Introduction to the Theory of Computation 2025 — Midterm 1

Solutions

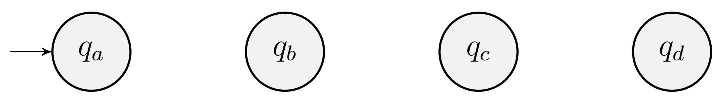
Problem 1 (20 pts). We have

$$\Sigma = \{0, 1\}.$$

(a) (5 pts) Let us consider the language

$$L_B = 0^*1^*0^*$$

Please give a DFA to recognize the language L_B with the following draft



That is, you should add links to connect nodes and may put some nodes as final states.

(b) (5 pts) On the diagram you draw in (a), please simulate the strings

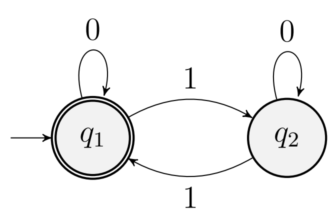
00010 and **00101** .

Note that you need to specify the strings are accepted or rejected.

(c) (10 pts) Now, we have another language

$$L_A = \left\{ \boldsymbol{w} \mid \sum_i w_i \text{ is even} \right\}$$

that can be recognized by the DFA



If we want a DFA that can recognize the language $L_A \cap L_B$, please utilize the following method if

$$M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$$

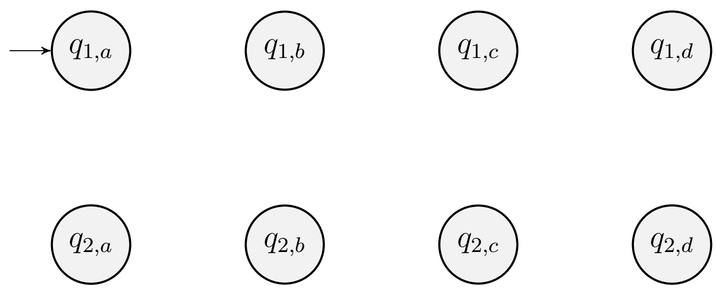
and

$$M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$$

recognize L_A and L_B respectively, the formal definition of M for $L_A \cap L_B$ is

$$\begin{aligned} Q &= \{(r_A, r_B) \mid r_A \in Q_A, r_B \in Q_B\} \\ &\Sigma \text{ is the same one} \\ \delta((r_A, r_B), a) &= (\delta_A(r_A, a), \delta_B(r_B, a)) \\ q_0 &= (q_A, q_B) \\ F &= \{(r_A, r_B) \mid r_A \in F_A \textbf{ and } r_B \in F_B\}. \end{aligned}$$

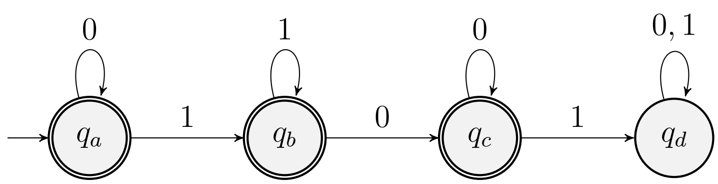
to generate the DFA with the following draft



Note that you are not required to simplify the diagram. Please answer this problem with those eight specific nodes.

Solution.

(a) Please check the following diagram.



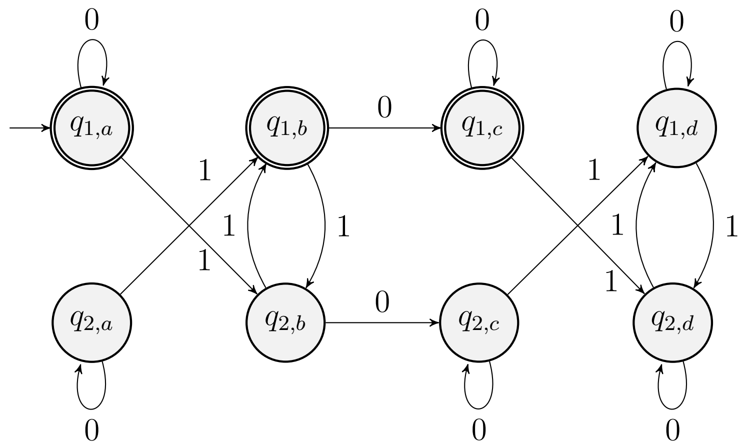
(b) For the string **00010**:

$$\rightarrow q_a \xrightarrow{0} q_a \xrightarrow{0} q_a \xrightarrow{0} q_a \xrightarrow{1} q_b \xrightarrow{0} q_c \rightarrow \text{accepted}$$

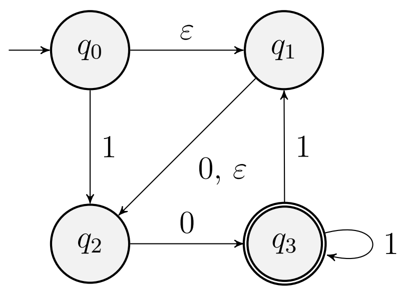
For the string **00101**:

$$\rightarrow q_a \xrightarrow{0} q_a \xrightarrow{0} q_a \xrightarrow{1} q_b \xrightarrow{0} q_c \xrightarrow{1} q_d \rightarrow \text{rejected}$$

(c) Please check the following diagram.



