MODULE-1

(6-8)m.

Central concepts of Automata theory:-

i) **Symbol:**

→ Basic building block.

→ It can be anything like character, number, picture etc.

Ex:- a, b, c, 1, 2, 3, O, Δ, ----

ii) **Alphabet set:-**

→ It is a finite collection of symbol.

→ It is represented by "Σ"

Ex:- $\Sigma = \{a, b\}$, $\Sigma = \{0, 1\}$

iii) **String:-**

→ It is a finite sequence of symbols from a given alphabet set.

→ It is represented by 'w'

Ex:- If $\Sigma = \{a, b\}$, then

$w = a, ab, ba, aab, -----$

iv) **Language:-**

→ It is a collection of strings.

→ It is represented by 'L'

Ex:- If $\Sigma = \{a, b\}$, then

$L = \{a, b, ab, aab, ------\}$

v) **Kleene star:-**

→ It is a unary operator on a given alphabet set that gives strings of all possible length.

→ Represented by Σ^*

$$\boxed{\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots \cup \Sigma^n}$$

Σ^0 → Strings of length 0 → Represented by ε → epsilon.

Σ^1 → length 1

Σ^n → length n.

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

then, $\Sigma^* = \{\varepsilon, a, b, ab, ba, \ldots\}$

vi) Kleene closure / plus:-

→ It is a unary operator on a given alphabet set that gives strings of all possible length except 'ε'.

→ Represented by Σ^+

$$\boxed{\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots \cup \Sigma^n}$$

Ex:- given, $\Sigma = \{a, b\}$, then

$\Sigma^+ = \{a, b, ab, ba, \ldots\}$.

Operations on strings:-

i) Concatenation of strings:-

→ Let us consider two strings 'w_1' & 'w_2', then the concatenation of two strings is given by $w_1.w_2$ is obtained by appending w_2 to the end of w_1. "." (Dot operator) is used for concatenation operation.

Ex:- $w_1 = aba$

$w_2 = bba$

$w_1.w_2 = ababba$

$w_2.w_1 = bbaaba$

ii) **Reverse of string:-**

Let us consider a string 'w'. The reverse of string w is given by w^R which is obtained by writing the string in reverse order.

Ex:- $w = abb \Rightarrow w^R = bba$

$w = 01101 \Rightarrow w^R = 10110$

iii) **Palindrome of string:-**

Let us consider a string w. Palindrome of string is obtained by concatenating the reverse of the string with original order.

Ex:- $w = 011$, $w^R = 110$ →

$w.w^R = 011110$

iv) **Prefix of string:-**

Set of all possible leading symbols.

Ex:- $w = aba$

$\{\epsilon, a; ab, aba\}$

v) **Suffix of string:-**

Set of all trailing symbols.

Ex:- $w = aba$

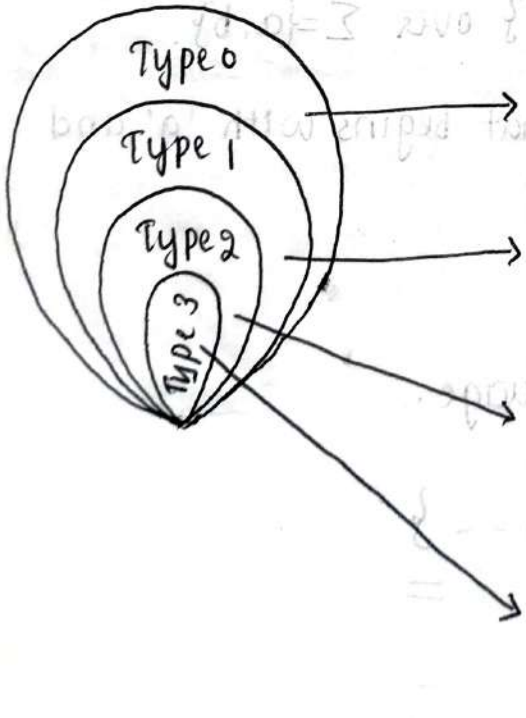
$\{\epsilon, a, ba, aba\}$

vi) **Length of string:-**

Let us consider a string w, then the length is obtained by counting no of symbols in that string (w) and it is denoted by $|w|$.

Ex:- $w = 010101 \Rightarrow |w| = 6$.

Chomsky Hierarchy:-



Grammer	Language	Machine.
Unrestricted	Recursively Enumerable	Turing machine.
Context sensitive	context sensitive	Linear bounded automata (LBA)
Context free (CFG)	Context free (CFL)	push down automata (PDA)
Restricted (regular)	Regular language (RL)	finite automata.

Q) Identify the language for the following:- 24/09/2024

1) $L = \{w : w \in \underbrace{(a+b)^*}_{} \}$ → regular expression.

Soln:-

$w \leftarrow$ It is made up of a's and b's

* There is no restriction on length

* Epsilon belongs to this language

"String of minimum length, belongs to the language"

$L = \{\varepsilon, a, b, ab, ba, abbab, \text{------}\}$

---

2) $L = \{w : w$ ~~belongs~~ begins with $b\}$ over $\Sigma = \{a, b\}$

Soln:-

$w \leftarrow$ String made up of a's and b's that starts with 'b'

* There is no restriction on length

* Epsilon does not belongs to the language.

$L = \{b, ba, bba, bab, babbab, \text{-------}\}$

---

3) $L = \{w : w$ begins with a & ends with $b\}$ over $\Sigma = \{a, b\}$.

Soln:- $w \leftarrow$ String made up of a's and b's that begins with 'a' and ends with 'b'.

* There is no restriction on length

* Epsilon does not belong to the language.

$L = \{ab, aab, abb, aaab, ababab, \text{-----}\}$

4) $L = \{w : w \in (0+1)^*$ and $|w| \bmod 2 = 0\}$.

Sol:- $w \leftarrow$ It is made up of 0's and 1's

* There is no restriction on length
* Epsilon belongs to the language

$L = \{\varepsilon, 10, 1000, 1100, 1110, 101010, \text{------}\}$

5) $L = \{w : n_a(w) = n_b(w)\}$ over $\Sigma = \{a, b\}$

Sol:- $w \leftarrow$ made up of a's and b's

* No restriction on length
* Epsilon belongs to the lang.

$L = \{\varepsilon, ab, aabb, aaabbb, \text{------}\}$

ε ↓ because no of a's and b's are 0.

6) $L = \{w : n_a(w) \bmod 3 = n_b(w) \bmod 2 = 0\}$ over $\Sigma = \{a, b\}$

Soln:- $w \leftarrow$ made up of a's and b's.

* No restriction on length
* $\varepsilon \in w$.

$L = \{\varepsilon, aaabb, ababaaabab, \text{------}\}$

(or)

$L = \{\varepsilon, bb, aaa, aaabb, abababaaab, \text{------}\}$

Finite Automata:-

(Deterministic finite automata)

(DFSM) DFA

(Deterministic finite state machine)

Finite Automata

(FA)

→ Abstract machine

(NDFSM) NFA

(Non-deterministic)

* Finite inputs
* Finite number of states
* Finite output

Representation of finite automata:-

25/09/2024

Definition: (Mathematical model)

M = 5 tuple representation.

$(Q, \Sigma, q_0, \delta, F)$

where $Q \rightarrow$ Finite set of all states. $\{q_0, q_1, q_2, \ldots q_n\}$.

$\Sigma \rightarrow$ alphabet set.

