- **Please give details of your calculation. A direct answer without explanation is not counted.**
- **Your answers must be in English.**
- **Please carefully read problem statements.**
- **During the exam you are not allowed to borrow others' class notes.**
- **Try to work on easier questions first.**

Problem 1 (25%)

1. Consider the following language of having only one string:

   $$\{a\}$$

   Assume $\Sigma = \{a\}$, what is the DFA diagram with the smallest number of states for this language? Give the formal definition.

2. Prove why yours has "the smallest number of states" ?

3. Now consider two languages:

   $$A = \{a\}, \text{ with } \Sigma_A = \{a\},$$
   $$B = \{b\}, \text{ with } \Sigma_B = \{b\},$$

   We would like to prove that

   $$A \cup B$$

   is regular. Before introducing NFA, in our lecture (or in the textbook), we construct a DFA with states like (q, r), where q is from that of the first machine and r is from the other. However, in that construction, we assume that A and B have the same Σ.
   Please modify your DFA for A in 1., so that it handles the language:

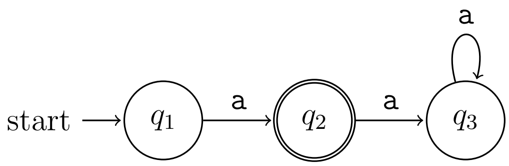
   $$\{a\}, \text{ with } \Sigma = \{a, b\}$$

4. After 3., we can use a similar setting for $\{b\}$, with $\Sigma = \{a, b\}$. Then use the procedure in our lecture to construct a DFA for $A \cup B$.

5. Give a DFA with the smallest number of states for $A \cup B$. Can you reduce the DFA in 4. to your DFA here?

Answer

1. D_A:



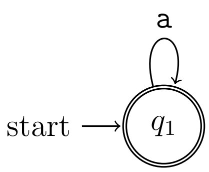
Formal definition: $M = (\{q_1, q_2, q_3\}, \{a\}, \delta, q_1, \{q_2\})$

δ is described as

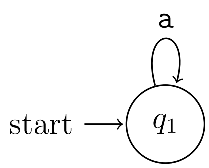
	a
q_1	q_2
q_2	q_3
q_3	q_3

2. Suppose that A can be recognized by DFA with number of states < 3 :

   i) If this number is 1, then it must be:

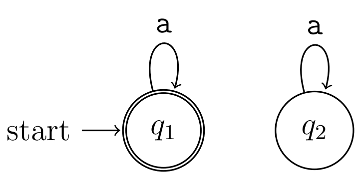


$\Rightarrow A = a^*$ or

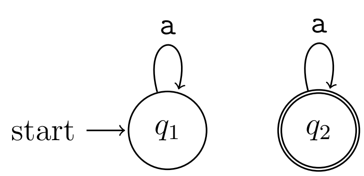


$\Rightarrow A = \phi$

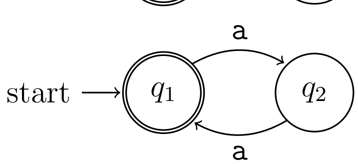
   ii) If this number is 2, then it must be one of the following situations:



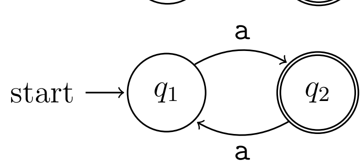
$\Rightarrow A = a^*$



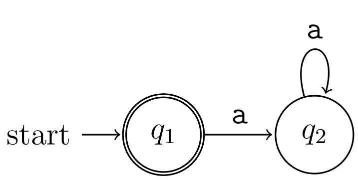
$\Rightarrow A = \phi$



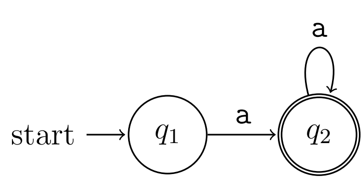
$\Rightarrow A = (aa)^*$



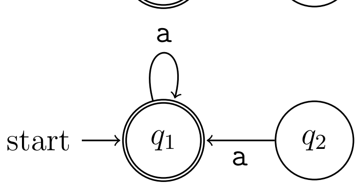
$\Rightarrow A = a(aa)^*$



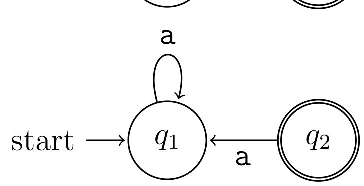
$\Rightarrow A = \phi$



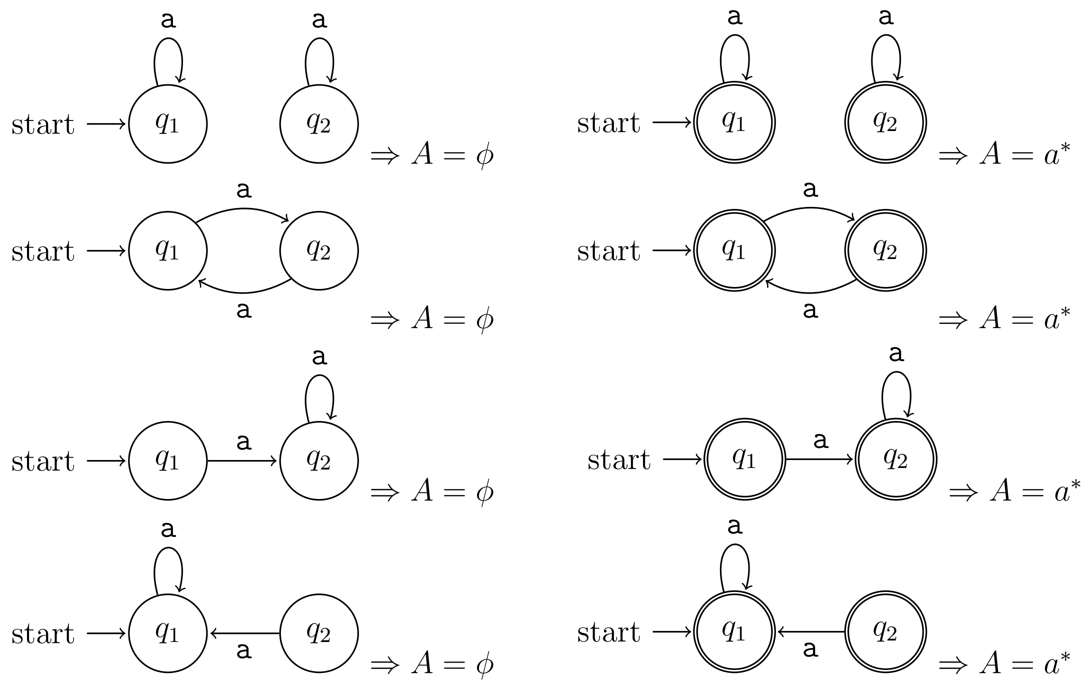
$\Rightarrow A = aa^*$



$\Rightarrow A = a^*$

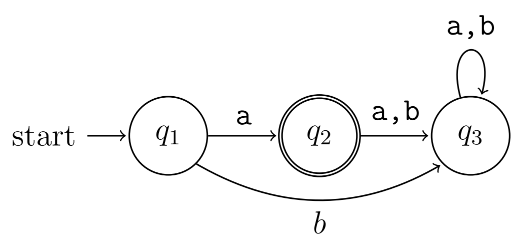


$\Rightarrow A = \phi$

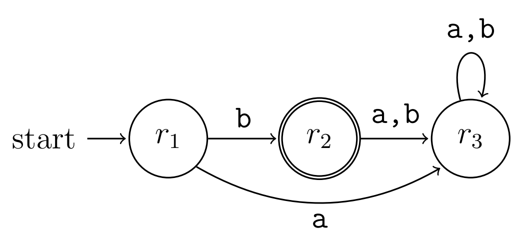


None of the above situations results in $A = \{a\}$, so the smallest number of states is 3.

