Homework 6

1. (i) Give the regular expressions for lexing a language consisting of whitespaces, identifiers (some letters followed by digits), numbers, operations `=`, `<` and `>`, and the keywords `if`, `then` and `else`. (ii) Decide whether the following strings can be lexed in this language?

   (a) `"if y4 = 3 then 1 else 3"`

   (b) `"if33 ifif then then23 else else 32"`

   (c) `"if x4x < 33 then 1 else 3"`

   In case they can, give the corresponding token sequences. (Hint: Observe the maximal munch rule and priorities of your regular expressions that make the process of lexing unambiguous.)

2. Suppose the grammar

$$\begin{array}{lcl} E & \rightarrow & F \mid F \cdot * \cdot F \mid F \cdot \backslash \cdot F \\ F & \rightarrow & T \mid T \cdot \texttt{+} \cdot T \mid T \cdot \texttt{-} \cdot T \\ T & \rightarrow & num \mid \texttt{(} \cdot E \cdot \texttt{)} \end{array}$$

   where E, F and T are non-terminals, E is the starting symbol of the grammar, and num stands for a number token. Give a parse tree for the string `(3+3)+(2*3)`.

3. Define what it means for a grammar to be ambiguous. Give an example of an ambiguous grammar.

4. Suppose boolean expressions are built up from

   1.) tokens for `true` and `false`,
   2.) the infix operations $\wedge$ and $\vee$,
   3.) the prefix operation $\neg$, and
   4.) can be enclosed in parentheses.

   (i) Give a grammar that can recognise such boolean expressions and (ii) give a sample string involving all rules given in 1.-4. that can be parsed by this grammar.