Student Information

Full Name : Adil Kaan akan
Id Number : 2171155

Answer 1

a.

We can prove with a method that is as follows.
The method uses an integer as a denominator and gives all integers that is less than that denominator as numerator. For the listing, the method uses reverse diagonals such that -1/2,-1/3,-2/3,-1/4,-1/5,-2/4,..

-1/2
-1/3 -2/3
-1/4 -2/4 -3/4
-1/5 -2/5 -3/5 -4/5
...

We can write infinitely many rows if we increment the denominator one by one. Since there are infinitely many rows, there are infinitely many rational number in interval (-1,0). Since the method allows us to list the number by using the diagonals, we can list them to have one to one correspondence between natural numbers. Therefore, we can say that the rational numbers in the open interval which is (-1,0) have the same cardinality with natural numbers and are countably infinite.

b.

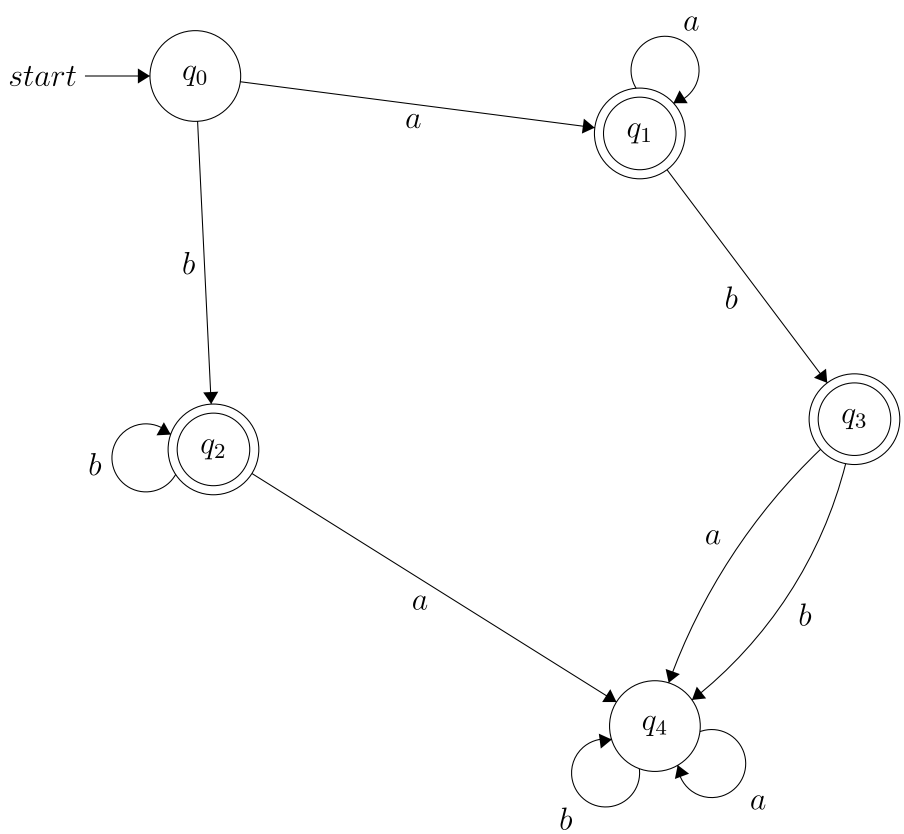
If the given language L is a regular language, then it must be accepted by the finite automaton. We can build a finite automata to accept the L^+. Since the L^+ is $L \circ L^*$, we can use the kleene star and concentanation. Since all finite languages are regular,we can build a finite automata that has finite states to accept finite language, we can say that L is regular. Since the kleene star operation and concatenation operation are closures, we can say that if we L is regular then L^* is regular and $L \circ L$ is also regular. By using that, we can say that $L \circ L^*$ is a regular language. Since the L^+ is $L \circ L^*$, the all of the finite languages, say L, all the L^+ are regular. Therefore, the set $D = \{L^+ :$ finite regular language over the unary alphabet $\sum = \{a\}$ and L^+ is not regular is emptyset.

c.