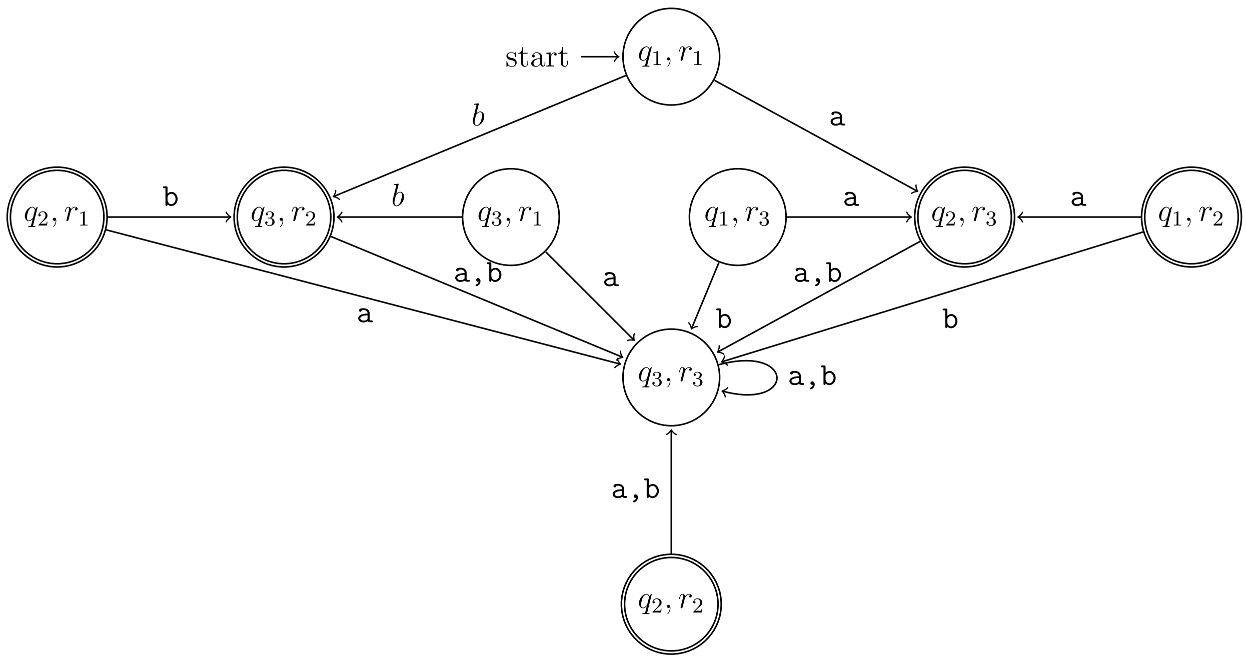
3. We can add edges for b on D_A to get D'_A:



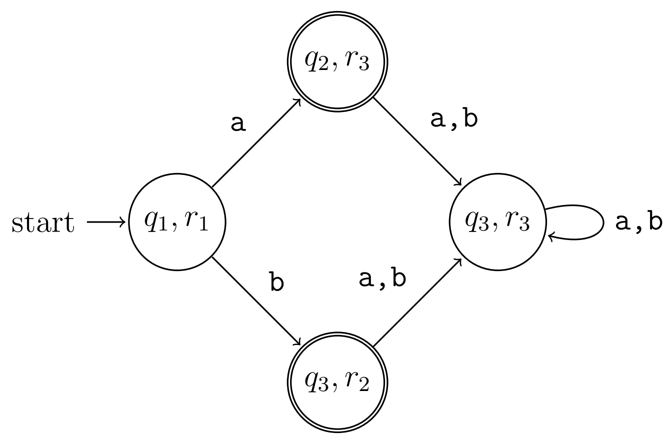
4. Similarly, we can get D'_B:



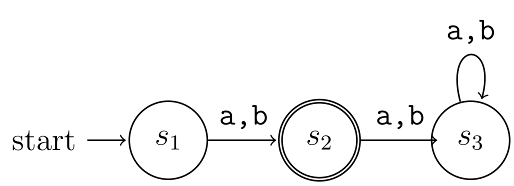
Then we can construct $D'_{A\cup B}$:



This figure can be simplified to the following one:



5. Furthermore, we can merge (q_2, r_3) and (q_3, r_2) as one state:



If there is a DFA with number of states < 3 for $A \cup B$, we can remove all links processing the character b. The resulting diagram is still a valid DFA and recognizes $\{a\}$. This contradicts the result in 2.. Therefore, the above diagram has the smallest number of states for $A \cup B$.

