1. **(15 points)** Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Figure 7 depicts two finite state machines that read Let L_a and L_b denote the languages recognized by DFA (a) and DFA (b) respectively.

   (a) **(6 points)** For each of L_a and L_b list three strings in Σ^* that are in the language and three strings in Σ^* that are not in the language.
   Solution:
   $\{abc, aabbcc, aaaabccc\} \subset L_a$
   $\{\epsilon, acabc, c\} \cap L_a = \emptyset$
   $\{\epsilon, aab, aaacaab\} \subset L_b$
   $\{aaab, aaacc, a^7\} \cap L_b = \emptyset$

   (b) **(3 points)** Write a short, English description of language, L_a. **Solution:**
   Language L_a is the set of all strings starting with one or more a followed by one or more b followed by one or more c.

   (c) **(6 points)** Write a short, English description of language, L_b. **Solution:**
   Language L_b is the set of all strings s such that (1) the $\#a(s) - 2\#b(s) - 3\#c(s) = 0$; (2) For all prefixes x of s, $0 \leq \#a(x) - 2\#b(x) - 3\#c(x) \leq 6$.

2. **(15 points)** (inspired by Sipser exercise 1.6 (p. 84): Give state diagrams of DFAs recognizing the following languages. For each language, the alphabet is $\{\mathtt{0}, \mathtt{1}\}$.

   (a) $\{w \mid w$ begins and ends with the same symbol$\}$.

   (b) $\{w \mid w$ contains three consecutive $\mathtt{1}$s$\}$.

   (c) $\{w \mid w$ contains neither the substring $\mathtt{010}$ nor the substring $\mathtt{101}\}$.

   Solution: See below.

3. **(20 points)** (inspired by Sipser exercise 1.7 (p. 84): Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. For each language, the alphabet is Σ with $\Sigma = \{\mathtt{0}, \mathtt{1}\}$.
   Solution: See below.

   (a) The set of strings w that end with the substring $\mathtt{010}$:

   $$\{w \mid \exists x \in \Sigma^*.\ w = x\mathtt{010}\}$$

   Use four states.

   (b) The set of strings w that contain the substring $\mathtt{010}$:

   $$\{w \mid \exists x, y \in \Sigma^*.\ w = x\mathtt{010}y\}$$

   Use four states.

   (c) The set of strings w that can be written as $x_1 \cdot x_2 \cdots x_k$ for some $k \geq 0$, with each x_i is an element of

   $$\{\mathtt{01}, \mathtt{10}, \mathtt{001}, \mathtt{0011}\}$$

   Use eleven states.

   (d) The set of strings whose length is a multiple of three plus a multiple of five:

   $$\{w \mid \exists m, n \in \mathbb{N}.\ |w| = 3m + 5n\}$$

   Use eight states.

