Homework 2

January 31, 2014

1 Question 1

Consider the following BNF grammar for `<Exp>`.

```
<Exp>     ::= <Exp> + <MulExp>
            | <Exp> - <MulExp>
            | <MulExp>
<MulExp> ::= <MulExp> * <NegExp>
            | <MulExp> / <NegExp>
            | <NegExp>
<NegExp> ::=  - <RootExp>
            | <RootExp>
<RootExp> ::= ( <Exp> )
            | 1 | 2 | 3 | 4
```

1. Extend the grammar by exponentiation such as `2^3`. Exponentiation should have higher priority than binary operations `+`, `-`, `/`, and `*`, but lower priority than negation operator `-`. Exponentiation is right associative. That is, `2^3^4` is the same as `2^(3^4)`, which yields 2^{81} and not just $4096 = (2^3)^4$.

2. Draw a parse tree for `-2^3^4` according to the new BNF grammar.

3. Draw abstract syntax trees for the the following terms `-2^3^4` and `-(2^3^4)`. (Note that parenthesis are omitted in the AST.)

2 Question 2

Give an **EBNF** grammar for floating point numbers that are made up of a fraction (e.g. 5.6 or 3.123 or -2.5) followed by an optional exponent (e.g. E10 or E-10). The exponent, if exists, is the letter "E" followed by an integer. For example, the following are floating point numbers: 3.5E3, 3.125E30, -2.5E2, -2.5E-2, 3.5. the following are not examples of floating point numbers: 3.E3, E3, 3.0E4.5.

For this exercise, let us assume that the the tokens are individual characters. Specifically, the alphabet is as follows:

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, E, -, .\}$$

Your grammar should be complete (i.e. it should not count on undefined non-terminals).