CSCI 2670 – Theory of Computation

Homework 1

Due Monday June 15, 2020 by 11:55 PM

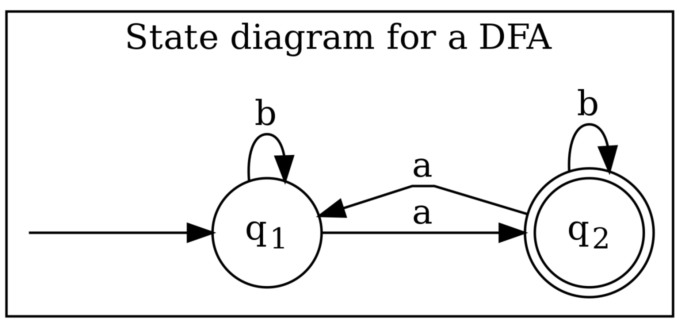
Whenever a problem requests a DFA or NFA you can provide either a state diagram (i.e., a picture of the machine) or the grafstate code.

The text uses the notation a^+ to indicate a string of 1 or more a's. In other words, a^+ is the same as aa^*.

25 points each

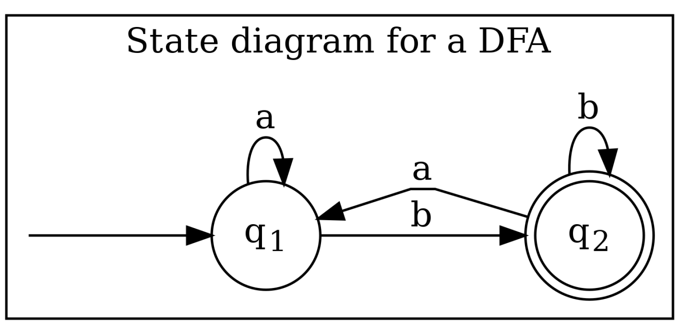
1. Problem 1.4 on page 83 Part f

   Here is a DFA that only accepts strings with an odd number of a's:



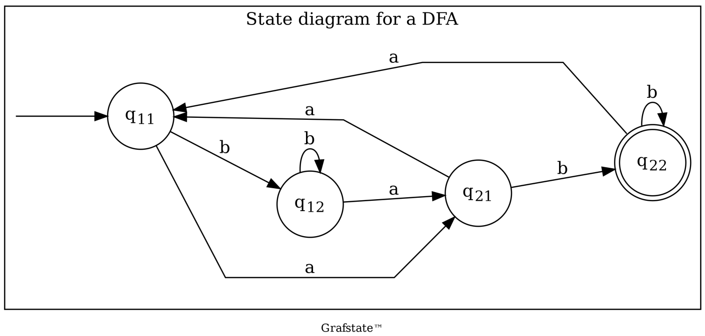
Grafstate™

Here is a DFA that only accepts strings ending in a b:



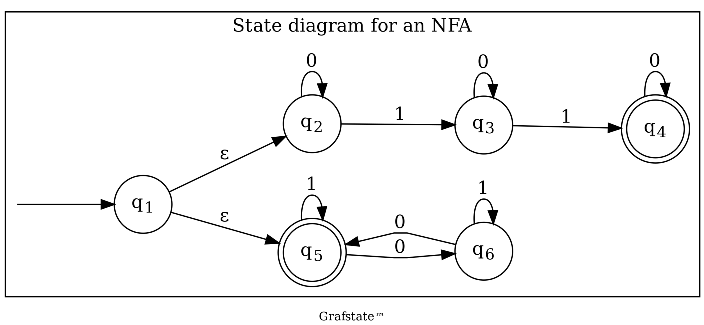
Grafstate™

Here is the combined DFA that only accepts strings with both these properties:



2. Problem 1.7 on page 84 Part c

Here is an NFA that only accepts strings that either contain an even number of 0's or exactly two 1's:



3. Problem 1.31 on page 88

The idea of this proof is that if we have an NFA M such that $L(M) = A$, then the NFA M^R made by reversing the directions of the arrows on the diagram of M will have the language A^R. This intuitively makes sense, but in my opinion still warrants a careful proof.