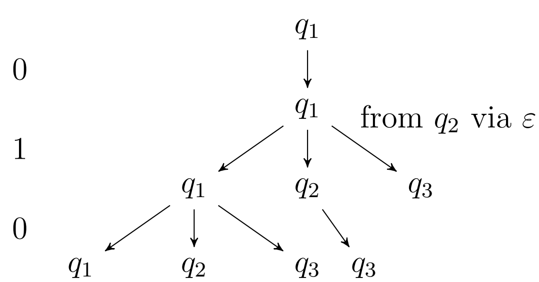
Problem 2 (40 pts). Let $\Sigma = \{0, 1\}, Q = \{q_0, q_1, q_2, q_3\}$. Consider the following NFA A.



(a) (5 pts) Please draw the computation of the NFA A on the input strings

10 and **0110**.

Note that you need to conclude whether they are **accepted** or **rejected**. A computation is like the following figure (copied from Figure 1.29 in the textbook, as an illustration and not related to the NFA in this subproblem.)



Please follow the rules in the textbook. Note that we handle the ε edge **immediately**, and you have to list all possible states that can be reached once processing each input character.

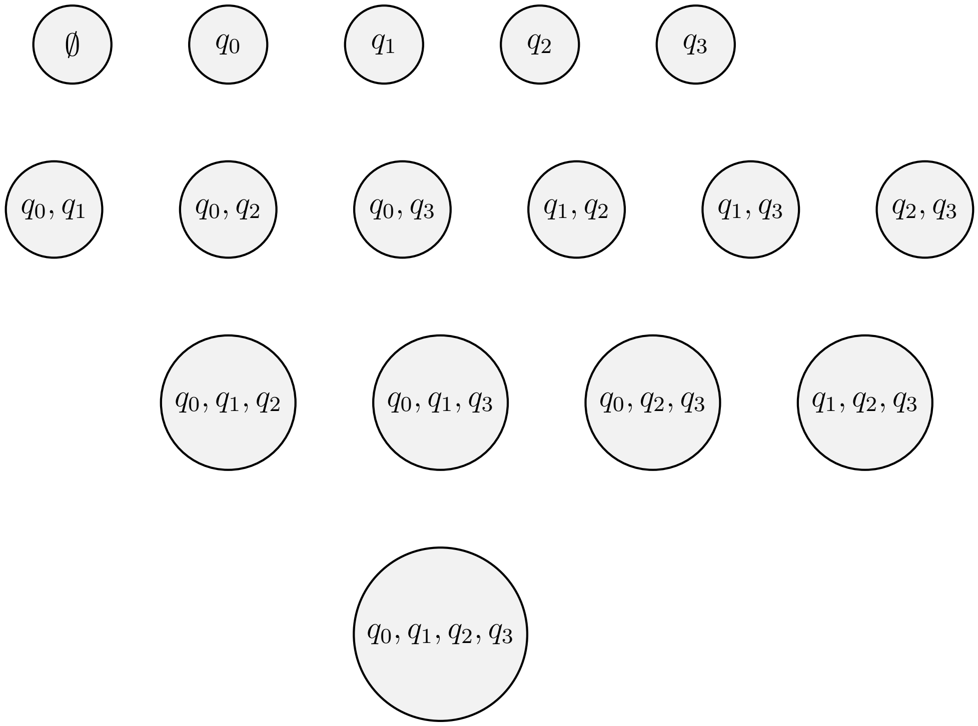
(b) (10 pts) Convert the NFA A to a DFA by the given procedure.

i. Each state is an element of the power set of Q (i.e., $P(Q)$).

ii. Started state: The state we can reach in the beginning.

iii. Accepted state: Any state including q_3 is an accepted state.

We have provided the following diagram illustrating all possible state combinations. Use this diagram for your conversion and show the result. Then remove useless states and show the resulting diagram.



(c) (10 pts) Convert the resulting DFA of the previous subproblem to a regular expression by constructing an equivalent GNFA using the following steps:

(i) Add a new started state q_s and an accepted state q_a to create the initial state diagram S_{initial}.

(ii) Remove the states in the following order (other states should already been removed in the previous subproblem):

$$\{q_0, q_1, q_2\}, \{q_2\}, \{q_3\}, \{q_2, q_3\}, \{q_1, q_2, q_3\}$$

during the conversion process to get the final state diagram S_{final}. (you can remove the $\emptyset$ state in any step or ignore it)

Please show all the details of the conversion process from S_{initial} to S_{final}.

Hint: Before Xiao-Ming started the conversion process, he peeked the answer sheet and realized that the regular expression should be:

$$10 \cup 00 \cup 0 \cup ((101 \cup 001 \cup 01)(001 \cup 01 \cup 1)^*(00 \cup \varepsilon \cup 0)).$$

(d) (10 pts) Convert the NFA A to a regular expression by constructing an equivalent GNFA using the following steps:

(i) Add a new started state q_s and an accepted state q_a to create the initial state diagram S_{initial}.

(ii) Remove the states $q_0, q_1, q_2, \ldots$ in **numerical order** during the conversion process to get the final state diagram S_{final}.

Please show all the details of the conversion process from S_{initial} to S_{final}.

Hint: Before Xiao-Ming started the conversion process, he peeked the answer sheet again and realized that the regular expression should be:

$$(1 \cup (0 \cup \varepsilon))0(1 \cup ((10 \cup 1 \cancel{\cup 11})0))^*.$$

Looks like Xiao-Ming accidentally added the string "11" due to his shortsightedness.. (however, it is still equivalent to the answer)

(e) (5 pts) After Xiao-Ming finished the computation for the previous two subproblems, he is wondering whether the regular expressions obtained from the previous two subproblems are equivalent. Please help him to **prove or disprove the equivalence** of the regular expressions.