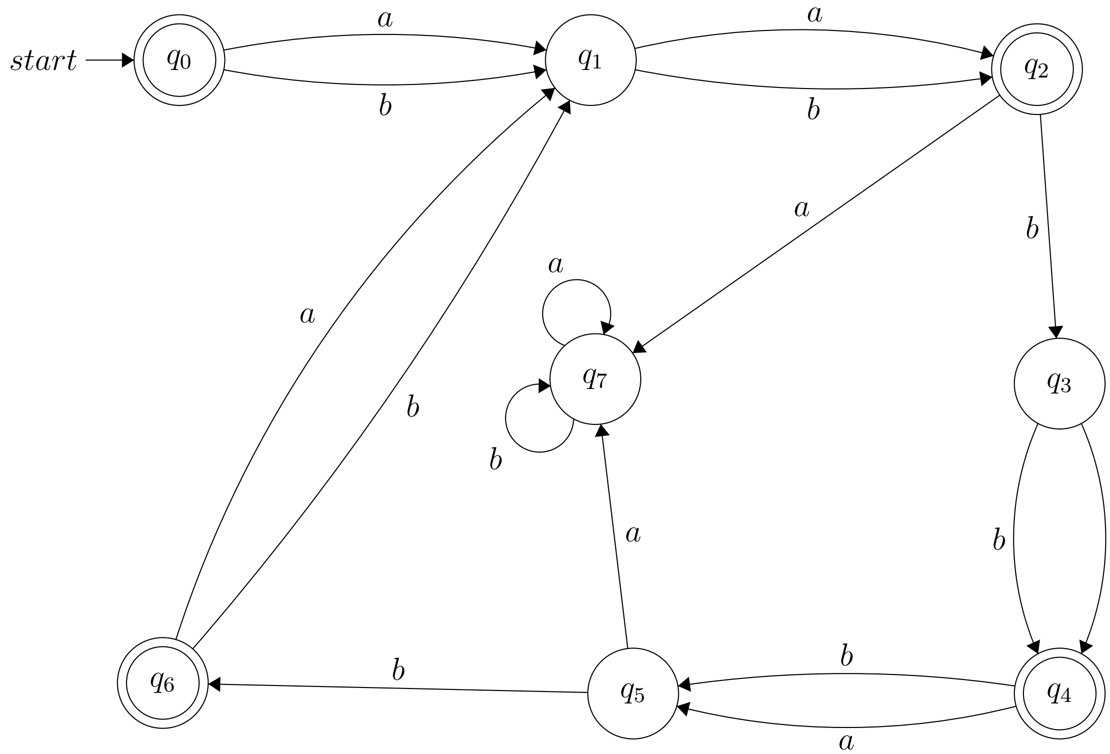
The set of all languages contains the regular languages and the languages that are not regulaer, actually it is union of them. It is uncountable since there are countably infinite alphabet and the cardinality of all languages is $2^{|\Sigma^*|}$ and that means uncountability. Since the regular languages are countably infinite and the set of all languages is uncountable, the set of non regular languages must be uncountable to the set of all languages be uncountable. Then, we can say that the set of languages that are not regular is uncountable.

