Fall Semester 2009, Wednesday 21 October 2009
22C:131 Midterm Exam

Instructions:

The maximum number of points is 100. To contest for maximum credit obey the rules:

1. Write your name on the booklet provided for answering the exam questions and be sure to return it back at the end of the class.
2. Solve all three problems that follows.

Problem I: (30 points). Let $\Sigma = \{a, b\}$ and C_k, $k \geq 1$, be the language consisting of all strings that contain an a exactly k places from the right-hand end.

1. (5 points) Give a regular expression that specify C_k for each $k \geq 1$.
   Solution sketch: $C_k = L(\Sigma^* a \Sigma^{k-1})$.
2. (15 points) Describe an NFA N_k with $k+1$ states that recognizes C_k.
   Solution sketch: The NFA N_k guesses when it has read an a that this a appears at most k symbols from the end. Then it counts $k-1$ more symbols and enters an accept state. Hence, N_k has an initial state q_0 and and additional states q_1, ..., q_k. State q_0 has transitions on both a and b back to itself, and on a also to state q_1. For $1 \leq i \leq k-1$, state q_i has transitions on a and b to state q_{i+1}. State q_k is an accept state with no transition arrows coming out of it.
3. (10 points) Give a formal definition of N_k.
   Solution sketch: $N_k = (Q, \Sigma, \delta, q_0, F)$ where:
   (a) $Q = \{q_0, q_1, \ldots, q_k\}$;
   (b) $\Sigma = \{a, b\}$;
   (c)
   $$\delta(q, x) = \begin{cases} \{q_0\}, & \text{if } q = q_0 \text{ and } x = b \\ \{q_0, q_1\}, & \text{if } q = q_0 \text{ and } x = a \\ \{q_{i+1}\}, & \text{if } q = q_i \text{ for } 1 \leq i < k \text{ and } x \in \Sigma \\ \emptyset, & \text{if } q = q_k \text{ or } x = \epsilon. \end{cases}$$
   (d) Start state is q_0;
   (e) The final set of states is $F = \{q_k\}$.

Problem II: (30 points). Consider the language $F = \{a^i b^j c^k | i, j, k \geq 0$ and if $i = 1$ then $j = k\}$ over the alphabet $\Sigma = \{a, b, c\}$.

1. (5 points) Give two strings $w_1, w_2 \in \Sigma^*$ such that $w_1 \notin F$ and $w_2 \in F$
   Solution sketch: $w_1 = ab^2c^3$, $w_2 = ab^2c^2$.

2. (10 points) Show that F is not regular.
   Hint: you may use pumping lemma or the fact that $L = \{ab^nc^n| \geq 0\}$ is not regular.
   Solution sketch: The language $L = \{ab^nc^n|n \geq 0\}$ is not regular. On the other hand $L = F \cap L(ab^*c^*)$ where $L(ab^*c^*)$ is the language specified by the regular expression ab^*c^*. Since regular languages are closed under intersection, if F were regular than L would be regular too. But we know that L is not regular so, F is not regular either.

3. (10 points) Show that F acts like a regular language. That is, give a pumping length p and demonstrates that F satisfies the three conditions of the pumping lemma for this value of p.
   Solution sketch: Consider $p = 2$ and $s \in F$, $|s| \geq 2$. We show that s can be pumped by the following cases, depending on i, the number of a-s in s.

   (a) If $i = 0$ the number of a-s in s is 0s, that is $s = b^*c^*$. We can pump this string by setting x to be ϵ, y to be the first letter of s and z to be the rest of s. That is $xy^kz = b^*c^* \in F$.

   (b) If $i = 1$ the number of a-s in s is 1, i.e., $s = ab^nc^n$ for $n \geq 1$. We can pump this string by setting x to be ϵ, y to be the first letter of s and z to be the rest of s. That is, $xy^kz = a^kb^nc^n \in F$ because $k \neq n$ does not exclude the inclusion $a^nb^nc^n \in F$.

