

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 DRIVERENDDDEF**<moduleDef>
- 6) <module> -> **DEF MODULE ID ENDDDEF TAKES INPUT SQBO**<input_plist>**SQBC SEMICOL**
<ret><moduleDef>
- 7) <ret> -> **RETURNS SQBO** <output_plist> **SQBC SEMICOL** | ε
- 8) <input_plist> -> **ID COLON**<dataType><iplist>
- 9) <iplist> -> **COMMA ID COLON** <dataType><iplist> | ε
- 10) <output_plist> -> **ID COLON** <type><oplist>
- 11) <oplist> -> **COMMA ID COLON** <type><oplist> | ε
- 12) <dataType> -> **INTEGER** | **REAL** | **BOOLEAN** | **ARRAY SQBO**<dynamic_range>**SQBC OF** <type>
- 13) <dynamic_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**<print>**BC SEMICOL**
- 19) <print> -> <var>|<boolConst>
- 20) <var> -> **ID** <whichId> | **NUM** | **RNUM**
- 21) <whichId> -> **SQBO**<index>**SQBC** | ε
- 22) <simpleStmt> -> <assignmentStmt> | <moduleReuseStmt>
- 23) <assignmentStmt> -> **ID** <whichStmt>
- 24) <whichStmt> -> <lvalueIDStmt> | <lvalueARRStmt>
- 25) <lvalueIDStmt> -> **ASSIGNOP** <expression_new> **SEMICOL**
- 26) <lvalueARRStmt> -> **SQBO**<index>**SQBC ASSIGNOP** <expression_new> **SEMICOL**
- 27) <index> -> **NUM** | **ID**
- 28) <moduleReuseStmt> -> <optional>**USE MODULE ID WITH PARAMETERS**<idList>
- 29) <optional> -> **SQBO** <idList> **SQBC ASSIGNOP** | ε
- 30) <idList> -> **ID**<idLists>
- 31) <idLists> -> **COMMA ID**<idLists> | ε
- 32) <expression_new> -> <U> | <arithmeticOrBooleanExpression>
- 33) <U> -> <op1><U’>
- 34) <U’> -> **BO**<arithmeticExpr>**BC** | <var>
- 35) <arithmeticOrBooleanExpression> -> <boolTerm> <followingBool>
- 36) <followingBool> -> <logicalOp> <boolTerm> <followingBool> | ε
- 37) <boolTerm> -> <arithmeticExpr> <boolean> | <boolConst>
- 38) <boolean> -> <relationalOp> <arithmeticExpr> | ε
- 39) <arithmeticExpr> -> <term> <followingArithExp>
- 40) <followingArithExp> -> <op1> <term> <followingArithExp> | ε
- 41) <term> -> <factor> <followingTerm>
- 42) <followingTerm> -> <op2> <factor> <followingTerm> | ε
- 43) <factor> -> **BO**<arithmeticOrBooleanExpression>**BC** | <var>
- 44) <op1> -> **PLUS** | **MINUS**
- 45) <op2> -> **MUL** | **DIV**
- 46) <logicalOp> -> **AND** | **OR**
- 47) <relationalOp> -> **LT** | **LE** | **GT** | **GE** | **EQ** | **NE**
- 48) <declareStmt> -> **DECLARE**<idList> **COLON** <dataType> **SEMICOL**
- 49) <conditionalStmt> -> **SWITCH BO ID BC START** <caseStmts> <default> **END**
- 50) <caseStmts> -> **CASE**<value> **COLON** <statements> **BREAK SEMICOL** <caseStmt>
- 51) <caseStmt> -> **CASE**<value> **COLON** <statements> **BREAK SEMICOL** <caseStmt> | ε
- 52) <value> -> **NUM** | <boolConst>
- 53) <default> -> **DEFAULT COLON** <statements> **BREAK SEMICOL** | ε
- 54) <iterativeStmt> -> **FOR BO ID IN** <range> **BC START** <statements> **END** |
WHILE BO<arithmeticOrBooleanExpression>**BC START**<statements>**END**
- 55) <range> -> **NUM RANGEOP NUM**
- 56) <boolConst> -> **TRUE**|**FALSE**