

Grammar for the ERPLAG Language

<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 <input_plist> SQBC SEMICOL <ret> <moduleDef>

<ret> ::= RETURNS SQBO <output_plist> SQBC SEMICOL | ε

<input_plist> ::= ID COLON <dataType> <sub_input_plist>

<sub_input_plist> ::= COMMA ID COLON <dataType> <sub_input_plist> | ε

<output_plist> ::= ID COLON <type> <sub_output_plist>

<sub_output_plist> ::= COMMA ID COLON <type> <sub_output_plist> | ε

<dataType> ::= INTEGER | REAL | BOOLEAN | ARRAY SQBO <dynamic_range> SQBC OF <type>

<dynamic_range> ::= <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 <extended_var> BC SEMICOL

<extended_var> ::= <var> | TRUE | FALSE

<var> ::= ID <whichId> | NUM | RNUM

<whichId> ::= SQBO <index> SQBC | ε

<simpleStmt> ::= <assignmentStmt> | <moduleReuseStmt>

<assignmentStmt> ::= ID <whichStmt>

<whichStmt> ::= <lvalueIDStmt> | <lvalueARRStmt>

<lvalueIDStmt> ::= ASSIGNOP <new_expression> SEMICOL

<lvalueARRStmt> ::= SQBO <index> SQBC ASSIGNOP <new_expression> SEMICOL

<index> ::= NUM | ID

<moduleReuseStmt> ::= <optional> USE MODULE ID WITH PARAMETERS <idList> SEMICOL

<optional> ::= SQBO <idList> SQBC ASSIGNOP | ϵ

<idList> ::= ID <sub_idList>

<sub_idList> ::= COMMA ID <sub_idList> | ϵ

<new_expression> ::= <u> | <expression>

<u> ::= PLUS <sub_u> | MINUS <sub_u>

<sub_u> ::= BO <arithmeticExpression> BC | <var>

<expression> ::= <arithmeticExpr> | <booleanExpr>

<arithmeticExpr> ::= <term> <sub_arithmeticExpr>

<sub_arithmeticExpr> ::= <op1> <term> <sub_arithmeticExpr> | ϵ

<term> ::= <factor> <sub_term>

<sub_term> ::= <op2> <factor> <sub_term> | ϵ

<factor> ::= BO <arithmeticExpr> BC | <var>

<op1> ::= PLUS | MINUS

<op2> ::= MUL | DIV

<booleanExpr> ::= <sub_booleanExpr> <logicalOp> <booleanExpr> | <sub_booleanExpr>

<sub_booleanExpr> ::= <arithmeticExpr> <relationalOp> <arithmeticExpr> | BO <booleanExpr> BC | TRUE | FALSE

<logicalOp> ::= AND | OR

<relationalOp> ::= LT | LE | GT | GE | EQ | NE

<declareStmt> ::= DECLARE <idList> COLON <dataType> SEMICOL

<conditionalStmt> ::= SWITCH BO ID BC START <caseStmt> <default> END

<caseStmt> ::= CASE <value> COLON <statements> BREAK SEMICOL <nullableCaseStmt>

<nullableCaseStmt> ::= CASE <value> COLON <statements> BREAK SEMICOL <caseStmt> | ϵ

<value> ::= NUM | TRUE | FALSE

<default> ::= DEFAULT COLON <statements> BREAK SEMICOL | ϵ

<iterativeStmt> ::= FOR BO ID IN <range> BC START <statements> END |
WHILE BO <booleanExpr> BC START <statements> END

<range> ::= NUM RANGEOP NUM