PHASE 2

Compiler Course Fall-2016

Saman Ferki
9231075

Parham Alvani
9231058

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