Computational Models, Spring 2013 Exrecise #2

DFA's, NFA's and non-regular languages

1. For each of the following languages draw an NFA, convert it to a DFA and draw the converted DFA.

   (a) $L = \{w : w \text{ contains the substring } 0011\}$ above $\Sigma = \{0, 1\}$

   (b) $L = \{w : \#_0(w) = 2 \text{ or } \#_1(w) = 1\}$ above $\Sigma = \{0, 1\}$

   (c) $L = \{a^i b^j c : i, j \geq 1\} \cup c^*$ above $\Sigma = \{a, b, c\}$

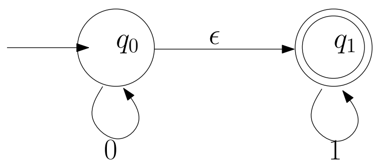
2. Write a regular expression for each of the following languages above. Give a thorough explanation (a formal proof is not needed)

   (a) $L = \{w : \ |w| \text{mod} 3 = 1\}$ above $\Sigma = \{a, b, c\}$
   $\mathbf{R = ((a + b + c)\ (a + b + c)\ (a + b + c))^*}$

   (b) $L = \{w \in L[a^*b^*]\ :\ |w| \text{mod} 3 = 2\}$ above $\Sigma = \{a, b\}$
   $\mathbf{R = aa(aaa)^*(bbb)^* + a(aaa)^*(bbb)^*b + (aaa)^*(bbb)^*bb}$

   (c) $L = \{w_1w_2 : w_1 \in \{0^*\} \text{ and } w_2 \in \{101\}^* \text{ and } |w| \text{mod} 2 = 0\}$ above $\Sigma = \{0, 1\}$
   $\mathbf{R = ((00)^*(101101)^* + 0(00)^*(101101)^*(101)}$

3. Consider the following NFA



   (a) What is the regular expression that expresses the language?
   $\mathbf{L = L[(0^*)1(1^*) + \epsilon]}$ **above** $\mathbf{\Sigma = \{0, 1\}}$

   (b) Construct the minimal DFA representing the language

4. We showed in class that Regular languages are closed under union. Are they closed under infinite union? Prove your answer formally.
   No, consider $\mathbf{L_i = \{a^ib^i\}}$ **which is regular for every i (finite language). Their union is** $\mathbf{\{a^nb^n}$ **s.t.** $\mathbf{n \in \mathbb{N}\}}$ **which is not regular**

5. Let L be a regular language state for each language if it is regular or not. Give a complete formal proof:

   (a) $L_a = \{ww' : w \in L \text{ and } w' \notin L\}$
   REGULAR - this is simply $\mathbf{LL^c}$

   (b) $L_b = \{y : \exists x, z\ s.t. |x| = |z| \text{ and } xyz \in L\}$
   REGULAR - ("keep middle") for each pair of states p and qsuch that p is reachable from $\mathbf{q_0}$ **in the same number of steps that a state in F is reachable from q (that is, the potential x and z) we will have an epsilon transition form a new initial state to p, and q will be the accepting state (we will have a replica for each pair of such states). Note that we don't need an algorithm to find the relevant pairs, only that such set of pairs exists (and is finite..)**

(c) $L_c = \{xy : |x| = |y| \text{ and } \exists\sigma \ s.t. \ x\sigma y \in L\}$
NON REGULAR - choose $\mathbf{L = 0^*12^*}$ thus $\mathbf{L_c \cup 0^*2^* = 0^n2^n}$ which is not regular (and we used closure of regular languages)

(d) $\text{Pref}(L) = \{x : \exists y \ s.t. \ xy \in L\}$
REGULAR - for each state that LEADS to an accepting state, add an epsilon transition to an accepting state

6. Prove that the following languages are not regular

   (a) $L_a = \{(ab)^n c^n : n \geq 0\}$ above $\Sigma = \{a, b, c\}$
   Use pumping lemma, for length p choose $\mathbf{w = xyz = (ab)^p c^p}$. The y may be either at the a, ab, b or bc part. Every part, if pumped yields a word not in the language

   (b) $L_b = \{a^i b^j c^k : i = j \text{ or } j = k\}$ above $\Sigma = \{a, b, c\}$
   assume that regular and get contradiction by $\mathbf{L_b \cap \{a^*b^*\} = \{a^nb^n\}}$

   (c) $L_c = \{ww : w \in \{0, 1\}^*\}$
   assume that regular and get contradiction by $\mathbf{L_c \cap \{0^*110^*\} = \{0^n110^n\}}$

   (d) $L_d = \{\text{balanced strings }\}$ above $\Sigma = \{(,)\}$
   A string is balanced if the number of (equals the number of) and while reading the string, the number of (is equal or greater than the number of).
   assume that regular and get contradiction by $\mathbf{L_d \cap \{(^*\ )^*\} = \{(^n\ )^n\}}$