Hint: Before Xiao-Ming started to prove the equivalence, TAs secretly told him that he should started by the following steps:

$$\text{From NFA: } (1 \cup 0 \cup \varepsilon)0(1 \cup 100 \cup 10 \cancel{\cup 110})^* \tag{i}$$
$$\text{From DFA: } 10 \cup 00 \cup 0 \cup ((101 \cup 001 \cup 01)(001 \cup 01 \cup 1)^*(00 \cup \varepsilon \cup 0)) \tag{ii}$$

For (i), we have

$$\begin{aligned}
\text{(i)} &= (10 \cup 00 \cup 0)(1 \cup 100 \cup 10 \cancel{\cup 1 \cdot 10})^* \\
&= (10 \cup 00 \cup 0)(1 \cup 100 \cup 10)^* \\
&= (10 \cup 00 \cup 0)(\varepsilon \cup (1 \cup 100 \cup 10)^+) \\
&= (10 \cup 00 \cup 0) \cup ((10 \cup 00 \cup 0)(1 \cup 100 \cup 10)^+) \\
&= 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)(1(00 \cup 0 \cup \varepsilon))^+).
\end{aligned}$$

Note that for convenience, we let R^+ be shorthand for RR^*. So $R^+ \cup \varepsilon = R^*$. (copied from Page 65 in the textbook.)

For (ii), with $(101 \cup 001 \cup 01) = (10 \cup 00 \cup 0)1$, we have

$$\text{(ii)} = 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)1(001 \cup 01 \cup 1)^*(00 \cup 0 \cup \varepsilon)).$$

That is, Xiao-Ming needs to check whether (i) and (ii) are equivalent.

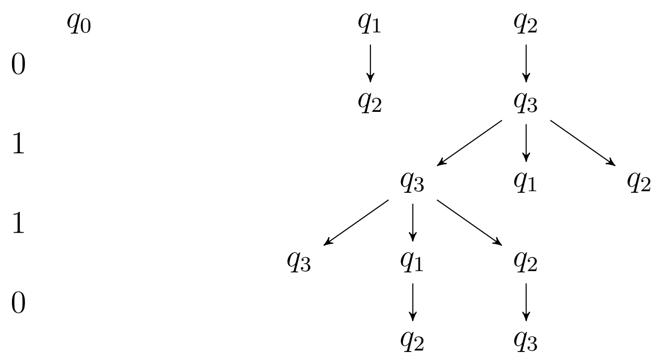
Solution.

(a) For the input string **10**:

$$\begin{array}{cccc}
 & q_0 & q_1 & q_2 \\
1 & \downarrow & & \\
 & q_2 & & \\
0 & \downarrow & & \\
 & q_3 & &
\end{array}$$

This NFA **accepts** the input string **10**.

For the input string **0110**:



This NFA **accepts** the input string **0110**.

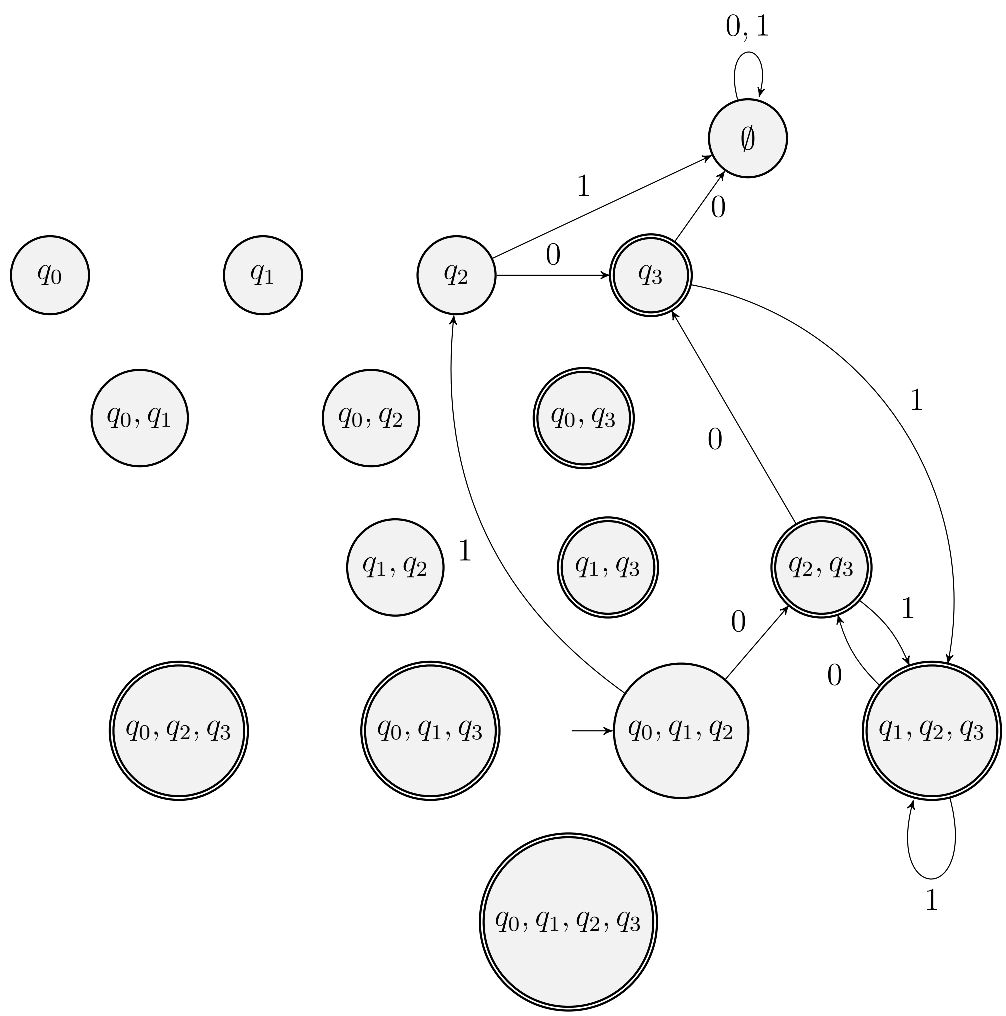
(b) Step 1: We have the combination of the states as

