

M:

Define alphabet & language with an example

Alphabet is define as finite, non empty set of symbols.
denoted by symbol Σ

Ex: $w = abab$, $\Sigma = \{a, b\}$

$w = ab$

$w = a$

$L = \{a, b, bb, ba, \dots\}$

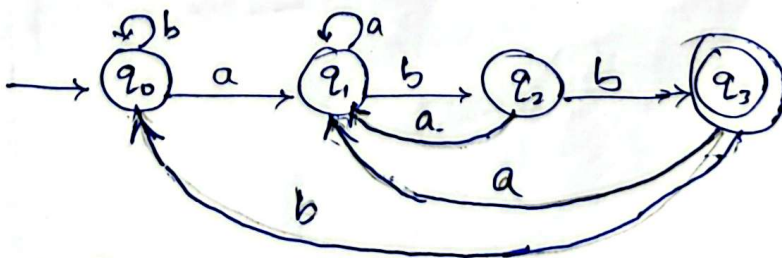
→ A language can be define as set of string obtain from Σ^* where Σ is set of alphabet of a Particular lang, formally a lang L over Σ is subset of Σ^* which is denoted by $L \subseteq \Sigma^*$

Ex: $L = \{\epsilon, 01, 0011\}$

Lang of string consisting of n no. of 0's followed by n no. of 1's

2. Draw a DFA to accept string of a's & b's ending with abb

→ $\Sigma = \{a, b\}$
 $L = \{abb, aabb, babb\}$

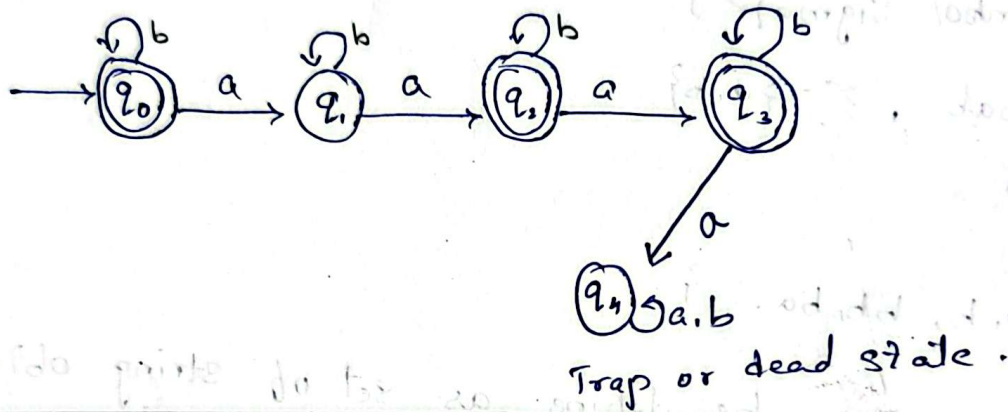


$F = \{q_3\}$

3. Draw a DFA to accept the strings of a's & b's having not more than 3 a's

Ans. $\Sigma = \{a, b\}$

$L = \{a, ba, aa, aaa\}$



4. Obtain regular expression to accept strings of a's & b's such that every block of four consecutive symbols contains at least two a's.

Ans. $\Sigma = \{a, b\}$

$L = \{abab, aaab, aaaa, \dots\}$

$RE = \{(aa(a+b)(a+b)) + ((a+b)(a+b)aa) + (a(a+b)a(a+b))\}$

$RE = ((aa(a+b)(a+b)) + ((a+b)aa(a+b) + (a+b)(a+b)aa) + (a(a+b)a(a+b)) + (a(a+b)(a+b)a) + ((a+b)a(a+b)a))$

5. Define Regular Language.

Ans. If a language is accepted by DFA's (Deterministic Finite Automata) NFA's & ϵ -NFA's then that language is called as Regular Language.

Regular lang can also be represented by using Regular Expression.

6. Obtain Regular Expression for the following.

a. Strings of a's & b's ending with b & has no substring aa.

