GRAMMAR ENCODING - GROUP_34
ADITYA KANTHI - SHIRISH KUMARAVEL - ARVIND RAM - ARYAN KAPOOR

Grammar[0]
\
|
|1) iterativeStmt ---> FOR BO ID IN for_range BC START statements END
|2) iterativeStmt ---> WHILE BO abExpr BC START statements END
Grammar[1]
\
|
|1) default_stmt ---> DEFAULT COLON statements BREAK SEMICOL
|2) default_stmt ---> EPS
Grammar[2]
\
|
|1) value ---> NUM
|2) value ---> TRUE_
|3) value ---> FALSE_
Grammar[3]
\
|
|1) caseStmts ---> CASE value COLON statements BREAK SEMICOL caseStmts2
Grammar[4]
\
|
|1) caseStmts2 ---> CASE value COLON statements BREAK SEMICOL caseStmts2
|2) caseStmts2 ---> EPS
Grammar[5]
\
|
|1) conditionalStmt ---> SWITCH BO ID BC START caseStmts default_stmt END
Grammar[6]
\
|
|1) declareStmt ---> DECLARE idList COLON dataType SEMICOL
Grammar[7]
\
|
|1) relationalOp ---> LT
|2) relationalOp ---> LE
|3) relationalOp ---> GT
|4) relationalOp ---> GE
|5) relationalOp ---> EQ
|6) relationalOp ---> NE
Grammar[8]
\
|

```
        |1 ) logicalOp ---> OR
        |2 ) logicalOp ---> AND
Grammar[9]
        \
        |
        |1 ) AnyTerm ---> arithmeticExpr   N8
        |2 ) AnyTerm ---> boolConst
Grammar[10]
        \
        |
        |1 ) abExpr ---> AnyTerm   N7
Grammar[11]
        \
        |
        |1 ) unary_op ---> PLUS
        |2 ) unary_op ---> MINUS
Grammar[12]
        \
        |
        |1 ) non_term ---> BO  arithmeticExpr  BC
        |2 ) non_term ---> var_const
Grammar[13]
        \
        |
        |1 ) U ---> unary_op  non_term
Grammar[14]
        \
        |
        |1 ) high_op ---> MUL
        |2 ) high_op ---> DIV
Grammar[15]
        \
        |
        |1 ) low_op ---> PLUS
        |2 ) low_op ---> MINUS
Grammar[16]
        \
        |
        |1 ) factor ---> BO  abExpr  BC
        |2 ) factor ---> NUM
        |3 ) factor ---> RNUM
        |4 ) factor ---> boolConst
        |5 ) factor ---> ID  factor2
Grammar[17]
        \
        |
        |1 ) factor2 ---> EPS
        |2 ) factor2 ---> SQBO  exprIndex  SQBC
```

Grammar[18]
\
|
|1) exprIndex ---> sign exprIndex2
|2) exprIndex ---> arrExpr
Grammar[19]
\
|
|1) exprIndex2 ---> index2
|2) exprIndex2 ---> BO arrExpr BC
Grammar[20]
\
|
|1) arrExpr ---> arrTerm arrExpr2
Grammar[21]
\
|
|1) arrExpr2 ---> EPS
|2) arrExpr2 ---> low_op arrTerm arrExpr2
Grammar[22]
\
|
|1) arrTerm ---> arrFactor arrTerm2
Grammar[23]
\
|
|1) arrTerm2 ---> high_op arrFactor arrTerm2
|2) arrTerm2 ---> EPS
Grammar[24]
\
|
|1) arrFactor ---> ID
|2) arrFactor ---> NUM
|3) arrFactor ---> boolConst
|4) arrFactor ---> BO arrExpr BC
Grammar[25]
\
|
|1) term2 ---> EPS
|2) term2 ---> high_op factor term2
Grammar[26]
\
|
|1) term ---> factor term2
Grammar[27]
\
|
|1) arithmeticExpr ---> term arithmeticExpr2

Grammar[28]
\
|
|1) arithmeticExpr2 ---> low_op term arithmeticExpr2
|2) arithmeticExpr2 ---> EPS
Grammar[29]
\
|
|1) expression ---> abExpr
|2) expression ---> U
Grammar[30]
\
|
|1) idList2 ---> COMMA ID idList2
|2) idList2 ---> EPS
Grammar[31]
\
|
|1) idList ---> ID idList2
Grammar[32]
\
|
|1) optional ---> SQBO idList SQBC ASSIGNOP
|2) optional ---> EPS
Grammar[33]
\
|
|1) moduleReuseStmt ---> optional USE MODULE ID WITH PARAMETERS
actual_para_list SEMICOL
Grammar[34]
\
|
|1) lvalueARRStmt ---> SQBO exprIndex SQBC ASSIGNOP expression SEMICOL
Grammar[35]
\
|
|1) lvalueIDStmt ---> ASSIGNOP expression SEMICOL
Grammar[36]
\
|
|1) sign ---> EPS
|2) sign ---> MINUS
|3) sign ---> PLUS
Grammar[37]
\
|
|1) whichStmt ---> lvalueIDStmt
|2) whichStmt ---> lvalueARRStmt

