Parham Alvani

9231058

phase 2

Compiler Course Fall-2016

Saman Ferki

9231075

# Lex Regular Expressions

t\_RECORD\_KW **=** r'record'

t\_IF\_KW **=** r'if'

t\_THEN\_KW **=** r'then'

t\_ELSE\_KW **=** r'else'

t\_SWITCH\_KW **=** r'switch'

t\_CASE\_KW **=** r'case'

t\_END\_KW **=** r'end'

t\_WHILE\_KW **=** r'while'

t\_DEFAULT\_KW **=** r'default'

t\_RETURN\_KW **=** r'return'

t\_BREAK\_KW **=** r'break'

t\_STATIC\_KW **=** r'static'

t\_NOT\_KW **=** r'not'

t\_AND\_KW **=** r'and'

t\_OR\_KW **=** r'or'

t\_TRUE **=** r'true'

t\_FALSE **=** r'false'

t\_SEMICOLON **=** r';'

t\_COLON **=** r':'

t\_DOT **=** r'\.'

t\_COMMA **=** r','

t\_BR\_OPEN **=** r'\{'

t\_BR\_CLOSE **=** r'\}'

t\_PR\_OPEN **=** r'\('

t\_PR\_CLOSE **=** r'\)'

t\_BK\_OPEN **=** r'\['

t\_BK\_CLOSE **=** r'\]'

t\_COMMENTS **=** r'\/\/.\*'

t\_INT\_T **=** r'int'

t\_BOOL\_T **=** r'bool'

t\_REAL\_T **=** r'real'

t\_CHAR\_T **=** r'char'

t\_REL\_OP **=** r'\.eq | \.gt | \.ge | \.lt | \.le | \.ne'

t\_MATH\_OP **=** r'\+ | \- | \\* | \/ | % | \?'

t\_EXP\_OP **=** r'='

t\_ID **=** r'\#[a-zA-Z]{2}[0-9]{2}'

t\_FAKE\_ID **=** r'\#[a-zA-Z]{2}[0-9]{2}[\w]+'

t\_CHARCONST **=** r"'\\?[\w'\\]'"

t\_REALCONST **=** r'\d\*\.\d+'

t\_NUMCONST **=** r'\d+'

t\_ignore **=** ' \t\r\f\v'

# Sample Source Code

record #po11 {

int #xx11, #yy11;

}

record #li11 {

record #po11 #xx11, #yy11;

}

int #at11 (int #ba12, #ca23[]; bool #do43, #el32; int #fo12)

{

int #gn11, #ho12[100];

real #el72;

record #po11 #aP11;

record #li11 #aL11;

record #li11 #tw33[2];

#aP11.#xx11 = 666;

#aP11.#yy11 = 667;

#el72 = 72.20;

#aL11.#xx11.#xx11 = 1; #aL11.#xx11.#yy11 = 2; #aL11.#yy11.#xx11 = 3; #aL11.#yy11.#yy11 = 4;

#tw33[0].#xx11.#xx11 = 42; #tw33[1].#yy11.#xx11 = 43;

#gn11 = #ho12[2] = 3\*\* #ca23; // hog is 3 times the size of array passed to cat

if (#do43 and #el32 or #ba12 .gt #ca23[3]) #do43 = not #do43;

else #fo12++;

if (#ba12 .le #fo12) {

while (#do43) {

static int #ho12; // hog in new scope

#ho12 = #fo12;

#do43 = #fr77(#fo12++, #ca23) .lt 666;

if (#ho12 .gt #ba12) break;

else if (#fo12 .ne 0) #fo12 += 7;

}

}

#fo12 = ?5;

switch (#fo12)

case 0: {

#fo12++;

break; }

case 1: {

#fo12--;

break; }

default:

break;

end

return (#fo12+#ba12 \*#ca23 [#ba12])/- #fo12;

}

// note that functions are defined using a statement

int #ma11(int #aa11, #bb11)

if (#aa11 .gt #bb11) return #aa11; else return #bb11;

# Grammar Rules

precedence = (

('left', 'OR\_KW', 'ORELSE'),

('left', 'AND\_KW', 'ANDTHEN'),

('left', 'EQ', 'NE'),

('left', 'LT', 'GT', 'LE', 'GE'),

('left', 'PLUS', 'MINUS'),

('left', 'REM'),

('left', 'MULT', 'DIV'),

('right', 'NOT\_KW', 'UMINUS', 'UMULT', 'RANDOM',

'MINUSMINUS', 'PLUSPLUS'),

('nonassoc', 'IFTHEN'),

('nonassoc', 'ELSE\_KW'),

)