Answer 2

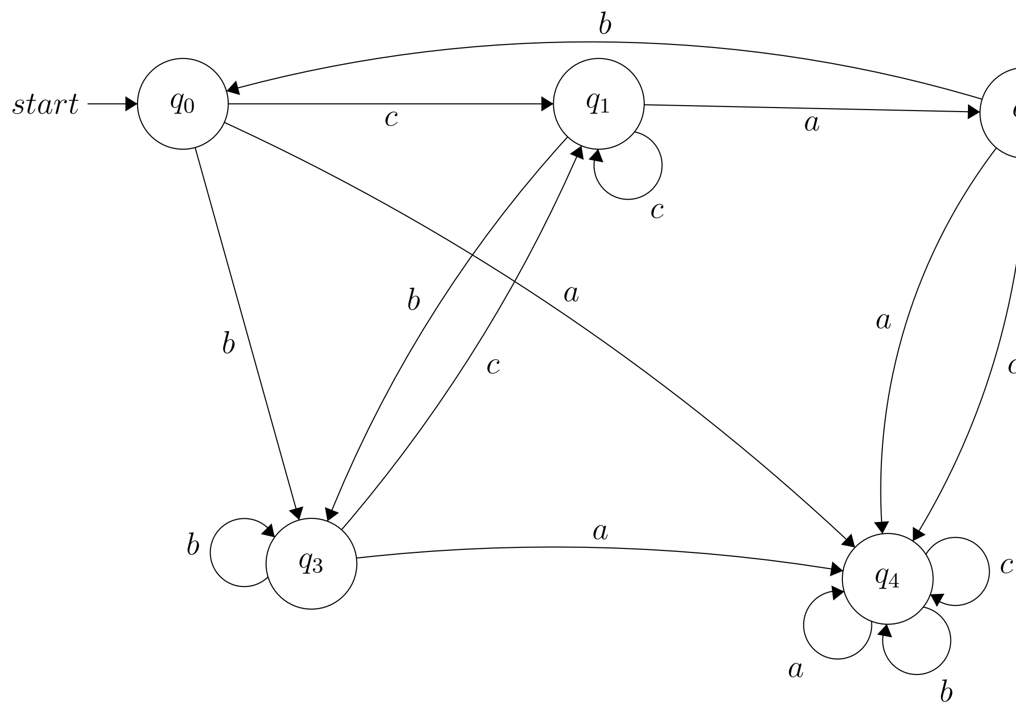
a.



b.


start
q_0
a
b
q_1
a
b
q_2
a
b
a
a
b
q_7
b
q_3
b
a
a
b
q_6
b
q_5
b
a
q_4

c.


b
start
q_0
c
q_1
a
q_2
c
b
a
b
c
a
c
b
q_3
a
q_4
c
a
b

Answer 3

a.

$(q_0, abbb) \vdash_N (q_1, bbb) \vdash_N (q_2, bb)$
$(q_0, abbb) \vdash_N (q_2, abbb) \vdash_N (q_4, bbb) \vdash_N (q_3, bb) \vdash_N (q_3, b), (q_4, e)$
$(q_0, abbb) \vdash_N (q_1, bbb) \vdash_N (q_3, bb) \vdash_N (q_3, b) \vdash_N (q_3, e)$

On the above, there are some of the ways which experessing the string "abbb" and all of them end on the q_3 or q_4 state. Since only accepting state is q_5 and there is no way to reach from q_3 andq_4 to q_5, there is no way to accept "abbb" string.

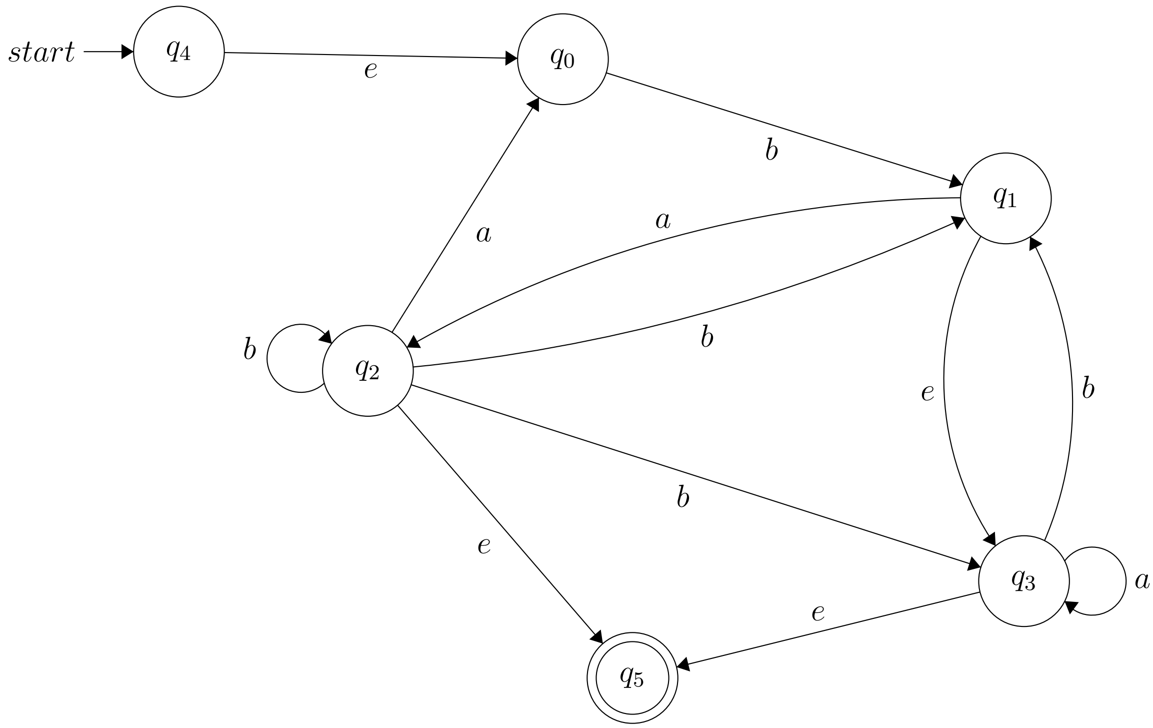
b.