$q_0 \rightarrow$ start state

$\delta \rightarrow$ transition function

$F \rightarrow$ set of all finite states. [and $F \in Q$]

(determines the automata that is DFA / NFA)

Deterministic finite automata (DFA):-

A DFA is a 5 tuple representation given by

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

where $M \leftarrow$ Required DFA

$Q \leftarrow$ finite set of all states.

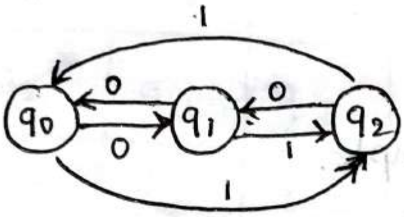
$\Sigma \leftarrow$ alphabet set

$q_0 \leftarrow$ start state, $q_0 \in Q$

$\delta \leftarrow$ transition function, $\boxed{\delta \leftarrow Q \times \Sigma \rightarrow Q}$

$F \leftarrow$ finite set of all final states.

Ex:-



$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_2$

Representation of DFA:-

3 types to represent DFA

(i) State transitions diagram

(ii) Transition function

(iii) Transition Table.

Notations:-

State :

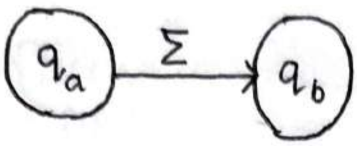
Start state :

Final state :

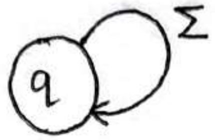
Dead state :

Note:- <u>Dead State</u>:- "State from which we are not reaching final state"

Transition:

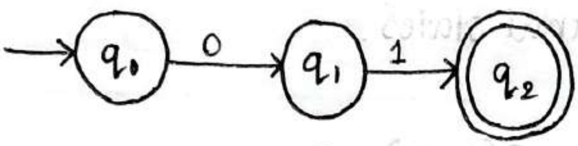


Self transition:

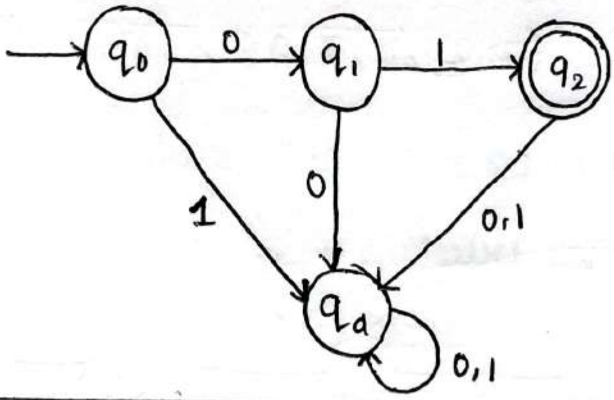


Ex:- Write a FA to accept the string 01 over $\Sigma = \{0, 1\}$

Soln:- Given, $L = \{01\}$



Note:- DFA representation of same example.



28/09/2024

Transition function (δ) for above DFA:-

$M_{DFA} = (Q, \Sigma, q_0, \delta, F)$

$Q : \{q_0, q_1, q_2, q_d\}$

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

$q_0 : q_0$

$F : \{q_2\}$

δ: $\delta(q_0, 0) = q_1$ $\delta(q_2, 0) = q_d$

$\delta(q_0, 1) = q_d$ $\delta(q_2, 1) = q_d$

$\delta(q_1, 0) = q_d$ $\delta(q_d, 0) = q_d$

$\delta(q_1, 1) = q_2$ $\delta(q_d, 1) = q_d$

Transition table (for same):-

$q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_1	q_d
q_1	q_d	q_2
q_2^*	q_d	q_d
q_d	q_d	q_d

represents start state

represents final state.

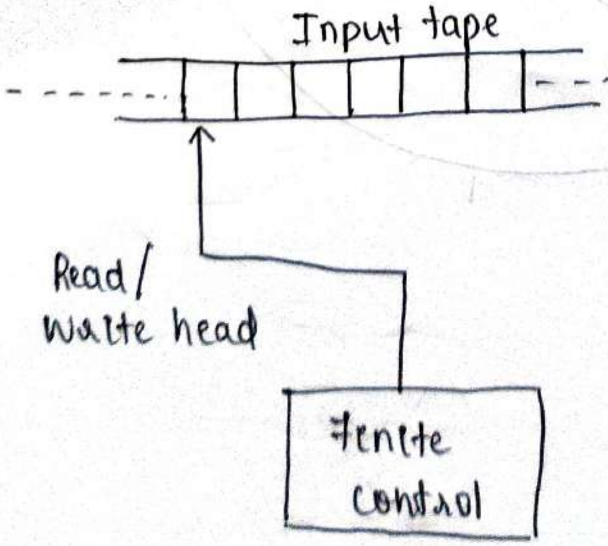
Working principle of finite automata:-

It works on 3 components:

1) Input Tape

2) Read (or) write head

3) Finite control.



Input tape:-

* Input tape is an infinite length tape consisting of number of cells.
* Each cell contains a single symbol of the string given.

Read (or) write head:-

* Read head basically reads the data from the input tape whereas write head writes the data into the input tape.
* It points to the starting symbol of the cell and the movement is from left to right.

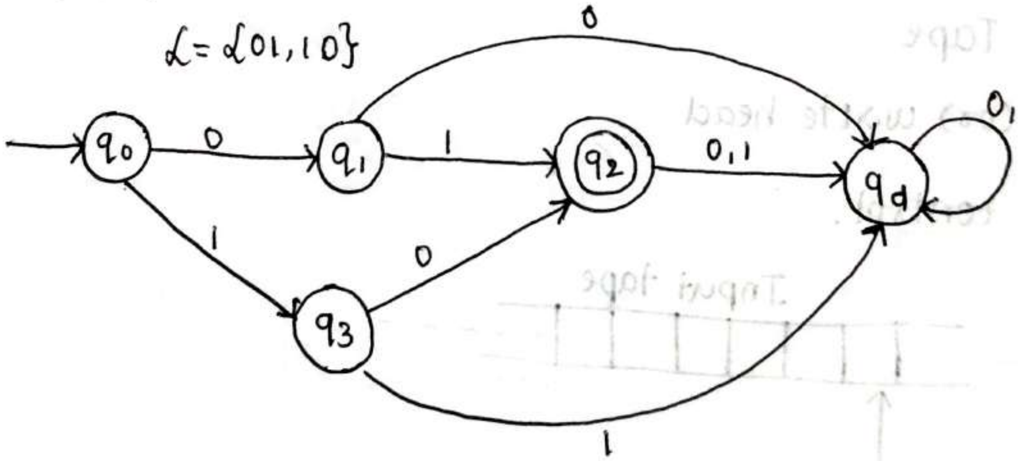
Finite control:-

* It basically controls the movement of read/write head
* It consists of the transition function.

(1)
Q) Construct a DFA to accept the language having a string either either 01 (or) 10.

Soln:-

$L = \{01, 10\}$



$Q : \{q_0, q_1, q_2, q_3, q_d\}$

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

$q_0 : q_0$

$F : \{q_2\}$

δ: transition function.

$\delta(q_0, 0) = q_1$

$\delta(q_0, 1) = q_3$

$\delta(q_1, 0) = q_d$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_d$

$\delta(q_2, 1) = q_d$

$\delta(q_3, 0) = q_2$

$\delta(q_3, 1) = q_d$

$\delta(q_d, 0) = q_d$

$\delta(q_d, 1) = q_d$

(or)

transition table.