$$\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_3\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_0, q_3\}, \{q_1, q_2\}, \{q_1, q_3\}, \{q_2, q_3\},$$
$$\{q_0, q_1, q_2\}, \{q_0, q_1, q_3\}, \{q_0, q_2, q_3\}, \{q_1, q_2, q_3\}, \{q_0, q_1, q_2, q_3\}$$

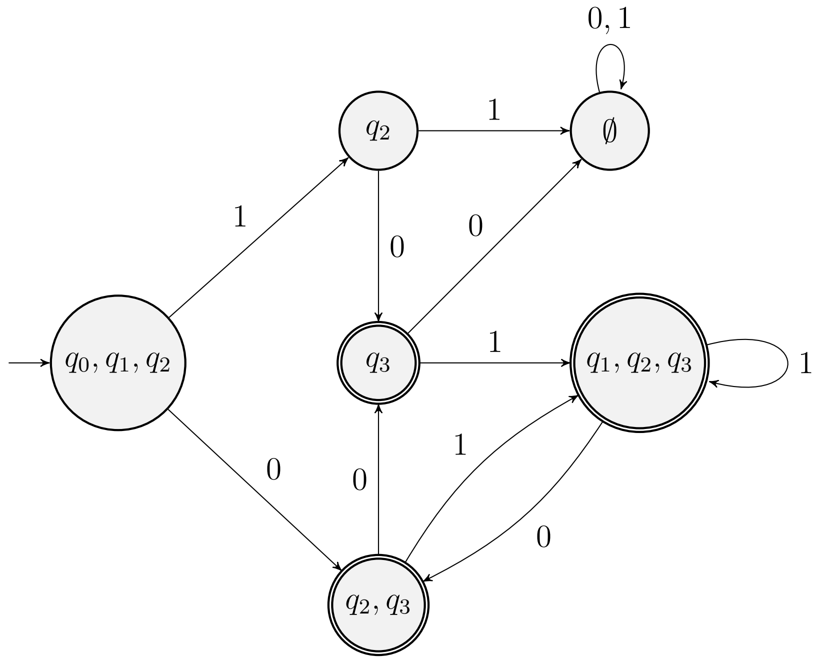
Step 2: We now follow the procedure outlined in our slides "chap1_NFA3.pdf":

i. Start state: $\{q_0, q_1, q_2\}$.

ii. Accept state: Any state that includes q_3 is considered an accept state.

