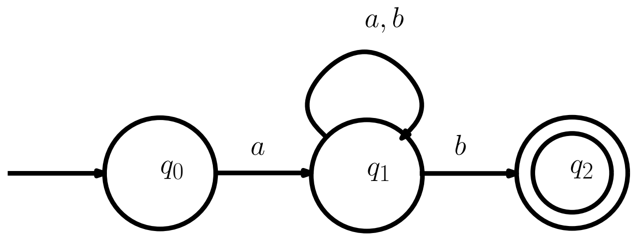
Computational Models, Spring 2016 Exercise #3

Context free grammars and Pushdown automata (and regular languages)

1. Give pushdown automata that recognize the following languages (only a drawing is required).
   (a) $L_1 = \{w \in \{0,1\}^* \ s.t.\ w$ contains at least three 1s$\}$
   (b) $L_2 = \{w \in \{0,1\}^* \ s.t.\ w = w^R$ and the length of w is odd$\}$
   (c) $L_3 = \{w \in \{0,1\}^* \ s.t.\ w = w^R\}$
   (d) $L_4 = \{a^i b^j c^k \ | i,j,k \geq 0$ and $i = j$ or $j = k\}$
   (e) $L_5 = \{a^i b^j c^k \ | i,j,k \geq 0$ and $i + j = k\}$
   (f) $L_6 = \{a^{2n} b^{3n} \ | n \geq 0\}$
2. For each of the following languages, give a context free grammars that derives the language (no need to prove your answer).
   (a) $L = \{a^i b^j c^i \mid 0 \leq i,j\}$
   (b) $L = \{a^i b^j c^k \ | i + k = j\}$
   (c) $L = \{w \mid w \in \{0,1\}^*$ and w has twice as many 1's as 0's$\}$
   (d) $L = \{a^i b^i \mid 0 \leq i\} \cup \{0\}^* \cup \{1\}^*$
   (e) $L = \{a^i b^j \mid i \neq j\}$
3. Prove that the following languages are not context free.
   (a) $L = \{a^n b^{n+1} c^{n+2} \mid n \geq 0\}$
   (b) $L = \{a^n b^m c^n d^m \mid m,n \geq 0\}$
   (c) $L = \{a^i b^j c^k \mid 0 < i < j < k < 2i\}$
4. Consider the following NFA:



   (a) Describe (in words) the language accepted by the NFA and give a regular expression for the language.
   (b) Prove your answer formally.
   Hint: In your proof, use induction on the length of the input. Be sure to state your induction hypothesis explicitly. Show (inductively) that the following three conditions hold for any w:
      i. $q_0 \in \hat{\delta}(q_0, w) \leftrightarrow w = \epsilon$
      ii. $q_1 \in \hat{\delta}(q_0, w) \leftrightarrow w$ begins with a
      iii. $q_2 \in \hat{\delta}(q_0, w) \leftrightarrow w$ begins with a and ends with b

5. Consider a *minimal* DFA A that works on an alphabet Σ and define $L = L(A)$. Assume that you are told that it has m states, but you know nothing more about A it is a “black box” and your only way of getting information about A is to feed in words and observe whether they are accepted.

   We will construct an algorithm for determining the transition diagram of A from (any finite number of) such observations.

   One extremely expensive method would be to first construct all possible minimal DFAs of size m, (a HUGE number of DFAs this would give you!) then start testing all words from Σ^* in alphabetical enumeration, weeding out all DFAs that on some word behave different from your black box. Then at some point only one of your DFAs is left, i.e., problem solved. Don't do it this way, but reconstruct the equivalence relation $\sim_L$ from the Myhill-Nerode theorem—that gives a much faster reconstruction.

   (a) First, explain (in words, no need to formally prove) that in order to check whether for any words u, v it holds that $u \sim_L v$ we need to test words of length at most m (how many such words are there?).

   (b) Now, explain how to choose m representative words, one for each equivalence class in $\sim_L$.

   (c) Finally, to reconstruct A identify each word chosen in the previous subsection with a different state of A and explain how to reconstruct the transition function.