Lab 6

Due: at the end of lab

1 Exercises

Write out solutions to the exercises below, following the formatting in the lecture slides.

1. Given the grammar from last week's lab:

```
<exp> ::= <exp> + <mulexp> | <mulexp>
<mulexp> ::= <mulexp> * <rootexp> | <rootexp>
<rootexp> ::= ( <exp> ) | a | b | c | d | e
```

   Rewrite and extend this grammar with a rule that provides an assignment operator (the $=$ character) in the language. The operator should have lower precedence than $+$, which suggests it should be grouped after the $+$ symbol and should therefore appear higher in the parse tree. $=$ is a right-associative operation. ($+$ and $*$ are left-associative so their production rules have recursive expansions on the left.)

   Verify to yourself that you can construct strings that you expect with the new rule, and that you can't craft expressions that you don't want in the langauge. (For simplicity's sake with this grammar, weird things like `a+b=c` are okay.)

2. With your extended grammar, draw parse trees for the following. You can abbreviate non-terminals like `<exp>` to `<e>` as long as your abbreviations are unique:

   - `e = c`
   - `a = b = c + d * e`

3. Given the ambiguous grammar with binary operators $\odot$ (odot) and ∇ (nabla):

```
<expr> ::= <o> ⊙ <o> | <expr> ⊙ <expr> | <o>
<o> ::= <n> | <o> ∇ <n>
<n> ::= ( <expr> ) | a | b | c
```

Draw **two possible** parse trees for $b \odot (a \nabla b) \odot c$

4. Write an unambiguous grammar equivalent to the language as above, with the following extra rules:

() has the highest priority. ∇ is left-associative and has a middle priority. $\odot$ is right-associative and has the lowest priority.