CS 4510 Automata and Complexity

Exam 1: Practice

NOTE: This practice exam is somewhat longer than the actual exam.
Topics you should be comfortable with:

- Proving a language is regular (we've learned four ways)
- Proving a language is irregular (we've learned one way)
- Converting between DFA's and NFA's
- Converting between regular expressions and NFA's
- Modifying finite automata to accept new languages
- Operations on regular languages

1. Short answer

   (a) Let L be a regular language, and let R be a non-regular language. For each of the following, write regular, non-regular, or unknown, and a short (one-sentence) justification:

   - $\emptyset$ is regular. (DFA is a single, nonaccepting state.)
   - $\{\varepsilon\}$ (Represented by the regular expression ε)
   - $\{x|x \in L \text{ or } x = \varepsilon\}$ (Union of two regular languages.)
   - $\{x|x \in L \text{ and } x \text{ contains at most three ones }\}$ (Intersection of two regular languages.)
   - $\{xy|x \in L \text{ and } y \notin L\}$ (Concatenation of two regular languages, since $\overline{L}$ is regular.)
   - $\{xy|x \in L \text{ and } y \in R\}$. (Unknown: $0^* \circ (0^{n^2})$ is regular, $1^* \circ 0^{n^2}$ is not.)
   - $\{x|x \notin R\}$ (Not regular. If this were regular, then R would be regular since it would be the complement of a regular language.)

   (b) Give an example of languages A, B such that neither A nor B are regular, but $A \cup B$ is regular.

   Solution: Let R be a non-regular language. Then $A = R$, $B = \overline{R}$ suffices.

   (c) What is wrong with the following proof that all languages are regular. (Do not just say "Some languages aren't regular." Find the flaw in the proof.)

   "Let L be a language. If there is a NFA for L, then L is regular. But we can construct a DFA for L. For each string $x = x_1x_2...x_n$ in the language, we can add the states $q_{x_1}, q_{x_2}, ..., q_{x_n}$, with $q_{x_i} \in \delta(q_{x_i - 1}, x_i) =$ and add q_{x_n} to the set of final states. But then we have constructed an NFA for L, so L is regular, completing the proof."

Solution If L is infinite, then this process creates an infinite number of states, which is not allowed. (In the same way that you can not code a program which is infinitely long.)

2. Language operations

   (a) Let $LAST\_ONE\_OFF(L) = \{x \in \Sigma^* \mid xa \in L \text{ for some } a \in \Sigma\}$. i.e. the set of strings that you get by deleting the last letter of the strings in L. Show that if L is regular, $LAST\_ONE\_OFF(L)$ is also regular.

   Solution Let D be a DFA for L with final states F. Then create D' by copying D, except that the final states are all those states of D which can transition to a final state on some input character. (Note that this means that some of the state that were accepting before are still accepting.)

   (b) Let $WEAVE(L, R)$ be the language $\{x_1y_1...x_ny_n | x_1...x_n \in L, y_1...y_n \in R, x_i, y_i \in \Sigma^*\}$. For example, suppose L is English words, and R is binary palindromes. Then $WEAVE(L, R)$ contains words like $ham110bu10r1011ger$, $hamburger110101011$, and $h1a1m0b1u0r1g0e1r1$. Show that if L and R are regular, then so is $WEAVE(L, R)$

   Solution Let $D_1 = (Q_1, \Sigma, q_{0-1}, F_1, \delta_1)$ and $D_2 = (Q_2, \Sigma, q_{0-2}, F_2, \delta_2)$ be DFA's for L and R respectively. Then we create an NFA $N = (Q, \Sigma, q_0, F, \delta)$ for $WEAVE(L, R)$ as follows:

   - $Q = Q_1 \times Q_2$
   - $q_0 = (q_{0-1}, q_{0-2})$
   - $F = F_1 \times F_2$
   - $\delta((q_i, q_j), a) = \{(\delta_1(q_i, a), q_j), (q_i, \delta_2(q_j, a)\}$

   In English, we keep track of the current state of each machine, and when we read an input character, we advance one of the machines one step while leaving the other where it was.

   (c) Let $SHIFT(L) = \{wx \mid xw \in L, w \in \Sigma^*, x \in \Sigma\}$, i.e. the set of words you get by taking a word from L and moving its first character to the end. If L were the set of English words, then $SHIFT(L)$ would contain words like $amburgerh, omplexityc$, etc. Show that, if L is regular, $SHIFT(L)$ is regular.

