UNIT 2 IMPORTANT QUESTIONS

SHORT QUESTIONS

1. Define Regular Expression

A:- → The language accepted by finite automata can be easily described by simple expressions called Regular Expressions. It is the most effective way to represent any language.

→ The languages accepted by some regular expression are referred to as regular languages.

2. Define the pumping lemma for regular expression.

A:- Formal Definition of pumping lemma for regular expression:

Let L be a regular language then there exists a constant 'p' such that for every string W in L

where $|W| \geq p$

W can be divided into 3 parts

$W = xyz$ such that the following conditions must be true

i) $|y| \geq 0$

ii) $|xy| \leq p$

iii) $\forall k \geq 0$, then the string xy^kz is also in L.

5. Define Grammar?

3. List the Identity Rules of Regular sets

A: Identity rules for regular sets:

1) $\phi + R = R$ (Union)
2) $\phi R + R\phi = \phi$ (Concat)
3) $\epsilon R = R\epsilon = R$
4) $\epsilon^* = \epsilon$ (and) $\phi^* = \epsilon$
5) $R + R = R$
6) $R^* R^* = R^*$
7) $RR^* = R^* R$
8) $(R^*)^* = R^*$
9) $\epsilon + RR^* = \epsilon + R^* R = R^*$
10) $(PQ)^* P = P(QP)^*$
11) $(P+Q)^* = (P^* + Q^*)^* = (P^* Q^*)^*$
12) $(P+Q)R = PR + QR$ & $R(P+Q) = RP + RQ$.

Scanned with CamScanner

4. List any two algebraic properties of regular expressions?

A) Let r_1, r_2 and r_3 be the 3 regular expressions

(1) Commutative law for union : $r_1 + r_2 = r_2 + r_1$

(2) Associative law for union:

$(r_1 + r_2) + r_3 = r_1 + (r_2 + r_3)$

Scanned with CamScanner

5 List the closure properties of regular languages

A:- 1) The Union of two regular languages is regular

2) The Intersection of two regular languages is regular.

3) The complement of a regular language is regular.

4) The difference of two regular languages is regular.

5) The reversal of a regular language is regular.

6) The closure (star) of a regular language is regular.

7) The concatenation of regular languages is regular

8) A Homomorphism (substitution of strings for symbols) for a regular language is regular.

9) The inverse homomorphism of a regular language is regular.

6 What are the steps involved in conversion of FA to regular expression

A: There are two methods involved in conversion of FA to regular expression.

1) State elimination method
2) Arden's theorem

1) State elimination method:

This method involves the following steps in finding the regular expression for the given finite automata

Step-I: The Initial state should not have any incoming edge.
If there exists any incoming edge to the initial state then create a new initial state having no incoming edge to it.

Step-II: The final state of the finite automata should not have any outgoing edge.

If their exist any outgoing edge from the final state then create a new final state then create a new final state having no outgoing edge from it with ϵ move.

Step-III: There must exist only one final state in the finite automata if there exist multiple final state in the FA then convert all the final states into non-final states and create a new single final state with ϵ move.

Step-IV: -Eliminate all the intermediate states one by one these states may be eliminated in any order

Other than the initial state and final state eliminate the remaining states one by one.

2) Arden's theorem:

If P and Q are two regular expressions over Σ and if P does not contain ϵ then then the following equation in R given by

$R = Q + RP$ has a unique solution i.e. $R = QP^*$

Scanned with CamScanner

7. Write any two applications of regular expression

A. Applications of regular expressions:-

→ Text Editors

→ Lexical Analysers

8. List the decision properties of regular languages.

A. Decision properties of regular language:-

1) Membership
2) Emptiness
3) Equivalence
4) Containment

9 Prove the Arden's theorem?

A> Arden's theorem:

If P and Q are two regular expressions over Σ and if P does not contain ϵ then the following equation in R given by $R = Q + RP$ has a unique solution i.e., $R = QP^*$

Case-1: $R = Q + RP$

$R = Q + [QP^*]P \quad [\because R = QP^*]$

$R = Q(\epsilon + P^*P)$ $[\because \epsilon + RR^* = \epsilon + R^*R = R^*]$

$R = QP^*$

Case-2:- $R = Q + RP$

$= Q + (Q+RP)P$ $[\because R = Q+RP]$

$= Q + QP + RPP$

$= Q + QP + (Q+RP)P^2$

$= Q + QP + QP^2 + RP^3$

$\vdots$

$= Q + QP + QP^2 + QP^3 + \ldots\ldots + QP^n + RP^{n+1}$

$= Q + QP + QP^2 + QP^3 + \ldots\ldots + QP^n + QP^*P^{n+1}$ $[\because R = QP^*]$

$= Q(\epsilon + P + P^2 + P^3 + \ldots\ldots + P^n + P^*P^{n+1})$

$R = QP^*$

10 Construct the Regular expression whose length is atleast 2

A:- $L = \{aa, ab, ba, bb, aaa \ldots\ldots\}$

$(a+b)(a+b)(a+b)^*$

where $* = 0$ to n

LONG QUESTIONS

11. Define regular expression? Find out the regular expression where $\Sigma = \{a,b\}$
a) where length is atleast 2
b) length is even
c) length is odd.

A: Regular Expression:
The language accepted by finite automata are easily described by simple expressions called regular expressions.
→ The Regular expression is the most effective way to represent regular language.

→ Regular expression contains 3 operators.

① Union (or) plus (+)

② Concatenation (·)

③ Kleene closure (*)

→ ϕ is a regular expression which denotes the empty set
$\phi \Rightarrow \{ \}$

→ ϵ is a regular expression and it denotes the set $\{\epsilon\}$ and it is a null string $\epsilon \Rightarrow \{\epsilon\}$.

→ For each 'a' in Σ, a is a regular expression and it denotes the set $\{a\}$

$$a \in \Sigma \Rightarrow \{a\}$$

→ $\phi, \epsilon, a \in \Sigma$ are called primitive regular expressions.

a) length is atleast 2

$L = \{aa, ab, ba, bb, aaa, \text{------}\}$

$\rightarrow (a+b)(a+b)(a+b)^*$

where $* = 0$ to n

b) length is even

$L = \{\epsilon, aa, ab, ba, bb, \text{------}\}$

$[(a+b)(a+b)]^*$

where $* = 0$ to n

c) length is odd

$L = \{a, b, aaa, \text{------}\}$

$[(a+b)(a+b)]^*(a+b)$

where $* = 0$ to n

12 Explain the process of equivalence between Finite automata and find out whether the given finite automata are equivalent or not.