Problem 2 (15%)

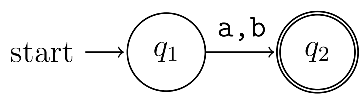
Consider the following regular expression

$$(a \cup b)^*$$

Assume $\Sigma = \{a, b\}$

1. What is the DFA with the smallest number of states for this language?

2. Starting with the following NFA.

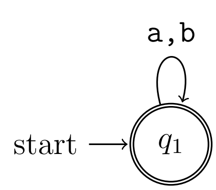
   

   Use the procedure in the textbook to prove that A^* is regular to generate a three-state NFA. Let's follow the textbook to have ϵ-links from the accept states to the original start state.[1] Give the formal definition of this three-state NFA.

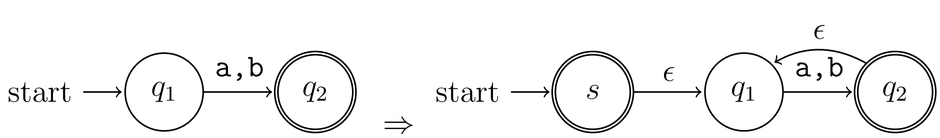
3. Convert the three-state NFA to an eight-state DFA by using the power set of NFA's states as the new set of states. Can the eight-state DFA be simplified to the DFA obtained in 1.?

Answer

1.



2.



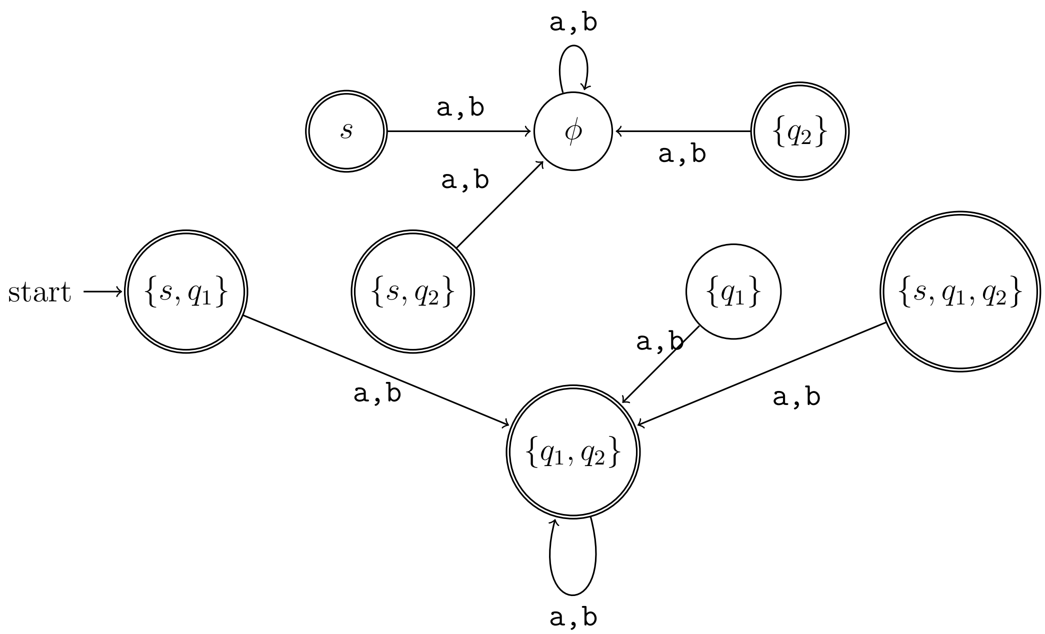
[1]In the lecture we have ϵ-links to the new start state. Don't do that.

Formal definition: $M = (\{s, q_1, q_2\}, \{\texttt{a}, \texttt{b}\}, \delta, s, \{s, q_2\})$

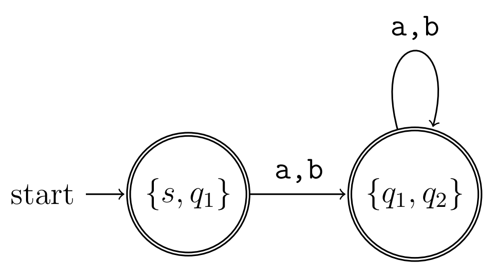
δ is described as

	a	b	ϵ
s	$\emptyset$	$\emptyset$	$\{q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\emptyset$
q_2	$\emptyset$	$\emptyset$	$\{q_1\}$

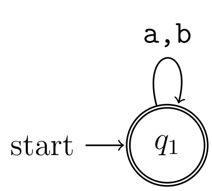
3.