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA with language A. Let δ^{-1} denote the inverse function of δ; in other words, the function defined by $\delta^{-1}(q, a) = \{p \in Q : \delta(p, a) = q\}$ for some $(q, a) \in Q \times (\Sigma \cup \{\epsilon\})$. Let $Q^R = Q \cup \{q_F\}$, and let $q_0^R = q_F$. Let $F^R = \{q_0\}$. Finally, let $\delta^R(q, a) =$

$$\begin{cases} F, & q = q_F \text{ and } a = \epsilon \\ \delta^{-1}(q, a), & \text{otherwise} \end{cases}$$

for some $(q, a) \in Q \times (\Sigma \cup \{\epsilon\})$. Let $M^R = (Q^R, \Sigma, \delta^R, q_0^R, F^R)$. We will now show that $L(M^R) = A^R$.

Let $w \in A^R$. Then, $w^R \in L(M)$, so M accepts w^R. Let $S = (s_1, ..., s_n)$ denote a sequence of states that M undergoes upon processing w^R, with $s_1 = q_0$ and $s_n \in F$. Let $C = (c_1, ..., c_{n-1})$ denote the corresponding sequence of characters and ϵ's that caused the transitions. Now, we will run M^R under input w. Start by taking the ϵ-transition from q_0^R to s_n (we know that this transition exists by our construction of d^R). Now, for every $1 \leq k \leq n-1$, we know that c_k caused M to transition from s_k to s_{k+1}. Thus, $s_{k+1} \in \delta(s_k, c_k)$, and $s_k \in \delta^{-1}(s_{k+1}, c_k) = \delta^R(s_{k+1}, c_k)$ by the definition of δ^R. M^R is therefore capable of undergoing the sequence of states $(q_0^R, s_n, ..., s_2, q_0)$ upon processing the sequence $(\epsilon, c_{n-1}, ..., c_1)$. Note that this sequence is simply w but padded with some number of ϵ-transitions. Therefore, M^R accepts w and $w \in L(M^R)$. This proves that $A^R \subseteq L(M^R)$.

Now, let $w \in L(M^R)$. Like before, let $S = (s_1, ..., s_n)$ denote a sequence of states that M^R undergoes upon processing w, with $s_1 = q_0^R$ and $s_n = q_0$, and let $C = (c_1, ..., c_{n-1})$ denote the corresponding sequence of characters and ϵ's that caused the transitions. We will now show that M accepts w^R. Note that for every $2 \leq k \leq n-1$, $s_{k+1} \in \delta^R(s_k, c_k) = \delta^{-1}(s_k, c_k)$. Therefore, $s_k \in \delta(s_{k+1}, c_k)$, which shows that M can undergo the sequence of states $(q_0, s_{n-1}, ..., s_2)$ under the sequence $(c_{n-1}, ..., c_2)$. We also know that $s_2 \in F$, since q_0^R only transitions to elements of F. Since the aforementioned sequence is simply w^R but padded with some number of ϵ-transitions, M accepts w^R and $w^R \in L(M) = A$. Therefore, $w \in A^R$, which proves that $L(M^R) \subseteq A^R$.

We have now shown that $L(M^R) = A^R$. This completes the proof that A^R is a regular language. $\square$

4. Consider the NFA in Example 1.38 on page 54 of the text.

   - Draw this NFA in grafstate. Submit your grafstate code.

Please find my grafstate code in the separate "grafstate_code.txt" file.

- Using grafstate, find an accepting computation path for the string 0011010. To do this, first draw the NFA in grafstate and select "Explore this NFA" then enter the string and select "Run this machine."

  The nondeterminism tree lists $(q_1, q_1, q_1, q_1, q_2, q_3, q_4, q_4)$ as an accepting computation path for this string.

Written solutions must be submitted as a pdf document on eLC. Create you solutions using LaTex. The grafstate code may be submitted as a separate text file.