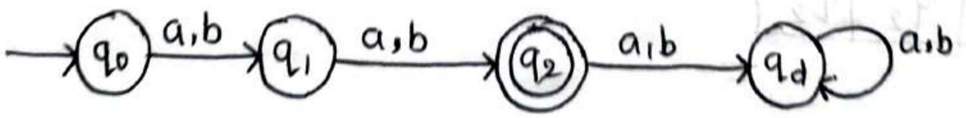
$q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_1	q_3
q_1	q_d	q_2
q_2^*	q_d	q_d
q_3	q_2	q_d
q_d	q_d	q_d

30/09/2024

(2)

Q) Construct DFA to accept all strings of length equal to 2 over $\{a, b\}$.

Sol: $L = \{aa, ab, ba, bb\}$



$Q: \{q_0, q_1, q_2, q_d\}$

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

$q_0 : q_0$

$F : \{q_2\}$

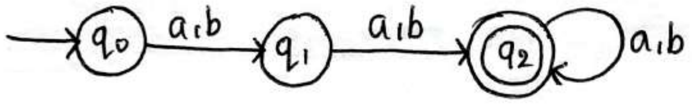
δ: transition table

$q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2^*	q_d	q_d
q_d	q_d	q_d

(3)

Q) DFA to accept all strings ≥ 2 over $\{a, b\}$.

Soln:- $L = \{aa, ab, ba, bb, aaa, aab, aba, aabbab, \ldots\}$



$Q : \{q_0, q_1, q_2\}$

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

$q_0 : q_0$

$F : \{q_2\}$

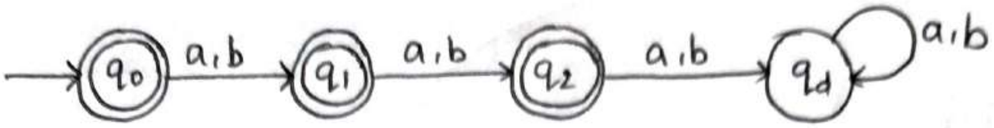
δ: transition table.

$q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_1	q_1
q_1	q_2	q_2
q_2^*	q_2	q_2

Note:- ε : q {start state itself is a final state}.

(4)
Q) DFA to accept all strings ≤ 2 over $\{a, b\}$.

Soln:- $L = \{\varepsilon, a, b, aa, ab, ba, bb\}$



$Q : \{q_0, q_1, q_2, q_d\}$

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

$q_0 : q_0$

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

δ = transition table.

$q \backslash \Sigma$	a	b
$\rightarrow q_0^*$	q_1	q_1
q_1^*	q_2	q_2
q_2^*	q_d	q_d
q_d	q_d	q_d

Note:- If 'n' is the minimum no of strings, then $(n+1)$ states are required to construct the machine.

* If length of string $\leq$ no || exactly = no $\Rightarrow$ dead state required.

else : no need.

(5)
Q) Write a DFA to accept all strings of length atmost 3 over $\{0, 1\}$

Soln:- $L = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111\}$

$Q: \{q_0, q_1, q_2, q_3, q_d\}$

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

$q_0: q_0$

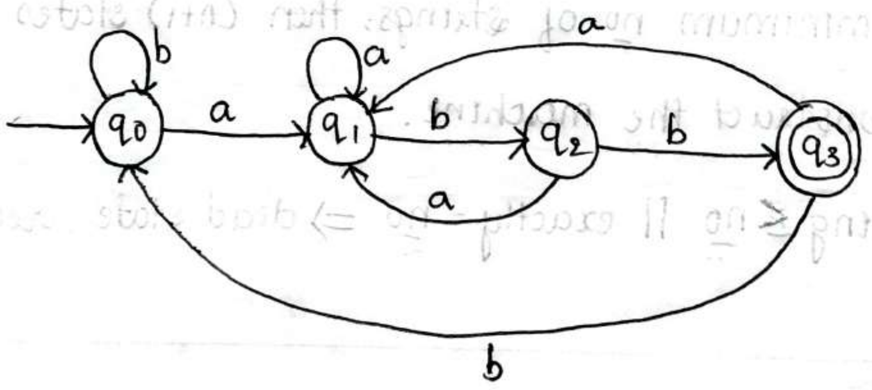
$F = \{q_0, q_1, q_2, q_3\}$

δ = transition table

$q \setminus \Sigma$	0	1
$\rightarrow q_0^*$	q_1	q_1
q_1^*	q_2	q_2
q_2^*	q_3	q_3
q_3^*	q_d	q_d
q_d	q_d	q_d

6) Write a DFA to accept all strings that ends with abb, over $\{a,b\}$.

Soln:- $L = \{abb, babb, aabb, ababb, baababb, \ldots\}$



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

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

$q_0: q_0$

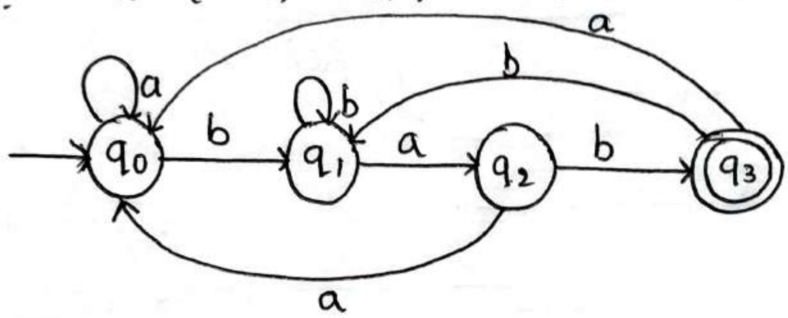
$F = \{q_3\}$

δ = transition table

q \ Σ	a	b
→q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3*	q_1	q_0

7) Construct DFA to accept strings that ends with bab over $\{a, b\}$.

Soln: $L = \{bab, abab, bbab, abbab, babbab, \ldots\}$



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

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

$q_0 : q_0$

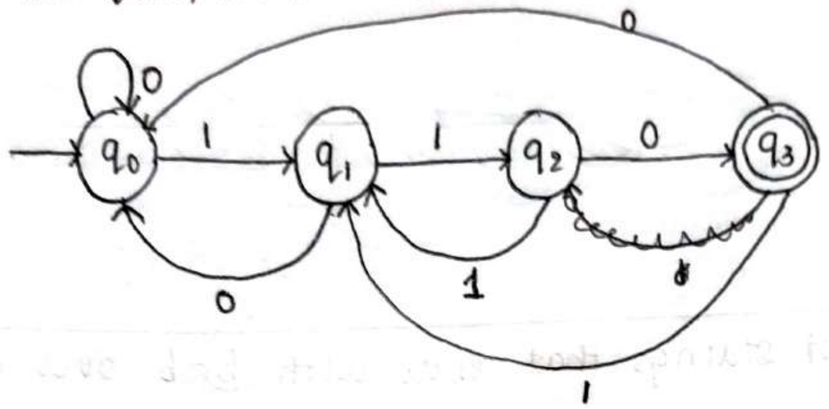
$F = \{q_3\}$

δ = Transition table

q \ Σ	a	b
→q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_3
q_3*	q_0	q_1

8) Write a DFA to accept all strings that ends with 110 over {0,1}

Soln: $L = \{110, 0110, 01110, 101110, \ldots\}$.