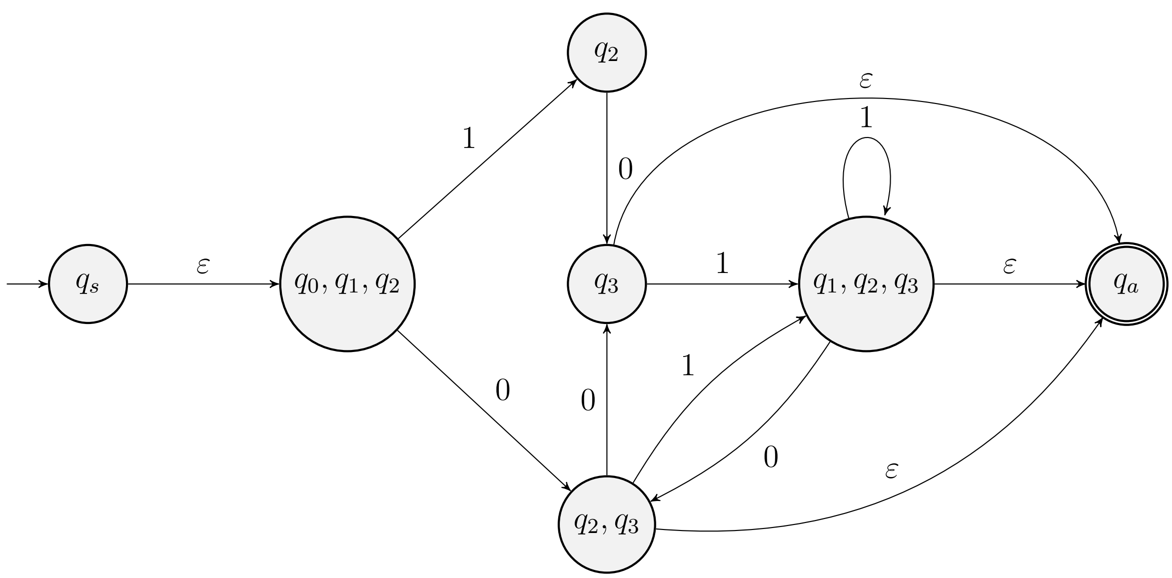


Step 3: After we remove the unused states, we have

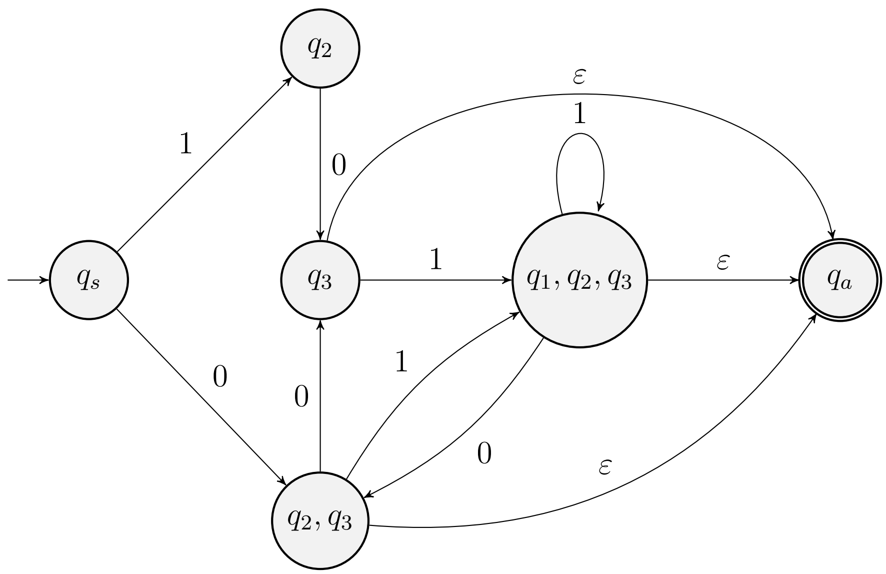


(c) We have the following steps:

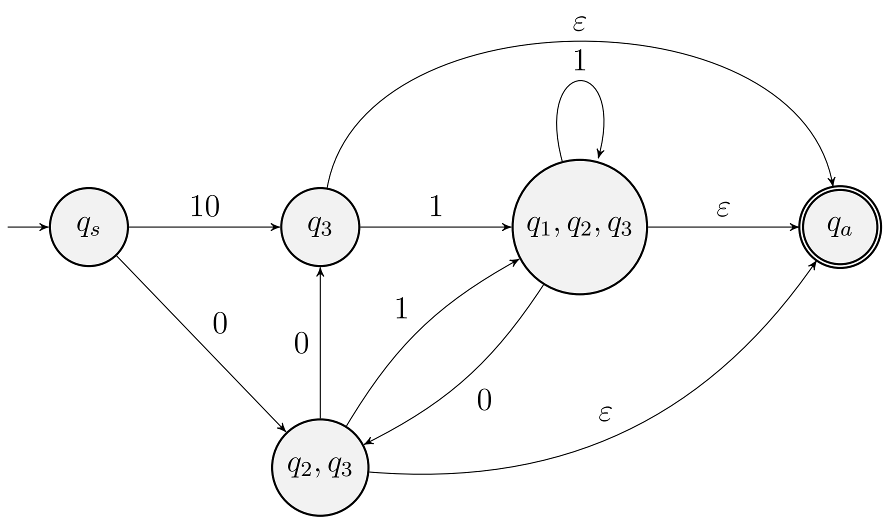
Step 1. Add new start q_s and accept q_a states.



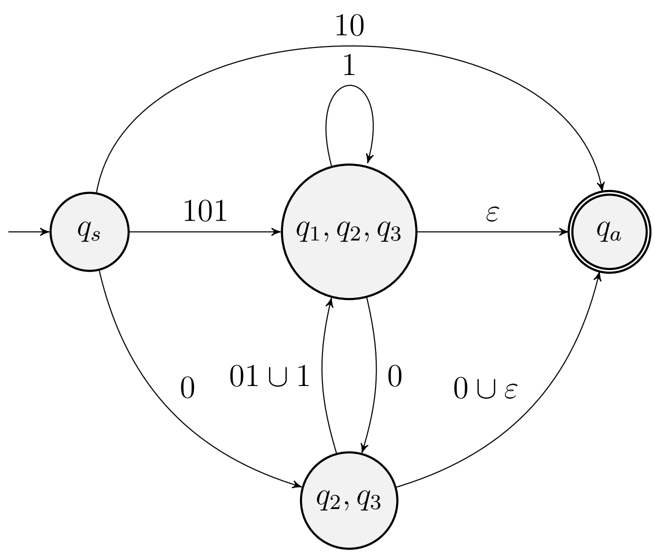
Step 2. Remove $\{q_0, q_1, q_2\}$.



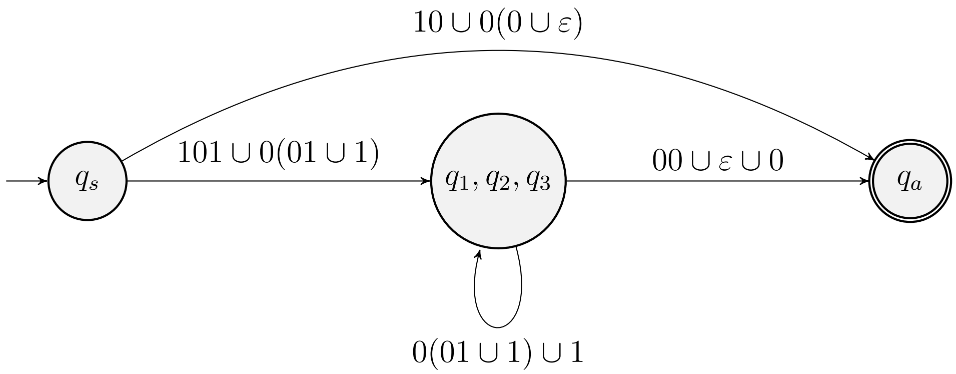
Step 3. Remove $\{q_2\}$.



