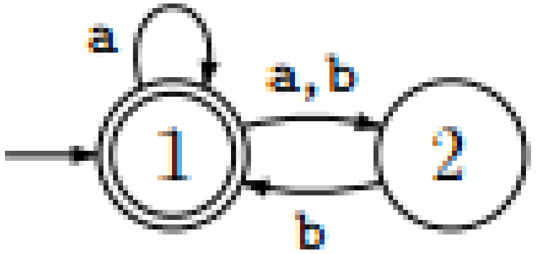
Recitation 11

Xiang Xiao

Q1

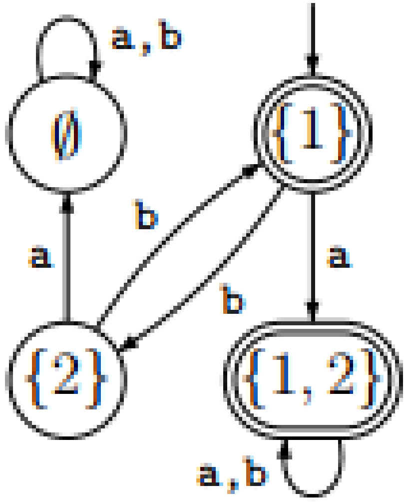
- Use the construction given in Theorem 1.39 to convert the NFA to equivalent DFA.



NFA: $N = (Q, \Sigma, \delta, q_0, F)$

DFA: $M = (Q', \Sigma', \delta', q_0', F')$

1. $Q' = P(Q)$. (subset of states of N)
2. For $R \in Q'$ and $a \in \Sigma$, $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$.
3. $q_0' = \{q_0\}$.
4. $F' = \{R \in Q' \mid$ R contains an accept state of N$\}$.



Q1 Answer

$$Q' = \mathcal{P}(Q)$$

$$Q' = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$$

$$\delta'(\emptyset, a) = \emptyset$$

$$\delta'(\emptyset, b) = \emptyset$$

$$\begin{aligned}\delta'(\{1\}, a) &= \delta(1, a) \\ &= \{1, 2\}\end{aligned}$$

$$\begin{aligned}\delta'(\{1\}, b) &= \delta(1, b) \\ &= \{2\}\end{aligned}$$

$$\begin{aligned}\delta'(\{2\}, a) &= \delta(2, a) \\ &= \emptyset\end{aligned}$$

$$\begin{aligned}\delta'(\{2\}, b) &= \delta(2, b) \\ &= \{1\}\end{aligned}$$

$$\begin{aligned}\delta'(\{1,2\}, a) &= \delta(1, a) \cup \delta(2, a) \\ &= \{1, 2\} \cup \emptyset \\ &= \{1, 2\}\end{aligned}$$

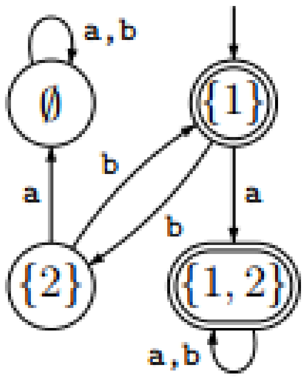
$$\begin{aligned}\delta'(\{1,2\},b) &= \delta(1,b) \cup \delta(2,b) \\ &= \{2\} \cup \{1\} \\ &= \{1,2\}\end{aligned}$$

