Homework 2: Grammar Precedence and Associativity (*Using jison*)

Stephen Wagstaff
CS 431
September 19, 2018

```
<e> ::= <e> + <e>
    |  <e> - <e>
    |  <e> * <e>
    |  <e> / <e>
    |  <e> ^ <e>
    |  <e> !
    |  <e> %
    |  ( <e> )
    |  - <e>
    |  E
    |  PI
    |  <NUMBER>
```

Note that your grammars do not consider white spaces and `<NUMBER>` *is anon-terminal that refers to numbers.*

Assignment

1. Your new grammar should have the following precedence. `+,-  *,/ ^ !,% -2`.
2. Your new grammar should enforce left associativity for `+, -, *, /` and right associativity for `^`
3. You should add non-terminals to resolve the ambiguity. In particular, you should add non-terminals called …
    - `<RootExp>`: any expression that does not include operators
    - `<NegExp>`: negation expression
    - `<UnaryExp>`: unary expressions ! and %
    - `<PowExp>`: exponentiation
    - `<MulExp>`: multiplication and division

```
<e> ::= <e> '+' <MulExp>
    | <e> '-' <MulExp>
    | <MulExp>
<MulExp> ::=  <MulExp> '*' <PowExp>
    | <MulExp> '/' <PowExp>
    | <PowExp>
<PowExp> ::= <UnaryExp> '^' <PowExp>
    | <UnaryExp>
<UnaryExp> ::= <NegExp> '!'
    |  <NegExp> '%'
    | <NegExp>
<NegExp> ::= '-' <RootExp>
    | <RootExp>
<RootExp> ::= '(' <e> ')'
    |  E
    |  PI
    |  <NUMBER>
```