule> <otherModules>

<moduleDeclarations> → <moduleDeclaration> <moduleDeclarations> | ε

<moduleDeclaration> → DECLARE MODULE ID SEMICOL

<otherModules> → <module> <otherModules> | ε

<driverModule> → DRIVERDEF DRIVER PROGRAM DRIVERENDDEF <moduleDef>

<module> → DEF MODULE ID ENDDEF TAKES INPUT SQBO <inputPlist> SQBC SEMICOL <ret> <moduleDef>

<ret> → RETURNS SQBO <outputPlist> SQBC SEMICOL | ε

<inputPlist> → ID COLON <dataType> <LR1>

<LR1> → COMMA ID COLON <dataType> <LR1> | ε

<outputPlist> → ID COLON <type> <LR2>

<LR2> → COMMA ID COLON <type> <LR2> | ε

<dataType> → ARRAY SQBO <arrRange> SQBC OF <type>

<arrRange> → <index> RANGEOP <index>

<type> → INTEGER | REAL | BOOLEAN

<moduleDef> → START <statements> END

<statements> → <statement> <statements> | ε

<statement> → <ioStmt> | <simpleStmt> | <declareStmt> | <ConditionalStmt> | <iterativeStmt>

<ioStmt> → GET_VALUE BO ID BC SEMICOL | PRINT BO <var> BC SEMICOL

<BooleanConst> → TRUE | FALSE

<varNum> → ID <whichId> | NUM | RNUM

<var> → <varNum> | <BooleanConst>

<whichId> → SQBO <index> SQBC | ε

<simpleStmt> → <assignmentStmt> | <moduleReuseStmt>

<assignmentStmt> → ID <whichStmt>

<whichStmt> → <lvalueIDStmt> | <lvalueARRStmt>

<lvalueIDStmt> → ASSIGNOP <expression> SEMICOL

<lvalueARRStmt> → SQBO <index> SQBC ASSIGNOP
<expression> SEMICOL

<index> → NUM | ID

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

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

<idList> → ID <LR3>

<LR3> → COMMA ID <LR3> | ε

<expression> → <arithBoolExpr> | <unaryExpr>

<unaryExpr> → <unaryOp> <NT>

<NT> → BO <arithmeticExpr> BC | <varNum>

<unaryOp> → PLUS | MINUS

<arithBoolExpr> → <arbitraryTerm> <LR6>

<LR6> → <logicalOp> <arbitraryTerm> <LR6> | ε

<arbitraryTerm> → <arithmeticExpr> <LR7> | <booleanConst>

<LR7> → <relationalOp> <arithmeticExpr> <LR7> | ε

<arithmeticExpr> → <term> <LR4>

<LR4> → <opLower> <term> <LR4> | ε

<term> → <factor> <LR5>

<LR5> → <opHigher> <factor> <LR5> | ε

<factor> → BO <arithmeticExpr> BC

<factor> → <varNum>

<op lower> ⟶ PLUS | MINUS

<op higher> ⟶ MUL | DIV

<logical Op> ⟶ AND | OR

<relational Op> ⟶ LT | LE | GT | GE | EQ | NE

<declare Stmt> ⟶ DECLARE <idList> COLON <dataType> SEMICOL

<conditional Stmt> ⟶ SWITCH BO ID BC START <caseStmt> <default> END

<case Stmt> ⟶ CASE <value> COLON <statements> BREAK SEMICOL <LRB>

<LRB> ⟶ CASE <value> COLON <statements> BREAK SEMICOL <LRB> | ε

<value> ⟶ NUM | TRUE | FALSE

<default> ⟶ DEFAULT COLON <statements> BREAK SEMICOL | ε

<iterative Stmt> ⟶ FOR BO ID IN <range> BC START <statements> END | WHILE BO <arithBoolExpr> BC START <statements> END