$(q_0, ababa) \vdash_N (q_1, baba) \vdash_N (q_2, aba) \vdash_N (q_4, ba) \vdash_N (q_3, a) \vdash_N (q_5, e)$ and also $(q_0, ababa) \vdash_N (q_2, ababa) \vdash_N (q_4, baba) \vdash_N (q_3, aba) \vdash_N (q_1 ba) \vdash_N (q_3, ba) \vdash_N (q_3, a) \vdash_N (q_5, e)$ Since both of the configurations end in the q_5 which accept state of the N, the string "ababa" accepted by the N.

Answer 4

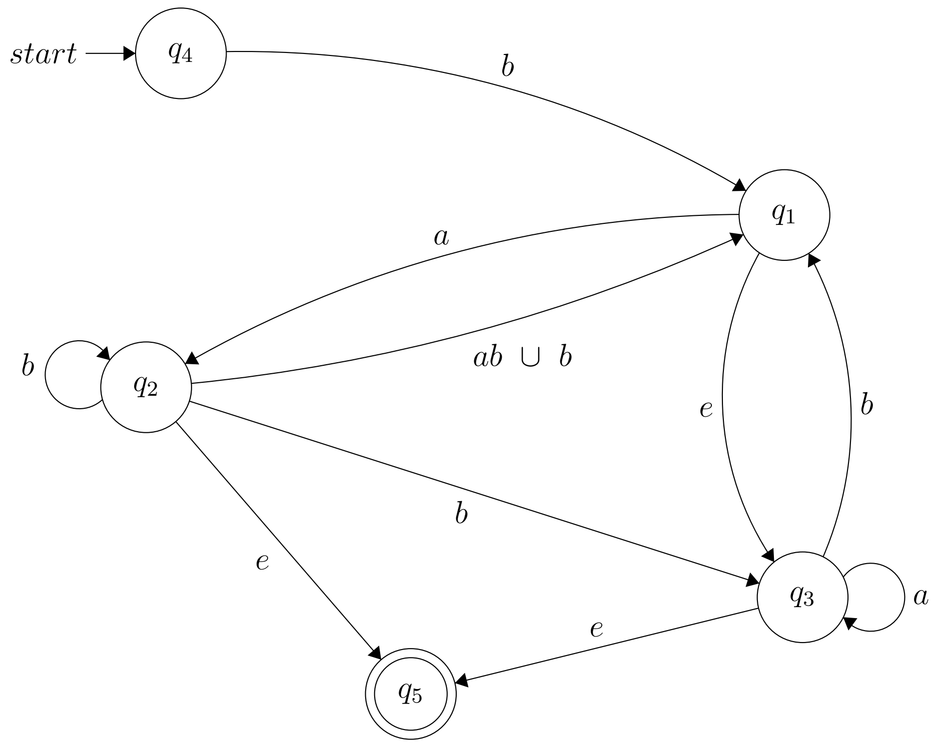
a.

