Homework 4

1. Why is every finite set of strings a regular language?

2. What is the language recognised by the regular expressions $(\varnothing^*)^*$.

3. If a regular expression r does not contain any occurrence of $\varnothing$ is it possible for $L(r)$ to be empty?

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

$$
\begin{aligned}
rev(\varnothing) &\stackrel{\text{def}}{=} \varnothing \\
rev(\epsilon) &\stackrel{\text{def}}{=} \epsilon \\
rev(c) &\stackrel{\text{def}}{=} c \\
rev(r_1 + r_2) &\stackrel{\text{def}}{=} rev(r_1) + rev(r_2) \\
rev(r_1 \cdot r_2) &\stackrel{\text{def}}{=} rev(r_2) \cdot rev(r_1) \\
rev(r^*) &\stackrel{\text{def}}{=} rev(r)^*
\end{aligned}
$$

and the set

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

prove whether

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

holds.

5. Give a regular expression over the alphabet $\{a, b\}$ recognising all strings that do not contain any substring bb and end in a.

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

$$
/* \ \ldots \ */
$$

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.)

7. Given the alphabet $\{a, b\}$. Draw the automaton that has two states, say q_0 and q_1. The starting state is q_0 and the final state is q_1. The transition function is given by

$$(q_0, a) \rightarrow q_0$$
$$(q_0, b) \rightarrow q_1$$
$$(q_1, b) \rightarrow q_1$$

What is the languages recognised by this automaton?

8. Give a deterministic finite automaton that can recognise the language $L(a^* \cdot b \cdot b^*)$.

9. (Optional) The tokenizer in `regexp3.scala` takes as argument a string and a list of rules. The result is a list of tokens. Improve this tokenizer so that it filters out all comments and whitespace from the result.

10. (Optional) Modify the tokenizer in `regexp2.scala` so that it implements the `findAll` function. This function takes a regular expressions and a string, and returns all substrings in this string that match the regular expression.