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

Σ: $\{0,1\}$

q_0: q_0

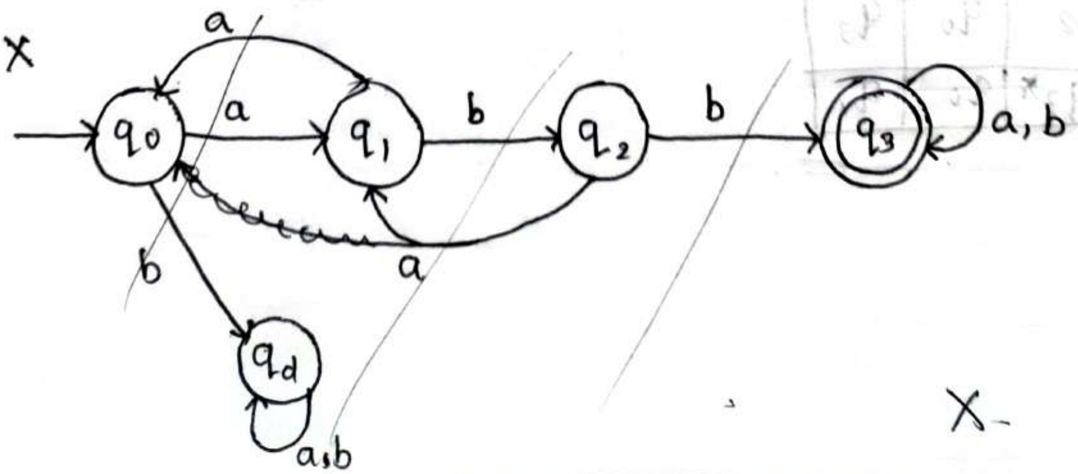
F: $\{q_3\}$

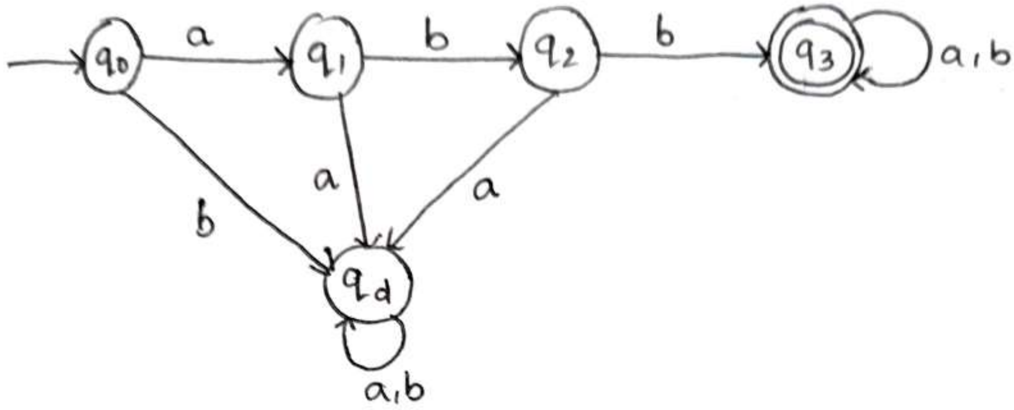
δ = transition table

q \ Σ	0	1
→ q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_3	q_1
q_3*	q_0	q_1

9) Construct a DFA to accept all strings that begins with abb over {a,b}.

Soln: $L = \{$ ~~aba~~ ~~aba~~ abb, abba, abbab, ... $\}$.





Q: $\{q_0, q_1, q_2, q_3, q_d\}$

Σ: $\{a, b\}$

q_0: q_0

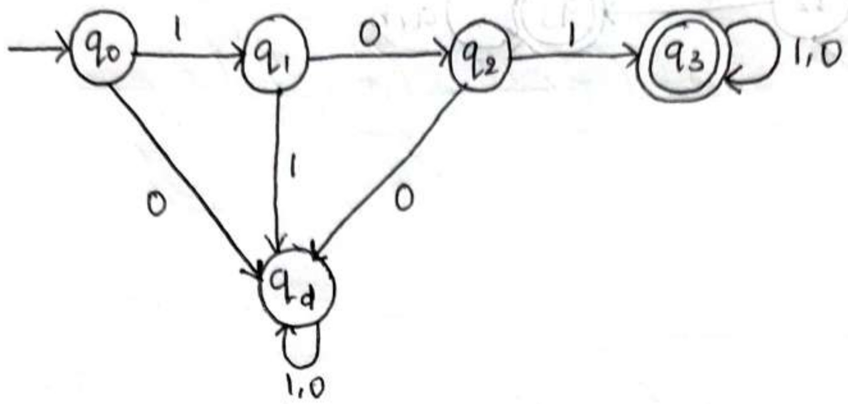
F: $\{q_3\}$

δ = transition table

$q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_1	q_d
q_1	q_d	q_2
q_2	q_d	q_3
q_3^*	q_3	q_3
q_d	q_d	q_d

10) DFA to accept all strings that begin with 101

Soln:- $L = \{101, 1010, 1011, \ldots\}$



$Q: \{q_0, q_1, q_2, q_3, q_d\}$

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

$q_0: q_0$

$F = \{q_3\}$

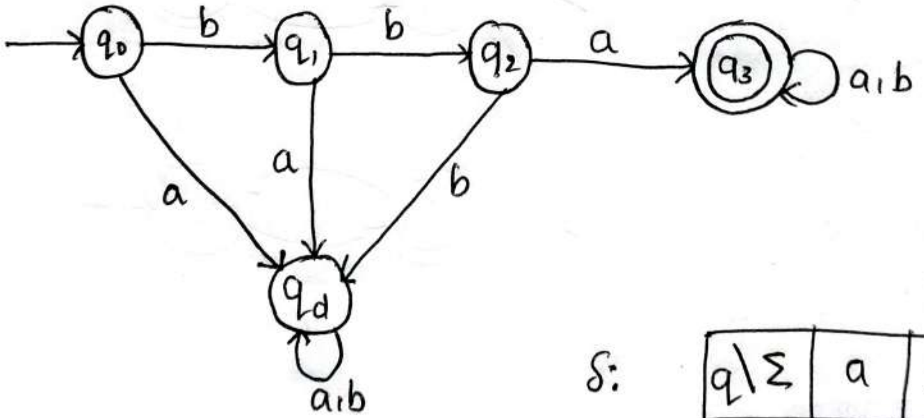
δ:

$q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_d	q_1
q_1	q_2	q_d
q_2	q_d	q_3
q_3^*	q_3	q_3
q_d	q_d	q_d

---

11) Construct a DFA to accept all strings that begin with bba

Soln:- $L = \{bba, bbab, bbaab, \ldots\}$



$Q: \{q_0, q_1, q_2, q_3, q_d\}$

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

$q_0: q_0$

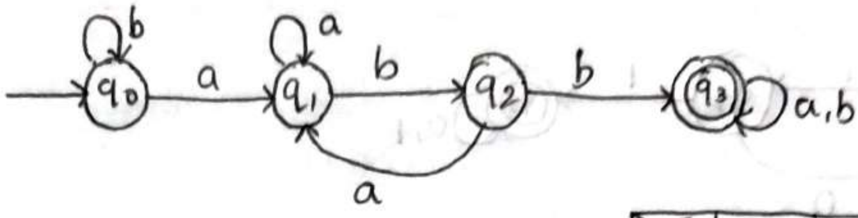
$F = \{q_3\}$

δ:

$q \backslash \Sigma$	a	b
$\rightarrow q_0$	q_d	q_1
q_1	q_d	q_2
q_2	q_3	q_d
q_3^*	q_3	q_3
q_d	q_d	q_d

12) DFA to accept all strings that contains (sub-string) abb

Sol:- $L = \{abb, aabb, abba, abaabb, ababbabba, \dots\}$



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

Σ: $\{a, b\}$

q_0: q_0

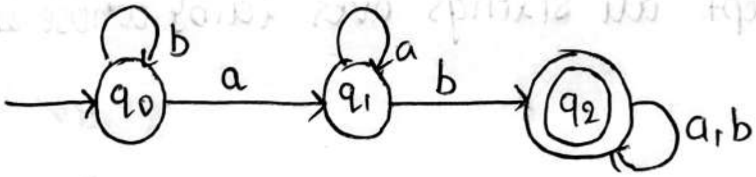
F = $\{q_3\}$

δ:

q \ Σ	a	b
→ q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3^*	q_3	q_3

13) DFA to accept all strings that has 'ab' as sub-string.

Soln:- $L = \{ab, aab, aba, aabbbb, \dots\}$



Q: $\{q_0, q_1, q_2\}$

Σ: $\{a, b\}$

q_0: q_0

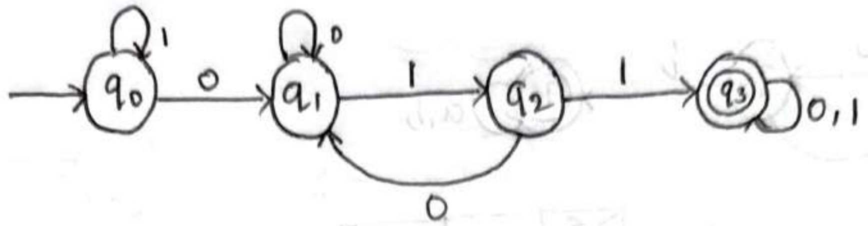
F = $\{q_2\}$

δ:

q \ Σ	a	b
→ q_0	q_1	q_0
q_1	q_1	q_2
q_2^*	q_2	q_2

14) DFA to accept the language that contains $\underline{011}$ as the substring over $\{0,1\}$.

Sol:- $L = \{011, 0011, 0110, 110111, \ldots\}$



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

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

$q_0: q_0$

$F: \{q_3\}$

δ:

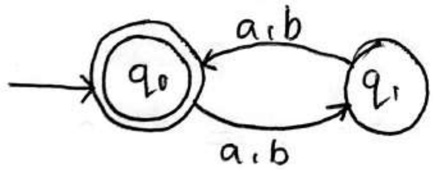
q \ Σ	0	1
→q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_3
q_3^*	q_3	q_3

01/10/2024

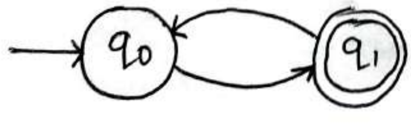
15) Construct a DFA to accept all strings over $\{a,b\}$ whose length is divisible by 2

Soln:-

Note:- (Standard notations for even and odd length).

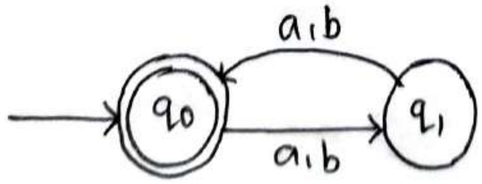


<Even length>



<Odd length>

Soln:- $L = \{\varepsilon, ab, aaab, ababbb, \ldots\}$



Acceptance & rejection of strings:

Ex: Consider abab

$\Rightarrow \delta(q_0, abab) = \delta(q_1, bab)$

$= \delta(q_0, ab)$

$= \delta(q_1, b)$

$= q_0$

$\because q_0 \in F \Rightarrow$ abab is accepted.

Consider, aaa

$\Rightarrow \delta(q_0, aaa) = \delta(q_1, aa)$

$= \delta(q_0, a)$

$= q_1$

$\because q_0 \notin F \Rightarrow$ aaa is rejected.

$Q: \{q_0, q_1\}$

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

$q_0: q_0$

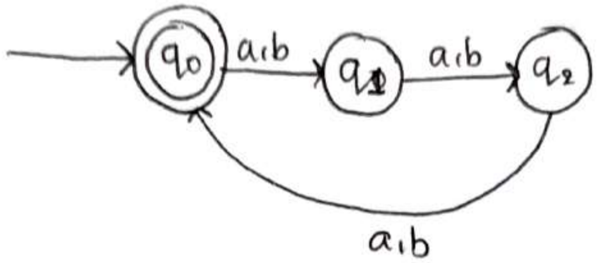
$F: \{q_0\}$

δ:

$Q \backslash \Sigma$	a	b
$\rightarrow q_0^*$	q_1	q_1
q_1	q_0	q_0

16) Write a DFA to accept strings over {a,b} whose length is divisible by 3.

Soln: $L = \{\varepsilon, aaa, aabbaa, \ldots\}$



Considering states → Consider the remainders

0	1	2
q_0	q_1	q_2

Note: If K is the divisor, then there are (K-1) no of states.

Ex: $\frac{\text{Any number}}{3} = 0,1,2$ (remainders)

$\frac{\text{Any number}}{4} = 0,1,2,3$ are the remainders.

Consider string, aaa

$\delta(q_0, aaa) = \delta(q_1, aa)$

$= \delta(q_2, a)$

$= q_0$

$\because q_0 \in F \Rightarrow$ aaa is accepted.

Consider string, aabb

$\delta(q_0, aabb) = \delta(q_1, aab)$

$= \delta(q_2, ab)$

$= \delta(q_0, b)$

$= q_1$

$\because q_1 \notin F \Rightarrow$ aabb is rejected.

$Q: \{q_0, q_1, q_2\}$

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

$q_0: q_0$

$F = \{q_0\}$

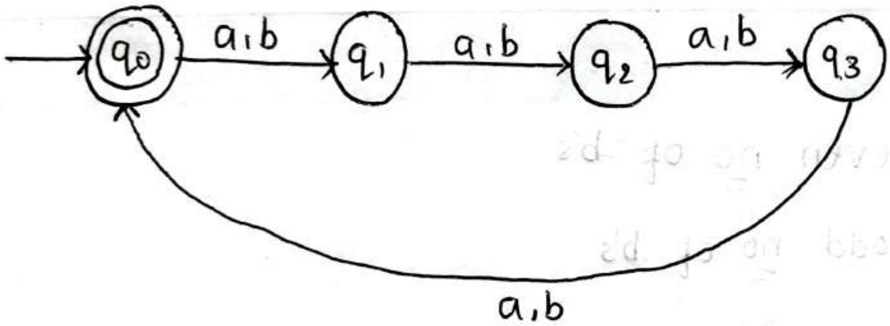
$\delta =$

$q \backslash \Sigma$	a	b
$\rightarrow q_0^*$	q_1	q_1
q_1	q_2	q_2
q_2	q_0	q_0

---

17). DFA to accept strings over $\{a, b\}$ whose length is divisible by 4.