'program : declarationList'

declarationList : declarationList declaration

| declaration

declaration : varDeclaration

declaration : funDeclaration

declaration : recDeclaration

recDeclaration : RECORD\_KW ID BR\_OPEN localDeclarations BR\_CLOSE

varDeclaration : typeSpecifier varDeclarationList SEMICOLON

scopedVarDeclaration : scopedTypeSpecifier varDeclarationList SEMICOLON

varDeclarationList : varDeclarationList COMMA varDeclarationInitialize

| varDeclarationInitialize

varDeclarationInitialize : varDeclarationId

| varDeclarationId COLON simpleExpression

varDeclarationId : ID

| ID BK\_OPEN NUMCONST BK\_CLOSE

scopedTypeSpecifier : STATIC\_KW typeSpecifier

| typeSpecifier

typeSpecifier : returnTypeSpecifier

| RECORD\_KW ID

returnTypeSpecifier : INT\_T

returnTypeSpecifier : REAL\_T

returnTypeSpecifier : BOOL\_T

returnTypeSpecifier : CHAR\_T

funDeclaration : typeSpecifier ID PR\_OPEN params PR\_CLOSE statement

| ID PR\_OPEN params PR\_CLOSE statement

params : paramList

params : empty

paramList : paramList SEMICOLON paramTypeList

| paramTypeList

paramTypeList : typeSpecifier paramIdList

paramIdList : paramIdList COMMA paramId

| paramId

paramId : ID BK\_OPEN BK\_CLOSE

| ID

statement : expressionStmt

statement : compoundStmt

statement : selectionStmt

statement : iterationStmt

statement : returnStmt

compoundStmt : BR\_OPEN localDeclarations statementList BR\_CLOSE

localDeclarations : localDeclarations scopedVarDeclaration

| empty

statementList : statementList statement

| empty

expressionStmt : expression SEMICOLON

| SEMICOLON

selectionStmt : IF\_KW PR\_OPEN simpleExpression PR\_CLOSE statement %prec IFTHEN

selectionStmt : IF\_KW PR\_OPEN simpleExpression PR\_CLOSE statement ELSE\_KW statement

selectionStmt : SWITCH\_KW PR\_OPEN simpleExpression PR\_CLOSE caseElement defaultElement END\_KW

caseElement : CASE\_KW NUMCONST COLON statement

| caseElement CASE\_KW NUMCONST COLON statement

defaultElement : DEFAULT\_KW COLON statement

| empty

iterationStmt : WHILE\_KW PR\_OPEN simpleExpression PR\_CLOSE statement

returnStmt : RETURN\_KW SEMICOLON

| RETURN\_KW expression SEMICOLON

breakStmt : BREAK\_KW SEMICOLON

expression : mutable EXP expression

expression : mutable PLUSEXP expression

expression : mutable MINUSEXP expression

expression : mutable MULTEXP expression

expression : mutable DIVEXP expression

expression : simpleExpression

expression : mutable PLUSPLUS

expression : mutable MINUSMINUS

simpleExpression : simpleExpression OR\_KW simpleExpression

simpleExpression : simpleExpression AND\_KW simpleExpression

simpleExpression : simpleExpression OR\_KW ELSE\_KW simpleExpression %prec ORELSE

simpleExpression : simpleExpression AND\_KW THEN\_KW simpleExpression %prec ANDTHEN

simpleExpression : NOT\_KW simpleExpression

simpleExpression : relExpression

relExpression : mathlogicExpression relop mathlogicExpression

| mathlogicExpression

relop : LE

relop : LT

relop : GT

relop : GE

relop : EQ

relop : NE

mathlogicExpression : mathlogicExpression PLUS mathlogicExpression

mathlogicExpression : mathlogicExpression MINUS mathlogicExpression

mathlogicExpression : mathlogicExpression MULT mathlogicExpression

mathlogicExpression : mathlogicExpression REM mathlogicExpression

mathlogicExpression : mathlogicExpression DIV mathlogicExpression

mathlogicExpression : unaryExpression

unaryExpression : MINUS unaryExpression %prec UMINUS

unaryExpression : RANDOM unaryExpression

unaryExpression : MULT unaryExpression %prec UMULT

unaryExpression : factor

factor : immutable

factor : mutable

mutable : ID

| mutable BK\_OPEN expression BK\_CLOSE

| mutable DOT ID

immutable : PR\_OPEN expression PR\_CLOSE

immutable : call

immutable : constant

call : ID PR\_OPEN args PR\_CLOSE

args : argList

args : empty

argList : argList COMMA expression

| expression

constant : NUMCONST

constant : REALCONST

constant : CHARCONST

constant : TRUE

constant : FALSE

# Parsing

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 10: varDeclarationList -> varDeclarationList, varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 7: recDeclaration -> RECORD\_KW ID {localDeclarations}