Ans. $\Sigma = \{a, b\}$

$L = \{b, ab, abab, bab, \dots\}$

RE = $(b+ab)^+$

b. String of 0's & 1's having no 2 consecutive 0's

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

$L = \{1, 01, 10, 111, 010, \dots\}$

RE = $\{(1+01)^*(0+1)\}$

7. Obtain RE for the String Beginning & ending with same letters over $\Sigma = \{a, b\}$

Ans. $\Sigma = \{a, b\}$

$L = \{abba, aba, bab, ababa, \dots\}$

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

8. Obtain RE for the string of a's & b's whose length is either even ^{or} multiple of 3

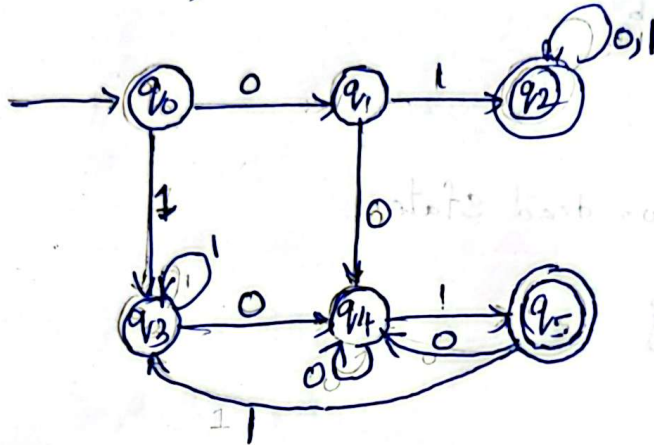
Ans. $\Sigma = \{a, b\}$

$L = \{\epsilon, aa, abbb, \dots\}$

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

Draw a DFA to accept set of all strings on alphabet $\Sigma = \{0,1\}$ that either begins or ends or both with Substring 01.

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



$M = \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$

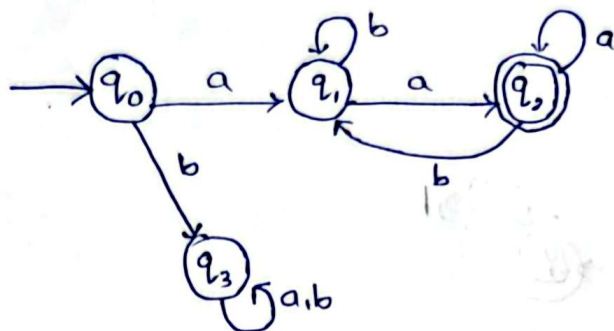
$\Sigma = \{0,1\}$

δ	0	1
q_0	q_1	q_3
q_1	q_4	q_2
q_2	q_2	q_2
q_3	q_4	q_3
q_4	q_4	q_5
q_5	q_4	q_3

q_0 is start state

F is $\{q_2, q_5\}$

3. Design a DFA for the language $L = \{awa \mid w \in (a+b)^n\}$ where $n \geq 0$.



Trap state or dead state

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

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

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

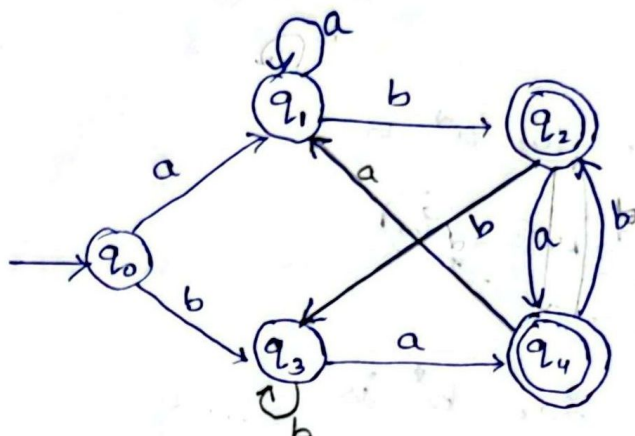
δ	a	b
$\rightarrow q_0$	q_1	ϕ
q_1	q_2	q_1
$* q_2$	q_2	q_1

Where q_0 is start state

$$F = \{q_2\}$$

4. Draw a DFA to accept string of a's & b's ending with ab or ba

Ans -



$$M = \{Q, \Sigma, \delta, q_0, F\}$$

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

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

δ	a	b
$\rightarrow q_0$	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_4
q_3	q_4	q_3
$*q_4$	q_2	q_4