Yes, we just need consider $\{s, q_1\}$ and $\{q_1, q_2\}$:



Merge the two states as one:



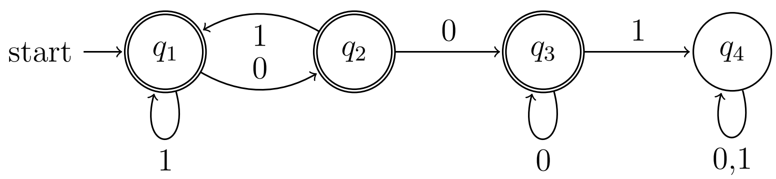
which is as the same result as 1.

Common Mistakes

1. Many did not give the correct formal definition of the NFA. This is not good.
2. From the eight-state DFA, you should show the two-state one before getting the one-state DFA. That is, you must give certain details.
3. You must give the formal definition. You cannot assume δ_1 of the original NFA is known and then copy the description of δ in the textbook.

Problem 3 (20%)

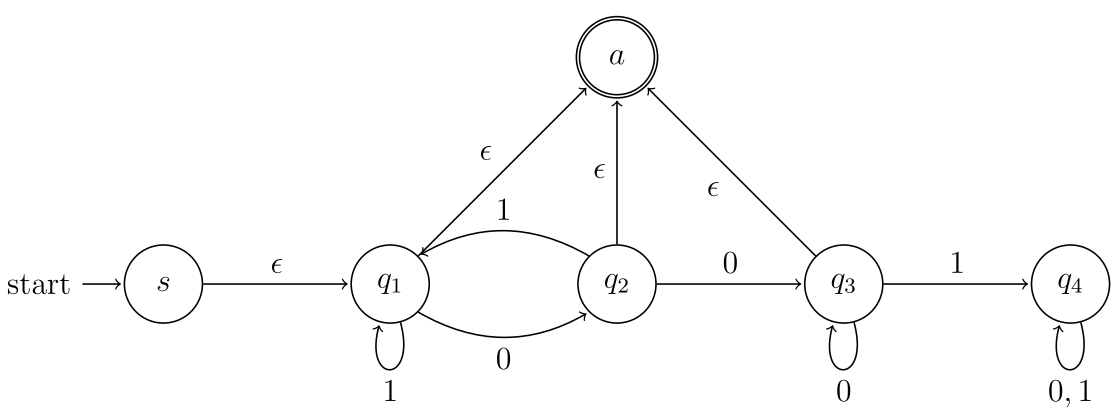
Consider the following state diagram:



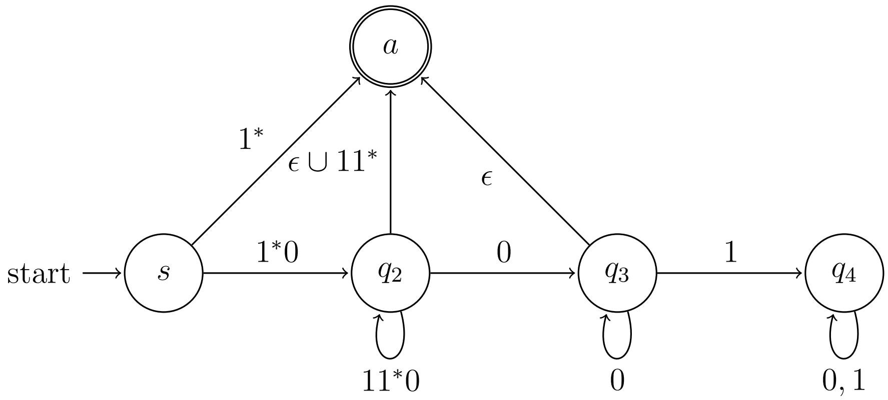
1. (5%) What is the language of this DFA?
2. (8%) Modify this DFA to GNFA and find the regular expression of the language. Please follow the order of q_1, q_2, q_3, q_4 to remove states.
3. (7%) Can you see that the results in 2. correspond to the language described in 1.? Prove it!

