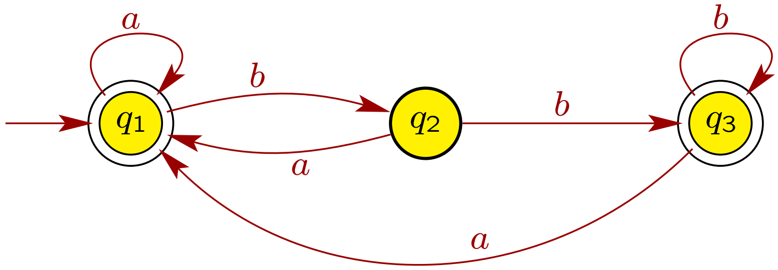
**CS 341: Foundations of Computer Science II**
**Prof. Marvin Nakayama**

# Homework 2 Solutions

1. For the state diagram of the DFA $M$ below, give its formal definition as a 5-tuple.



**Answer:** For the given state diagram, we formally express the DFA as $M = (Q, \Sigma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

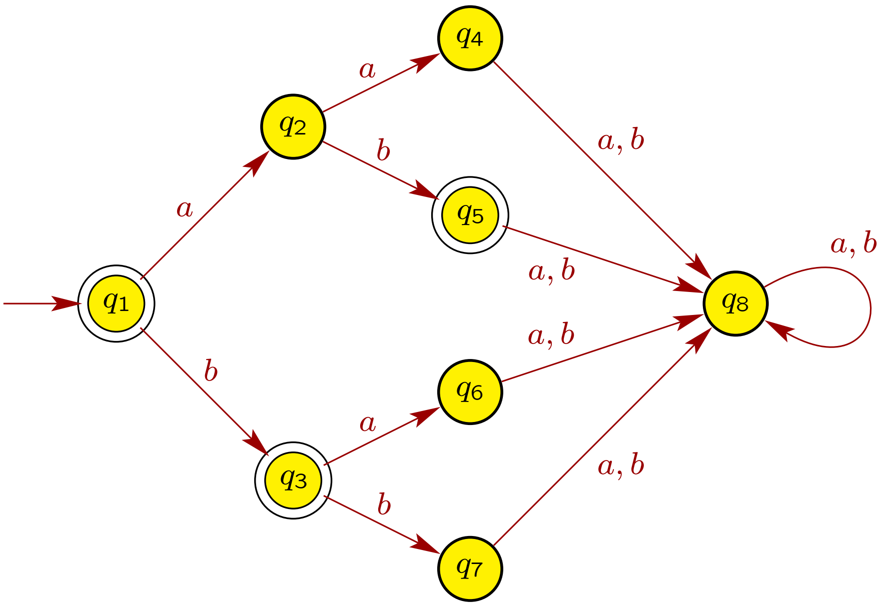
|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_1$ | $q_3$ |
| $q_3$ | $q_1$ | $q_3$ |

- $q_1$ is the start state
- $F = \{q_1, q_3\}$ is the set of accept states.

2. For each of the following languages over the alphabet $\Sigma = \{a,\ b\}$, give a DFA that recognizes the language. Give both a state diagram and 5-tuple specification for each DFA. (Each regular language has infinitely many correct DFAs, but you only need to give one.)

(a) $A = \{\varepsilon,\ b,\ ab\}$.

**Answer:** A state diagram of one DFA that recognizes the language $A$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_8\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_3$ |
| $q_2$ | $q_4$ | $q_5$ |
| $q_3$ | $q_6$ | $q_7$ |
| $q_4$ | $q_8$ | $q_8$ |
| $q_5$ | $q_8$ | $q_8$ |
| $q_6$ | $q_8$ | $q_8$ |
| $q_7$ | $q_8$ | $q_8$ |
| $q_8$ | $q_8$ | $q_8$ |

- $q_1$ is the start state
- $F = \{q_1, q_3, q_5\}$ is the set of accept states.

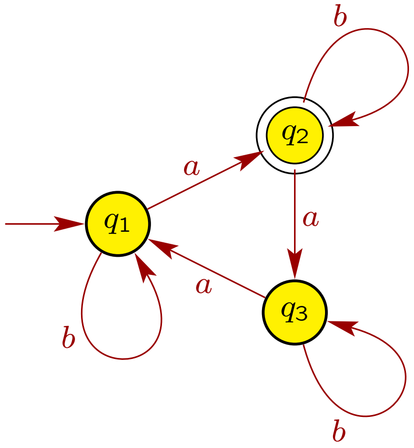
There are simpler DFAs that recognize this language. Can you come up with one with only 4 states?