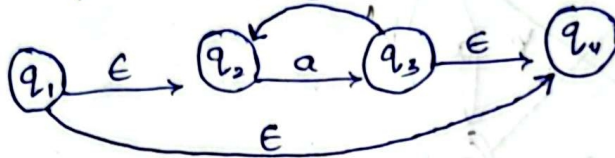
q_0 is start state

$$F = \{q_2, q_4\}$$

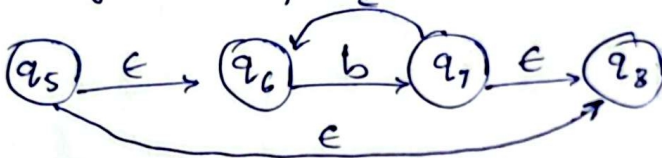
6. Obtain

5. Obtain ϵ -NFA for the Regular Expression $a^* + b^* + c^*$.

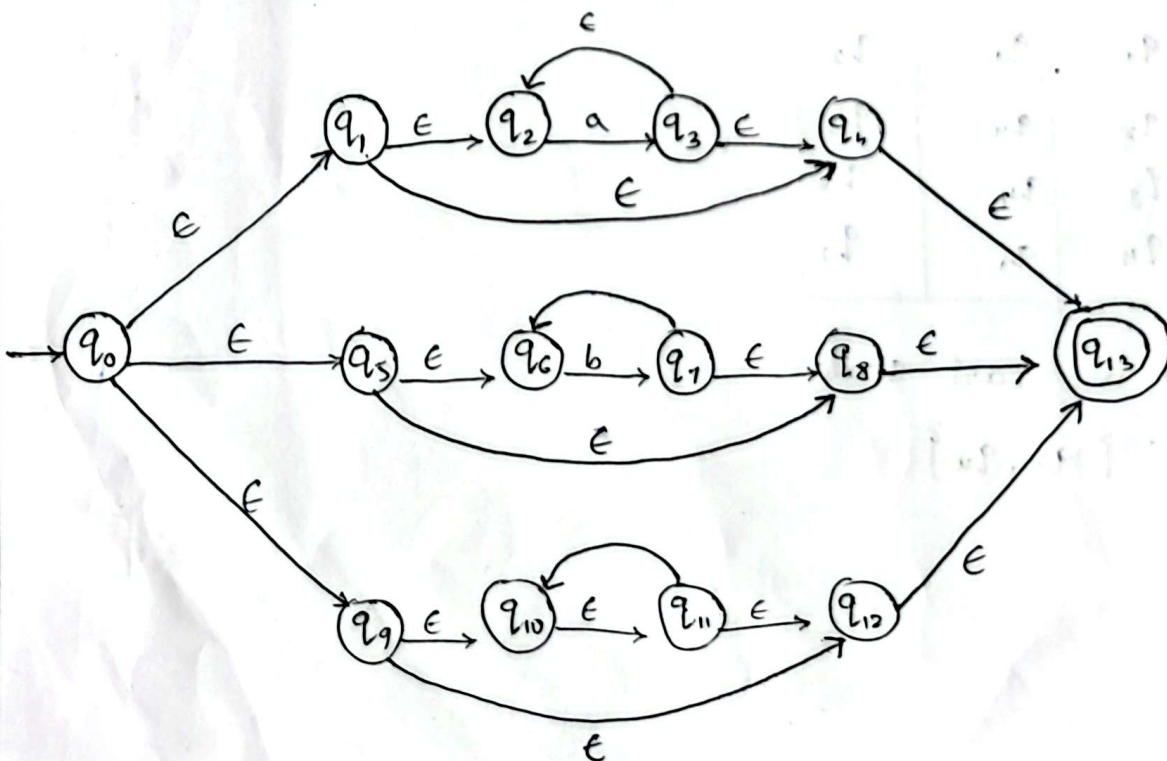
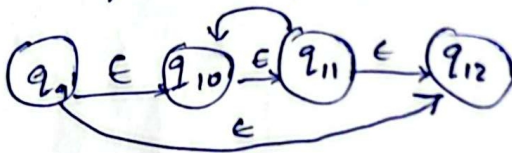
Ans. The regular expression for a^* is



The regular expression for b^* is



The regular expression for c^* is



6. Define regular expression. Obtain regular expression for the language containing at least one a , ϵ at least one b where $\Sigma = \{a, b, c\}$

Ans. A RE is recursively defined as.

