Homework 6

Please submit your solutions via email. Please submit only ASCII text or PDFs. Every solution should be preceeded by the corresponding question text, like:

Qn:	**…a difficult question from me…**
A:	**…an answer from you …**
Q$n+1$	**…another difficult question…**
A:	**…another brilliant answer from you…**

Solutions will only be accepted until 20th December! Please send only one homework per email.

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.

5. Given the regular expressions

$$\begin{array}{ll} 1) & (ab + a) \cdot (\mathbf{1} + b) \\ 2) & (aa + a)^* \end{array}$$

there are several values for how these regular expressions can recognise the strings (for 1) ab and (for 2) aaa. Give in each case *all* the values and indicate which one is the POSIX value.

6. Parsing combinators consist of atomic parsers, alternative parsers, sequence parsers and semantic actions. What is the purpose of (1) atomic parsers and of (2) semantic actions?

7. **(Optional)** This question is for you to provide regular feedback to me: for example what were the most interesting, least interesting, or confusing parts in this lecture? Any problems with my Scala code? Please feel free to share any other questions or concerns. Also, all my material is ~~crap~~ imperfect. If you have any suggestions for improvement, I am very grateful to hear.