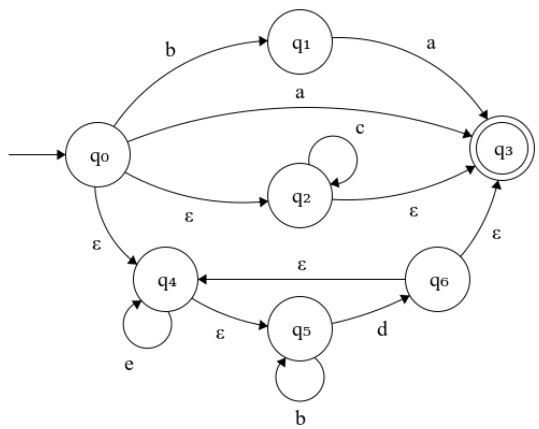
Quiz Solutions (Automata 2022)

Automata Theory Monsoon 2022, IIIT Hyderabad

September 1, 2022

1. [3 points] Construct an NFA for $R = a \cup (e^*b^*d)^* \cup ba \cup c^*$.

   Solution:



2. [3 points] Prove that regular languages are closed under dropout.

$$Dropout(A) = \{xz \mid xyz \in A, y \in \Sigma \text{ and } x, z \in \Sigma^*\}$$

   Solution: Since A is a regular language, it must be recognized by a DFA. Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA that recognises A. Now, we will construct an NFA $N = (Q', \Sigma \cup \{\epsilon\}, \delta', q'_o, F')$ that recognizes Dropout(A).

   The **idea** behind the construction is that N simulates M on its input, non-deterministically guessing the point at which the dropped out symbol occurs. At that point N guesses the symbol to insert in that place, without reading any actual input symbol at that step. Afterwards, it continues to simulate M.

   We implement this idea in N by keeping two copies of M , called the top and bottom copies. The start state is the start state of the top copy. The accept states of N are the accept states of the bottom copy. Each copy contains the edges that would occur in M. Additionally, include ϵ edges from each state q in the top copy to every state in the bottom copy that q can reach.

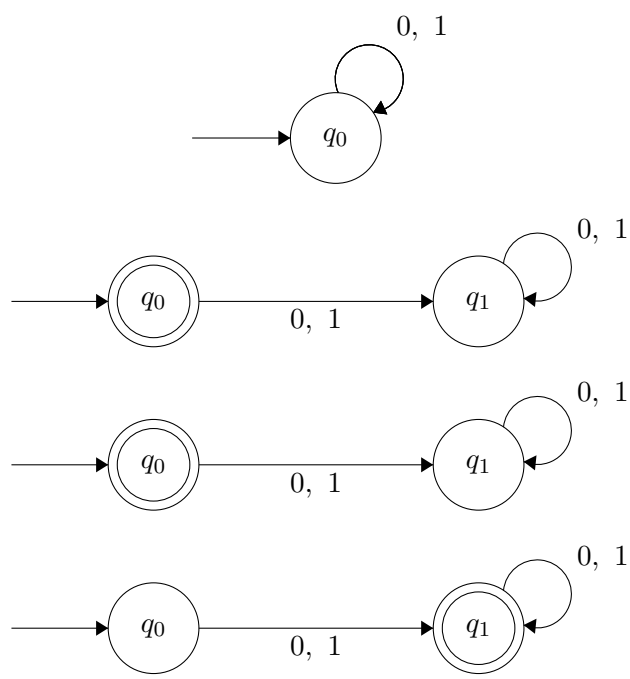
   We describe N formally. The states in the top copy are written with a T and the bottom with a B, thus: (T, q) and (B, q).

- $Q' = \{T, B\} \times Q$
- $q_0' = (T, q_0)$
- $F' = \{B\} \times F$
- $\delta'((T,q),a) = \begin{cases} \{(T, \delta(q,a))\} & a \in \Sigma \\ \{(B, \delta(q,b)) \mid b \in \Sigma\} & a = \epsilon \end{cases}$
- $\delta'((B,q),a) = \begin{cases} \{(B, \delta(q,a))\} & a \in \Sigma \\ \emptyset & a = \epsilon \end{cases}$

3. [4 points] Suppose we have $\Sigma = \{0, 1\}$. Show that the following languages are regular by drawing a DFA for them:

   (a) $L_1 = \Phi$
   (b) $L_2 = \Phi^*$
   (c) $L_3 = \{\epsilon\}^*$
   (d) $L_4 = \Sigma^* \setminus \{\epsilon\}$

   Solution: The solutions in order (a, b, c, d) are as follows:



4. [5 points] Consider G, a CFG in Chomsky Normal Form with b variables. Let $L(G)$ be the language generated by G. Show that if $w \in L(G)$ such that $|w| \geq 2^b$ then G can generate infinite distinct strings.

   Solution: Since G is a CFG in Chomsky normal form, every derivation can generate atmost two non-terminals, so that in any parse tree using G, an internal node can have atmost two children. This implies that every parse tree with height k has at most $2^k - 1$ internal nodes.

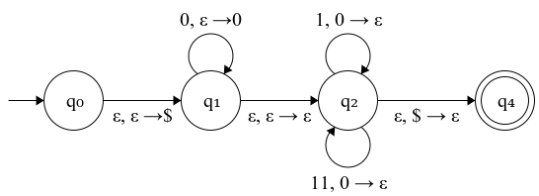
   If G generates some string with a derivation having at least 2^b steps, the parse tree of that string will have at least 2^b internal nodes. Based on the above argument, this parse tree has height is at

least $b+1$, so that there exists a path from root to leaf containing $b+1$ variables. By pigeonhole principle, there is one variable occurring at least twice. So, we can use the technique in the proof of the pumping lemma to construct infinitely many strings which are all in $L(G)$.

5. [5 points] Let $\Sigma = \{0, 1\}$. Consider the language $L = \{0^n 1^m | n \leq m \leq 2n\}$. Is L regular or context-free? If you think it is regular, write the regular expression for it and draw the corresponding DFA/NFA. If you think it is context-free, write the grammar rules and draw the corresponding PDA.

   <u>Solution</u>: The language is not regular but is context free and requires a stack to keep in track of the number of 1s. Thus, we shall construct a PDA for this language.

   The grammar for this language would be,

$$S \to 0S1|0S11|\epsilon \tag{1}$$

   The corresponding PDA of this language would be,