   Solution We assume L is regular, so there exists a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that determines L. We will also assume a binary alphabet. We will create a new NFA $N = (Q', \Sigma, \delta', q'_0, F')$ that determines $SHIFT(L)$. We will make two copies of M, call them $M_0 = (Q_0, \Sigma, \delta_0, q_{00}, F_0)$ and $M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$. We will create two new states q'_0 and q_f. We will have $Q' = Q_0 \cup Q_1 \cup \{q'_0, q_f\}$. The set of accepting states for N will be $F' = \{q_f\}$. We copy all transitions from δ_0 and δ_1 into δ'. We know that M is a DFA and we have a binary alphabet so there must be exactly two transitions exiting the state q_0, specifically $\delta(q_0, 0) = q_i$ for some state q_i and $\delta(q_0, 1) = q_j$ for some state q_j. Let the states q_{i0} and q_{j1} be the counterparts to q_i and q_j in M_0 and M_1 respectively. We will add the following transition $\delta'(q_0', \varepsilon) = \{q_{i0}, q_{j1}\}$. We will also add $\delta'(q_{k0}, 0) = \{q_f\}$ for each state $q_{k0} \in F_0$ and $\delta'(q_{k1}, 1) = \{q_f\}$ for each state $q_{k1} \in F_1$. This is a NFA that accepts $SHIFT(L)$ given that L is regular, so $SHIFT(L)$ is regular if L is regular.

3. Regular expressions

   Give regular expressions that represent the following:

   - All strings that start with 1 and end with 00. $(1(\Sigma^*)00)$
   - All strings that start with 1 or end with 00 (or both). $(1\Sigma^* \cup \Sigma^*00)$
   - All strings that start with 1 or end with 00, but not both. $(1\Sigma^*(01 \cup 10 \cup 11) \cup (11 \cup 10 \cup 1) \cup (0\Sigma^*00 \cup 00)$

4. Prove that the following languages are not regular:

   - $L_1 = \{xw_1xw_2x | w_i \in \Sigma^*, x \in \Sigma, |w_1| = |w_2|\}$, i.e. all strings whose first, middle, and last character are the same.

     Solution 1. Assume L is regular, and let p be the pumping length.
     2. Let $s = 10^p10^p1$. Note $|s| \geq p$.
     3. Let $s = xyz$, where $|xy| \leq p$, $|y| > 0$. Then $y = 10^n$, $0 \leq n < p$, or $y = 0^m$, $0 < m \leq p$.
     4. Consider xz.
       - Case 1: $y = 10^n$. Then xz begins with a 0 but ends in 1. So $xz \notin L$, which is a contradiction.
       - Case 2: $y = 0^m$. In this case, $xz = 10^{p-m}10^p1$. Since the string starts and ends with a 1, there must be a 1 in the middle as well. However, there is only one 1 remaining in the string. $w_1 = 0^{p-m}$ and $w_2 = 0^p$. Since $m > 0$, $|w_1| \neq |w_2|$. But then $xz \notin L$, which is a contradiction.

     So we find that $xz \notin L$, which contradicts our assumption that L is regular. Thus, L is not regular.

   - $L_2 = \{w_1w_2 \mid |w_1| = |w_2|$, and w_1 is larger than w_2 when interpreted as a binary number$\}$. Examples: 10011000, 10, 1101.

     Solution 1. Assume L is regular, and let p be the pumping length.
     2. Let $s = 10^p1^{p-1}$. Since $10^{p-1} > 01^{p-1}$, $s \in L$. Also note $|s| \geq p$.
     3. Let $s = xyz$, where $|xy| \leq p$, $|y| > 0$.
     4. Consider xz. Note that if y has odd length, $xz \notin L$ and we are done. So suppose y has length n and n is even. Then the second half of xz is the binary number $1^{p-\frac{n}{2}}$. This is the largest possible binary number with $p - \frac{n}{2}$ digits, so it is not possible that the left hand side is a larger binary number. Thus $xz \notin L$. This contradicts the pumping lemma, so L is not regular, Q.E.D.

     Another proof: Choose $s = 0^p10^p0$. Then $x = 0^a, y = 0^b$ such that $b > 0$ and

$a+b \le p$. Choose $i=2$, so $xy^iz = xyyz = 0^{p+b}10^p0$. The left number can only be zero, and the right number is greater than zero, so $xyyz \notin L$ and we have a contradiction.