Grammar[38]
\
|
|1) assignmentStmt ---> ID whichStmt
Grammar[39]
\
|
|1) simpleStmt ---> assignmentStmt
|2) simpleStmt ---> moduleReuseStmt
Grammar[40]
\
|
|1) program ---> moduleDeclarations otherModules driverModule otherModules
Grammar[41]
\
|
|1) moduleDeclarations ---> moduleDeclaration moduleDeclarations
|2) moduleDeclarations ---> EPS
Grammar[42]
\
|
|1) moduleDeclaration ---> DECLARE MODULE ID SEMICOL
Grammar[43]
\
|
|1) otherModules ---> module otherModules
|2) otherModules ---> EPS
Grammar[44]
\
|
|1) driverModule ---> DRIVERDEF DRIVER PROGRAM DRIVERENDDEF moduleDef
Grammar[45]
\
|
|1) module ---> DEF MODULE ID ENDDEF TAKES INPUT SQBO input_plist SQBC
SEMICOL ret moduleDef
Grammar[46]
\
|
|1) ret ---> RETURNS SQBO output_plist SQBC SEMICOL
|2) ret ---> EPS
Grammar[47]
\
|
|1) input_plist ---> ID COLON dataType input_plist2
Grammar[48]
\
|

|1) input_plist2 ---> COMMA ID COLON dataType input_plist2
|2) input_plist2 ---> EPS
Grammar[49]
\
|
|1) output_plist ---> ID COLON type output_plist2
Grammar[50]
\
|
|1) output_plist2 ---> COMMA ID COLON type output_plist2
|2) output_plist2 ---> EPS
Grammar[51]
\
|
|1) dataType ---> INTEGER
|2) dataType ---> REAL
|3) dataType ---> BOOLEAN
|4) dataType ---> ARRAY SQBO arr_range SQBC OF type
Grammar[52]
\
|
|1) arr_range ---> arr_index RANGEOP arr_index
Grammar[53]
\
|
|1) var_const ---> ID
|2) var_const ---> NUM
|3) var_const ---> RNUM
Grammar[54]
\
|
|1) type ---> INTEGER
|2) type ---> REAL
|3) type ---> BOOLEAN
Grammar[55]
\
|
|1) moduleDef ---> START statements END
Grammar[56]
\
|
|1) statements ---> statement statements
|2) statements ---> EPS
Grammar[57]
\
|
|1) statement ---> ioStmt
|2) statement ---> simpleStmt

|3) statement ---> declareStmt
|4) statement ---> conditionalStmt
|5) statement ---> iterativeStmt
Grammar[58]
\
|
|1) ioStmt ---> PRINT BO print_var BC SEMICOL
|2) ioStmt ---> GET_VALUE BO ID BC SEMICOL
Grammar[59]
\
|
|1) print_var ---> ID N1
|2) print_var ---> NUM
|3) print_var ---> RNUM
|4) print_var ---> boolConst
Grammar[60]
\
|
|1) boolConst ---> TRUE_
|2) boolConst ---> FALSE_
Grammar[61]
\
|
|1) N1 ---> SQBO sign index2 SQBC
|2) N1 ---> EPS
Grammar[62]
\
|
|1) arr_index ---> sign index2
Grammar[63]
\
|
|1) index2 ---> ID
|2) index2 ---> NUM
Grammar[64]
\
|
|1) actual_para_list ---> sign K N9
Grammar[65]
\
|
|1) actual_para_list2 ---> SQBO exprIndex SQBC
|2) actual_para_list2 ---> EPS
Grammar[66]
\
|
|1) N7 ---> logicalOp AnyTerm N7
|2) N7 ---> EPS

Grammar[67]
\
|
|1) N8 ---> relationalOp arithmeticExpr
|2) N8 ---> EPS
Grammar[68]
\
|
|1) N9 ---> COMMA sign K N9
|2) N9 ---> EPS
Grammar[69]
\
|
|1) K ---> NUM
|2) K ---> RNUM
|3) K ---> boolConst
|4) K ---> ID actual_para_list2
Grammar[70]
\
|
|1) for_index ---> for_sign for_index2
Grammar[71]
\
|
|1) for_index2 ---> NUM
Grammar[72]
\
|
|1) for_sign ---> EPS
|2) for_sign ---> MINUS
|3) for_sign ---> PLUS
Grammar[73]
\
|
|1) for_range ---> for_index RANGEOP for_index