Soln: $L = \{\varepsilon, aaab, aabbabab, \ldots\}$



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

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

$q_0: q_0$

$F = \{q_0\}$

$\delta =$

$q \backslash \Sigma$	a	b
$\rightarrow q_0^*$	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_0	q_0

Consider string, aabb

$\delta(q_0, aabb) = \delta(q_1, abb)$

$= \delta(q_2, bb)$

$= \delta(q_3, b)$

$= q_0$

$\because q_0 \in F \Rightarrow$ aabb is accepted

Consider string, aba

$\delta(q_0, aba) = \delta(q_1, ba)$

$= \delta(q_2, a)$

$= q_3$

$\because q_3 \notin F \Rightarrow$ aba is rejected.

Types of problems:-

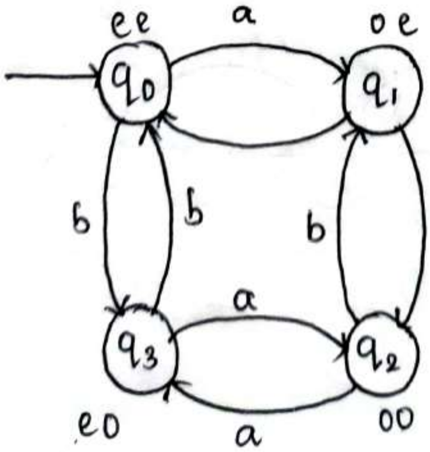
i) Even no of a's and even no of b's

ii) Even no of a's and odd no of b's

iii) Odd no of a's and odd no of b's

iv) odd no of a's and even no of b's

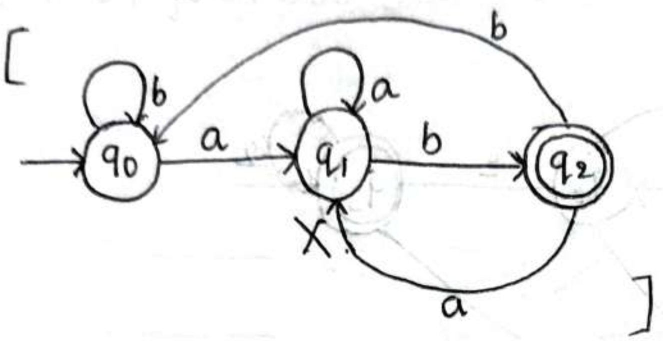
Standard representation is

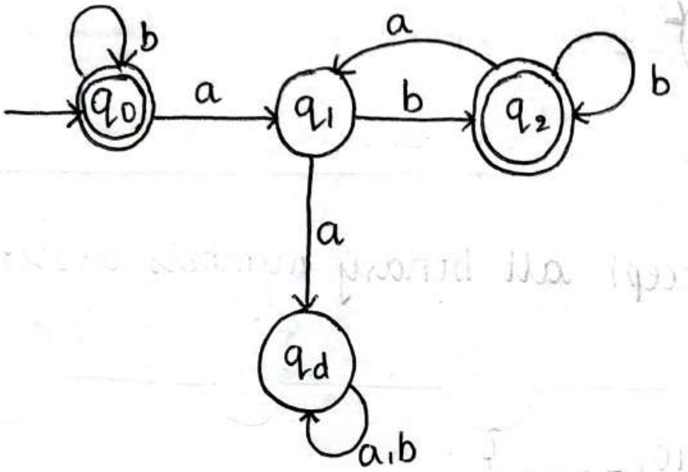


{ based on type, make the state as final }

18) Construct a DFA to access all strings made up of a's and b's in which every a is followed by a b. 03/10/2024

Soln:- $L = \{\varepsilon, ab, bab, abab, bbabab, bbbb, \ldots\}$



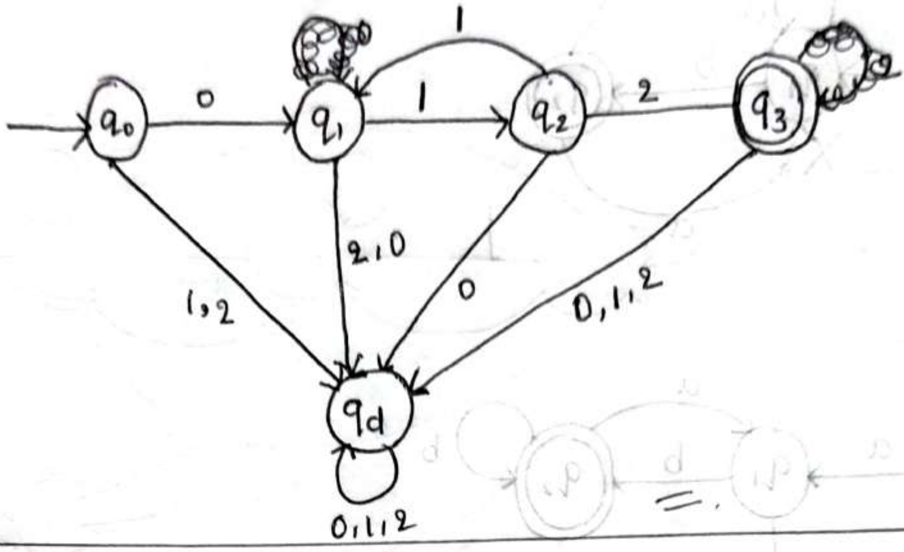


δ:

$q \backslash \Sigma$	a	b
$\rightarrow q_0^*$	q_1	q_0
q_1	q_d	q_2
q_2^*	q_1	q_2
q_d	q_d	q_d

19) Construct DFA which accepts that begins with 0 followed by odd no of ones ~~and ends with~~ followed by 2. {only one 0 at beginning, 2 at the end and odd no of 1's in middle}

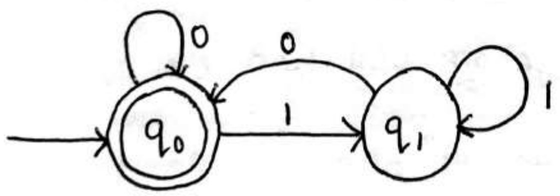
Soln: $L = \{012, 01112, 0111112, \ldots\}$.



Imp.

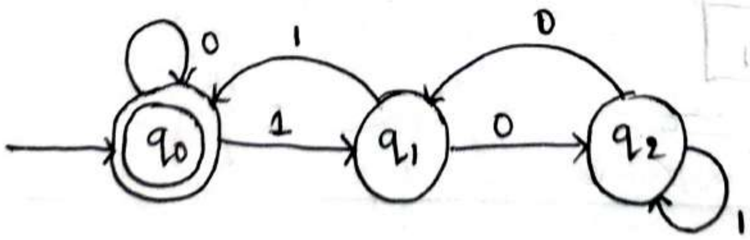
20) Construct a DFA to accept all binary numbers divisible by 2.

Soln: $L = \{0, 10, 100, 1010, 10010, \ldots\}$.



21) DFA to accept all binary numbers divisible by 3.

Soln: $L = \{0, 11, 0110, \ldots\}$.



=.