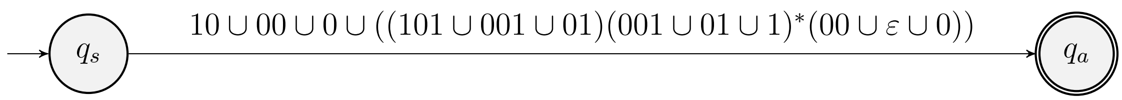
Step 4. Remove $\{q_3\}$.



Step 5. Remove $\{q_2, q_3\}$.

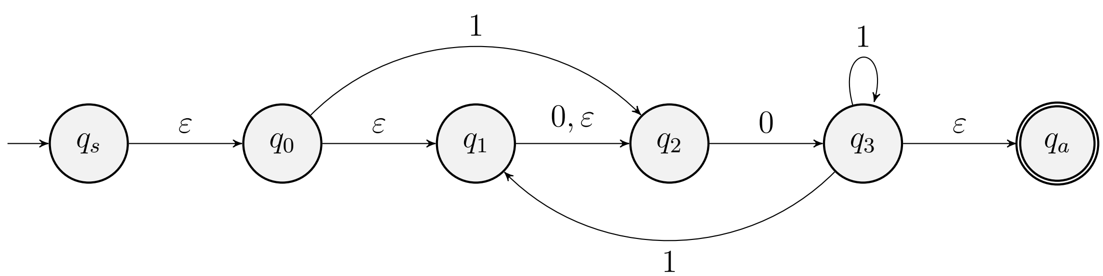


Step 6. Remove $\{q_1, q_2, q_3\}$.

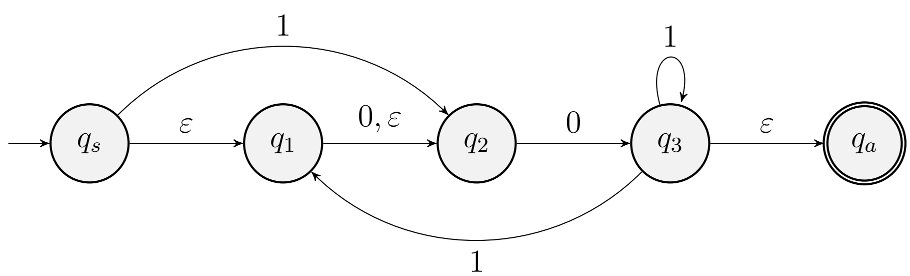


(d) We have the following steps:

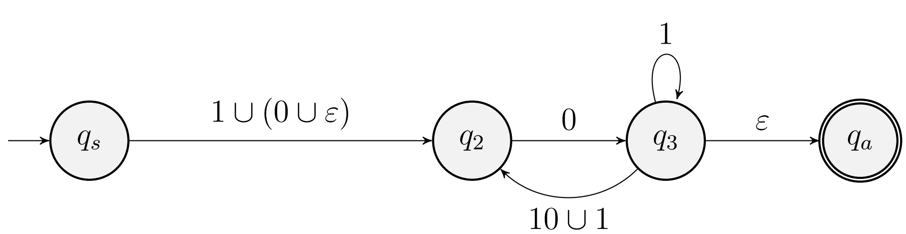
Step 1. Add new start q_s and accept q_a states.



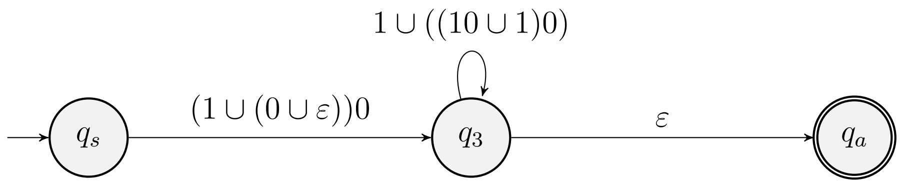
Step 2. Remove q_0.



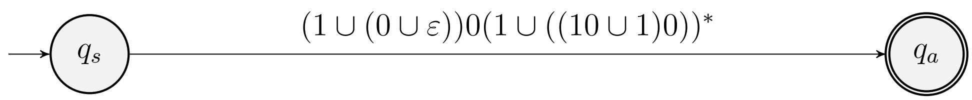
Step 3. Remove q_1.



Step 4. Remove q_2.



Step 5. Remove q_3.



(e) The regular expressions obtained from the previous two subproblems are equivalent.

$$\text{From NFA: } (1 \cup 0 \cup \varepsilon)0(1 \cup 100 \cup 10 \cancel{\cup 110})^* \tag{iii}$$
$$\text{From DFA: } 10 \cup 00 \cup 0 \cup ((101 \cup 001 \cup 01)(001 \cup 01 \cup 1)^*(00 \cup \varepsilon \cup 0)) \tag{iv}$$

For (iii), we have

$$\begin{aligned}
\text{(iii)} &= (10 \cup 00 \cup 0)(1 \cup 100 \cup 10 \cancel{\cup 1 \cdot 10})^* \\
&= (10 \cup 00 \cup 0)(1 \cup 100 \cup 10)^* \\
&= (10 \cup 00 \cup 0)(\varepsilon \cup (1 \cup 100 \cup 10)^+) \\
&= (10 \cup 00 \cup 0) \cup ((10 \cup 00 \cup 0)(1 \cup 100 \cup 10)^+) \\
&= 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)(1(00 \cup 0 \cup \varepsilon))^+).
\end{aligned}$$

For (iv), with $(101 \cup 001 \cup 01) = (10 \cup 00 \cup 0)1$, we have

$$\text{(iv)} = 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)1(001 \cup 01 \cup 1)^*(00 \cup 0 \cup \varepsilon)).$$

Let

$$u_N = (1(00 \cup 0 \cup \varepsilon))^+$$

and

$$u_D = 1(001 \cup 01 \cup 1)^*(00 \cup 0 \cup \varepsilon),$$

we have

$$\text{(iii)} = 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)u_N)$$

and

$$\text{(iv)} = 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)u_D).$$

Which means we need to check whether u_N and u_D are equivalent.

