Homework 2

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

2. 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 counter-example.

3. Given the regular expressions $r_1 = \epsilon$ and $r_2 = \varnothing$ and $r_3 = a$. How many strings can the regular expressions r_1^*, r_2^* and r_3^* each match?

4. 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.)

5. Decide whether the following two regular expressions are equivalent $(\epsilon + a)^* \equiv^? a^*$ and $(a \cdot b)^* \cdot a \equiv^? a \cdot (b \cdot a)^*$.

6. 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?

7. Prove that for all regular expressions r we have

$$\mathit{nullable}(r) \quad \text{if and only if} \quad [] \in L(r)$$

   Write down clearly in each case what you need to prove and what are the assumptions.

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

9. Assume the set Der is defined as

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

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

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

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

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

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

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 out why?

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