Finite automata 'A'

State/Σ	c	d
$\rightarrow Q_1^*$	Q_1	Q_2
Q_2	Q_3	Q_1
Q_3	Q_2	Q_3

Finite automata 'B'

State/Σ	c	d
$\rightarrow Q_4^*$	Q_4	Q_5
Q_5	Q_6	Q_4
Q_6	Q_7	Q_6
Q_7	Q_6	Q_4

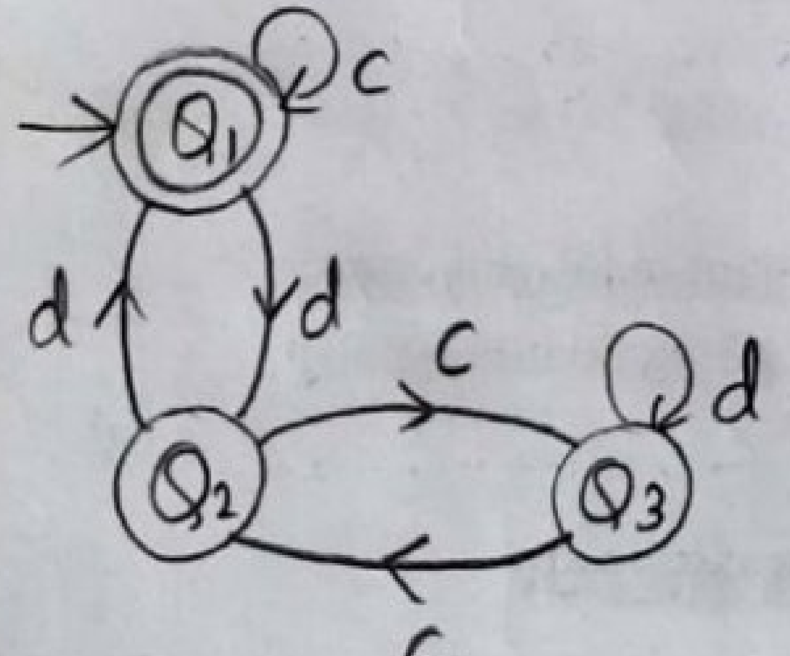
A: Steps to identify equivalence:

1) For any pair of states $\{q_i, q_j\}$ the transition for input $a \in \Sigma$ is defined by $\{q_a, q_b\}$ where $\delta\{q_i, a\} = q_a$ and $\delta\{q_j, a\} = q_b$

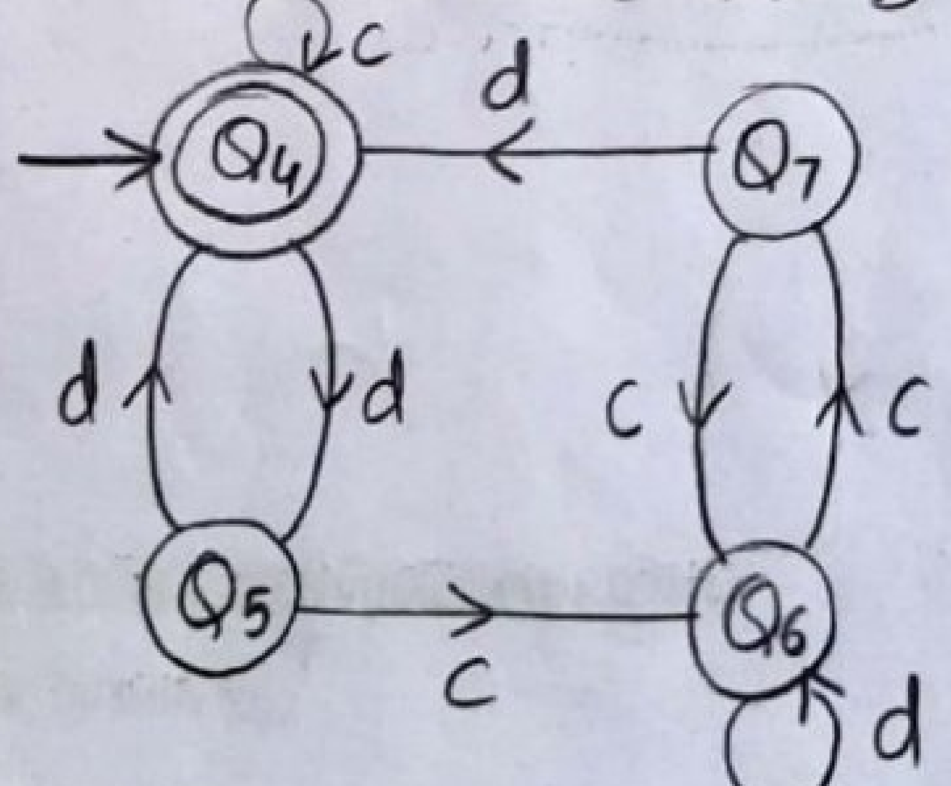
The two automata are not equivalent if for a pair $\{q_a, q_b\}$ one is INTERMEDIATE state and the other is FINITE state.

2) If Initial State is final state of one automata, then in the second automata also initial state must be final state for them to be equivalent.

Finite automata 'A'