Answer

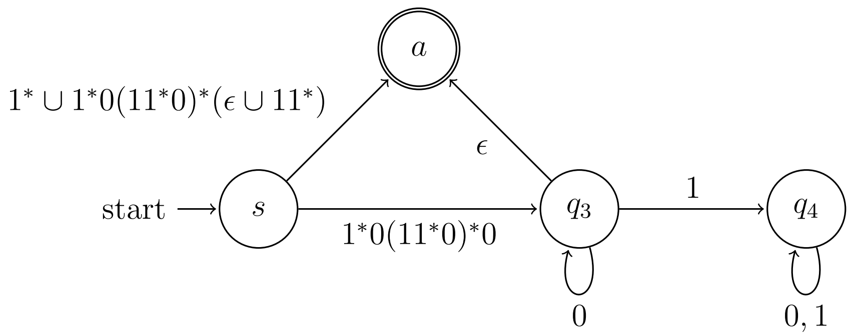
1. The language L is $\{w \mid w \in \{0,1\}^* \text{ and } w \text{ does not contain } 001\}$
   Once it contains 001, it goes to q_4 and is never accepted.
2. Add state s and state a:



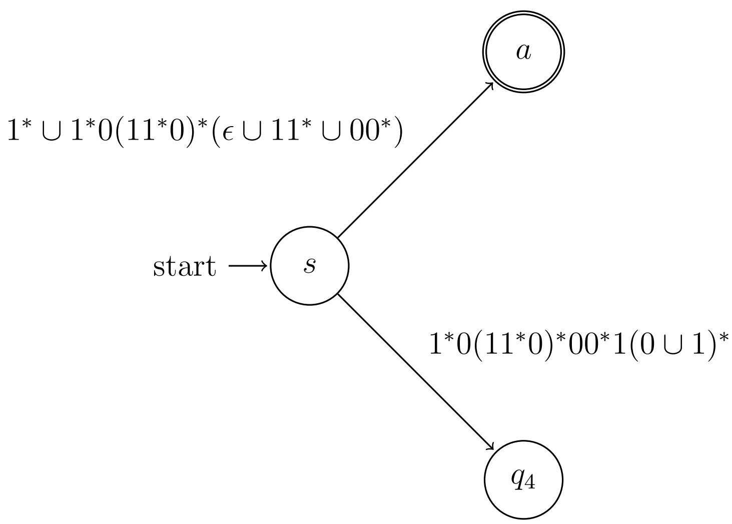
i) Remove q_1:



ii) Remove q_2:



iii) Remove q_3:



Therefore, the regular expression of the DFA is: $1^* \cup 1^*0(11^*0)^*(\epsilon \cup 11^* \cup 00^*)$
$=1^* \cup 1^*0(11^*0)^*(1^* \cup 0^*)$

3. Yes.

   The expression $0(11^*0)^*$ starts and ends with 0. It has no consecutive 0s since all 0s are separated by at least one 1. Thus both $1^*0(11^*0)^*1^*$ and $1^*0(11^*0)^*0^*$ cannot contain 001.

So any string satisfying the regular expression cannot contain 001.

For any string that does not contain 001, we can remove the first consecutive 1s, so that $w = 1^*w'$ and w' either is empty or starts with 0.

i) If $w' = \epsilon$, then $w = 1^*$.

ii) If w' ends in 1, then w' cannot have consecutive 0s. By removing the last consecutive 1s, the remaining string ends in 0, and can be represented by $0(11^*0)^*$. Thus w can be expressed by $1^*0(11^*0)^*1^*$.

iii) If w' ends in 0, then all the consecutive 0s must be at the end. Thus we can remove until the remaining string does not contain 00 and ends with 0. Thus w can be expressed by $1^*0(11^*0)^*0^*$.

Thus all strings not containing 001 can be expressed by the regular expression $1^*(\epsilon \cup 1^*0(11^*0)^*(1^* \cup 0^*))$.

So the regular expression is the same as the description in 1.

Common Mistakes

1. Many only explain that 001 cannot appear in the obtained expression, but you also need to check if a string without 001 can be represented by your expression.

Problem 4 (20%)

1. Assume L is regular. Is the following language regular?

$$\bar{L} = \{w \mid w \in L \text{ or } w^R \in L\},$$

where w^R is the reverse of w. Please prove your answer.

