Theory of Computation

Fall 2015, Midterm Exam. Solutions

Nov. 16, 2015

1. (20 pts) Regular or not? If regular, construct an automaton, regular expression or a grammar. If not regular, use pumping lemma for regular languages.

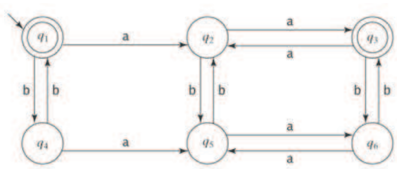
   (a) Let A be the set of all strings of the form $\{0^k 1^l \mid k + l = 8\}$ over alphabet $\{0, 1\}$.
   Solution: Regular. A simple finite state machine with finite number of states which takes care of the order of symbols and counts the total number of characters.

   (b) Let B be the set of all strings of the form $\{0^k 1^l \mid k - l = 8\}$ over alphabet $\{0, 1\}$.
   Solution: Not regular. Language L consists of all strings of the form 0^*1^* where the number of 0's is eight more than the number of 1's. We can show that L is not regular by applying pumping lemma. Let w = $0^{m+8}1^m$. Since $|xy| \leq m$, y must equal 0^k for some $k > 0$. We can pump y out once, which will generate the string $0^{m+8-k}1^m$, which is not in L because the number of 0's is less than 8 more than the number of 1's.

   (c) Let $C = \{(0^m 1^n)^p \mid m, n, p \geq 0\}$ over alphabet $\{0, 1\}$.
   Solution: Regular. $C = (0^*1^*)^*$

   (d) Let D be the set of strings over $\{0, 1\}$ that can be written in the form $1^k 0 y$ where y contains at least k 1's, for some $k \geq 1$.
   Solution: Assume to the contrary that B is regular. Let p be the pumping length given by the pumping lemma. Consider the string $s = 1^p 0^p 1^p \in B$. The pumping lemma guarantees that s can be split into 3 pieces $s = abc$, where $|ab| \leq p$. Hence, $y = 1^i$ for some $i \geq 1$. Then, by the pumping lemma, $ab^2c = 1^{p+i}0^p 1^p \in B$, but cannot be written in the form specified, a contradiction.

2. (10 pts) Minimize the following DFA. Show your derivation in detail.



Solution

Initially, we have *classes* = {[1, 3], [2, 4, 5, 6]}.

At step 1:

((1, a), [2, 4, 5, 6])((3, a), [2, 4, 5, 6]) No splitting required here.

((1, b), [2, 4, 5, 6]) ((3, b), [2, 4, 5, 6])

((2, a), [1, 3]) ((4, a), [2, 4, 5, 6])((5, a), [2, 4, 5, 6])((6, a), [2, 4, 5, 6])

((2, b), [2, 4, 5, 6]) ((4, b), [1, 3]) ((5, b), [2, 4, 5, 6]) ((6, b), [1, 3])

These split into three groups: [2], [4, 6], and [5]. So classes is now {[1, 3], [2], [4, 6], [5]}.

At step 2, we must consider [4, 6]:

((4, a), [5]) ((6, a), [5])

((4, b), [1]) ((6, b), [1])

No further splitting is required.

The minimal machine has the states: {[1, 3], [2], [4, 6], [5]}, with transitions as shown above.

3. (10 pts) Let $M_A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_{0B}, F_B)$ be two DFAs accepting languages A and B, respectively, where $A, B \subseteq \Sigma^*$. Construct an NFA $M = (Q, \Sigma, \delta, q_0, F)$ to accept $A \| B = \{w \mid \exists u \in A, v \in B, w$ is a *shuffle* of u and $v\}$. For instance, $a123bc4$ is a shuffle of the strings abc and 1234.

SOLUTION: Let $M_A = (Q_A, \Sigma, \delta_A, q_{0A}, F_A)$ and $M_B = (Q_B, \Sigma, \delta_B, q_{0B}, F_B)$ be two DFAs accepting the languages A and B respectively. Then we define an NFA $M = (Q, \Sigma, \delta, q_0, F)$ for $S(A, B)$ as follows.

Let $Q = Q_A \times Q_B$, $q_0 = (q_{0A}, q_{0B})$ and $F = F_A \times F_B$. Define $\delta((q_A, q_B), s) = \{(\delta_A(q_A, s), q_B)\} \cup \{(q_A, \delta_B(q_B, s))\}$, i.e., at each step, the machine changes q_A according to δ_A or q_B according to δ_B. It reaches a state in $F_A \times F_B$ if and only if the moves according to δ_A take it from q_{0A} to a state in F_A, and the ones according to δ_B take it from q_{0B} to a state in F_B. Hence M accepts exactly the language $S(A, B)$.