1. ϕ : is a RE denoting an empty language
2. ϵ : is a RE " a language containing empty string.
3. a : is a RE that denotes the language containing only a
4. R : is a RE denoting the language LR is $\&\&\&\&$
 S is a RE " " " " LS then
 - a. $R+S$ is a RE denoting the language $LRULS$
be RS be the RE denoted by language LR, LS
 - b. R^* be the RE corresponding to the lang LR^*
 - c. The expressions obtain by applying an of the rules from 1 to 4 are Regular Expression



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

$$L = \{ab, ba, \dots\}$$

$$R_E = (a+b+c)^* a (a+b+c)^* b (a+b+c)^* + (a+b+c)^* b (a+b+c)^* a (a+b+c)^*$$

② Define the following with an example:

i. String

Ans. Seq. of Symbols obtained from Alphabet is called String.

String is defined as finite sequence of symbols from the alphabet Σ

Ex: $\Sigma = \{a, b\}$

$$W = abab$$

$$W = ab$$

$$W = a$$

$$L = \{a, b, bb, ba, \dots\}$$

ii. String length:

the length of a string w denoted by $|w|$ is number of symbols in w

Ex: $w = abbab$

$$|w| = 5$$

iii Prefix of string:

- A substring x of a string xy is called Prefix always occurs from beginning of the string

Ex: For string HELLO the Prefixes are

HELLO: $\epsilon, H, HE, HEL, HELL, HELLO$

iv Suffix of string:

- A substring x of a string yx is called Suffix. the suffix always occurs at the end of string

Ex: For the string HELLO suffixes are:

ϵ HELLO: $\epsilon, O, LO, LLO, ELLO, HELLO$

v. String Reversal:

- The reversal of a string w denoted by w^R is obtained by writing the symbols of the string w in reversal order

Ex: $w = ab$

$w^R = ba$

10M:

Define

DFA : The DFA is

The deterministic finite Automata has 5 types including 5 components that is

$$M = \{Q, \Sigma, \delta, q_0, F\}$$

M is name of the machine

Q is non-empty finite set of states

Σ is non-empty finite set of input alphabets

δ is a transition function mapping from

$Q \times \Sigma$ to Q denoted by

$$\delta: Q \times \Sigma \rightarrow Q$$

$q_0 \in Q$ is the start state of the Machine

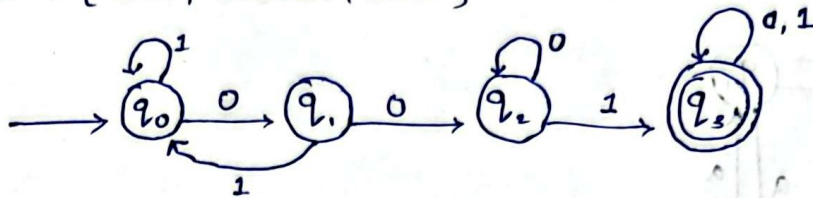
$F \subseteq Q$ is set of final states

Define DFA & Draw the DFA to accept the following language.

a. $L = \{w \in \{0,1\}^* : w \text{ has } 001 \text{ as substring}\}$

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

$L = \{001, 010101, 0001\}$



$M = \{Q, \Sigma, \delta, q_0, F\}$

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

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

δ	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_3
$* q_3$	q_3	q_3

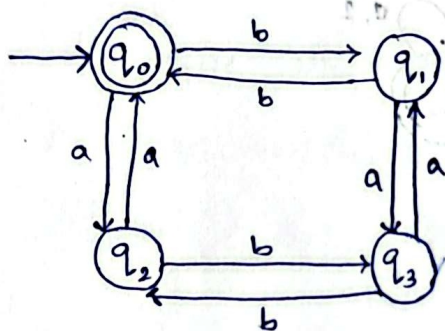
q_0 is start state

F is $\{q_3\}$ final state

b. $L = \{w \in \{0,1\}^* : w \text{ has even number of a's \& even no. of b's}\}$