2. Is the following language regular?

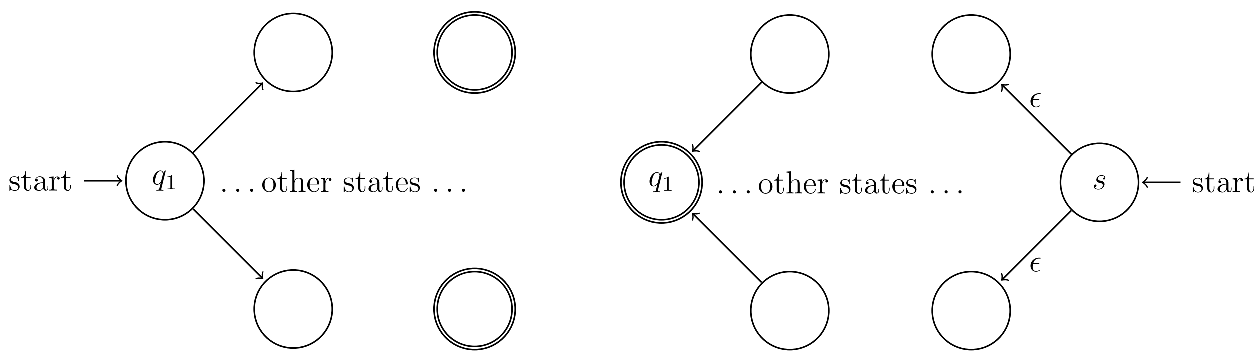
$$L = \{a^{n!} \mid n \geq 0\}$$

Please prove your answer.

Answer

1. For L, consider the following way to generate language L^R, where $L^R = \{w^R \mid w \in L\}$.

    1. Find a DFA M (Figure 1a) which recognizes language L.
    2. Add a node s as the new initial state, which connects an edge to each accept state with empty character ϵ. an edge from each accept state to
    3. Remove accept states and let originally initial state as the only accept state.
    4. Reverse the direction of all the original edges.
    5. So we get a new NFA (Figure 1b).



(a) original DFA M (b) new NFA

Figure 1: A NFA generated from DFA M by the procedure in answer of Problem 4.1

This new NFA can recognize at least L^R because $\forall w \in L$, w must end on some accept state of M with some path, and the reverse of this path is contained in it, hence, w^R will be accepted by it. In addition, because all of the paths and the corresponding reversed paths are one to one, it can just recognize L^R. Therefore, L^R is regular.

In addition, according to the definition of L^R, $w^R \in L \Leftrightarrow w \in L^R$. So $\bar{L} = \{w \mid w \in L \text{ or } w^R \in L\} = \{w \mid w \in L \text{ or } w \in L^R\} = L \cup L^R$ is also regular.

2. No.
Assume that L is regular. Given a pumping number p, we can construct a string $s = a^{p!}$ such that the length of s is certainly longer than p. According to the Pumping Lemma, s can be split into three pieces: $s = xyz$ s.t. $|xy| \leq p$ and $|y| > 0$. So $|y| \leq p$.
If $p = 1$, then $s = a$. If $|y| > 0, x, y, z$ can only be ϵ, a, ϵ. So for $i = 5, xy^5z = a^5 \notin L$.
If $p > 1$,then

$$|xy^2z| = p! + |y| \leq p! + p \leq p! + p! < p!p + p! = p!(p+1) = (p+1)!$$

So $p! + |y| < (p+1)! \neq q!\ \forall q > p$. So, for $i = 2, xy^2z = a^{p!+|y|} \notin L$.

Therefore, L is not regular.

Common Mistakes

1. In proving that L^R is regular, many wrongly assume there is one single accept state in L.

2. For $a^{p!}$, many did not consider the case of $p = 1$. To bypass the $p = 1$ issue, you can choose $s = 1^{q!}$, where $q > p$.

Problem 5 (20%)

In our proof of that

$$A = \{0^n 1^n \mid n \geq 0\}$$

is not regular, we demonstrate that we can use only two of the three conditions:

i. $xy^iz \in A, \forall i$

ii. $|y| > 0$

1. For proving that $\{1^{n^2} \mid n \geq 0\}$ is not regular, is it possible to use also only the above two conditions?