22) Construct a DFA to accept all decimal numbers devisible by 2. 07/10/2024

Soln:- Here, $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$

Devisible by 2 : <u>Remainders</u>

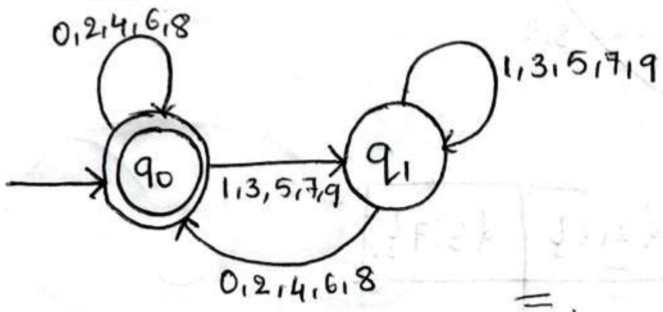
0 , 1

$\downarrow$ $\downarrow$

q_0 q_1

Consider sets based on remainders

$\{0,2,4,6,8\}$, $\{1,3,5,7,9\}$



---

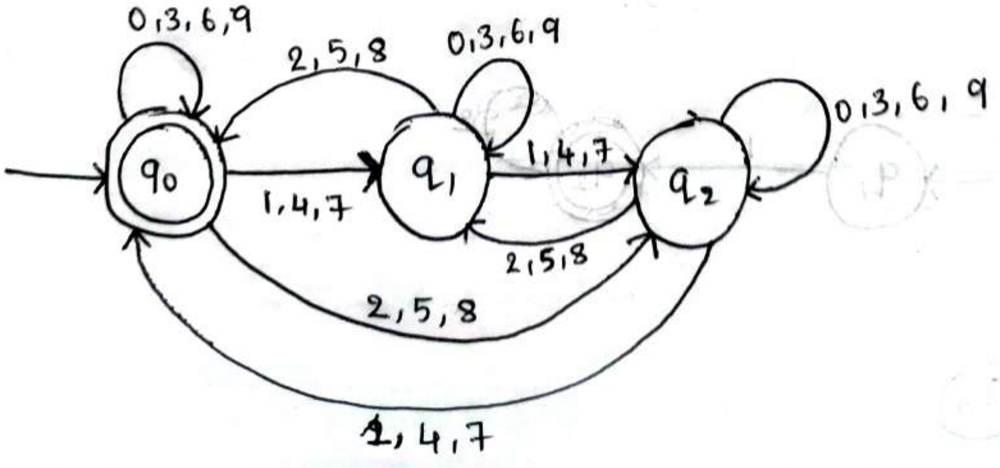
23) Devisibility by 3. (DFA).

Soln:- $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$.

Sets:-

$\{0,3,6,9\}$, $\{1,4,7\}$, $\{2,5,8\}$.

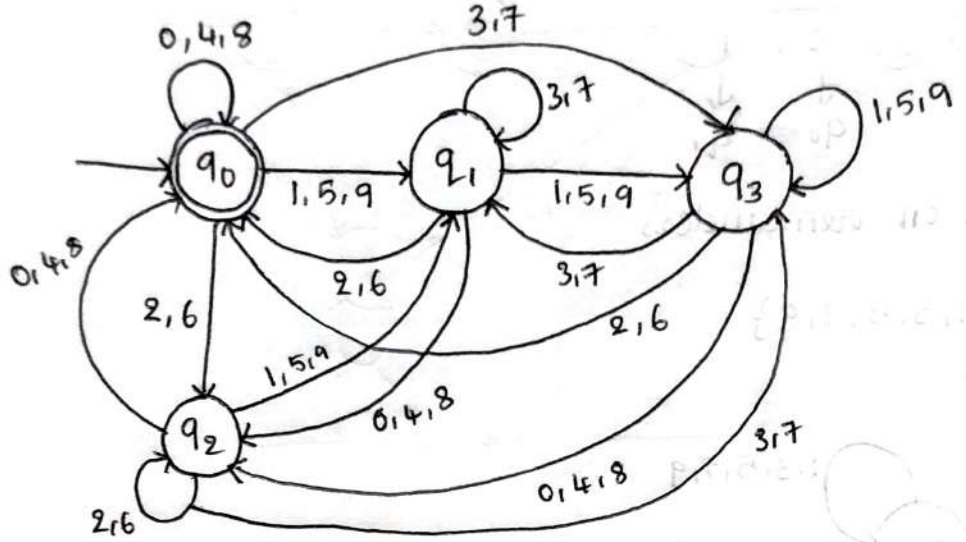
(0) (1) (2).



24). Divisibility by 4.

Sol: $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$.

Sets: $\{0,4,8\}$, $\{1,5,9\}$, $\{2,6\}$, $\{3,7\}$.
(0) (1) (2) (3).

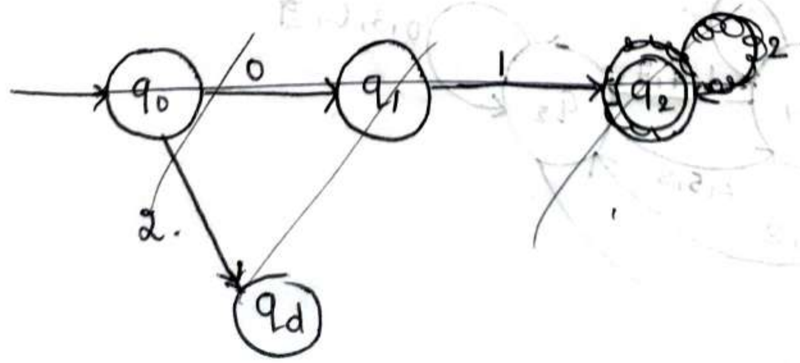


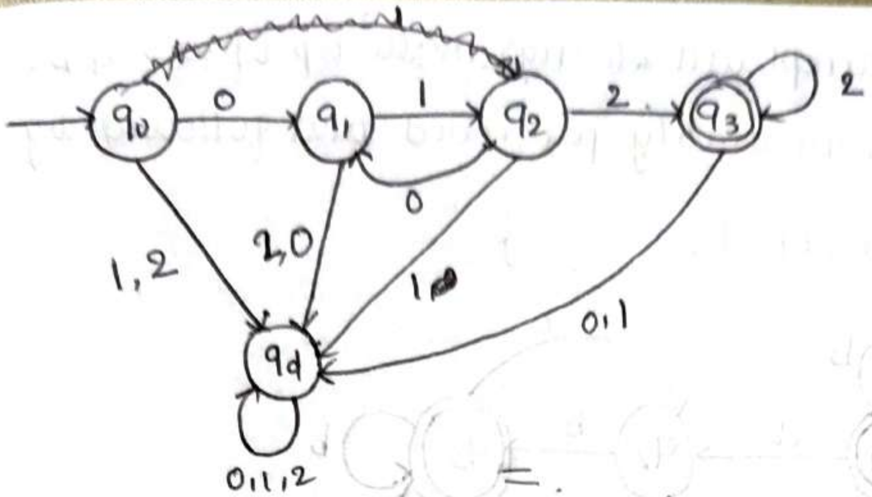
δ:

$q \backslash \Sigma$	$\{0,4,8\}$	$\{1,5,9\}$	$\{2,6\}$	$\{3,7\}$.
$\rightarrow q_0^*$	q_0	q_1	q_2	q_3.
q_1	q_2	q_3	q_0	q_1
q_2	q_0	q_1	q_2	q_3.
q_3	q_2	q_3	q_0	q_1

25). Given, $L = \{(01)^i (2)^j \mid i,j \geq 1\}$

Sol: $\{012, 0122, 0101222, ----\}$.