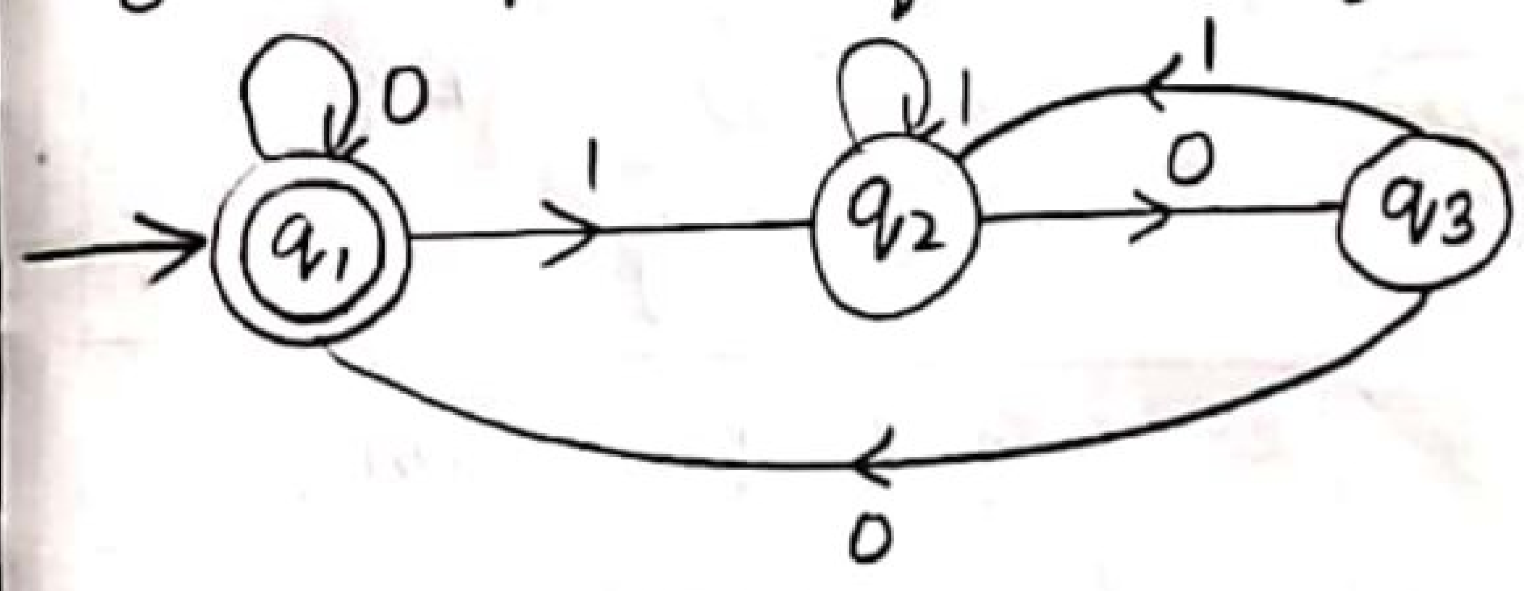
Finite automata 'B'



States	c	d
(Q_1, Q_4)	(Q_1, Q_4)	(Q_2, Q_5)
(Q_2, Q_5)	(Q_3, Q_6)	(Q_1, Q_4)
(Q_3, Q_6)	(Q_2, Q_7)	(Q_3, Q_6)
(Q_2, Q_7)	(Q_3, Q_6)	(Q_1, Q_4)

Both finite automatas are equivalent.

13) State and prove Arden's theorem. Find out the regular expression from the given FA



A:- Arden's theorem:

If P and Q are two regular expressions over Σ. and if P doesnot contain ϵ then the following equation in R given by

$R = Q + RP$ has a unique solution i.e. $R = QP^*$

Proof:

Case-1: $R = Q + RP$

$= Q + QP^*P \quad [\because R = QP^*]$

$= Q(\epsilon + P^*P)$

$R = QP^*$

Case-2: $R = Q + RP$

$= Q + (Q + RP)P \quad [\because R = Q + RP]$

$R = Q + QP + RPP$

$R = Q + QP + (Q + RP)P^2$

$R = Q + QP + QP^2 + RP^3$

$\vdots$

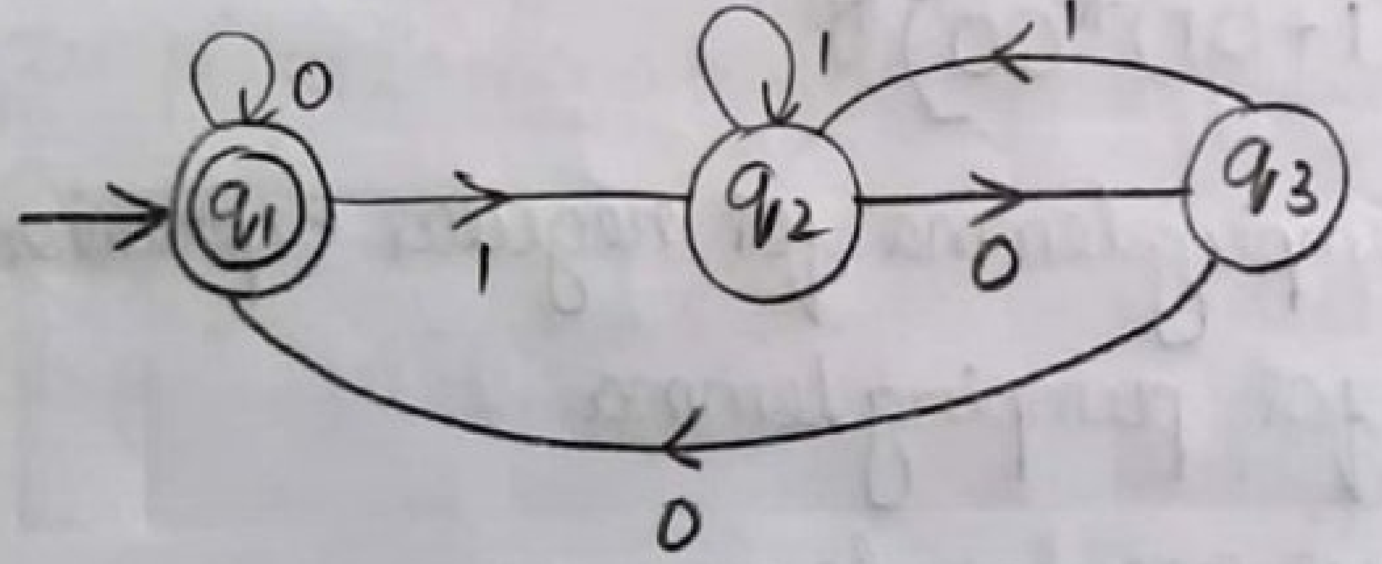
$R = Q + QP + QP^2 + \cdots + QP^n + RP^{n+1}$

$R = Q + QP + QP^2 + \cdots + QP^n + QP^*P^{n+1}$

$[\because R = QP^*]$

$R = Q[\epsilon + P + P^2 + \cdots + P^n + P^*P^{n+1}]$

$R = QP^*$



$q_1 = q_1 0 + q_3 0 + \epsilon \longrightarrow (1)$

$q_2 = q_1 1 + q_2 1 + q_3 1 \longrightarrow (2)$

$q_3 = q_2 0 \longrightarrow (3)$

$q_2 = q_1 1 + q_2 1 + q_3 1$

Substitute q_3 in q_2

$q_2 = q_1 1 + q_2 1 + [q_2 0] 1$

$q_2 = q_1 1 + q_2 1(\epsilon + 0)$

$\underset{R}{\underline{q_2}} = \underset{Q}{\underline{q_1 1}} + \underset{R}{\underline{q_2}}\,\underset{P}{\underline{(1+01)}}$