(b) For any string $w \in \Sigma^*$, let $n_a(w)$ denote the number of $a$'s in $w$. For example, $n_a(abaaba) = 4$. Define the language

$$B = \{\, w \in \Sigma^* \mid n_a(w) \bmod 3 = 1 \,\},$$

i.e., $w \in B$ if and only if the number of $a$'s in $w$ is $3k + 1$ for some $k \geq 0$. Also, recall that for two integers $a$ and $b$, $a \bmod b$ returns the remainder after dividing $a$ by $b$. For example, $8 \bmod 3 = 2$ and $12 \bmod 3 = 0$.

**Answer:** A state diagram of one DFA that recognizes the language $B$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

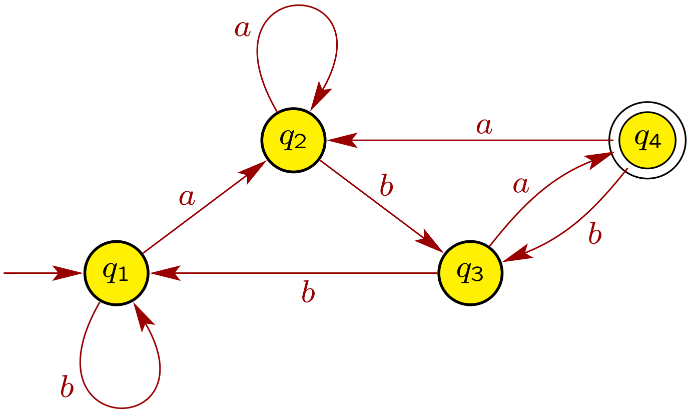
- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_1$ |
| $q_2$ | $q_3$ | $q_2$ |
| $q_3$ | $q_1$ | $q_3$ |

- $q_1$ is the start state
- $F = \{q_2\}$ is the set of accept states.

(c) $C = \{\, w \in \Sigma^* \mid w = saba$ for some string $s \in \Sigma^* \,\}$, i.e., $C$ is the language of strings in $\Sigma^*$ that end in $aba$.

**Answer:** A state diagram of one DFA that recognizes the language $C$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

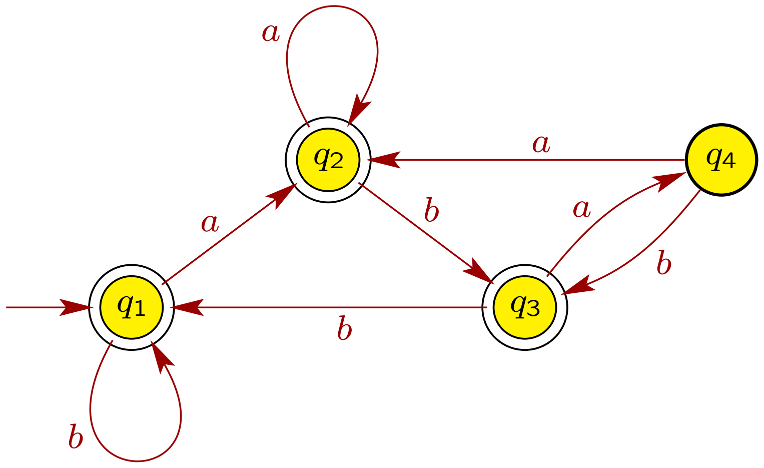
- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

| | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_1$ |
| $q_2$ | $q_2$ | $q_3$ |
| $q_3$ | $q_4$ | $q_1$ |
| $q_4$ | $q_2$ | $q_3$ |

- $q_1$ is the start state
- $F = \{q_4\}$ is the set of accept states.

(d) $D = \overline{C}$, where $C$ is the language in the previous part; i.e., $D$ is the language of strings in $\Sigma^*$ that do not end in $aba$.