14. $\Sigma = \{a, b\}$

$L = \{aabb, aaaa, baba, abab\}$



$M = \{Q, \Sigma, \delta, q_0, F\}$

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

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

$\delta =$

	δ	a	b
$\xrightarrow{*}$	q_0	q_2	q_1
	q_1	q_3	q_0
	q_2	q_0	q_3
	q_3	q_1	q_2

$F = \{q_0\}$

4. Convert the following ϵ -NFA to DFA



Ans. Let us obtain ϵ -closure of each state

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

$$\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$$

$$\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$$

$$\epsilon\text{-closure}(q_2) = \{q_2\}$$

Transitions:

Consider the state $\{q_0, q_1, q_2\}$

$$\text{if } p = a \quad \delta(\{q_0, q_1, q_2\}, a) = \epsilon\text{-closure}\{\delta(\{q_0, q_1, q_2\})a\}$$

$$= \epsilon\text{-closure}\{\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a)\}$$

$$= \epsilon\text{-closure}\{(q_0 \cup \phi \cup \phi)\}$$

$$= \epsilon\text{-closure}\{q_0\}$$

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

$$\text{if } p = b \quad \delta(\{q_0, q_1, q_2\}, b) = \epsilon\text{-closure}\{\delta(\{q_0, q_1, q_2\})b\}$$

$$= \epsilon\text{-closure}\{\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b)\}$$

$$\epsilon\text{-closure}\{\phi \cup q_1 \cup \phi\}$$

$$\epsilon\text{-closure}\{q_1\}$$

$$= \{q_1, q_2\}$$

if $P = c$ $\delta(\{q_0, q_1, q_2\}c) = \epsilon\text{-closure}\{\delta(q_0, q_1, q_2)c\}$
 $\epsilon\text{-closure}\{\delta(q_0, c) \cup \delta(q_1, c) \cup \delta(q_2, c)\} = \delta(q_2, c)$
 $\epsilon\text{-closure}\{\emptyset \cup \emptyset \cup q_2\}$
 $\epsilon\text{-closure}\{q_2\}$
 $= q_2$

Consider the state $\{q_1, q_2\}$

if $P = a$ $\delta(\{q_1, q_2\}a) = \epsilon\text{-closure}\delta(\{q_1, q_2\}a)$

$\epsilon\text{-closure}\{\delta(q_1, a) \cup \delta(q_2, a)\}$

$\epsilon\text{-closure}\{\emptyset \cup \emptyset\}$
 $= \emptyset$

if $P = b$ $\delta(\{q_1, q_2\}b) = \epsilon\text{-closure}\delta(\{q_1, q_2\}b)$

$= \epsilon\text{-closure}\{\delta(q_1, b) \cup \delta(q_2, b)\}$

$= \epsilon\text{-closure}\{q_1 \cup \emptyset\}$

$= \epsilon\text{-closure}\{q_1\}$

$= \{q_1, q_2\}$

if $P = c$ $\delta(\{q_1, q_2\}c) = \epsilon\text{-closure}\delta(\{q_1, q_2\}c)$