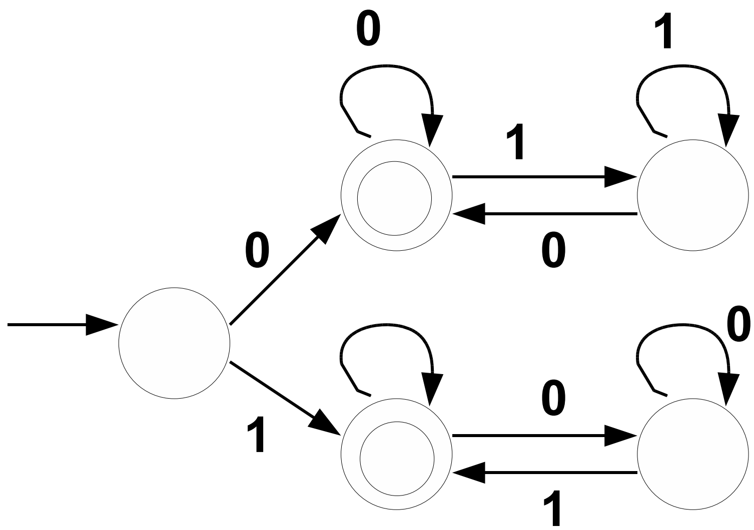

Figure 1: Solution for question 2a

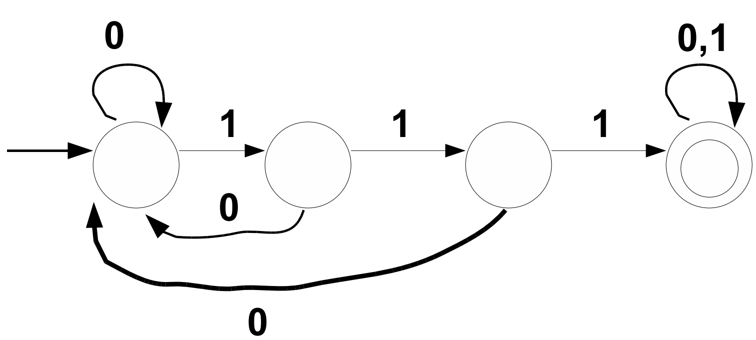

Figure 2: Solution for question 2b

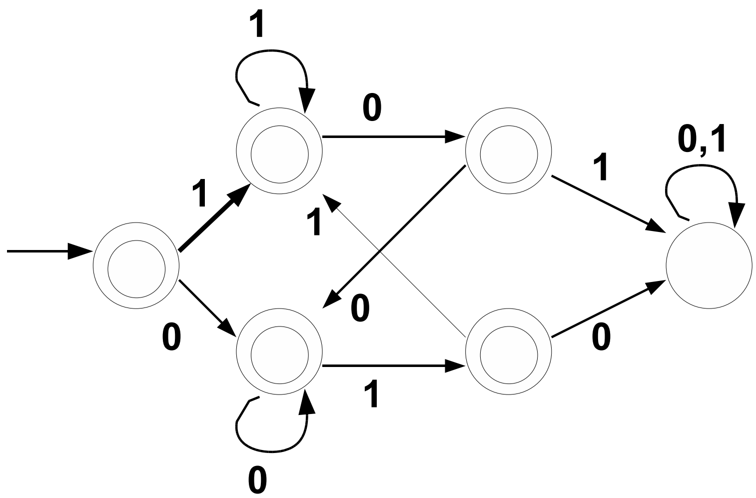

Figure 3: Solution for question 2c

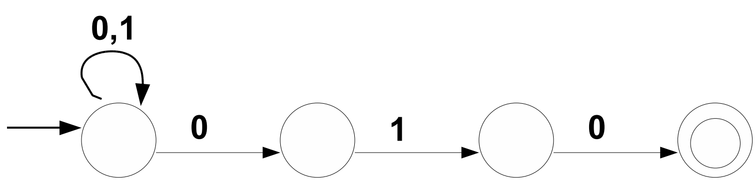

Figure 4: Solution for question 3a

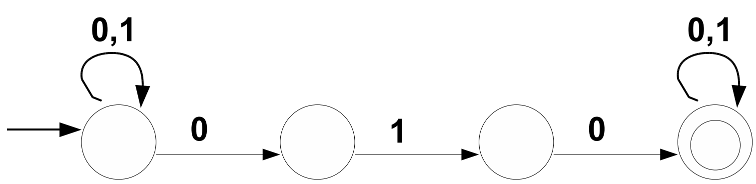

Figure 5: Solution for question 3b

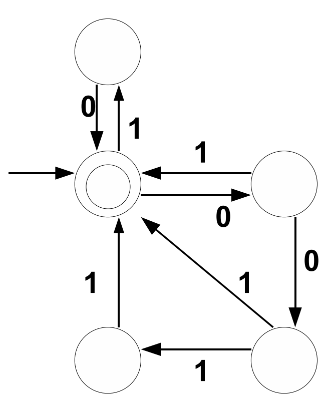

Figure 6: Solution for question 3c

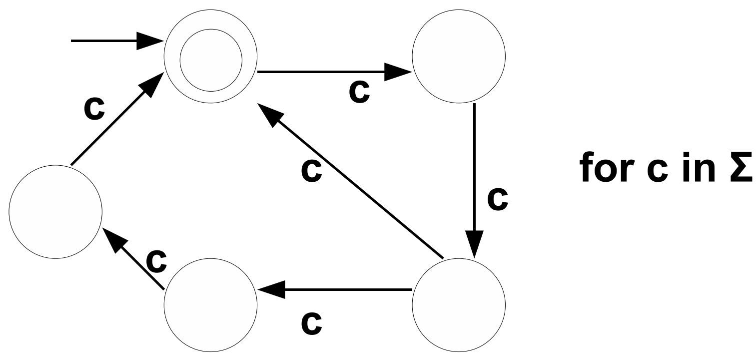

Figure 7: Solution for question 3d

4. **(20 points)**: Let $\Sigma = \{\mathtt{0}, \mathtt{1}\}$. Let L_k be the set of all strings whose k^{th} from end symbol is a $\mathtt{1}$:

$$L_k \quad = \quad \{w \in \Sigma^* \mid \exists x \in \Sigma^*, y \in \Sigma^{k-1}.\ w = x\mathtt{1}y\}$$

In the September 12 notes, we claimed that any DFA that recognizes L_k must have at least 2^k states. Prove this claim.