**Answer:** Because $D = \overline{C}$, the complement of $C$, we can convert the DFA for $C$ into a DFA for $D$ by swapping the accept and non-accept states:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

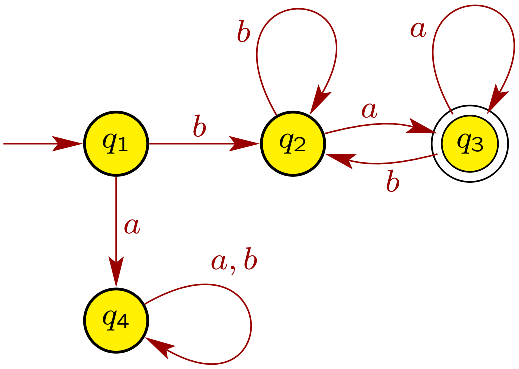
- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

| | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_1$ |
| $q_2$ | $q_2$ | $q_3$ |
| $q_3$ | $q_4$ | $q_1$ |
| $q_4$ | $q_2$ | $q_3$ |

- $q_1$ is the start state
- $F = \{q_1, q_2, q_3\}$ is the set of accept states.

(e) $E = \{ w \in \Sigma^* \mid w \text{ begins with } b \text{ and ends with } a \}$.

**Answer:** A state diagram of one DFA that recognizes the language $E$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

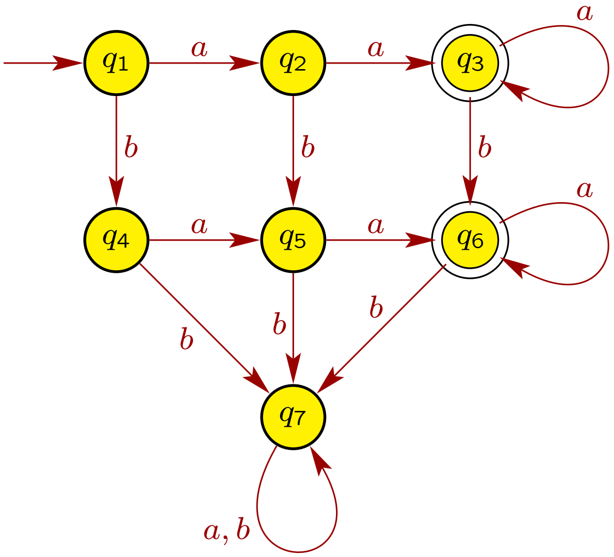
- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_4$ | $q_2$ |
| $q_2$ | $q_3$ | $q_2$ |
| $q_3$ | $q_3$ | $q_2$ |
| $q_4$ | $q_4$ | $q_4$ |

- $q_1$ is the start state
- $F = \{q_3\}$ is the set of accept states.

(f) For any string $w \in \Sigma^*$, let $n_b(w)$ denote the number of $b$'s in $w$. Define the language $F = \{ w \in \Sigma^* \mid n_a(w) \geq 2, n_b(w) \leq 1 \}$.

**Answer:** A state diagram of one DFA that recognizes the language $F$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

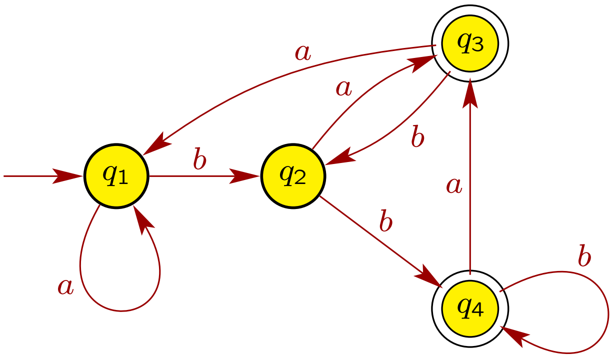
- $Q = \{q_1, q_2, \ldots, q_7\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_4$ |
| $q_2$ | $q_3$ | $q_5$ |
| $q_3$ | $q_3$ | $q_6$ |
| $q_4$ | $q_5$ | $q_7$ |
| $q_5$ | $q_6$ | $q_7$ |
| $q_6$ | $q_6$ | $q_7$ |
| $q_7$ | $q_7$ | $q_7$ |

- $q_1$ is the start state
- $F = \{q_3, q_6\}$ is the set of accept states.

(g) $G = \{\, w \in \Sigma^* \mid |w| \geq 2,$ second-to-last symbol of $w$ is $b \,\}$. If string $w = w_1 w_2 \cdots w_n$ where each $w_i \in \Sigma$, then the second-to-last symbol of $w$ is $w_{n-1}$.