$${q_0}' = \{q_0\}$$

$${q_0}' = \{1\}$$

$$F' = \{R \in Q' |\ R \text{ contains an accept state of } N\}$$

$$F' = \{\{1\}, \{1,2\}\}$$

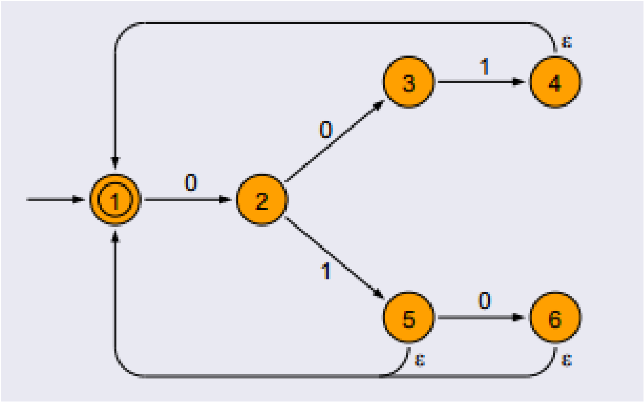


Q2

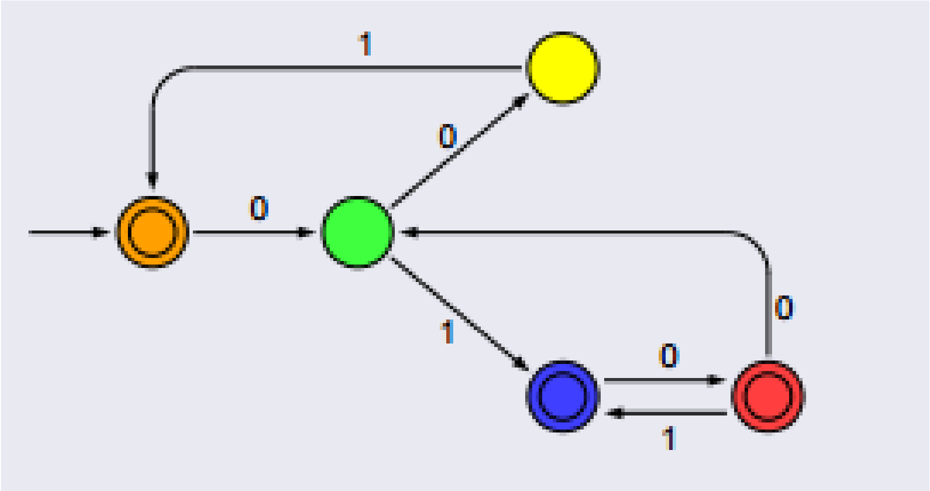
- Give an NFA that recognizing the language $(01 \cup 001 \cup 010)^*$
- Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

Q2 Answer

a.



b.



Q3

- Give regular expressions of the following language.
  - {w | w starts with 0 and has odd length, or starts with 1 and has even length}

    $$(0\cup 1\Sigma)(\Sigma\Sigma)^*$$

  - {w | w doesn’t contain the substring 110}

    $$(0\cup(10)^*)^*1^*$$

  - {w | w is any string except 11 and 111}

    $$\Sigma^*0\Sigma^*\cup 1111\Sigma^*\cup 1\cup \varepsilon$$

  - {w | w contains an even number of 0s, or contains exactly two 1s}

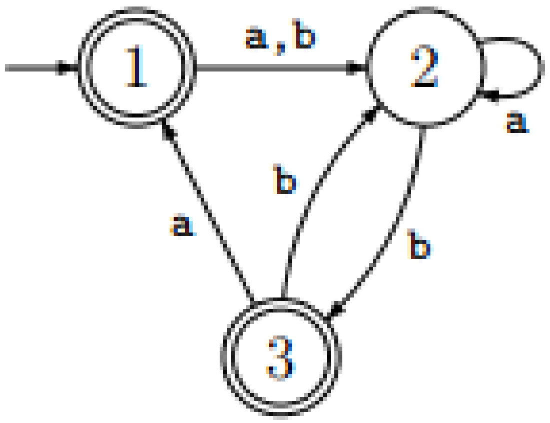
    $$(1^*01^*01^*)^*\cup 0^*10^*10^*$$

Q4

- Convert the following finite automata to regular expressions.

- Transfer DFA to GNFA
  - Add new start state and final state. Make original final states non-final.
  - Perform union on edge with multiple labels.
  - Select a state (not start or accept state) and remove it from the machine. Make sure the same language still recognized.
  - Perform union on edge with multiple labels.
  - Select a state (not start or accept state) and remove it from the machine.

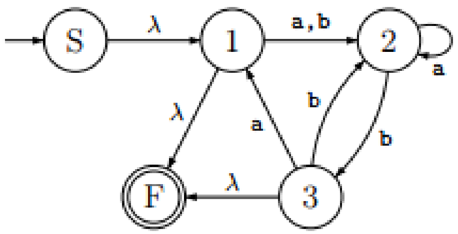
  ……..



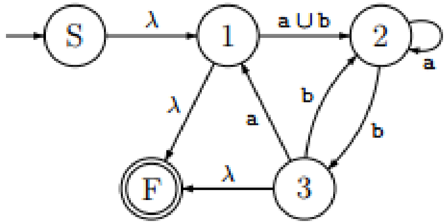
Answer: $\varepsilon \cup ((a \cup b)a^* b)((a(a \cup b)) \cup b)a^* b)^* (a \cup \varepsilon))$

Q4 Answer

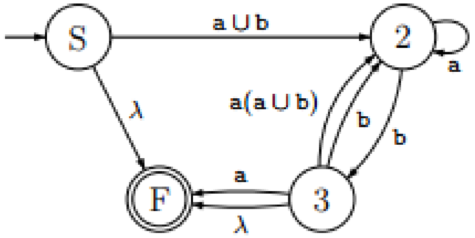
Add new start state and final state. Make original final states non-final.



