CS 301 Languages and Automata, UIC
Fall 2012, Assignment 2

Unless otherwise noted, the alphabet for all questions below is assumed to be $\Sigma = \{0, 1\}$.

1. This question tests your comfort with "boundary cases" of DFA's. Draw the state diagrams of DFA's recognizing each of the following languages.
   (a) $L = \{\epsilon\}$ for ϵ the empty string.
   (b) $L = \emptyset$.
   (c) $L = \{0, 1\}^*$.

2. This question demonstrates how sometimes NFAs are easier to design than DFAs.
   (a) Draw the state diagram for a DFA recognizing language $L_1 = \{x \mid x \text{ contains at least two 1s}\}$.
   (b) Draw the state diagram for a DFA recognizing language $L_2 = \{x \mid x \text{ contains at most one } 0\}$.
   (c) Draw the state diagram for a DFA recognizing language $L_3 = L_1 \cup L_2$. Hint: One option is to use the construction of Theorem 1.25 in the text.
   (d) Draw the state diagram for a NFA recognizing language L_3 (this should be simpler than your DFA from part 3 above).

3. This question illustrates differences and commonalities between NFAs and DFAs.
   (a) Show that if M is a DFA that recognizes language B, then swapping the accept and nonaccept states in M yields a new DFA M' recognizing the complement of B, $\overline{B}$. Which operation does this imply the regular languages are closed under?
   (b) Prove by counterexample that if M is a NFA that recognizes languages B, then swapping the accept and nonaccept states in M does not necessarily yield an NFA recognizing $\overline{B}$. With this in mind, is the class of languages recognized by NFA's closed under complement (explain your answer!)?

4. This question tests your ability to prove a language is regular using the closure properties of regular languages. Given languages A and B, define the operation $\cdot$ as

$$A \cdot B := \{x \mid x \in A \text{ and } x \text{ does not contain any string in } B \text{ as a substring.}\}.$$

   Prove that the class of regular languages is closed under the $\cdot$ operation.

5. This question tests your understanding of the equivalence between DFAs and NFAs. Consider NFA $M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_1\})$ for δ defined as:

δ	0	1	ϵ
q_1	$\{q_1, q_2\}$	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\{q_1\}$	$\{q_1\}$

   Draw both the state diagrams for M and for a DFA M' equivalent to M based on the construction of Theorem 1.39 in the text (recall the latter proves that DFAs and NFAs are equivalent).

6. This question demonstrates that although DFAs and NFAs are equivalent in terms of the sets of languages they recognize, they are *provably not* equivalent in terms of efficiency (i.e. DFAs may require **many** more states to recognize a language than an NFA). Consider the language $C_k = \Sigma^* 0 \Sigma^{k-1}$ for $k \geq 1$. Convince yourself that an NFA with $k+1$ states for recognizing C_k exists (no need to include this in your assignment answer). Now, prove that for any k, C_k cannot be recognized by a DFA with less than 2^k states.