Rule 6: declaration -> recDeclaration

Rule 3: declarationList -> declaration

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 19: typeSpecifier -> RECORD\_KW ID

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 10: varDeclarationList -> varDeclarationList, varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 7: recDeclaration -> RECORD\_KW ID {localDeclarations}

Rule 6: declaration -> recDeclaration

Rule 2: declarationList -> declarationList declaration

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 34: paramId -> ID

Rule 32: paramIdList -> paramId

Rule 33: paramId -> ID [ ]

Rule 31: paramIdList -> paramIdList , paramId

Rule 30: paramTypeList -> typeSpecifier paramIdList

Rule 29: paramList -> paramTypeList

Rule 22: returnTypeSpecifier -> BOOL\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 34: paramId -> ID

Rule 32: paramIdList -> paramId

Rule 34: paramId -> ID

Rule 31: paramIdList -> paramIdList , paramId

Rule 30: paramTypeList -> typeSpecifier paramIdList

Rule 28: paramList -> paramList; paramTypeList

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 34: paramId -> ID

Rule 32: paramIdList -> paramId

Rule 30: paramTypeList -> typeSpecifier paramIdList

Rule 28: paramList -> paramList; paramTypeList

Rule 26: params -> paramList

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 15: varDeclarationId -> ID [ NUMCONST ]

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 10: varDeclarationList -> varDeclarationList, varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 21: returnTypeSpecifier -> REAL\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 19: typeSpecifier -> RECORD\_KW ID

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 19: typeSpecifier -> RECORD\_KW ID

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 19: typeSpecifier -> RECORD\_KW ID

Rule 17: scopedTypeSpecifier -> typeSpecifier

Rule 15: varDeclarationId -> ID [ NUMCONST ]

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 45: statementList -> empty

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 105: constant -> REALCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 94: mutable -> mutable[expression]

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 94: mutable -> mutable[expression]

Rule 95: mutbale -> mutable.ID

Rule 95: mutbale -> mutable.ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 94: mutable -> mutable[expression]

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 89: unaryExpression -> MULT unaryExpression

Rule 86: mathlogicExpression -> unaryExpression

Rule 83: mathlogicExpression -> mathlogicExpression MULT mathlogicExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 68: simpleExpression -> simpleExpression AND\_KW simpleExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 77: relop -> GT

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 94: mutable -> mutable[expression]

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 67: simpleExpression -> simpleExpression OR\_KW simpleExpression

Rule 93: mutable -> ID

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 71: simpleExpression -> NOT\_KW simpleExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 93: mutable -> ID

Rule 65: expression -> mutable PLUSPLUS

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 49: selectionStmt -> IF\_KW (simpleExpression) statement ELSE\_KW statement

Rule 37: statement -> selectionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 75: relop -> LE

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 45: statementList -> empty

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 16: scopedTypeSpecifier -> STATIC\_KW typeSpecifier

Rule 14: varDeclarationId -> ID

Rule 12: varDeclarationInitialize -> varDeclarationId

Rule 11: varDeclarationList -> varDeclarationInitialize

Rule 9: scopedVarDeclaration -> scopedTypeSpecifier varDeclarationList;

Rule 42: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 45: statementList -> empty

Rule 93: mutable -> ID

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 93: mutable -> ID

Rule 65: expression -> mutable PLUSPLUS

Rule 103: argList -> expression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 102: argList -> argList, expression

Rule 100: args -> argList

Rule 99: call -> ID(args)

Rule 97: immutable -> call

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 76: relop -> LT

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 77: relop -> GT

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 58: breakStmt -> BREAK\_KW ;

Rule 40: statement -> breakStmt

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 80: relop -> NE

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 60: expression -> mutable PLUSEXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 48: selectionStmt -> IF\_KW (simpleExpression) statement