Since

$$(ab)^+ = (ab \cup (ab)^2 \cup (ab)^3 \cup \cdots) = a(\varepsilon \cup ba \cup (ba)^2 \cup \cdots)b = a(ba)^*b,$$

we have

$$u_N = 1((00 \cup 0 \cup \varepsilon)1)^*(00 \cup 0 \cup \varepsilon) = u_D.$$

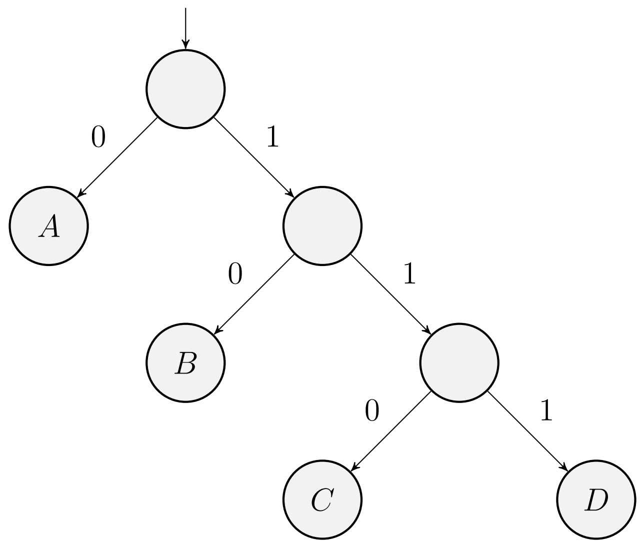
Thus,

$$\begin{aligned}
\text{(iii)} &= 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)u_D) \\
&= 10 \cup 00 \cup 0 \cup ((10 \cup 00 \cup 0)u_N) \\
&= \text{(iv)}.
\end{aligned}$$

Problem 3 (10 pts). In information theory, Huffman code is a useful coding to compress the bit string lengh. Now we consider to encode the following four characters

$$A, B, C, \text{ and } D.$$

with the corresponding coding tree



Therefore, the string

$$ABAACAAD$$

can be encoded into

$$\begin{array}{cccccccc} A & B & A & A & C & A & A & D \\ 0 & 10 & 0 & 0 & 110 & 0 & 0 & 111. \end{array}$$

Xiao-Ming encodes many character strings into bit strings with the above Huffman coding tree. Let

$$\Sigma = \{0, 1\}.$$

Please design a DFA with ≤ 3 nodes that

$$\begin{cases} \text{accepts a bit string if this string is Huffman encoded,} \\ \text{rejects a bit string if this string is not Huffman encoded,} \end{cases}$$

to help Xiao-Ming checking his results.

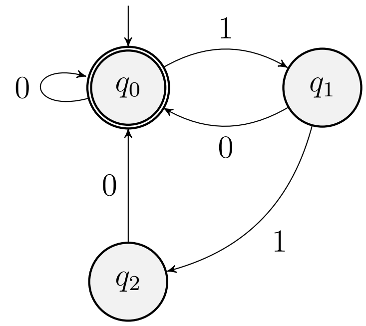
Solution.

By the coding tree, we know that each character is encoded into at most three bits, and we can further observe that

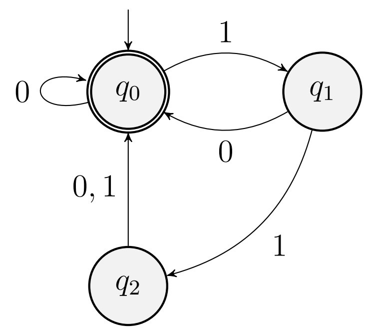
If the bit "0" occurs, this must be in the end of the encoded string from a character.

That is, when we read a 0 bit, then next bit must be the started bit of another encoded character. Thus, the transition of reading 0 should go back to the initial state.

With the above observation, let us set the node q_0, q_1, and q_2 as the status of reading zero, one, and two 1s, respectively. By handling the character A, B, C, we have the diagram



In the final, we handle the case "111," i.e., D, and it should also be sent back to the initial state. Hence, we have the following DFA adiagram.



Problem 4 (30 pts). Are the languages regular? Your solution should satisfy the following rules:

- If the language is regular, **please give a DFA** with no more than **4** states to recognize it. Note that if your DFA is recognizing an equivalent language to the language in the subproblem, you should **prove the equivalence** (to prove that language A is equivalent to language B, you must prove that $A \subseteq B$ and $B \subseteq A$).
- If the language is not regular, please use the pumping lemma to prove it is not regular. In using the pumping lemma, your s and i must be clearly given. You cannot just roughly say the existence of them.

You must give clear explanation instead of just giving the answer.