$q_2 = q_1 1[1+01]^*$ —— ④ $[\because R = QP^*]$

Substitute Eq ④ in ③

$q_3 = q_2 0$

$q_3 = q_1 1[1+01]^* 0$ —— ⑤

Substitute Eq ⑤ in Eq ①

$q_1 = q_1 0 + [q_1 1[1+01]^* 0]0 + \epsilon$

$q_1 = q_1 0 + q_1 1[1+01]^* 00 + \epsilon$

$\underset{R}{q_1} = \underset{R}{q_1}\underset{P}{[0+1[1+01]^* 00]} + \underset{Q}{\epsilon}$

$q_1 = \epsilon[0+1(1+01)^* 00]^*$ $[\because R = QP^*]$

$q_1 = [0+1(1+01)^* 00]^*$

14. a) Explain the pumping lemma for regular expression and applications for pumping lemma

A: Pumping Lemma for regular Language:

→ Pumping Lemma is used to prove that the given language is not a regular language.

→ It cannot be used to prove the language is regular.

→ Whenever the given language is regular then we are able to construct DFA, NFA, ε-NFA automata to accept the given language.

Formal definition of the pumping lemma:
Let L be a regular language then there exist a constant 'P' such that for every string W in L

where $|W| \geq P$

W can be divided into 3 parts

$W = xyz$ such that the following conditions must be true

i) $|y| \geq 0$

ii) $|xy| \leq P$

iii) $\forall\, k \geq 0$, then the string xy^kz is also in L.

To Prove that a language is not regular using pumping lemma: We prove using contradiction

Step-I: Assume that L is regular.

Step-II: It has a pumping length i.e. P

→ All strings longer than P can be pumped

→ Now find a string W in L such that $|W| \geq P$

→ Divide W into xyz.

→ Show that None of these 3 ($|y| \geq 0$, $|xy| \leq P$, $k \geq 0$ $xy^kz \in L$) conditions can satisfy

→ Then W cannot be pumped.

Applications of pumping Lemma:

→ Pumping Lemma is to be applied to show that certain languages are not regular

→ It should never be used to show a language is regular

i) If L is regular, it satisfies pumping lemma.

ii) If L does not satisfy pumping lemma, then it is not a regular

b) Prove that the given language $L = \{0^n 1^n / n >= 1\}$ is not a regular

A: Let the set L is defined by

$L = \{01, 0011, 000111, \text{-------}\}$

choose a string from the defined language and apply pumping lemma.

$W = 0011$

$x = 0 \,,\, y = 01, z = 1$

For $K \geq 0$ then $xy^kz \notin L$ Hence

$K = 2$

$= 0(01)^2 1$

$= 001011 \notin L$

then, the given language is not a regular language

15. Discuss pumping lemma for regular expression and applications of pumping lemma.
Show that the given language $L = \{a^P / P \text{ is prime}\}$ is not a regular.

Scanned by CamScanner

A: Pumping Lemma for regular Language:

→ Pumping Lemma is used to prove that the given language is not a regular language.

→ It cannot be used to prove the language is regular.

→ Whenever the given language is regular then we are able to construct DFA, NFA, ∈-NFA automatas to accept the given language.

Formal definition of the pumping Lemma:
Let L be a regular language then there exist a constant 'P' such that for every string W in L
where $|W| \geq P$
W can be divided into 3 parts
$W = xyz$ such that the following conditions must be true
i) $|Y| \geq 0$
ii) $|xy| \leq P$
iii) $\forall K \geq 0$, then the string xy^Kz is also in L.

To Prove that a language is not regular using pumping Lemma: We prove using contradiction

Step-I: Assume that L is regular.

Step-II: It has a pumping length i.e. P

→ All strings longer than P can be pumped

→ Now find a string W in L such that $|W| \geq P$

→ Divide W into xyz.

→ Show that None of these 3 ($|Y| \geq 0$, $|xy| \leq P$, $K \geq 0$ $xy^Kz \in L$) conditions can satisfy

→ Then W cannot be pumped.

Applications of pumping lemma:

→ Pumping lemma is to be applied to show that certain languages are not regular

→ It should never be used to show a language is regular

i) If L is regular, it satisfies pumping lemma.

ii) If L does not satisfy pumping lemma, then it is not a regular

Given,

$L = \{a^P / P \text{ is prime}\}$

Let the set L is defined by

$L = \{aa, aaa, aaaaa, aaaaaaa, \ldots\}$

choose a string from the defined language and apply pumping lemma

$W = aaa$

$x = a, y = a, z = a$

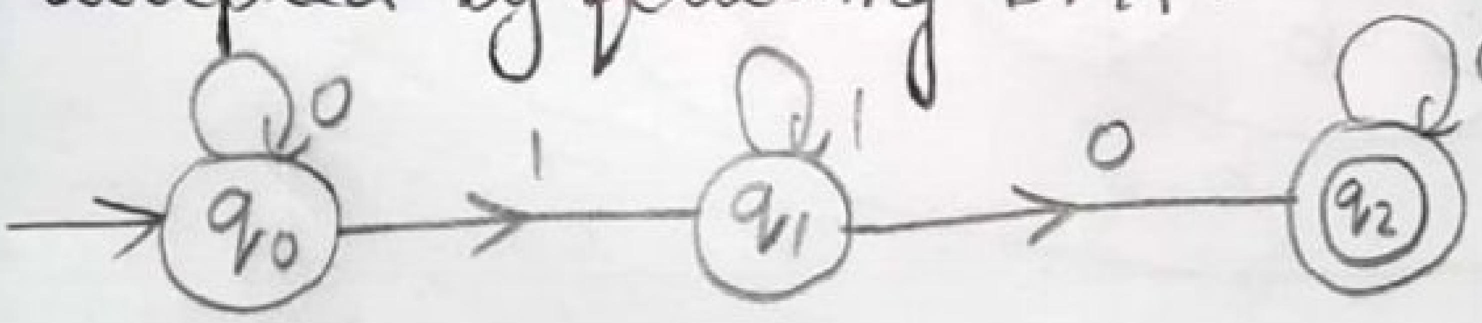
$k = 2 \Rightarrow xy^k z$

$= a(a)^2 a$

$= aaaa \notin L$

It is not a regular language.

