CSc 30400 Introduction to Theory of Computer Science
3rd Homework Set

1. Figure 1 presents the state diagrams of a NFA M_1 and a NFA$_\varepsilon$ M_2. Answer the following questions about each of these machines.

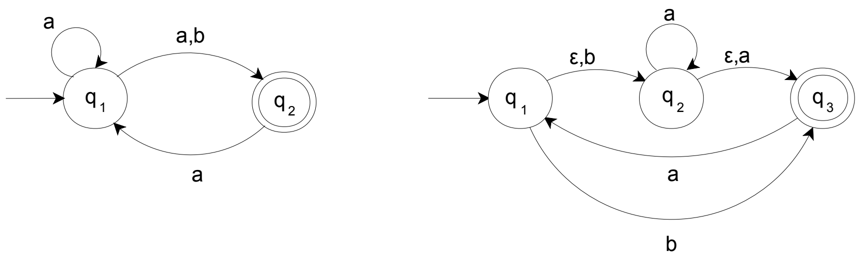

Figure 1: M_1 and M_2 of exercise 1.

   (a) What is the start state?
   (b) What is the set of accepting states?
   (c) What sequence of sets of states does the machine go through on input aaab?
   (d) Does the machine accept the string aaab?
   (e) Does the machine accept the string ε?

2. The formal definition of a NFA M is $(\{q_0, q_1, q_2\}$, {a,b}, δ, q_1, $\{q_2\})$, where δ is given by table 1. Give the state diagram of this machine.

	a	b
q_0	$\{q_0\}$	$\emptyset$
q_1	$\{q_0, q_1\}$	$\{q_2\}$
q_2	$\{q_1, q_2\}$	$\{q_0\}$

Table 1: The transition function δ of exercise 2.

3. For the NFA$_\varepsilon$ of figure 2 first construct an equivalent NFA and then from this an equivalent DFA (use the slides about NFA$_\varepsilon$ - NFA - DFA equivalence). Now construct the minimum equivalent DFA (you can use either the slides about minimizing DFAs or the method that we described in class).

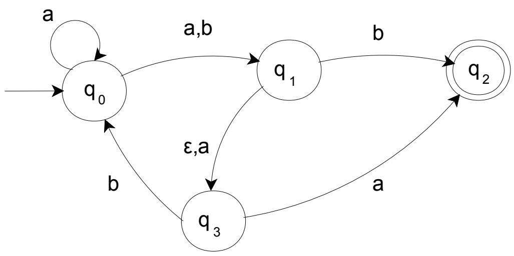

Figure 2: The NFA$_\varepsilon$ of exercise 3.

4. Read Theorem 1.25 on pages 45 - 46 and make sure that you understand the proof. Then prove that if A and B are languages accepted by two DFAs M_A and M_B then there is a DFA M that recognizes their intersection $A \cap B$.

5. Design an NFA$_\varepsilon$ on the alphabet $\Sigma = \{0, 1\}$ that recognizes the language:

   (a) $L_1 = \{w|w$ contains an odd number of 0s or an even number of 1s$\}$

   (b) $L_2 = \{w|w$ ends with 10$\}$

   (c) $L_3 = \{w|w$ contains an even number of 1s and ends with 10$\}$

   (d) $L_4 = \{w|w$ contains a pair of 1s that are separated by an odd number of symbols$\}$ (Hint: First construct a DFA that recognizes the language $L = \{w|w$ has odd length$\}$. Then consider how to create the formula $\cdots 1\underbrace{\cdots}_{odd}1\cdots$. The NFA has 4 states.)

6. Try to design DFAs that recognize the languages of ex. 5 (do not convert the NFA to an equivalent DFA).

   Hints:

   (a) First construct two DFAs M_1 and M_2 that recognize the languages $L_{11} = \{w|w$ contains an odd number of 0s$\}$ and $L_{12} = \{w|w$ contains an even number of 1s$\}$. Then use the construction given in the proof of Theorem 1.25 of the book.

   (b) This DFA will have 3 states in total.

   (c) Observe that $L_3 = L_{12} \cap L_2$. Use the construction of exercise 4.

   (d) First observe that I can have as many 0s as I want as a prefix.

   - If the word begins with $1\overbrace{0\cdots0}^{odd}1$ then it should accept.

- Try to figure out what happens if a word begins with $1\overbrace{0\cdots0}^{even}1$

When the formula is found I can have as many 0s and 1s I want as a suffix. The DFA has 5 states.

7. Convert the NFAs of exercise 5 to equivalent DFAs. Then (if it's not) find the minimum equivalent DFA. For each of the questions of exercise 6 that you couldn't answer try to figure out how the produced DFA works.

8. Problem 1.31 of book: For any string $w = w_1w_2\cdots w_n$ the **reverse** of w, written $w^{\mathcal{R}}$, is the string w in reverse order, $w_n\cdots w_2w_1$. For any language A, let $A^{\mathcal{R}} = \{w^{\mathcal{R}}|w \in A\}$.

   (a) Show that if there is a DFA M with exactly one final state that recognizes a language A then there is a NFA M' that recognizes $A^{\mathcal{R}}$. Hint: The sequence of states that M goes through on an accepting input $w_1\cdots w_n$ is $q_0, q_1, \ldots, q_f$ where q_0 is the start state and q_f is the final state. Try to figure out the path that M' should go through on input $w_n\cdots w_1$ in order to accept.

   (b) Consider the case where A is recognized by a DFA M with two or more final states. Design a NFA$_\varepsilon$ M' that recognizes $A^{\mathcal{R}}$.

9. Problem 1.38 of book: An **all**-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if *every* possible state that M could be in after reading input x is a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* state among these possible states is an accept state. Prove that all-NFAs are equivalent with DFAs. Hint: First observe that a DFA is an all-NFA (why?). For the opposite direction, from each all-NFA you should construct an equivalent DFA.