Rule 37: statement -> selectionStmt

Rule 49: selectionStmt -> IF\_KW (simpleExpression) statement ELSE\_KW statement

Rule 37: statement -> selectionStmt

Rule 44: statementList -> statementList statement

Rule 41: compoundStmt -> {localDeclarations statementList}

Rule 36: statement -> compoundStmt

Rule 55: iterationStmt -> WHILE\_KW (simpleExpression) statement

Rule 38: statement -> iterationStmt

Rule 44: statementList -> statementList statement

Rule 41: compoundStmt -> {localDeclarations statementList}

Rule 36: statement -> compoundStmt

Rule 48: selectionStmt -> IF\_KW (simpleExpression) statement

Rule 37: statement -> selectionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 104: constant -> NUMCONST

Rule 98: immutable -> constant

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 88: unaryExpression -> RANDOM unaryExpression

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 59: expression -> mutable EXP expression

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 45: statementList -> empty

Rule 93: mutable -> ID

Rule 65: expression -> mutable PLUSPLUS

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 58: breakStmt -> BREAK\_KW ;

Rule 40: statement -> breakStmt

Rule 44: statementList -> statementList statement

Rule 41: compoundStmt -> {localDeclarations statementList}

Rule 36: statement -> compoundStmt

Rule 51: caseElement -> CASE\_KW NUMCONST: statement

Rule 43: localDeclarations -> localDeclarations scopedVarDeclaration

Rule 45: statementList -> empty

Rule 93: mutable -> ID

Rule 66: expression -> mutable MINUSMINUS

Rule 46: expressionStmt -> expression;

Rule 35: statement -> expressionStmt

Rule 44: statementList -> statementList statement

Rule 58: breakStmt -> BREAK\_KW ;

Rule 40: statement -> breakStmt

Rule 44: statementList -> statementList statement

Rule 41: compoundStmt -> {localDeclarations statementList}

Rule 36: statement -> compoundStmt

Rule 52: caseElement -> caseElement CASE\_KW NUMCONST: statement

Rule 58: breakStmt -> BREAK\_KW ;

Rule 40: statement -> breakStmt

Rule 53: defaultElement -> DEFAULT\_KW: statement

Rule 50: selectionStmt -> SWITCH\_KW (simpleExpression) caseElement defaultElement END\_KW

Rule 37: statement -> selectionStmt

Rule 44: statementList -> statementList statement

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 93: mutable -> ID

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 94: mutable -> mutable[expression]

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 83: mathlogicExpression -> mathlogicExpression MULT mathlogicExpression

Rule 81: mathlogicExpression -> mathlogicExpression PLUS mathlogicExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 96: immutable -> (expression)

Rule 91: factor -> immutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 87: unraryExpression -> MINUS unaryExpression

Rule 86: mathlogicExpression -> unaryExpression

Rule 85: mathlogicExpression -> mathlogicExpression DIV mathlogicExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 57: returnStmt -> RETURN\_KW expression ;

Rule 39: statement -> returnStmt

Rule 44: statementList -> statementList statement

Rule 41: compoundStmt -> {localDeclarations statementList}

Rule 36: statement -> compoundStmt

Rule 24 funDeclaration -> typeSpecifier ID (params) statement

Rule 5: declaration -> funDeclartion

Rule 2: declarationList -> declarationList declaration

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 20: returnTypeSpecifier -> INT\_T

Rule 18: typeSpecifier -> returnTypeSpecifier

Rule 34: paramId -> ID

Rule 32: paramIdList -> paramId

Rule 34: paramId -> ID

Rule 31: paramIdList -> paramIdList , paramId

Rule 30: paramTypeList -> typeSpecifier paramIdList

Rule 29: paramList -> paramTypeList

Rule 26: params -> paramList

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 77: relop -> GT

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 73: relExpression -> mathlogicExpression relop mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 57: returnStmt -> RETURN\_KW expression ;

Rule 39: statement -> returnStmt

Rule 93: mutable -> ID

Rule 92: factor -> mutable

Rule 90: unaryExpression -> factor

Rule 86: mathlogicExpression -> unaryExpression

Rule 74: relExpression -> mathlogicExpression

Rule 72: simpleExpression -> relExpression

Rule 64: expression -> simpleExpression

Rule 57: returnStmt -> RETURN\_KW expression ;

Rule 39: statement -> returnStmt

Rule 49: selectionStmt -> IF\_KW (simpleExpression) statement ELSE\_KW statement

Rule 37: statement -> selectionStmt

Rule 24 funDeclaration -> typeSpecifier ID (params) statement

Rule 5: declaration -> funDeclartion

Rule 2: declarationList -> declarationList declaration

Rule 1: program -> declarationList