**Answer:** A state diagram of one DFA that recognizes the language $G$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- transition function $\delta$ is given by

|  | $a$ | $b$ |
|---|---|---|
| $q_1$ | $q_1$ | $q_2$ |
| $q_2$ | $q_3$ | $q_4$ |
| $q_3$ | $q_1$ | $q_2$ |
| $q_4$ | $q_3$ | $q_4$ |

- $q_1$ is the start state
- $F = \{q_3, q_4\}$ is the set of accept states.

3. Show that, if $M$ is a DFA that recognizes language $B$, swapping the accept and non-accept states in $M$ yields a new DFA that recognizes $\overline{B}$, the complement of $B$. Conclude that the class of regular languages is closed under complement.

**Answer:**
Suppose language $B$ over alphabet $\Sigma$ has a DFA

$$M = (\,Q,\ \Sigma,\ \delta,\ q_1,\ F\,).$$

Then, a DFA for the complementary language $\overline{B}$ is

$$\overline{M} = (\,Q,\ \Sigma,\ \delta,\ q_1,\ Q - F\,).$$

The reason why $\overline{M}$ recognizes $\overline{B}$ is as follows. First note that $M$ and $\overline{M}$ have the same transition function $\delta$. Thus, since $M$ is deterministic, $\overline{M}$ is also deterministic. Now consider any string $w \in \Sigma^*$. Running $M$ on input string $w$ will result in $M$ ending in some state $r \in Q$. Since $M$ is deterministic, there is only one possible state that $M$ can end in on input $w$. If we run $\overline{M}$ on the same input $w$, then $\overline{M}$ will end in the same state $r$ since $M$ and $\overline{M}$ have the same transition function. Also, since $\overline{M}$ is deterministic, there is only one possible ending state that $\overline{M}$ can be in on input $w$.

Now suppose that $w \in B$. Then $M$ will accept $w$, which means that the ending state $r \in F$, i.e., $r$ is an accept state of $M$. But then $r \notin Q - F$, so $\overline{M}$ does not accept $w$ since $\overline{M}$ has $Q - F$ as its set of accept states. Similarly, suppose that $w \notin B$. Then $M$ will not accept $w$, which means that the ending state $r \notin F$. But then $r \in Q - F$, so $\overline{M}$ accepts $w$. Therefore, $\overline{M}$ accepts string $w$ if and only $M$ does not accept string $w$, so $\overline{M}$ recognizes language $\overline{B}$. Hence, the class of regular languages is closed under complement.

4. We say that a DFA $M$ for a language $A$ is *minimal* if there does not exist another DFA $M'$ for $A$ such that $M'$ has strictly fewer states than $M$. Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ is a minimal DFA for $A$. Using $M$, we construct a DFA $\overline{M}$ for the complement $\overline{A}$ as $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$. Prove that $\overline{M}$ is a minimal DFA for $\overline{A}$.

   **Answer:**
   We prove this by contradiction. Suppose that $\overline{M}$ is not a minimal DFA for $\overline{A}$. Then there exists another DFA $D$ for $\overline{A}$ such that $D$ has strictly fewer states than $\overline{M}$. Now create another DFA $D'$ by swapping the accepting and non-accepting states of $D$. Then $D'$ recognizes the complement of $\overline{A}$. But the complement of $\overline{A}$ is just $A$, so $D'$ recognizes $A$. Note that $D'$ has the same number of states as $D$, and $\overline{M}$ has the same number of states as $M$. Thus, since we assumed that $D$ has strictly fewer states than $\overline{M}$, then $D'$ has strictly fewer states than $M$. But since $D'$ recognizes $A$, this contradicts our assumption that $M$ is a minimal DFA for $A$. Therefore, $\overline{M}$ is a minimal DFA for $\overline{A}$.

5. Give a proof that the class of regular languages is closed under intersection.

   **Answer:**
   **Basic Idea:** Recall that Theorem 1.25 establishes that the class of regular languages is closed under union. The approach that we will use to show that the class of regular languages is closed under intersection is to modify the proof of Theorem 1.25. Specifically, Theorem 1.25 establishes that if $A_1$ is regular and $A_2$ is regular, then their union $A_1 \cup A_2$ is regular. The proof of Theorem 1.25 builds a DFA $M_3'$ for $A_1 \cup A_2$ by simultaneously running a DFA $M_1$ for $A_1$ and a DFA $M_2$ for $A_2$, where the union DFA $M_3'$ accepts if and only if $M_1$ accepts or $M_2$ accepts (or both accept). To instead build a DFA for the intersection $A_1 \cap A_2$, we can build a DFA $M_3$ by running $M_1$ and $M_2$ simultaneously, with the intersection DFA $M_3$ accepting if and only if both $M_1$ and $M_2$ accept. Below are the details.

