Homework 4

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. 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.

2. If a regular expression r does not contain any occurrence of $\mathbf{0}$, is it possible for $L(r)$ to be empty? Explain why, or give a proof.

3. Define the tokens and regular expressions for a language consisting of numbers, left-parenthesis $($, right-parenthesis $)$, identifiers and the operations $+$, $-$ and $*$. Can the following strings in this language be lexed?

   - $(a + 3) * b$
   - $)() + + - 33$
   - $(a/3) * 3$

   In case they can, can you give the corresponding token sequences.

4. Assume that s^{-1} stands for the operation of reversing a string s. Given the following *reversing* function on regular expressions

$$\begin{array}{rcl} rev(\mathbf{0}) & \stackrel{\text{def}}{=} & \mathbf{0} \\ rev(\mathbf{1}) & \stackrel{\text{def}}{=} & \mathbf{1} \\ rev(c) & \stackrel{\text{def}}{=} & c \\ rev(r_1 + r_2) & \stackrel{\text{def}}{=} & rev(r_1) + rev(r_2) \\ rev(r_1 \cdot r_2) & \stackrel{\text{def}}{=} & rev(r_2) \cdot rev(r_1) \\ rev(r^*) & \stackrel{\text{def}}{=} & rev(r)^* \end{array}$$

and the set

$$Rev\, A \stackrel{\text{def}}{=} \{s^{-1} \mid s \in A\}$$

prove whether

$$L(rev(r)) = Rev(L(r))$$

holds.

5. Assume the delimiters for comments are `/*` and `*/`. Give a regular expression that can recognise comments of the form

   `/* ... */`

   where the three dots stand for arbitrary characters, but not comment delimiters. (Hint: You can assume you are already given a regular expression written `ALL`, that can recognise any character, and a regular expression `NOT` that recognises the complement of a regular expression.)

6. Simplify the regular expression

   $$(\mathbf{0} \cdot (b \cdot c)) + ((\mathbf{0} \cdot c) + \mathbf{1})$$

   Does simplification always preserve the meaning of a regular expression?

7. The Sulzmann & Lu algorithm contains the function *mkeps* which answers how a regular expression can match the empty string. What is the answer of *mkeps* for the regular expressions:

   $$(\mathbf{0} \cdot (b \cdot c)) + ((\mathbf{0} \cdot c) + \mathbf{1})$$
   $$(a + \mathbf{1}) \cdot (\mathbf{1} + \mathbf{1})$$
   $$a^*$$

8. What is the purpose of the record regular expression in the Sulzmann & Lu algorithm?

9. Recall the functions *nullable* and *zeroable*. Define recursive functions *atmostempty* (for regular expressions that match no string or only the empty string), *somechars* (for regular expressions that match some non-empty string), *infinitestrings* (for regular expressions that can match infinitely many strings).

10. **(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.