16. State Arden's theorem. Construct the regular expression corresponding to the language accepted by following DFA

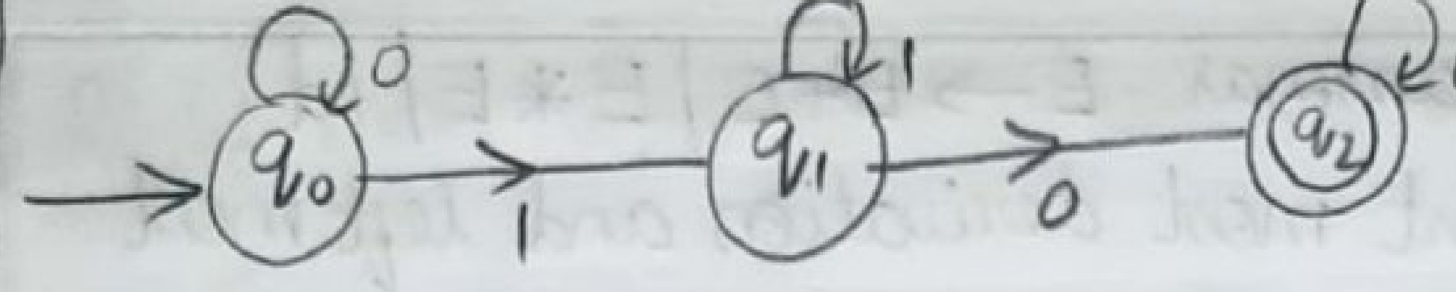


A:- Arden's theorem:

If P and Q are two regular expressions over Σ and if P does not contain ϵ then the following equation in R is given by

$R = Q + RP$ has a unique solution i.e,

$R = QP^*$



$q_0 = \epsilon + q_0 0$ — (1)

$q_1 = q_1 1 + q_0 1$ — (2)

$q_2 = q_1 0 + q_2 0 + q_2 1$ — (3)

$\underbrace{q_2}_{R} = \underbrace{q_1 0}_{Q} + \underbrace{q_2}_{R}\underbrace{(0+1)}_{P}$

$q_2 = q_1 0 (0+1)^*$ — (4) $[\because R = QP^*]$

Solving Eq (2)

$\underbrace{q_1}_{R} = \underbrace{q_0 1}_{Q} + \underbrace{q_1}_{R}\underbrace{1}_{P}$

$q_1 = q_0 1 (1)^*$ — (5) $[\because R = QP^*]$

Solving Eq (1)

$\underbrace{q_0}_{R} = \underbrace{\epsilon}_{Q} + \underbrace{q_0}_{R}\underbrace{0}_{P}$

$q_0 = \epsilon (0)^*$

$q_0 = 0^*$ — (6)

Substitute Eq (6) in (5)

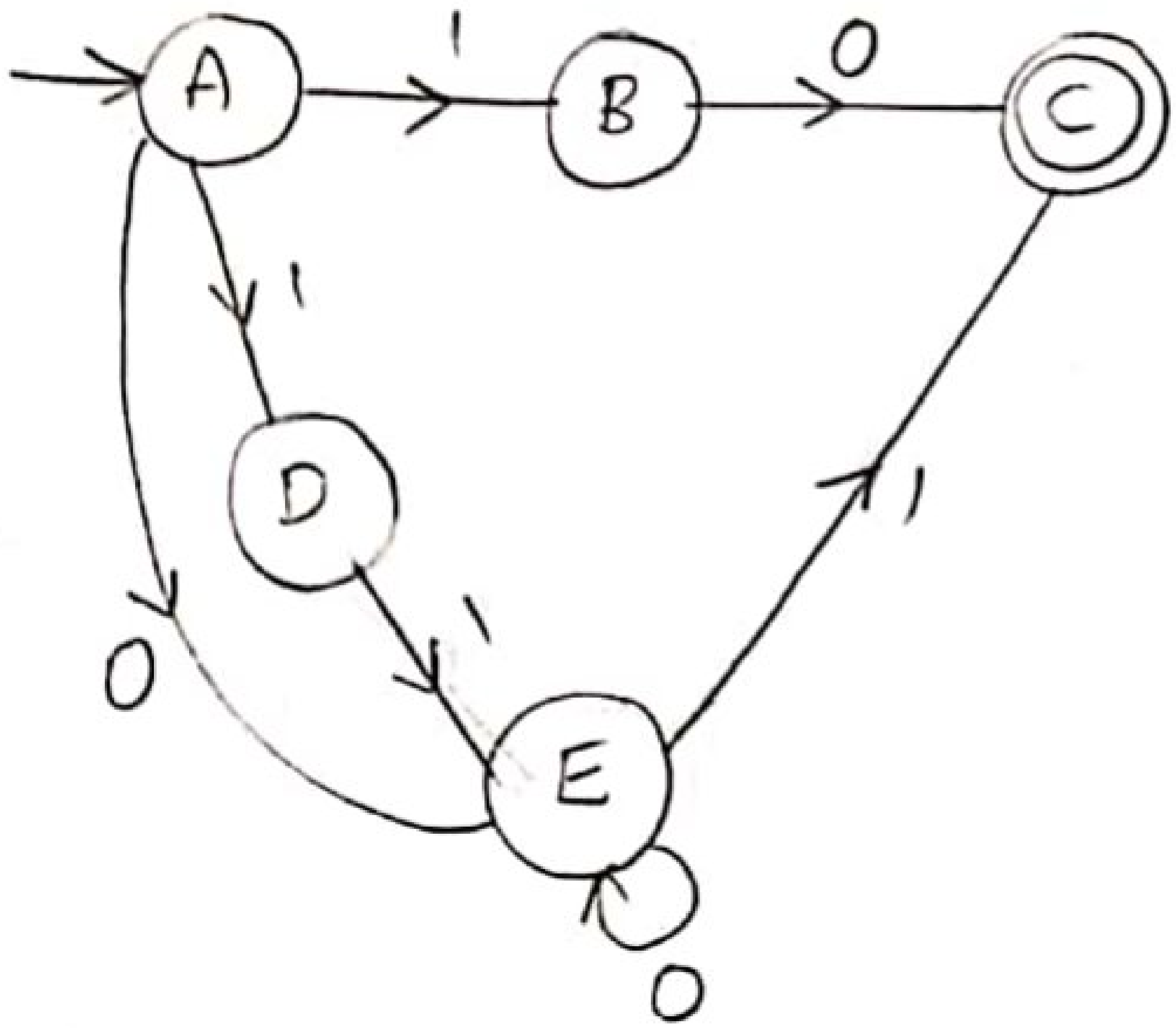
$q_1 = 0^* 1 1^*$ — (7)

Substitute Eq (7) in (4)

$\boxed{q_2 = 0^* 1 1^* 0 (0+1)^*}$

17 a) Design a FA from the given RE is $10+(0+11)0^*1$

A:



b) List out the closure properties of regular expression

A: Closure properties on regular languages are defined as certain operations on regular language which are guaranteed to produce regular language. Closure refers to some operations on a language, resulting in a new language that is of same "type" as originally operated on i.e, regular.