(a) (10 pts) Given the alphabet

$$\Sigma = \{0, 1\},$$

we consider the language

$$L_1 = \{0^p u 1^p \mid p \geq 1, u \in \Sigma^*\}.$$

(b) (10 pts) The language

$$L_2 = \{0^p 1^q \mid p, q \in \mathbb{N} \cup \{0\} \text{ and } p < q\}.$$

(c) (10 pts) The language

$$L_3 = \{0^q \mid q = 2^p, p \in \mathbb{N} \cup \{0\}\}.$$

Note that in this subproblem, Xiao-Ming has written part of the proof:

> L_3 is not regular. For any pumping length $p > 0$, we can take
>
> $$q = 2^k > 2p, \text{ for some } k > 0,$$
>
> such that
>
> $$s = 0^q \in L_3.$$

He hopes that **he can finish the proof by using an $i \neq 2$.** Please help him complete the proof.

Solution.

(a) L_1 is regular. Since

$$L_1 = \{0u1 \mid u \in \Sigma^*\} \cup \{0(0u1)1 \mid u \in \Sigma^*\} \cup \left\{0(0^2u1^2)1 \mid u \in \Sigma^*\right\} \cup \cdots,$$

we have

$$L_1 \subseteq \{0u1 \mid u \in \Sigma^*\} \cup \{0u1 \mid u \in \Sigma^*\} \cup \cdots = \{0u1 \mid u \in \Sigma^*\}$$

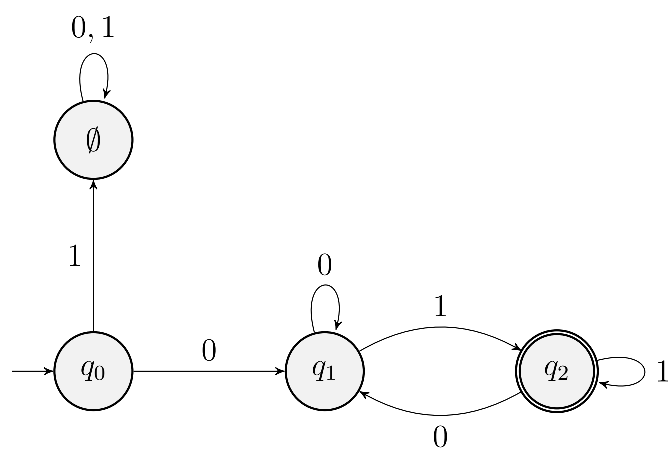
and

$$\{0u1 \mid u \in \Sigma^*\} \subseteq \{0u1 \mid u \in \Sigma^*\} \cup \left\{0^2u1^2 \mid u \in \Sigma^*\right\} \cup \left\{0^3u1^3 \mid u \in \Sigma^*\right\} \cup \cdots = L_1.$$

Thus,

$$L_1 = \{0u1 \mid u \in \Sigma^*\},$$

and we only need to check whether it starts with 0 and ends with 1.



(b) L_2 is not regular. For any pumping length $p > 0$, we can take

$$s = 0^p1^{p+1} \in L_2.$$

Let $s = xyz$ and

$$|xy| \leq p, |y| > 0,$$

for every x, y, z. That is, we have

$$y = 0^k, \text{ where } k \geq 1.$$

When we take $i = 2$,

$$xy^iz = 0^{p-k}0^{2k}1^{p+1} = 0^{p+k}1^{p+1}$$

is not in L_2 due to

$$p + k \geq p + 1.$$

By pumping lemma, we show that L_2 is not regular.

(c) L_3 is not regular. For any pumping length $p > 0$, we can take

$$q = 2^k > 2p, \text{ for some } k > 0, \tag{v}$$

such that

$$s = 0^q \in L_3.$$

Let $s = xyz$ and

$$|xy| \leq p, |y| > 0,$$

for every x, y, z, which implies

$$y = 0^a, 1 \leq a \leq p.$$

When we take $i = 0$, we have

$$0^{q-p} \leq xy^0z \leq 0^{q-1}. \tag{vi}$$

By (v), we have

$$q - p > 2^k - 2^{k-1} = 2^{k-1},$$

so (vi) becomes

$$0^{2^{(k-1)}} < 0^{q-p} \leq xy^0z \leq 0^{2^k-1},$$

which implies

$$xy^0z \notin L_3.$$

Another Solution L_3 is not regular. For any pumping length $p > 0$, we can take

$$q = 2^k > 2p, \text{ for some } k > 0,$$

such that

$$s = 0^q \in L_3.$$

Let $s = xyz$ and

$$|xy| \leq p, |y| > 0,$$

for every x, y, z, which implies

$$y = 0^a, 1 \leq a \leq p < 2^{k-1}.$$

When we take $i = 4$, we have

$$xy^4z = 0^{q+3a}.$$

if $q + 3a = 2^m$, for some $m > k$, we can imply

$$\begin{aligned} 3a &= 2^m - 2^k = 2^k(2^{m-k} - 1) \\ \Rightarrow a &= \frac{2^k(2^{m-k} - 1)}{3}. \end{aligned}$$

With the property

$$a < 2^{k-1},$$

we have that

$$\begin{aligned} &\frac{2^k(2^{m-k} - 1)}{3} < 2^{k-1} \\ \Rightarrow &2^k(2^{m-k} - 1) < 3 \cdot 2^{k-1} \\ \Rightarrow &2 \cdot (2^{m-k} - 1) < 3. \end{aligned}$$

This inequality only holds if $m = k + 1$. However, when $m = k + 1$, we have

$$3a = 2^{k+1} - 2^k = 2^k,$$

which is impossible if a is an integer. Therefore,

$$xy^4z = 0^{q+3a} \notin L_3.$$

Common mistakes: Some of students only claim that

$$3a = 2^m - 2^k = 2^k(2^{m-k} - 1)$$

cannot be hold, but lack of the detailed proof.