CSE 322: Formal Models in Computer Science **Homework #2**
April 10, 2009 Due: **Friday**, April 17, 2009

Reading Assignment: Sipser 1.2,1.3

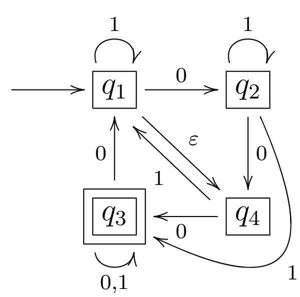
Problems:

1. Given two strings x and y of exactly the same length, we can create a new string called $shuffle(x, y)$ that consists of characters of x and y alternating one after another starting with the rst character of x. That is, if $x = x_1 \dots x_k$ and $y = y_1 \dots y_k$, then

$$shuffle(x, y) = x_1 y_1 x_2 y_2 \dots x_k y_k \tag{1}$$

For languages A and B, define $SHUFFLE(A, B) = \{shuffle(x, y) | x \in A, y \in B \text{ and } |x| = |y|\}$. Given DFAs that accept A and B, give an intuitive description and then a formal description of how to build a DFA that accepts $SHUFFLE(A, B)$. (Note that in the above we did not specify that A and B have the same alphabet. Also note that the DFA for $SHUFFLE(A, B)$ only gets one symbol at a time, that is, on input $x_1 y_1 x_2 y_2 \dots x_k y_k$, it reads as x_1 then y_1 then x_2, etc, and not (for example) in pairs like $x_1 y_1$ then $x_2 y_2$, etc.)

2. Convert the following NFA to a DFA using the subset construction discussed in class. For your answer you should draw the state diagram for your DFA. (You may obmit, if they exist, states which are not reachable from the start state.)



3. In this problem you will prove that regular languages are closed under certain operations. For all the three parts, assume L and M are regular languages. (If you prove these by giving a construction of a DFA or NFA, present a correct construction of the DFA or NFA, including a formal description and also give an informal (yet convincing) discussion of why the construction works. You do not need to give a formal proof of correctness using induction.) Work over a fixed alphabet Σ.

   (a) Prove that $L^R = \{x^R | x \in L\}$ is also regular. Recall that the R operation reverses the order of the string.

   (b) Prove that $L^+ = \{x_1 x_2 \cdots x_k | k \geq 1, x_i \in L\}$ is also regular. Note that L^+ differs from L^* in that it does not necessarily contain the empty string (it only contains the empty string if L itself contains the empty string.)

(c) Prove that $\bar{L} = \{x \in \Sigma^* | x \notin L\}$ is also regular. $\bar{L}$ it the *complement* of L in Σ^*.

4. For the language denoted by each of the following regular expressions, give two strings that are members and two strings that are non-members - a total of four strings for each part. Assume the alphabet is $\Sigma = \{a, b\}$ in all parts:

   (a) $\Sigma^* ab \Sigma^* a \Sigma^*$

   (b) $(aba \cup bab) \cup \emptyset$

   (c) $(\varepsilon \cup a)b$

5. **Extra Credit** (do it for the glory, not the points!) An odd-NFA M is a 5-tuple $(Q, \Sigma, \delta, s, F)$ that accepts a string $x \in \Sigma^*$ if the number of possible states that M could be in (the number of states "alive") after reading input x, which are also in F, is an odd number. (In other words, the set of all possible states has an odd number of states from F.) Note, in contrast, a regular NFA accepts a string if at least one state among these final states is an accept state.

   Prove that odd-NFAs accept the set of regular languages. Note that there are two directions to this proof: one is to show that odd-NFAs accept regular languages and the other direction is that if the language is regular, then it is accepted by an odd-NFA.

6. **Extra Credit** (do it for the glory, not the points!) Prove that if L is regular, then $half(L)$ is regular, where the operation $half$ is defined as follows:

$$half(L) = \{x| \text{ for some } y \text{ such that } |x| = |y|, xy \text{ belongs to } L\} \tag{2}$$