Scanned by CamScanner

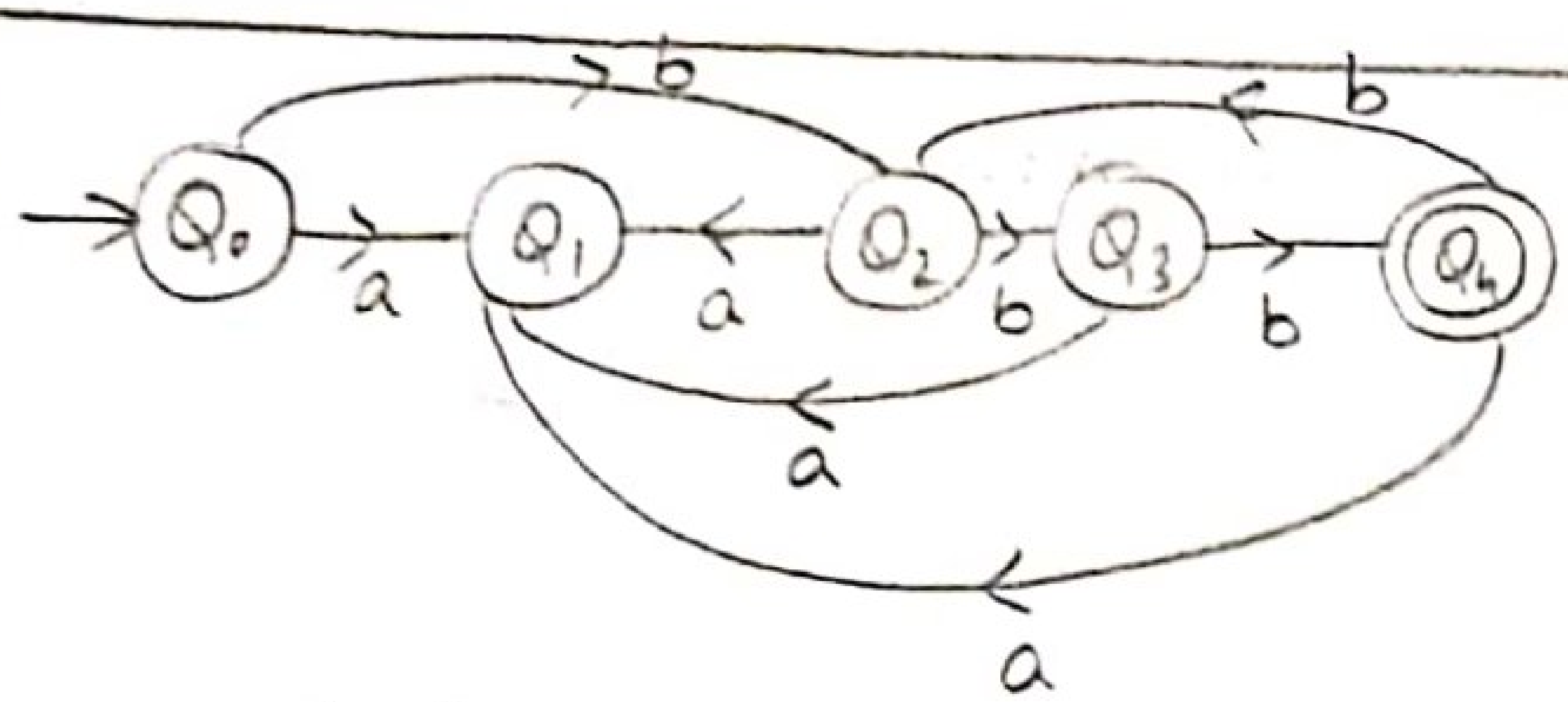
Regular languages are closed under following operations
1) Kleen closure
2) Positive closure
3) Complement
4) Reverse Operator
5) Union
6) Intersection
7) Set Difference operator
8) Homomorphism
9) Inverse Homomorphism

→ There are few more properties like symmetric difference operator, prefix operator, substitution which are closed under closure properties of regular language.

18. Construct the minimal DFA for the given DFA

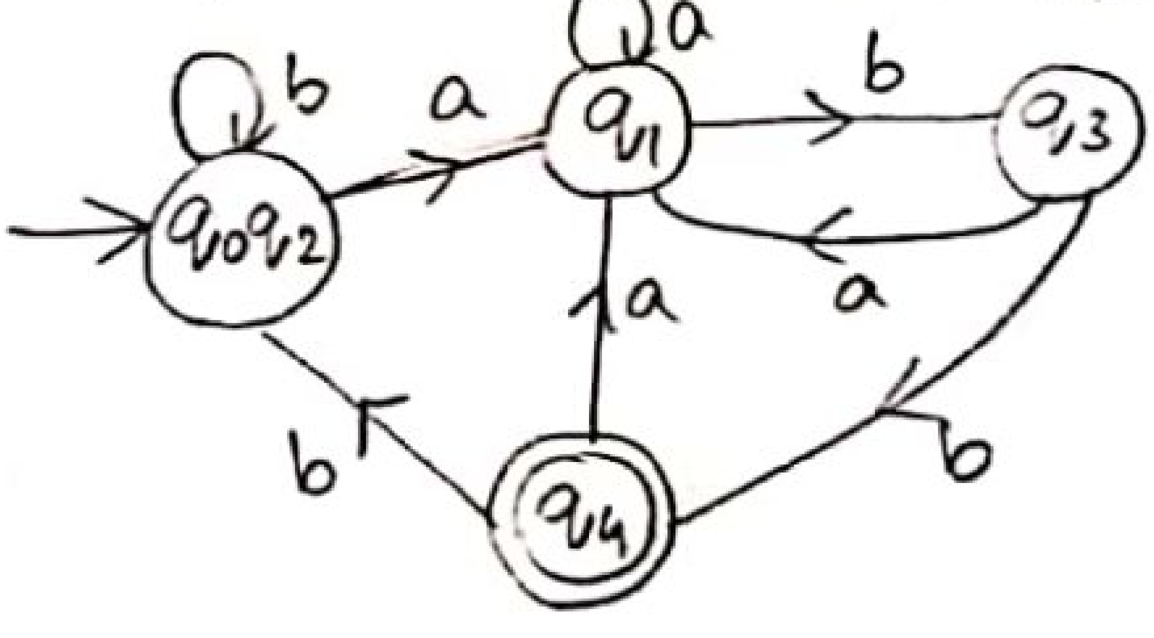
Q/Σ	a	b
→ Q_0	Q_1	Q_2
Q_1	Q_1	Q_3
Q_2	Q_1	Q_2
Q_3	Q_1	Q_4
Q_4^*	Q_1	Q_2