$\epsilon\text{-closure}\{\delta(q_1, c) \cup \delta(q_2, c)\}$

$\epsilon\text{-closure}\{\emptyset \cup q_2\}$

$\epsilon\text{-closure}\{q_2\}$

$= \{q_2\}$

Consider the state $\{q_2\}$

i/p = a $\delta(q_2)a$ ϵ -closure $\delta(q_2)a$

ϵ -closure $\{\delta(q_2a)\}$

ϵ -closure $\{\emptyset\}$

= $\emptyset$

i/p = b $\delta(q_2)b$ ϵ -closure $\delta(q_2)b$

ϵ -closure $\{\delta(q_2b)\}$

ϵ -closure $\{\emptyset\}$

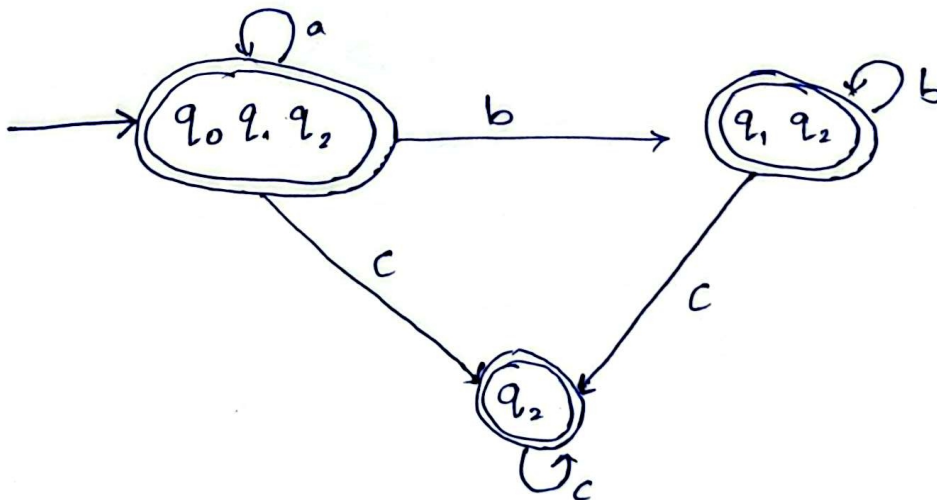
= $\emptyset$

i/p = c $\delta(q_2)c$ ϵ -closure $\delta(q_2)c$

ϵ -closure $\{\delta(q_2c)\}$

ϵ -closure $\{q_2\}$

= q_2



3. Minimize the state of following DFA:

δ	a	b
$\rightarrow A$	B	F
B	G	C
* C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

Ans. Partition of state $\{A, B, C, D, E, F, G, H\}$ into 2 grp considering non final state & final state as shown

$$\pi_0 = \{ \overset{G_1}{[A, B, D, E, F, G, H]}, \overset{G_2}{[C]} \}$$

a

From the grp G_1 , D & F on input symbol A entering into G_2 so make them separate new group the group is π_1 .

$$\pi_1 = \{ \overset{G_1}{[A, B, E, G, H]}, \overset{G_2}{[D, F]}, \overset{G_3}{[C]} \}$$

b

from the grp G_1 , B & H on input symbol B entering into G_3 so make them as separate grp the resulting grp is π_2

$$\pi_2 = \{ [A, E, H], [B, H], [D, F], [C] \}$$

$\begin{array}{c} G_1 \quad G_2 \quad G_3 \quad G_4 \\ \begin{array}{c} \text{---} \text{---} \text{---} \text{---} \\ | \quad | \quad | \\ \text{a} \end{array} \end{array}$

From the grp G_1 , A, E on input a entering into G_2
 So make them as separate grp the resulting grp is

π_3

$$\pi_3 = \{ [A, E], [B, H], [D, F], [G], [C] \}$$

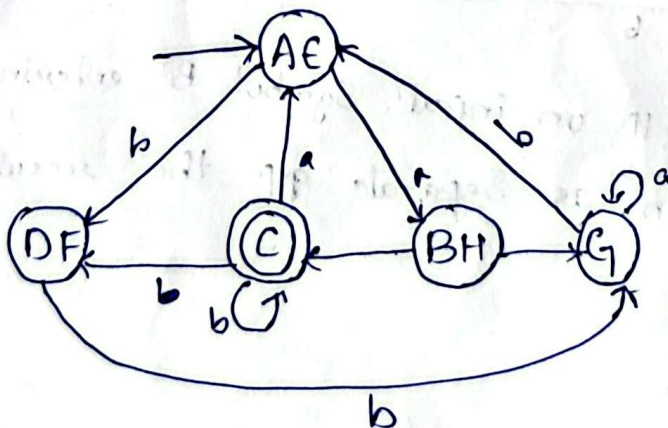
The final Minimize state of DFA are

$$\{ [A, E], [B, H], [D, F], [G], [C] \}$$

Transition Table:

δ	a	b
[A, E]	[B, H]	[D, F]
[B, H]	[G]	[C]
[D, F]	[C]	[G]
[G]	[G]	[A, E]
[C]	[A, E]	[C]