<range> ⟶ NUM RANGEOP NUM

Non-Terminal	First	Follow
<program>	DECLARE	$
<moduleDeclarations>	DECLARE	DEF, DRIVERDEF, $
<moduleDeclaration>	DECLARE	DECLARE, DEF, DRIVERDEF, $
<otherModules>	DEF, ε	DRIVERDEF, $
<driverModule>	DRIVERDEF	DRIVER DEF, DEF, $
<module>	DEF	DRIVERDEF, $
<ret>	RETURNS	START
<inputPlist>	ID	SQBC
<LR1>	COMMA	SQBC
<outputPlist>	ID	SQBC
<LR2>	COMMA, ε	SQBC
<dataType>	ARRAY	SQBC, SEMICOL
<arrRange>	NUM, ID	SQBC
<type>	INTEGER, REAL, BOOLEAN	SQBC, SEMICOL
<moduleDef>	START	DRIVERDEF, $
<statements>	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE, ε	END, BREAK
<statement>	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE
<SOStmt>	GET_VALUE, PRINT	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE
<booleanConst>	TRUE, FALSE	BC
<varNum>	ID, NUM, RNUM	BC, SEMICOL
<var>	ID, NUM, RNUM, TRUE, FALSE	BC

<whichId>	SQBO, ε	BC, SEMICOL
<simpleStmt>	ID, SQBO, USE	END, BREAK
<assignmentStmt>	ID	END, BREAK
<whichStmt>	ASSIGNOP, SQBO	END, BREAK
<lvalueIDStmt>	ASSIGNOP	END, BREAK
<lvalueARRStmt>	SQBO	END, BREAK
<index>	NUM, ID	RANGEOP, SQBC
<moduleReuseStmt>	SQBO, USE	END, BREAK
<optional>	SQBO, ε	USE
<idList>	ID	SEMICOL, SQBC, COLON
<LR8>	COMMA, ε	SEMICOL, SQBC, COLON
<expression>	ID, NUM, RNUM, BO, TRUE, FALSE, PLUS, MINUS	SEMICOL
<unaryExpr>	PLUS, MINUS	SEMICOL
<NT>	BO, ID, NUM, RNUM	SEMICOL
<unaryOp>	PLUS, MINUS	SEMICOL
<arithBoolExpr>	BO, ID, NUM, RNUM, TRUE, FALSE	SEMICOL, BC
<LR6>	AND, OR, ε	SEMICOL, BC
<arithExpr>	BO, ID, NUM, RNUM, TRUE, FALSE	SEMICOL, BC
<LR7>	LT, LE, GT, GE, EQ, NE, ε	SEMICOL, BC
<arithmeticExpr>	BO, ID, NUM, RNUM	SEMICOL, BC, NE, LT, GE, LE, GT, EQ.
<LR4>	PLUS, MINUS, ε	BC, SEMICOL, LT, LE, GE, NE, EQ, GT
<term>	BO, ID, NUM, RNUM	PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, SEMICOL, BC
<LR5>	MUL, DIV, ε	PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, SEMICOL

<factor>	BO, ID, NUM, RNUM	MUL, DIV, PLUS, MINUS, BC, LT, LE, GT, GE, EQ, NE, SEMICOL, BC
<lowPrecedenceOp>	PLUS, MINUS	BO, ID, RNUM, NUM
<highPrecedenceOp>	MUL, DIV	BO, ID, RNUM, NUM
<logicalOp>	AND, OR	BO, ID, NUM, RNUM, TRUE, FALSE
<relationalOp>	LT, LE, GE, GT, EQ, NE	BO, ID, NUM, RNUM
<declareStmt>	DECLARE	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE
~~<switchStmt>~~	~~SWITCH~~	
<conditionalStmt>	SWITCH	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE
<caseStmts>	CASE	DEFAULT, END
<LRB>	CASE, E	DEFAULT
<value>	NUM, TRUE, FALSE	COLON
<default>	DEFAULT, E	END
~~<iterativeStmt>~~	~~FOR, WHILE~~	~~END, BREAK~~
<range>	NUM	BC
<iterativeStmt>	~~FOR~~ FOR, WHILE	GET_VALUE, PRINT, ID, SQBO, DECLARE, SWITCH, FOR, WHILE