AC32008 Theory of Computation
Tutorial Sheet 1 - Sets, Languages, Proofs and DFAs

1. Let $A = \{ab, c\}$ and $B = \{c, ca\}$ be two languages over the alphabet $\Sigma = \{a, b, c\}$. Calculate

   (a) AB;

   (b) B^2A;

   (c) $A^2 \cup B^2$.

2. Describe deterministic finite automata which accept each of the following languages over $\Sigma = \{0, 1\}$. Give a transition diagram or a table showing the transition function.

   (a) The set of all strings ending in 00.

   (b) The set of all strings that do not contain 10 as a substring.

   (c) The set of all strings with three consecutive 0's.

   (d) The set of all strings that start and end with the same symbol.

   (e) The set of all strings of length at most 4.

3. Suppose that $\hat{\delta}$ is the transition function over strings of a DFA. Prove, by induction over y, that for any input strings x and y, $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$.

4. Suppose that we have two DFAs, $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $M_2 = \{Q_2, \Sigma, \delta_2, q_2, F_2)$. Let M_1 accept the language L_1 and M_2 accept the language M_2. We can construct a DFA $M = (Q, \Sigma, \delta, q_0, F)$ to recognise the language $L_1 \cup L_2$ in the following way:

   (a) $Q = \{(r_1, r_2) | r_1 \in Q_1, r_2 \in Q_2\}$. That is, Q is the Cartesian product of Q_1 and Q_2.

   (b) δ is defined as follows: for each $(r_1, r_2) \in Q$ and each $a \in \Sigma$, let

   $$\delta((r_1, r_2), a) = (\delta_1(r_1, a), \delta_2(r_2, a)).$$

   (c) q_0 is the pair (q_1, q_2).

   (d) $F = \{(r_1, r_2) | r_1 \in F_1, r_2 \in F_2\}$. That is, F is the Cartesian product of F_1 and F_2.

   Use this construction to describe (giving a transition diagram) DFAs accepting the following languages

   (a) The set of all strings over the alphabet $\{0, 1\}$ that have an even number of 0's and one or two 1's.

   (b) The set of all strings over the alphabet $\{0, 1\}$ that have an odd number of 0's and end with a 1.

5. (**Harder**) The string over alphabet $\{(,)\}$ is *balanced* if it has an equal number of ('s and)'s and if every prefix w of that string has at least as many ('s as)'s. Some examples of balanced strings are (), (()), ()(), ()(()), while strings (() and)(() are not balanced. Let a set of strings R be defined using the following 4 rules:

   (a) ϵ is in R.

   (b) If w is in R, then (w) is in R.

   (c) If w and x are in R, so is wx.

   (d) There are no other strings in R.

   Prove by induction on the length of a string that every string in R is balanced.