We should add a new start state which is q_4 and a new final state which is q_5 in order to do state elimination. Moreover, we should add empty transitions from new start states, q_4, and to new final state ,q_5. After doing these operations the corresponding NFA is as follows:

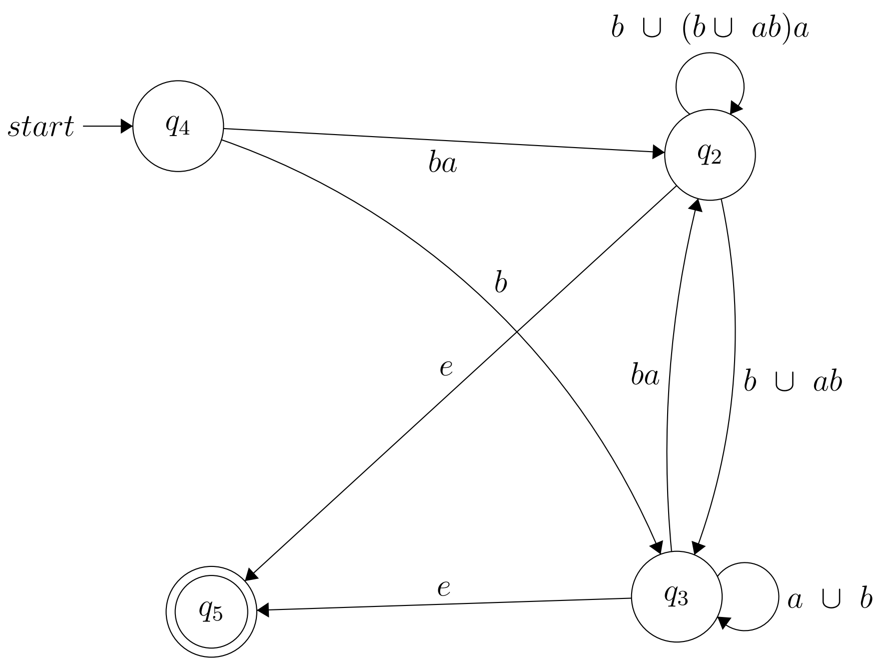


b.

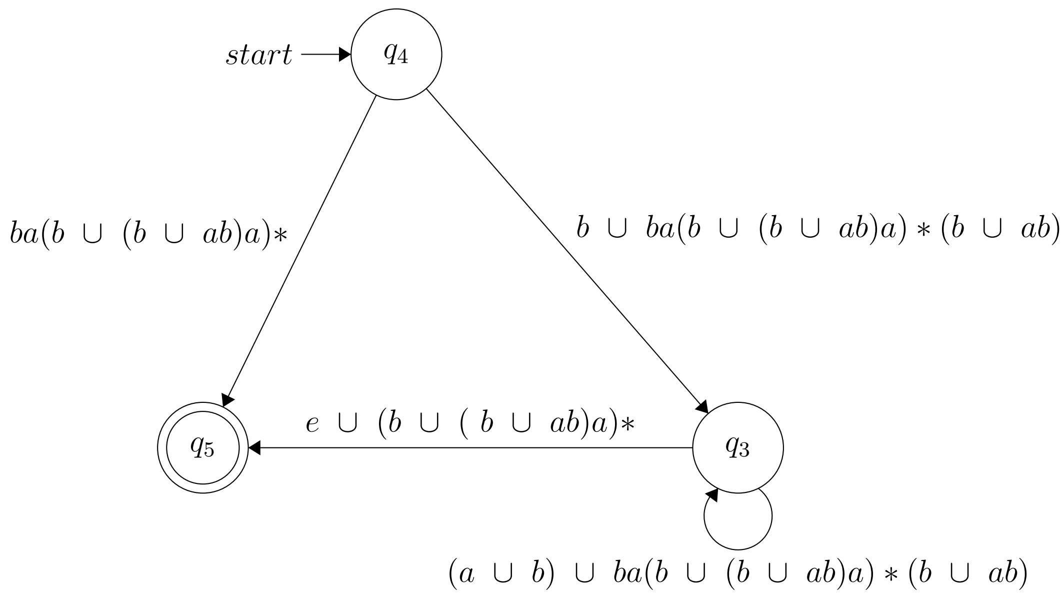
Now i will eliminate states one by one and draw the GFA corresponding to the eliminated state.
After eliminating q_0 state, we get GFA which follows.