4. (10 pts) Define C to be all strings consisting of some positive number of 0's, followed by some string twice, followed again by some positive number of 0's. For example 1100 is not in C, since it does not start with at least one 0. However 0001011010000000 is in C since it is three 0's, followed by 101 twice, followed by seven 0's. Prove that C is not regular using the Myhill-Nerode Theorem. (Hint: Consider strings 01^k0 for each natural number k.)
Solution: We will show that there are infinitely many strings, any two of which are distinguishable with respect to C. This will mean there are infinitely many indistinguishability classes. By the Myhill-Nerode Theorem, we can then conclude that C is not regular. Our strings will be 01^k0 for each natural number k. Let k_1 and k_2 be distinct natural numbers. $01^{k_1}01^{k_1}00$ is in L. If $01^{k_1}01^{k_2}00$ were in L, then it must be $0ss0$ or $0ss00$ for some string s. So s must contain at least one zero. Thus $01^{k_1}01^{k_2}00$ must be $0ss0$. So s must end with a 0, and that is the only 0 in s. But then s must be both $1^{k_1}0$ and $1^{k_2}0$. This is impossible since those strings have different lengths. So each 01^k0 is in a different indistinguishability class and C is not regular.

5. (10 pts) Consider the language $F = \{a^ib^jc^k | i, j, k \geq 0$ and if $i = 1$ then $j = k\}$. Argue that F satisfies the pumping lemma for regular languages.
Solution: The pumping lemma says that for any string s in the language, with length greater than the pumping length p, we can write $s = xyz$ with $|xy| \leq p$, such that xy^iz is also in the language for every $i \geq 0$. For the given language, we can take $p = 2$. Consider any string $a^ib^jc^k$ in the language.

   - If $i = 1$ or $i > 2$, we take $x = \epsilon$ and $y = a$. If $i = 1$, we must have $j = k$ and adding any number of a's still preserves the membership in the language. For $i > 2$, all strings obtained by pumping y as defined above, have two or more a's and hence are always in the language.
   - For $i = 2$, we can take $x = \epsilon$ and $y = aa$. Since the strings obtained by pumping in this case always have an even number of a's, they are all in the language.
   - Finally, for the case $i = 0$, we take $x = \epsilon$, and $y = b$ if $j > 0$ and $y = c$ otherwise. Since strings of the form b^jc^k are always in the language, we satisfy the conditions of the pumping lemma in this case as well.

6. (10 pts) A grammar is called *right-linear* if each of its productions is of the form $A \to aB$ or $A \to c$, where a is a terminal, A, B are nonterminals, and c is either a terminal or an ϵ. Find a right-linear grammar $G = (V, T, P, S)$ to generate the regular language represented by regular expression $(00 \cup 1)^*$. (Hint: Find an NFA for the regular expression, and then convert the NFA to a right-linear grammar.)
Solution:

   - NFA $M = (\{A, B\}, \{0, 1\}, \{\delta(A, 0) = B, \delta(A, 1) = A), \delta(B, 0) = A\}, A, \{A\})$.
   - $A \to 1A \quad A \to 0B \quad B \to 0A \quad A \to \epsilon$

7. (20 pts) Prove or disprove the following statements.

   (a) If $A \subseteq B$ then $A^* \subseteq B^*$.
   Solution: True. If $x_1x_2...x_k \in A^*$ (where $x_i \in A$) then $x_i \in B, 1 \leq i \leq k$. Hence, $x_1x_2...x_k \in B^*$.

   (b) If $A \subseteq \Sigma^*$ is regular, $B \subseteq \Sigma^*$ is a finite language, then $A \setminus B = \{w \in A \mid w \notin B\}$ is regular.
   Solution: True. Note that B is regular, so is $\bar{B}$. $A \setminus B = A \cap \bar{B}$, which is regular.

   (c) There is a non-regular language L such that L^* is regular.
   Solution: True. Consider $L = \{0^{n^2} \mid n \geq 0\}$ which is not regular. Note that $0 \in L$. L^*=0^+, which is regular.

   (d) The following grammar is ambiguous : $S \rightarrow 0S1 \quad | \quad 01S \quad | \epsilon$
   Solution: The string 01 may be generated in two ways:

   - $S \rightarrow 0S1 \rightarrow 0(\epsilon)1 = 01$
   - $S \rightarrow 01S \rightarrow 01(\epsilon) = 01$

8. (10 pts) Given a morphism h and a language L, it is known that $h(h^{-1}(L)) \subseteq L \subseteq h^{-1}(h(L))$, prove that neither containment is necessarily an equality. That is, show that there is a language A, $h(h^{-1}(A)) \neq A$ and there is another language B, $B \neq h^{-1}(h(B))$.
   Solution:

   - $(h(h^{-1}(A)) \neq A)$
     Let $\Sigma = \{0,1\}, \Gamma = \{a,b\}$. Consider $h(0) = h(1) = a$. Let $A = \{a,b\}$. Clearly $h^{-1}(A) = \{0,1\}$. Hence, $h(h^{-1}(A)) = \{a\} \neq A$.
   - $B \neq h^{-1}(h(B))$.
     Let $\Sigma = \{0,1\}, \Gamma = \{a,b\}$. Consider $h(0) = h(1) = a$. Let $B = \{0\}$. Clearly $h(B) = \{a\}$. Hence, $h^{-1}(h(B)) = \{0,1\} \neq B$.