Lynne Coblammers
EECS 665
Lab 5 Report
2015.10.11

If you were permitted to use Yacc's built-in operator precedence support for binary operations, what parts of your specification would change and why?

Currently, my specifications include eight separate rules for binary operators, one for each level or precedence, and eight separate rules for terms, which allow the binary operators to be evaluated correctly according to precedence. The lowest level of precedence (bitwise or) calls into rules with higher level of precedence up to the highest (multiplication, division, and mod). The rules for the two highest levels are as follows:

```
binop1 : MUL
   | DIV
   | MOD
binop2 : ADD
   | SUB
term2 : term2 binop2 term1
   | term1
term1 : term1 binop1 exprNoBinop
   | exprNoBinop
```

However, if I were to use Yacc's built-in operator precedence support, I could have a single rule for all binary operators and a single rule for binary operator expressions. To differentiate their levels of precedence, I would list them along with their associativity in order from lowest precedence to highest. For example, if I the only operators were ADD, SUB, MUL and DIV, my rules would be as follows:

```
%left ADD SUB
%left MUL DIV

binop: ADD
        | SUB
        | MUL
        | DIV

term: term binop term
        | exprNoBinop
```