CS360 Fall 2013-Assignment 2

Due Wednesday, October 9th, 11:59am

1. [2×10 **points**]For each of the languages, L_1, L_2 below, construct a DFA that computes that languages. Represent your DFA's as tuples, $M = (\Sigma, Q, q_0, \delta, F)$, with a transition table for δ, and draw the transition diagram (graph representation) of M. Provide a brief justification of your construction.

   (a) $L_1 = \{x \in \{a, b, c\}^* : ab \text{ is not a substring of } x\}$.

   (b) $L_2 = \{x \in \{a, b\}^* : n_a(x) \text{ is even, or } n_b(x) \text{ is even}\}$, (where, for a letter $\sigma \in \Sigma$, and a word $w \in \Sigma^*$, $n_\sigma(w) =$ *the number of occurrences of* σ *in* w.

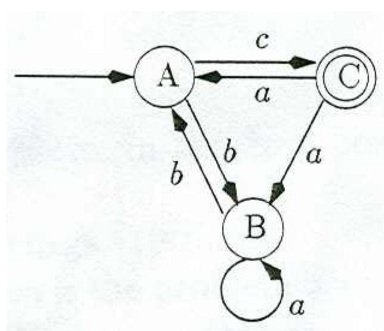
2. (a) [**10 points**] Let $M = (\Sigma, Q, q_0, \delta, F)$ be a DFA with a single accepting state (that is, $|F| = 1$). Prove that for every $x, y, \in L(M)$,

   $$\{z \in \Sigma^* : xz \in L(M)\} = \{z \in \Sigma^* : yz \in L(M)\}$$

   (b) [**15 points**] Find a regular language, L, (describe it either as $L(A)$ for some DFA, A, or as $L(r)$ for some regular expression r) that can not be computes by any DFA that has a single accepting state. Prove your claims.

   (c) [**10 points**] Prove that for every DFA, M, there exist a finite number of DFA's, $A_1, \ldots, A_k$, each having a single accepting state, such that $L(M) = \bigcup_{i=1}^{k} L(A_i)$.

3. [**15 points**] Convert the following NFA into a DFA. That is, construct a DFA that computes the same language computed by the NFA described by the transition diagram below. Explain the steps you took in that transformation.



4. We discussed in class how do construct NFA's for keyword searches. Let us elaborate about that construction. Given any finite set of words $\{w_1, \ldots w_k\}$ over the alphabet $\{a, b\}$, let $|w_1| \ldots |w_k|$, denote their lengths (respectively),

   (a) [**15 points**] Construct an NFA, $N_{(w_1, \ldots w_k)}$ that has $1 + \sum_{i=1}^{k} |w_i|$ states that accepts a word w if and only if it contains at least one of these k words as a substring (namely, for some $u, v \in \{a, b\}^*$ and some $i \leq k$, $w = uw_iv$).

(b) **[15 points]** Construct a DFA, A_{w_1} that has $1 + |w_1|$ states that accepts a word w if and only if it contains w_1 as a substring (namely, $w = uw_1v$, for some $u, v \in \{a, b\}^*$).

(c) **[Bonus 10 points]** Construct a DFA, $A_{(w_1,\ldots w_k)}$ that has at most $1 + \sum_{i=1}^{k} |w_i|$ states that accepts the language $L(N_{(w_1,\ldots w_k)})$.