Then when eliminate the q_1 the corresponding GFA as follows



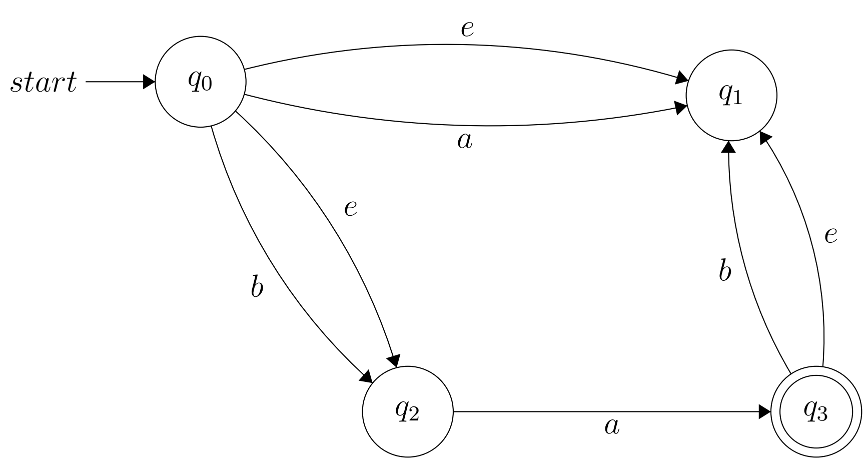
After eliminating q_2, corresponding GFA as follows



After eliminating q_3 we will get the final regular expression which is α
$= ba(b\cup(b\cup ab)a)*\cup b\cup ba(b\cup(b\cup ab)a)*(b\cup ab)((a\cup b)\cup ba(b\cup(b\cup ab)a)*(b\cup ab))*e\cup ba(b\cup(b\cup ab)a)*$

Answer 5

The given NFA which N is



a.

If we use the algorithm, we should write the states which are

$E(q_0) = q_0, q_1, q_2$
$E(q_1) = q_1$
$E(q_2) = q_2$
$E(q_3) = q_1, q_3$

We can find those by looking the states, adding the states that is reachable with empty transitions.

The state which is q_0, q_1, q_2 should be initial states since q_0 was the inital state of N.

Then we will look the transtion function.

$\delta(q_0, q_1, q_2, a) = E(q_1) \cup E(q_3) \cup \emptyset = q_1, q_3$
$\delta(q_0, q_1, q_2, b) = E(q_2) \cup \emptyset \cup \emptyset = q_2$

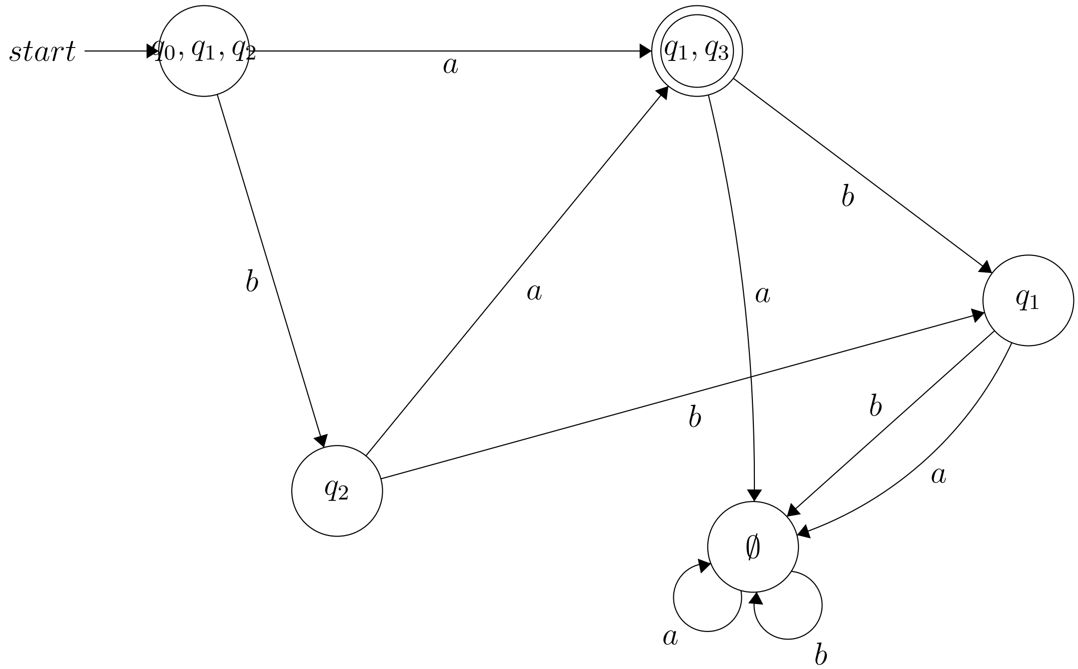
$\delta(q_1, q_3, a) = \emptyset \cup \emptyset = \emptyset$
$\delta(q_1, q_3, b) = \emptyset \cup E(q_1) = q_1$

