CS 4510: Automata and Complexity

Spring 2015

Home work 2 // Due: Friday, January 30, 2015

1. (10 points) (Problem 1.38, page 89 of third edition Sipser text.)

   An **all**-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts a string $x \in \Sigma^*$ 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 recognize the class of regular languages.

   Solution: A DFA is clearly an **all**-NFA.

   In the other direction, given an **all**-NFA construct a DFA as in the case of an ordinary NFA except for the following modification: the final states of the DFA is now $\mathcal{P}(F)$, where F is the set of final states of the **all**-NFA. (This is because the **all**-NFA accepts an input x iff the set of states that it is in after reading input x is a subset of F.)

2. (10 points) Let L be a regular language. Is the language L_1 defined below regular? Why?

   $L_1 = \{w \in \{0,1\}^* \mid w \text{ is either in } L^R \text{ or in } \bar{L} \text{ but not in both}\}$.

   Solution: Yes. L_1 is the exclusive OR of the languages $\overline{L}$ and L^R. Since L is regular, because of closure properties, there is a DFA M_1 for $\overline{L}$, a DFA M_2 for L^R, and a DFA for the exclusive OR of $L(M1)$ and $L(M_2)$.

   (Note: If A and B are sets, their exclusive OR is the set of strings w such that w is either in A or in B but not in both A and B.)

3. (10 points) Consider the DFA $M = (\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_2\})$, where δ is as defined below:

$$\begin{aligned} \delta(q_1, 0) &= q_1 \\ \delta(q_1, 1) &= q_2 \\ \delta(q_2, 0) &= q_2 \\ \delta(q_2, 1) &= q_2 \end{aligned}$$

   Convert this into a regular expression using the algorithm covered in class. In particular, show the following regular expressions: $R_{ij}^{(k)}$ for $1 \leq i, j \leq 2$ for $k = 0, 1$ and $R_{12}^{(2)}$. (Simplify the regular expressions using the rules they obey.)

4. (15 points) Let L be a regular language. Suppose there exists a set of pairs of strings $\{(x_i, y_i) \mid 1 \leq i \leq n\}$ such that the following two conditions hold:

   (a) $x_i y_i \in L$ for all $1 \leq i \leq n$, and

   (b) $x_i y_j \notin L$ for all $1 \leq i \neq j \leq n$.

   Show that any NFA for L must have n states.

   Solution: Consider any NFA N for L. For each $1 \leq i \leq n$, let S_i denote the set of states reachable from the start state on string x_i. For $x_i y_i$, for some i, there must be some state $s \in S_i$ from which a final state f can be reached since $x_i y_i \in L$. Suppose $s \in S_j$ for some $j \neq i$. Then, we have an accepting path for $x_j y_i$ (from start state to the state s in S_j and from s to the state f on y_i). This contradicts the second condition. So, for each $1 \leq i \leq n$, the set S_i has a state that is not contained in the set S_j for $i \neq j$.

5. (15 points) Identify the regular set and the non-regular set from the following two subsets of $\{0,1\}^*$. Prove your answer.

   (a) $B = \{1^k y \mid y \text{ contains at least } k \text{ 1s, for } k \geq 1\}$.

   (b) $C = \{1^k y \mid y \text{ contains at most } k \text{ 1s, for } k \geq 1\}$.

   Solution: B is the same as the language $A = 1(0 \cup 1)^* 1 (0 \cup 1)^*$ which is regular. This is because any string in B starts with a 1 and has at least one 1 appearing in it later.

   C is not a regular Language. Let p be the pumping lemma constant. Choose the string $1^p 0 1^p$ in C. Choose $i = 0$.

6. Let L be a language defined over a finite alphabet Σ. Two strings $x, y \in \Sigma^*$ are said to be *distinguishable* by L iff there exists some string z such that exactly one of xz or yz is in L.

   (15 points) Consider the language $L = \{x \in \Sigma^* \mid x = x^R\}$. (Here x^R is the reverse of the string x.) Show that any two distinct strings $x, y \in \Sigma^*$ are distinguishable by L.

   Solution: Let x and y be two distinct strings.

   Case 1: $|x| = |y|$. Then, the string $z = x^r$ serves to distinguish the two strings.

   Case 2: $|x| \neq |y|$. Assume, without loss of generality, that $|x| < |y|$.

   Then, y can be written as $y_1 y_2$ where $|y_1| = |x|$ and y_2 is not empty. Let w be a string such that $|w| = |y_2|$ and $w \neq y_2$. Then, the string $z = ww^r x^r$ serves to distinguish the strings x and y.