Exercise sheet 4

1. Design a non-deterministic finite state automaton over the alphabet $\Sigma = \{0, 1\}$ that will recognize the following languages (try to design a deterministic one too to get a feel for how much easier it is to design a non-deterministic one).

   a) Strings with the nth last character 0, for any natural number n.
   b) Strings that begin with 01.

2. If w is a string, let w^* denote the string $ww...w$, i.e. w repeated any finite number of times. Design non-deterministic finite state automata to recognize the following languages over $\Sigma := \{0, 1\}$.

   a) Strings of the form 0^*
   b) Strings of the form 11^*
   c) Strings of the form $101(010)^*11$

3. If L_1 and L_2 are regular languages, design a non-deterministic finite state automaton that recognizes:

   a) $L_1 \cup L_2$
   b) $L_1 \circ L_2 := \{xy \mid x \in L_1,\ y \in L_2\}$ (i.e. the concatentation of a string from L_1 with a string from L_2)
   c) $L_1^* := \{x_1 x_2 \dots x_n \mid x_i \in L_1\}$ (i.e. concatenation of finitely many strings from the language)

4. Prove that any language that can be recognized by a non-deterministic finite state automaton can also be recognized by a deterministic one. Therefore, a language is regular if and only if it can be recognized by a non-deterministic finite state automaton.

5. Prove that if a language, L, is regular then the language, L', obtained by reversing every string of L is also regular.

6. If $(Q_1, \Sigma, \delta_1, q_1, F_1)$ and $(Q_2, \Sigma, \delta_2, q_2, F_2)$ are non-deterministic finite state automata that recognize the languages L_1 and L_2 respectively, then we have already seen that we can design an automaton $(Q, \Sigma, \delta, q_0, F)$ to recognize $L_1 \cup L_2$ by taking $Q = Q_1 \cup Q_2 \cup \{q_0\}$, where q_0 is a new state,

$F = F_1 \cup F_2$, and

$$\delta(q, c) = \begin{cases} \delta_1(q, c) & q \in Q_1 \\ \delta_2(q, c) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0, c = \epsilon \\ \emptyset & \text{otherwise} \end{cases}$$

Why did we need an entirely new initial state? What if we took either q_1 or q_2 to be the initial state? Does it then accept or reject more strings than it should?