$\delta(q_2, a) = E(q_3) = q_1, q_3$
$\delta(q_2, b) = \emptyset$

$\delta(q_1, a) = \emptyset$
$\delta(q_1, b) = \emptyset$

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

With using new states and new transitions, we get the equivalent DFA which follows



b.

If we look the accepted state, which is q_1, q_3 the language A is only accepted strings which are "a" or "ba". The complement of that language is the set of strings which are not "a" and "ba". The regular expression of the complement of the language which is $\bar{L}$
$= e \cup b \cup (a(a \cup b) \cup bb \cup ba(a \cup b))(a \cup b)^*$
which is also $\bar{L} = \{w \in \sum^* : w \textit{ is not } "a" \textit{ and } "ba"\}$

Answer 6

We can think $L_1 - L_2$ as $L_1 \cap \overline{L_2}$ and also as $\overline{(\overline{L_1} \cup L_2)}$

Assume that the DFA $M_1 = \{K_1, \sum_1, \delta_1, s_1, F_1\}$ recognizes language one which is L_1 and $M_2 = \{K_2, \sum_2, \delta_2, s_2, F_2\}$ recognizes language two which is L_2

After getting the complement of the M_1 which is DFA that recognizes L_1, we get the $\overline{M_1}$.
Then add the new initial state which is lets say s_0.
Then add empty transitions to $\overline{M_1}$'s inital state and M_2's initial state.
When we did these operations, we will get $\overline{M_1} \cup M_2$. After that, we can use the algorithm that can change NFA to DFA. By using this algorithm we can get $M_0 = \overline{M_1} \cup M_2$
After that, if we get the complemt of M_0, we get $M_0 = \overline{(\overline{M_1} \cup M_2)}$ which recognizes $\overline{(\overline{L_1} \cup L_2)}$ which is equivalent to $L_1 - L_2$.
Since we find M_0 which recognizes $L_1 - L_2$, we can easily say that $L_1 - L_2$ is regular language since it is recognized by finite automaton which is M_0.

Answer 7

a.

Assume the given language which is $L = \{w \in \{a, b\}^* : f(a, w) = n^2 for some n \in N\}$

The question want us to use pumping lemma.

By using pumping lemma we can say that there is an integer $k \geq 1$ such that for any string w, $|w| > k$, and the string can be splitted such as w = xyz such that the length of xy , $|xy| \leq k$, is less than or equal to k and $xy^iz \in$ L for all i ≥ 0

Then, take $w = b^{k-n^2}a^2a^{n^2-2}b^l$ where l is a positive number, $l > 0$. It is obvius that w is in the language L because it contains n^2 a's. We can the string w as $x = b^{k-n^2}a^2$ y = a^{n^2-2} and $z = b^l$. Then, we can use pumping lemma to claim $xy^iz \in$ L for all i $\geq$ 0 because the length of the is greater than k,$|w| = k + l > l$, the length of the string xy is less than or equal to k,$|xy| = k \leq k$, and we assumed that the L is regular language. But, if we take i = 0, we will get xz = $b^{k-n^2}a^2b^l$. By pumping lemma that string must be in the language, however, that is not possible since the string only contains 2 a's, since 2 is not a square of a natural number. The fact that L is not a regular language is proved by the contradiction and pumping lemma.