   (c) If $i > 1$ the number of a-s in s is greater than 1, i.e., $s = a^kb^*c^*$. We can pump this string by setting x to be ϵ, y to be the first letter of s and z to be the rest of s. That is, $xy^kz = a^kb^*c^* \in F$ because $k > 1$.

   In all these cases using the division: $x = \epsilon$, y is the first symbol of s, z is the rest we have:
   (i) $xy^iz \in F$ for any $i \geq 0$, (ii) $|y| = 1$, i.e. $|y| > 0$, and (iii) $|xy| \leq 2$.
   as required by the pumping lemma.

4. (5 points) Explain why (2) and (3) above do not contradict pumping lemma.
   Solution sketch: pumping lemma is not violated because it states only that regular languages satisfy the three conditions. Pumping lemma does not states that nonregular languages fail to satisfy the three conditions.

Problem III: (40 points). Let $\Sigma = \{0, 1\}$. For $w \in \Sigma^*$ denote by $\sim w$ the "one's complement" of w. That is, $\sim w$ is obtained from w by exchanging '0'-s and '1'-s, and $\sim \epsilon = \epsilon$. For example $\sim 0110 = 1001$. Consider the language $L = \{w \in \Sigma^* | \sim w = w^R\}$ where w^R is the reversal of w. For example, $0101, 1010 \in L$ because $\sim 0101 = 1010 = 0101^R$ and $\sim 1010 = 0101 = 1010^R$ but $0110 \notin L$ because $\sim 0110 = 1001$ and $0110^R = 0110$.

1. (20 points) Construct a CFG G that generates L (20 points). **Hint:** observe that if $w \in L$ then $|w| = 2k$ for an integer $k \geq 0$
   Solution sketch:
   (a) First observe that L has no string of odd length because in a string w of odd length w and w^R must have the same middle letter while the complement changes it, and therefore $w^R \neq \sim w$.
   (b) For strings of even length, in order to have $\sim w = w^R$ the first letter of w must be the complement of its last letter, the second must be the complement of the next to the last, etc. Therefore we must envision the generation of a string from both ends towards the middle. This requires the rules $S \to \epsilon | 0S1 | 1S0$. Hence the grammar G is $G = (\{S\}, \{0, 1\}, \{S \to \epsilon | 0S1 | 1S0\}, S)$.

2. (20 points) Prove that the grammar G you have constructed generates L.
   Hint: Use set equality and induction on the length of strings $w \in L$.
   Solution sketch: $L(G) = L$**:**
   Since languages are sets, their equality is established by proving the two inclusions: $L(G) \subseteq L$ and $L \subseteq LG$.
   (a) The inclusion $L(G) \subseteq L$ is a consequence of the construction of G.
   (b) The inclusion $L \subseteq L(G)$ established by proving that if $w \in L$ then $w \in L(G)$, using inductions on the length of strings $w \in L$.
      i. **Induction basis:** strings of length 0 are in the language because $\sim \epsilon = \epsilon$.
      ii. **Induction step:**
         - Assume that all strings $w \in L$, $|w| = 2n$, $n \geq 0$, are generated by G.
         - Show that if $w \in L$, $|w| = 2(n + 1)$ then it is also generated by G. For a string $w \in L$ of length $2(n + 1)$ the first and last letters must be opposite. Thus, $w = 0w'1$ or $w = 1w''0$. Since $|w'| = |w''| = 2n$, by assumption there are derivations $S \overset{*}{\Rightarrow}_1 w'$ and $S \overset{*}{\Rightarrow}_2 w''$. Then $S \overset{S \to 0S1}{\Rightarrow} 0S1 \overset{*}{\Rightarrow}_1 0w'1$ or $S \overset{S \to 1S0}{\Rightarrow} 1S0 \overset{*}{\Rightarrow}_2 1w''0$ is a derivation of w using the grammar G.