A:



$Q \setminus \Sigma$	a	b
$\rightarrow Q_0$	Q_1	Q_2
Q_1	Q_1	Q_3
Q_2	Q_1	Q_2
Q_3	Q_1	Q_4
Q_4^*	Q_1	Q_2

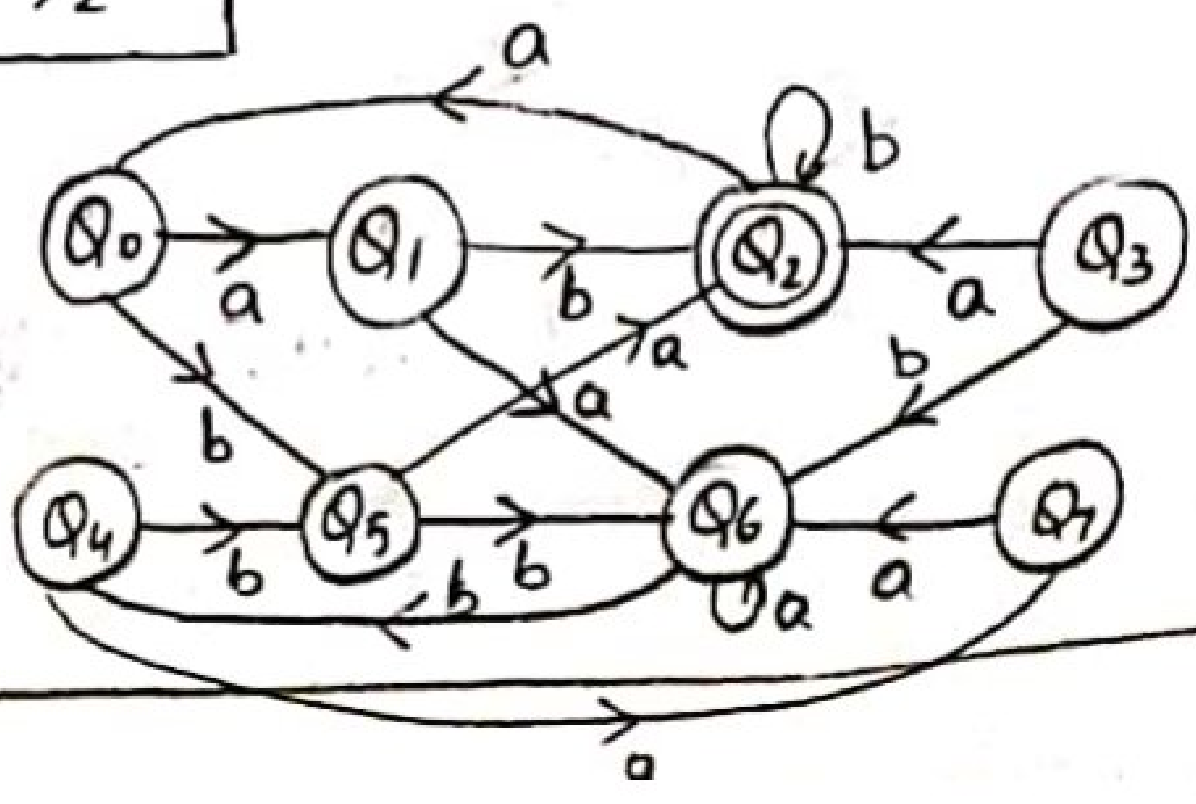
0 Equivalent $[q_0 q_1 q_2 q_3]$ $[q_4]$
1 Equivalent $[q_0 q_1 q_2]$ $[q_3]$ $[q_4]$
2 Equivalent $[q_0 q_1]$ $[q_2]$ $[q_3]$ $[q_4]$
3 Equivalent $[q_0 q_2]$ $[q_1]$ $[q_3]$ $[q_4]$



19) Construct the minimal DFA for the given DFA

$Q \backslash \Sigma$	a	b
$\rightarrow Q_0$	Q_1	Q_5
Q_1	Q_6	Q_2
Q_2^*	Q_0	Q_2
Q_3	Q_2	Q_6
Q_4	Q_7	Q_5
Q_5	Q_2	Q_6
Q_6	Q_6	Q_4
Q_7	Q_6	Q_2

A: Given DFA:



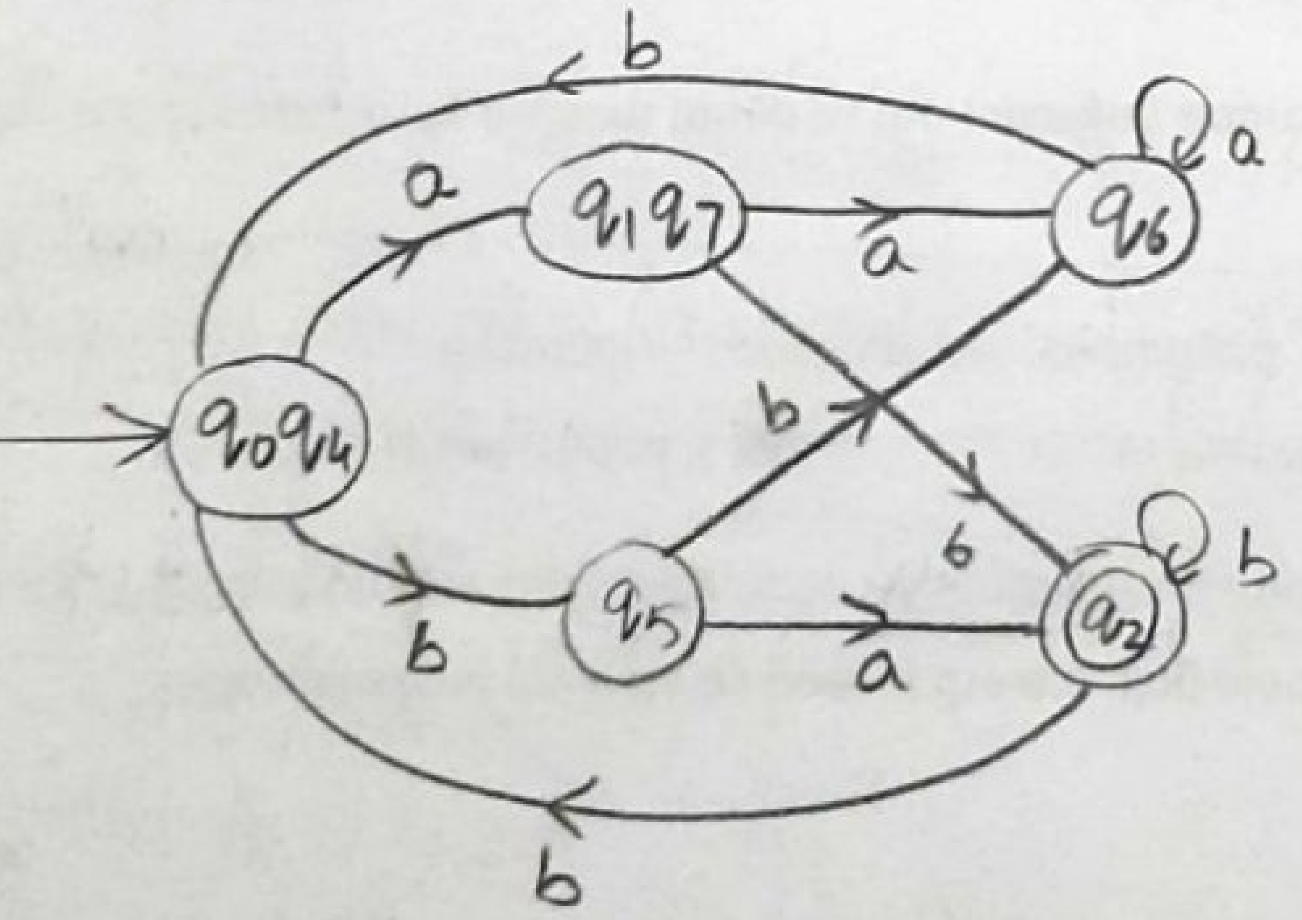
	a	b
→Q_0	Q_1	Q_5
Q_1	Q_6	Q_2
Q_2^*	Q_0	Q_2
Q_4	Q_7	Q_5
Q_5	Q_2	Q_6
Q_6	Q_6	Q_4
Q_7	Q_6	Q_2

0- equivalent $[q_0 q_1\ q_4 q_5 q_6 q_7]\ [q_2]$

1 equivalent $[q_0 q_4 q_6]\ [q_1 q_7]\ [q_5]\ [q_2]$

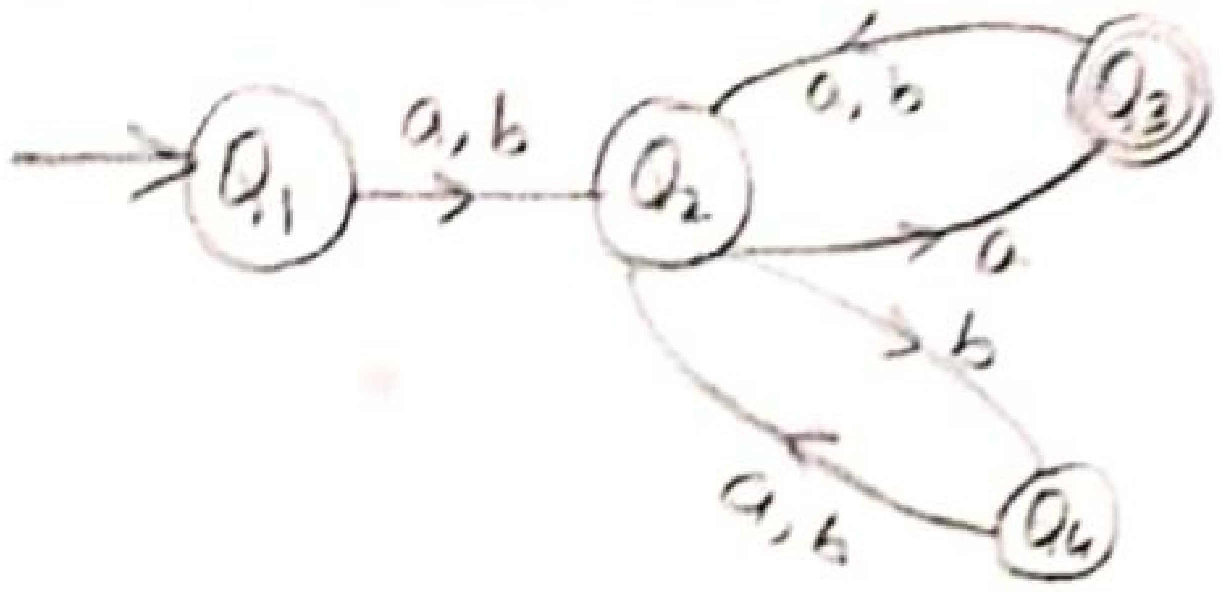
2 equivalent $[q_0 q_4]\ [q_6]\ [q_1 q_7]\ [q_5]\ [q_2]$

3 equivalent $[q_0 q_4]\ [q_6]\ [q_1 q_7]\ [q_5]\ [q_2]$



20 Construct the Regular Expression for the given FA

Q \ Σ	a	b
Q_1	Q_2	Q_2
Q_2	Q_3	Q_4
Q_3	Q_2	Q_2
Q_4	Q_2	Q_2



$Q_1 = \epsilon$ ——— (1)

$Q_2 = aQ_1 + bQ_1 + aQ_3 + bQ_3 + (a+b)Q_4$ ——— (2)

$Q_3 = aQ_2$ ——— (3)

$Q_4 = bQ_2$ ——— (4)

$Q_2 = (a+b)Q_1 + (a+b)Q_3 + bQ_4$

Substitute Q_1, Q_3 & Q_4 in Q_2

$Q_2 = (a+b)Q_1 + (a+b)aQ_2 + bQ_2(a+b)$

$Q_2 = (a+b)\epsilon + (a+b)a\,Q_2 + Q_2 b(a+b)$

$Q_2 = (a+b) + Q_2[a(a+b) + b(a+b)]$

$$\boxed{Q_2 = (a+b)[a(a+b) + b(a+b)]^*}$$