Solution:

Suppose the claim does not hold, i.e. there exists a DFA $A_k = (Q_k, \Sigma, \delta, q_0, F)$ recognizing L_k with $|Q_k| < 2^k$. Since there are 2^k strings of length k over $\Sigma = \{0, 1\}$, there exists two strings $s_1, s_2 \in \Sigma^k$ such that $s_1 \neq s_2$ and A_k is in the same state after processing both s_1 and s_2. Without loss of generarilty, assume that s_1 and s_2 differ in symbol i, i.e. $s_1^i = 0$ and $s_2^i = 1$ with $0 \leq i \leq k-1$. Then A_k is in the same state after processing strings $s_1 \circ 0^i$ and $s_2 \circ 0^i$, but $s_1 \circ 0^i \notin L_k$ and $s_2 \circ 0^i \in L_k$, a contradiction. Therefore, $|Q_k| \geq 2^k$. $\square$

5. **(20 points, extra credit)**: In this exercise, you will prove the equivalence of NFA acceptances as defined in Sipser and the formulation that I gave in class. This problem statement is long because I first summarize both Sipser's and my definitions of acceptance. If you are comfortable with both, you can skip to the end of the problem statement where I ask you to prove the equivalence of the two formulations.

Let $N_{mrg} = (Q, \Sigma, \Delta, q_0, F)$ be an NFA as defined in the September 12 notes.

Q is a finite set of states;

Σ is a finite alphabet;

$\Delta \subseteq Q \times \Sigma_\epsilon \times Q$ is the transition relation;

$q_0 \in Q$ is the initial state; and

$F \subseteq Q$ is the set of accepting states.

where $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$.

Here are the formulas that we used to define NFA acceptance (see the slides for explanations). $close_\epsilon(q)$ is the subset of Q where $p \in close_\epsilon(q)$ iff

$$p = q$$
$$\exists q' \in close_\epsilon(q).\ (q', \epsilon, p) \in \Delta$$

We extended $close_\epsilon$ to sets as

$$close_\epsilon(G) \quad = \quad \bigcup_{q \in G} close_\epsilon(q)$$

Then we defined:

$$\begin{array}{rcll} step(q,c) & = & close_\epsilon(\{q' \mid (q,c,q') \in \Delta\}), & c \in \Sigma \\ step(G,c) & = & \bigcup_{q \in G} step(q,c), & G \subseteq Q, c \in \Sigma \\ \Delta(G,\epsilon) & = & close_\epsilon(G), & G \subseteq Q \\ \Delta(G, x \cdot c) & = & step(\Delta(G,x),c), & G \subseteq Q, x \in \Sigma^*, c \in \Sigma \end{array}$$

Finally, we said that N_{mrg} accepts s iff

$$\Delta(\{q_0\}, s) \cap F \neq \emptyset$$

Now, for Sipser's version. Given $N_{mrg} = (Q, \Sigma, \Delta, q_0, F)$ as described above, let $N_{ms} = (Q, \Sigma, \delta, q_0, F)$ with $\delta : Q \times \Sigma_\epsilon \rightarrow 2^Q$ where 2^Q is the power set of Q and

$$\delta(q,c) \quad = \quad \{q' \mid (q,c,q') \in \Delta\}$$

Sipser says that NFA N_{ms} accepts string s iff we can find $y_1, y_2, \ldots, y_m \in \Sigma_\epsilon$ and $r_0, r_2, \ldots r_m \in Q$ such that

- $s = y_1 \cdot y_2 \cdots y_m$;
- $r_0 = q_0$;
- for i in $0 \ldots m-1$, $r_{i+1} \in \delta(r_i, y_i + 1)$;
- $r_m \in F$.

Prove that N_{ms} accepts a string (i.e. acceptance by Sipser's condition), iff N_{mrg} accepts the string (i.e. acceptance by the condition given in class).