2. Is it possible to prove that a language is not regular by using only

   i. $xy^iz \in A, \forall i$

   ii. $|xy| \leq p$

   or

   i. $|y| > 0$

   ii. $|xy| \leq p$

Answer

1. Yes.

   For the language $\{0^n 1^n \mid n \geq 0\}$, what we have proved is that

   $\forall p > 0$ ($\exists s \in A, |s| \geq p$ such that

   $\{\forall x, y, z$ with $s = xyz,$ the following two conditions cannot be both true,

   i. $\forall i \geq 0, xy^iz \in A$

   ii. $|y| > 0$

   $\}$)

We will do the prove by a similar way.

Let $s = 1^{p^2} = xyz$.

We have

$$|x| + |y| + |z| = p^2$$

If $|y| > 0$, we have the following 2 cases.

- Case 1: $0 < |y| \leq p$

  Then when $i = 2$,

  $$|xy^2z| = p^2 + |y|$$

  and

  $$p^2 < p^2 + |y| < (p+1)^2 \ ,$$

  so $xy^2z \notin A$,which violates the first condition.

- Case 2: $|y| > p$

  Choose

  $$i = |y|p^2 + 1$$

  Then

  $$|xy^iz| = p^2 + (i-1)|y| \quad = p^2 + |y|^2p^2 = (1 + |y^2|)p^2$$

  However,

  $$\begin{aligned}(|y|p)^2 &< (1 + |y|^2)p^2 \\ &< 1 + 2|y|p + |y|^2p^2 \\ &= (1 + |y||p|)^2 \ ,\end{aligned}$$

  so $xy^{|y|p^2+1}z \notin A$, which violates the first condition.

2. No.

   (a) If now the two conditions are changed, we will show that the statement

   $\{\forall x, y, z$ with $s = xyz,$ the following two conditions cannot be both true,

   i. $\forall i \geq 0, xy^iz \in A$

   ii. $|xy| \leq p$

   $\}$

   can never be true. Therefore we cannot use them for the proof.

The opposite of {...} is

$$\{\exists x, y, z \text{ with } s = xyz, \text{the following two conditions are both true,}$$

$$\text{i. } \forall i \geq 0, xy^i z \in A$$

$$\text{ii. } |xy| \leq p$$

$$\}$$

This is always true because we can always pick $x = \epsilon$, $y = \epsilon$, $z = s$. As a result,

$$\text{i. } \forall i \geq 0, xy^i z = z = s \in A$$

$$\text{ii. } |xy| = 0 \leq p$$

(b) If now the two conditions are changed, we will show that the statement

$$\{\forall x, y, z \text{ with } s = xyz, \text{the following two conditions cannot be both true,}$$

$$\text{i. } |y| > 0$$

$$\text{ii. } |xy| \leq p$$

$$\}$$

can never be true. Therefore we cannot use them for the proof.

The opposite of {...} is

$$\{\exists x, y, z \text{ with } s = xyz, \text{the following two conditions are both true,}$$

$$\text{i. } |y| > 0$$

$$\text{ii. } |xy| \leq p$$

$$\}$$

This is always true because we can always pick $x = \epsilon$, $s = yz$ with $|y| = p$. As a result,

$$\text{i. } |y| = p > 0$$

$$\text{ii. } |xy| = p \leq p$$

Alternative solution of 1.

1. $\forall p > 0$, we choose $s = 1^{p^2}$.

   If $|y| > 0$, we can split s into xyz, where

$$x = 1^j, 0 \leq j \leq p^2.$$

$$y = 1^k, 0 < k \leq p^2.$$

$$z = 1^{p^2-j-k}, 0 < j + k \leq p^2.$$

$$|x| = j, |y| = k, |z| = p^2 - j - k.$$

We would like to prove that

$$\forall y, \exists i \geq 0 \text{ s.t. } xy^i z = 1^j 1^{ik} 1^{p^2-j-k} = 1^{p^2+k(i-1)} \notin A.$$

Choose $i = 2p + k + 2$, then

$$\begin{aligned} p^2 + k(i-1) &= p^2 + k(2p + k + 1) \\ &= p^2 + 2kp + k^2 + k. \end{aligned}$$

$$\begin{aligned} (p+k)^2 &< p^2 + 2kp + k^2 + k \\ &< p^2 + 2kp + k^2 + 2p + 2k + 1 \\ &= (p + k + 1)^2. \end{aligned}$$

So $p^2 + 2kp + k^2 + k$ is not a square number.

Therefore, $xy^{2p+k+2}z = 1^{p^2+2kp+k^2+k} \notin A$.