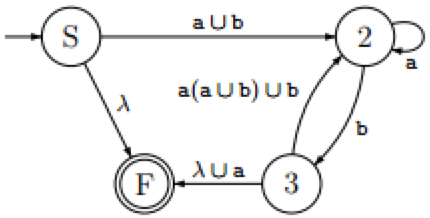
Perform union on edge from state 1 to state 2.



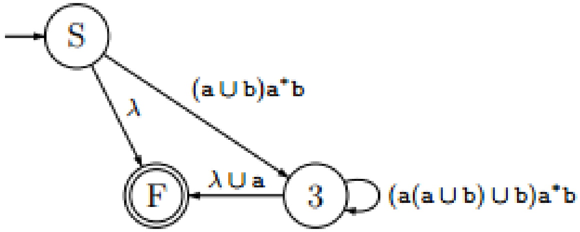
Eliminate state 1.



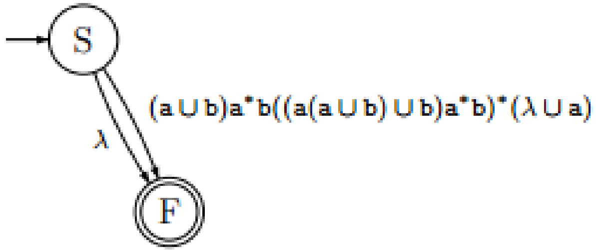
Perform unions on edges from state 3 to state 2 and from state 3 to the final state.



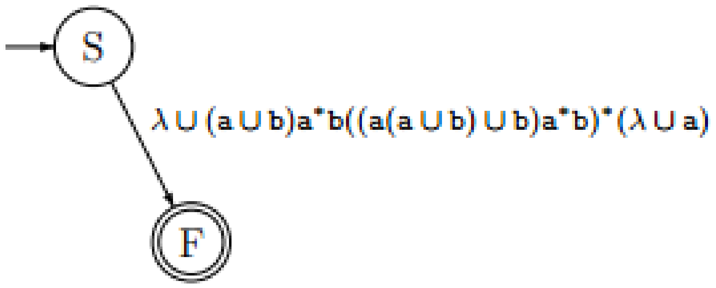
Eliminate state 2.



No unions necessary, so eliminate state 3.



Perform union on remaining edges.



Answer: $\varepsilon \cup ((a \cup b)a^*b)((a(a \cup b)) \cup b)a^*b)^*(a \cup \varepsilon))$

Q5

- Using the pumping Lemma to show the following language are not regular.

  A2 = {www | w ∈ {a, b}*}

Proof:

Assume to the contrary that A2 is regular. Let p be the pumping length given by the pumping lemma. Let s be the string $a^p b a^p b a^p b$.

Because s is a member of A2 and s has length more than p, the pumping lemma guarantees that s can be split into three pieces, s = xyz, satisfying the three conditions of the lemma.
However, condition 3 implies that y must consist only of a*s*, so xyyz ∉ A2 and one of the first two conditions is violated.
Therefore, A2 is non-regular.

Q6

- Let B_n = { a^k | k is a multiple of n}. Show that for each n >= 1, the language B_n is regular.

For each n>=1, we built a DFA with the n states q_0, q_1, …, q_{n-1} to count the number of consecutive a's modulo n read so far. For each character a that is input, the counter increments by 1 and jumps to the next state in M. It accept the string if and only if the machine stops at q_0. That means the length of the string consists of all a's and its length is a multiple of n.

More formally, the set of states of M is Q = {q_0, q_1, …, q_{n-1}}. The state q_0 is the start state and the only accept state. Define the transition function as $\delta(q_i, a) = q_j$ where j = (i + 1) mod n.

Q7

- Let Cn = {x | x is a binary number that is multiple of n}. Show that for each n >= 1, the language Cn is regular.

By simulating binary division, we create a DFA M with n states that recognizes Cn. M has n states which keep track of the n possible remainders of the division process. The start state is the only accept state and corresponds to remainder 0.

The input string is fed into M starting from the most significant bit. For each input bit, M doubles the remainder that its current state records, and then adds the input bits. Its new state is the sum modulo n. We double the remainder because that corresponds to the left shift of the computed remainder in the long division algorithm. If an input string ends at the accept state (corresponding to remainder 0), the binary number has no remainder on division by n and is therefore a member of Cn.

The formal definition of M is $(\{q_0, q_1, \ldots, q_{n-1}\}, \{0,1\}, \delta, q_0, \{q_0\})$. For each $q_i \in Q$ and $b \in \{0,1\}$, define $\delta(q_i, b) = q_j$, where $j = (2i + b) \bmod n$.