CSE 322: Formal Methods in Computer Science Spring 2003

PROBLEM SET 2

Due Friday, April 18, 2003, in class

Instructions: Same as those for Problem Set 1.
There are **FIVE** problems in this problem set.

---

1. Give nondeterministic finite automata (with ϵ-transitions if needed) that recognize the following languages. Try to take advantage of nondeterminism as much as possible.
   (a) The set of strings of 0's and 1's such that there are two 0's separated by a number of positions that is a multiple of 3. (Note that 0 is an allowed multiple of 3.)
   (b) The set of strings of 0's and 1's which contain a 1 among the *last six* positions.

2. Convert the NFAs described below over alphabet $\Sigma = \{0, 1\}$ into DFAs that recognize the same language. (Reading Example 2.2.4 on page 72 of the textbook might be helpful for this exercise.)
   (a) $N_1 = (K_1 = \{q_0, q_1\}, \Sigma, \Delta_1, q_0, \{q_0\})$ where the transition relation Δ_1 is given by $\Delta_1 = \{(q_0, 0, q_0), (q_0, 0, q_1), (q_0, 1, q_1), (q_1, 1, q_0)\}$.
   (b) $N_2 = (K_2 = \{s_0, s_1, s_2\}, \Sigma, \Delta_2, s_0, \{s_1\})$ where the transition relation Δ_2 is given by $\Delta_2 = \{(s_0, \epsilon, s_1), (s_0, 0, s_2), (s_1, 0, s_0), (s_2, 0, s_1), (s_2, 1, s_1), (s_2, 1, s_2)\}$.

3. (a) Prove or disprove: If N is an NFA recognizing B, then swapping the final and non-final states in N yields a new NFA that recognizes the complement of B.
   (b) Give a counterexample to show that the following construction fails to prove the closure of regular languages under the Kleene star operation. Let $N = (K, \Sigma, \Delta, s, F)$ be an NFA for language A. Then construct an NFA $N' = (K', \Sigma, \Delta', s', F')$ which supposedly recognizes A^* as follows.
   - $K' = K$.
   - $s' = s$.
   - $F' = F \cup \{s\}$.
   - $\Delta' = \Delta \cup \{(q, \epsilon, s) \mid q \in F\}$.

   (In other words, you must present an NFA N for which the constructed NFA N' does not recognize the Kleene star of N's language.)

4. For an integer $k \geq 1$, define the language L_k over alphabet $\Sigma = \{0, 1\}$ as follows:
$$L_k = \{x \mid \text{ the } k\text{'th symbol from the right in } x \text{ equals } 1\} .$$
   (a) Design an NFA with $k + 1$ states that recognizes L_k.
   (b) Design a DFA for the same language L_k. How many states does your DFA have?

(c) Prove that every DFA that recognizes L_k must have at least 2^k states. (Hint: Call a pair of strings x and y "L_k-distinguishing" if for some z, one of the two strings xz and yz belongs to L_k and the other one doesn't. Let q_x and q_y be the states in which the DFA ends up in when run on inputs x and y respectively. Prove that q_x must be different from q_y if the pair (x, y) is L_k-distinguishing. Then, find a collection of 2^k strings every pair of which is L_k-distinguishing, and use it to complete the proof.)

5. Suppose that L is regular, i.e. is recognized by a DFA. Then:

(a) Prove that $L^R = \{w^R \mid w \in L\}$ is also regular. (Here w^R denotes the *reverse* of the string w, i.e. if $w = w_1 w_2, \ldots, w_n$ then $w^R = w_n w_{n-1} \ldots w_2 w_1$.)

(b) Prove that $\mathsf{MIN}(L) = \{w \in L \mid \text{no proper prefix of } w \text{ is in } L\}$ is also regular.

(c) * (Extra Credit) Define $\mathsf{HALF}(L) = \{x \mid \text{for some } y \text{ such that } |x| = |y|,\ xy \text{ belongs to } L\}$. Prove that $\mathsf{HALF}(L)$ is regular if L is regular.
(For all of the parts above, presenting a *correct* construction of an NFA/DFA for the languages in question, along with an informal yet convincing explanation of why it does the right thing, will suffice. Fully formal proofs by induction are *not* required.)