Midterm Review

October 10, 2011

Important Things

Here are some things you should make sure to study:

- Definition of DFA, NFA, Regular Expressions
- How to convert between DFAs, NFAs, and Regular Expressions
- Closure properties of regular languages
- Countability and Uncountability
- Pumping Lemma
- Definition of CFG

Practice Problems

1. Are the following true or false? Let L_1 and L_2 be arbitrary languages over an alphabet $\Sigma = \{a, b\}$.

   (a) If L_1 is infinite, then $L_1 \circ L_2$ is infinite
   Solution: False. consider $L_1 = \{a\}^*$ and $L_2 = \emptyset$

   (b) $(L_1^*)^* = L_1^*$
   Solution: True.

   (c) $L_1^* = (L_1 L_1)^*$
   Solution: False. Consider $L_1 = \{a\}$.

   (d) $(L_1 \circ L_2)^* = L_1^* \circ L_2^*$
   Solution: False. Consider $L_1 = \{a\}$ and $L_2 = \emptyset$.

   (e) If $L_1 \subset L_2$, then $L_1^* \subset L_2^*$
   Solution: True.

   (f) If L_1 is regular, then it is context-free.
   Solution: True.

   (g) If L_1 is finite, then it is regular.
   Solution: True.

   (h) If L_1 is non-regular, then $\overline{L_1}$ is non-regular.
   Solution: True.

2. Draw a DFA that recognizes the following languages. Assume $\Sigma = \{a, b\}$

   (a) $\{w \mid |w| = 5\}$

   (b) All strings except the empty string

   Solution: See attached images.

3. Draw an NFA the recognizes the language a^*b^*aba

4. Are the following languages regular or not?

   (a) $\{x = y + z \mid x, y, z$ are binary integers, and x is the sum of y and $z\}$ (the alphabet here is $\{0, 1, =, +\}$). *Solution:* Non-regular. We show it does not satisfy the pumping lemma. Suppose the pumping length is p. Then consider the string $1^p = 1^p + 0^p$. Then xy has to just consist of 1's, and so if we pump down the left hand side of the equality will be less than the right hand side.

   (b) $\{w \mid w$ is the binary representation of a number greater than $3\}$ *Solution:* Regular. Consider the regular expression $0^*1(0 \cup 1)(0 \cup 1)(0 \cup 1)^*$.

5. Give a context free grammar for the following languages:

   (a) The set of strings over $\{a, b\}$ with more a's than b's.
   Solution:
   $S \to TaT$
   $T \to \epsilon \mid TaTbT \mid TbTaT \mid Ta \mid aT$

   (b) $\{a^i b^j c^k \mid i = j$ or $j = k$ and $i, j, k \geq 0\}$. *Solution:*
   $S \to C \mid D$
   $C \to EF$
   $E \to \epsilon \mid aEb$
   $F \to \epsilon \mid Fc$
   $D \to GH$
   $G \to \epsilon \mid Ga$
   $H \to \epsilon \mid bHc$

6. Sometimes it is easier to prove properties of regular languages by basing the argument on DFAs. Other times it is easier to work with NFAs or Regular Expressions. Each of the following are easier to prove using one of these three representations (in the humble opinion of one of your TFs). Prove the following statements.

   (a) If L_1 is a regular language, then $\overline{L_1}$ is regular.
   Solution: Take the DFA for L_1 and swap the accept states and the non-accept states. It's not at all clear how to complement a R.E. And for an NFA, we have to worry about an accept state being connected by an epsilon transition to a non-accept state.

   (b) For all languages L, define $DROPOUT(L) = \{w \mid w$ is some string from L with exactly one character deleted $\}$. Prove that if L is regular, then $DROPOUT(L)$ is regular.
   Solution: Proceed by structural induction on the regular expression for R. We will construct a function f that given a regular expression for a language, returns

the *DROPOUT* form of that language. For the base cases we have $f(\epsilon) = \emptyset$, $f(\sigma) = \epsilon$, and $f(\emptyset) = \emptyset$.
Then we have:
$f(R_1 \circ R_2) = f(R_1) \circ R_2 \cup R_1 \circ f(R_2)$.
$f(R_1 \cup R_2) = f(R_1) \cup f(R_2)$
$f(R_1^*) = R_1^* f(R_1) R_1^*$

The last one is subtle. Suppose $x \in f(R_1*)$ then we started with some string of the form $w_1 \cdots w_n$ for $n \geq 1$ where each $w_i \in R_1$ and then deleted a character. (Why do we put on the constraint $n \geq 1$ here? Because if we had started with $n = 0$, we wouldn't have been able to delete a character). Now then, when we delete this character, it has to come from one of the w_i. That w_i with that character removed is in $f(R_1)$, and then the remaining things to its left and right are just in R_1^*