Homework 2

1. What is the difference between *basic* regular expressions and *extended* regular expressions?

2. What is the language recognised by the regular expressions $(\mathbf{0}^*)^*$.

3. Review the first handout about sets of strings and read the second handout. Assuming the alphabet is the set $\{a, b\}$, decide which of the following equations are true in general for arbitrary languages A, B and C:

$$\begin{array}{rcl} (A \cup B)@C & =^? & A@C \cup B@C \\ A^* \cup B^* & =^? & (A \cup B)^* \\ A^*@A^* & =^? & A^* \\ (A \cap B)@C & =^? & (A@C) \cap (B@C) \end{array}$$

   In case an equation is true, give an explanation; otherwise give a counterexample.

4. Given the regular expressions $r_1 = \mathbf{1}$ and $r_2 = \mathbf{0}$ and $r_3 = a$. How many strings can the regular expressions r_1^*, r_2^* and r_3^* each match?

5. Give regular expressions for (a) decimal numbers and for (b) binary numbers. Hint: Observe that the empty string is not a number. Also observe that leading 0s are normally not written—for example the JSON format for numbers explicitly forbids this. So 007 is not a number according to JSON.

6. Decide whether the following two regular expressions are equivalent $(\mathbf{1} + a)^* \equiv^? a^*$ and $(a \cdot b)^* \cdot a \equiv^? a \cdot (b \cdot a)^*$.

7. Given the regular expression $r = (a \cdot b + b)^*$. Compute what the derivative of r is with respect to a, b and c. Is r nullable?

8. Give an argument for why the following holds: if r is nullable then $r^{\{n\}} \equiv r^{\{..n\}}$.

9. Define what is meant by the derivative of a regular expressions with respect to a character. (Hint: The derivative is defined recursively.)

10. Assume the set *Der* is defined as

$$Der\, c\, A \stackrel{\text{def}}{=} \{s \mid c :: s \in A\}$$

    What is the relation between *Der* and the notion of derivative of regular expressions?

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

12. Do $(a + b)^* \cdot b^+$ and $(a^* \cdot b^+) + (b^* \cdot b^+)$ define the same language?

13. Define the function *zeroable* by recursion over regular expressions. This function should satisfy the property

$$zeroable(r) \text{ if and only if } L(r) = \{\} \qquad (*)$$

The function *nullable* for the not-regular expressions can be defined by

$$nullable(\sim r) \stackrel{\text{def}}{=} \neg(nullable(r))$$

Unfortunately, a similar definition for *zeroable* does not satisfy the property in $(*)$:

$$zeroable(\sim r) \stackrel{\text{def}}{=} \neg(zeroable(r))$$

Find a counter example?

14. Give a regular expressions that can recognise all strings from the language $\{a^n \mid \exists k.\, n = 3k + 1\}$.

15. Give a regular expression that can recognise an odd number of as or an even number of bs.

16. **(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.

    If *you* want to share anything (code, videos, links), you are encouraged to do so. Just drop me an email.