Final Minimize DFA is:

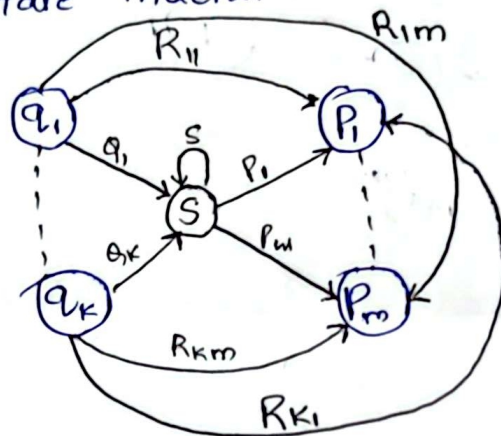


4.a. Explain state elimination method for conversion of finite automata to regular expression.

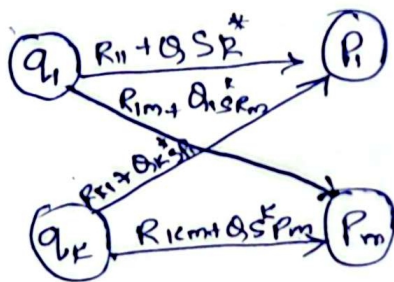
Ans. There are 2 Methods to obtain regular expression to finite automata.

1. State Elimination Method.
2. Kleene's theorem.

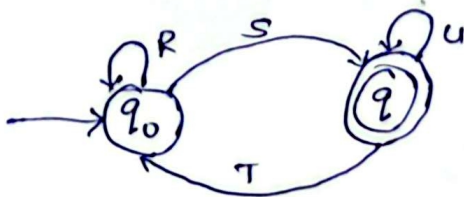
1. SE: for each accepting state q , applying the reduction state process to produce an equivalent finite state machine.



After elimination of state S

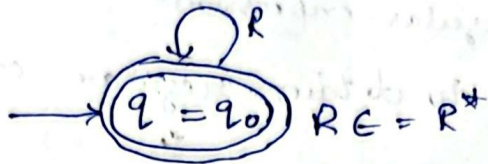


Case 2: $q \neq q_0$ (start state & final state is not same)



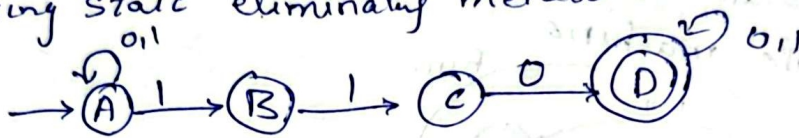
$$RE = (R + Su^*T)^* Su^*$$

Case 3: $q = q_0$ (if start state & final state is same)

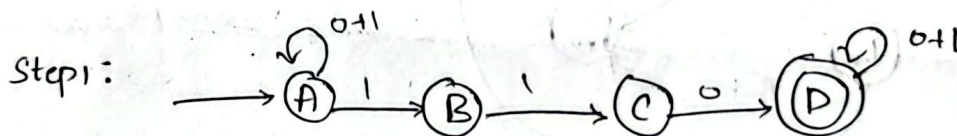


The derived RE is the sum of all the expression derived from the reduce finite automata for each accepting state

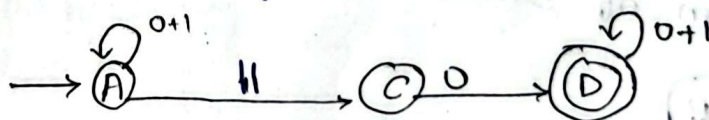
b. Obtain RE for the finite automata shown below using state eliminating method.



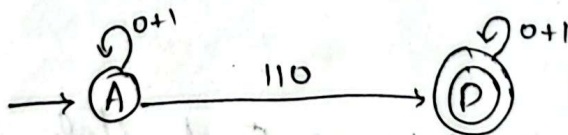
Ans. $RE = (0+1)^* 110(0+1)^*$



After removing state B



After removing state C



Final RE will be given by

$$RE = (0+1)^* 110 (0+1)^*$$