   **Proof:** Suppose $A_1$ and $A_2$ are defined over the same alphabet $\Sigma$. Suppose DFA $M_1$ recognizes $A_1$, where $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Suppose DFA $M_2$ recognizes $A_2$, where $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$. Define DFA $M_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$ for $A_1 \cap A_2$ as follows:

   - Set of states of $M_3$ is

     $$Q_3 = Q_1 \times Q_2 = \{ \, (x, y) \mid x \in Q_1, \, y \in Q_2 \, \}.$$

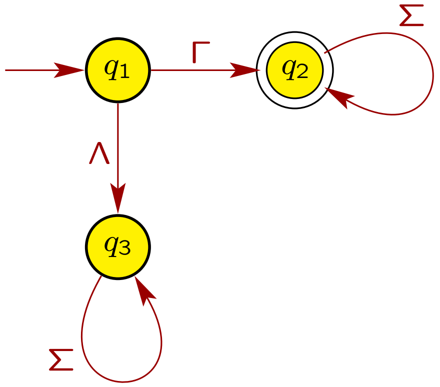
- The alphabet of $M_3$ is $\Sigma$.
- $M_3$ has transition function $\delta_3 : Q_3 \times \Sigma \to Q_3$ such that for $x \in Q_1$, $y \in Q_2$, and $\ell \in \Sigma$,
$$\delta_3(\,(x,y),\,\ell\,) \;=\; (\,\delta_1(x,\ell),\; \delta_2(y,\ell)\,)\,.$$
- The initial state of $M_3$ is $s_3 = (q_1, q_2) \in Q_3$.
- The set of accept states of $M_3$ is
$$F_3 \;=\; \{\,(x,y) \in Q_1 \times Q_2 \mid x \in F_1 \text{ and } y \in F_2\,\} = F_1 \times F_2.$$

Since $Q_3 = Q_1 \times Q_2$, the number of states in the new DFA $M_3$ is $|Q_3| = |Q_1| \cdot |Q_2|$. Thus, $|Q_3| < \infty$ since $|Q_1| < \infty$ and $|Q_2| < \infty$.

6. Let $\Sigma = \{\,\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, \mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots, \mathtt{9}\,\}$ be the alphabet consisting of lower-case Roman letters and Arabic numerals. Consider the language
$$L = \{\,w \in \Sigma^* \mid w \text{ begins with a lower-case Roman letter}\,\}.$$

(a) Give a DFA for $L$. For your DFA, give both a state diagram and 5-tuple for it.

**Answer:** Define $\Gamma = \{\,\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}\,\}$ as the set of lower-case Roman letters, and $\Lambda = \{\,\mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots, \mathtt{9}\,\}$ as the set of Arabic numerals, so $\Sigma = \Gamma \cup \Lambda$ with $\Gamma \cap \Lambda = \emptyset$. The state diagram of one DFA that recognizes $L$ is below:



We formally express the DFA as a 5-tuple $(Q, \Sigma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{\,\mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, \mathtt{0}, \mathtt{1}, \mathtt{2}, \ldots, \mathtt{9}\,\}$
- transition function $\delta$ is given by

| | $\Gamma$ | $\Lambda$ |
|---|---|---|
| $q_1$ | $q_2$ | $q_3$ |
| $q_2$ | $q_2$ | $q_2$ |
| $q_3$ | $q_3$ | $q_3$ |

- $q_1$ is the start state

- $F = \{q_2\}$ is the set of accept states.

(b) Let $J$ be the set of valid variable names in the Java programming language. Is $L \subseteq J$? Is $J \subseteq L$? Explain your answers.

**Answer:** Note that the string `if` $\in L$ but `if` $\notin J$ (because `if` is a reserved keyword), so $L \not\subseteq J$. Also, the string `AB` $\in J$ but `AB` $\notin L$, so $J \not\subseteq L$.