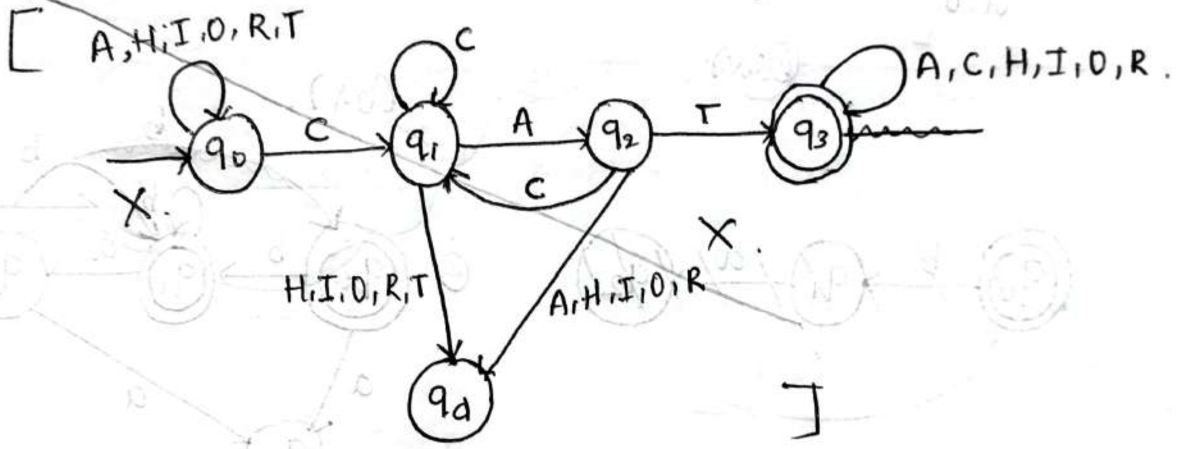




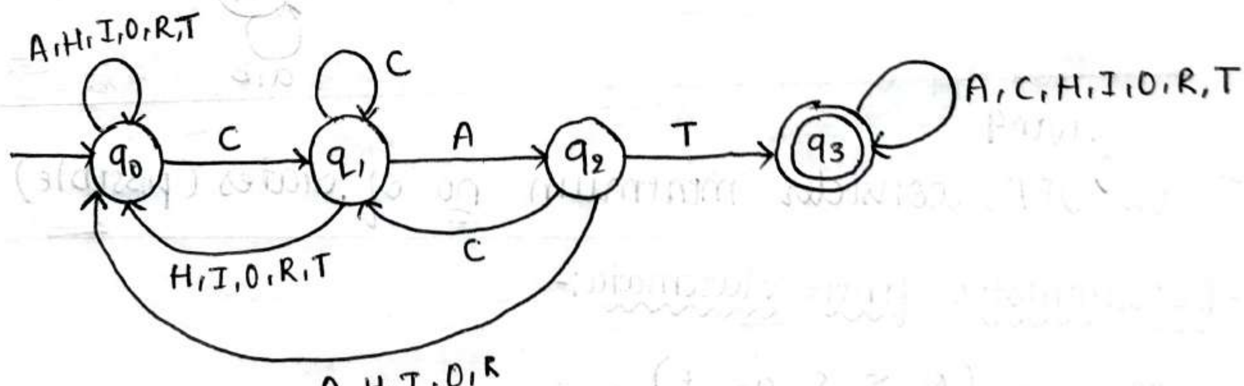
26) Design a DFA to read strings made up of the word CHARIOT and recognize those strings that contain CAT as substring.

$\Sigma = \{A, C, H, I, O, R, T\}$.

Soln:- [



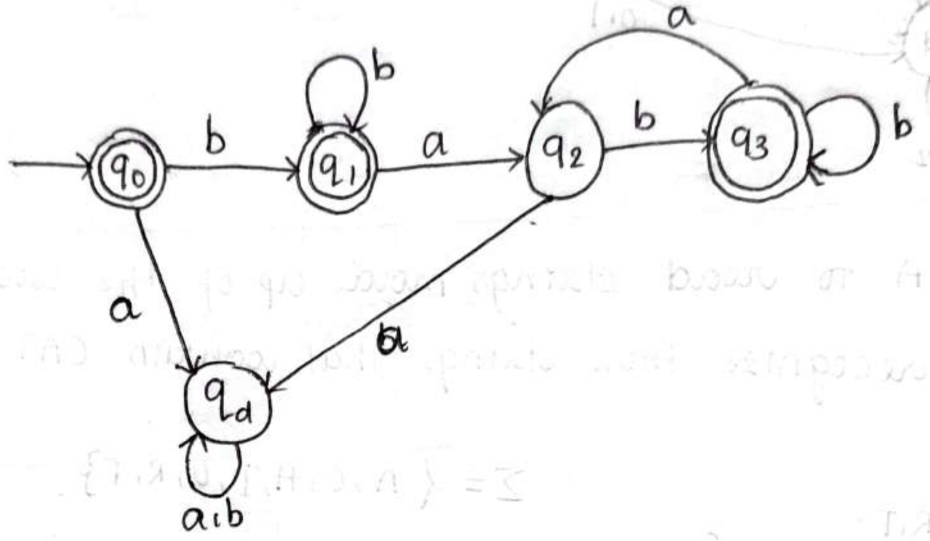
]



Note:- "Sub-string ~~do~~ should not contain dead state".

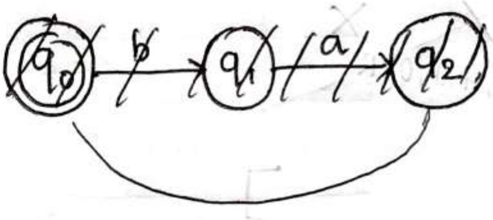
27) Construct DFA to accept all strings made up of a's & b's in which every 'a' is immediately preceeded and followed by 'b'.

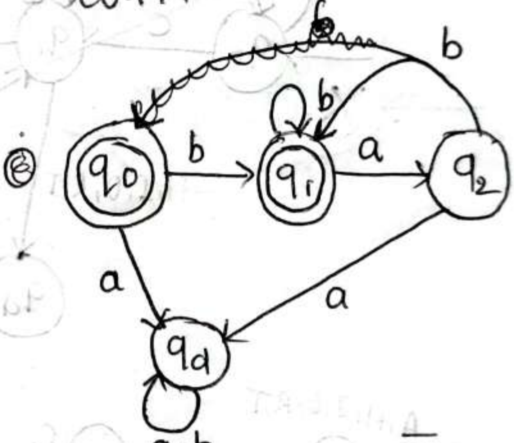
Soln:- $L = \{\varepsilon, b, bb, bab, bbab, \ldots\}$.



(case).

(or).





Imp.

Note:- For solving DFA, consider minimum no of states (possible).

Imp

Non-Deterministic finite Automata:-

$$M_{NFA} = (Q, \Sigma, \delta, q_0, F)$$

where Q: a finite set of all states

Σ: a finite alphabet set.

q_0: start state

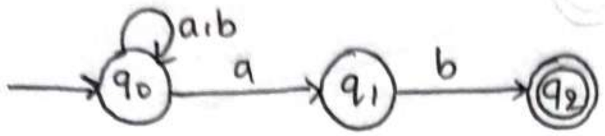
δ: Transition function, $Q \times \Sigma \rightarrow 2^Q$

F: a finite set of all final states.

Construct NFA for the following:-

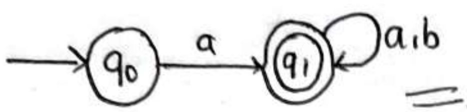
1) To accept all strings ending with ab.

Soln: $L = \{ab, aab, bab, aaab, \dots\}$



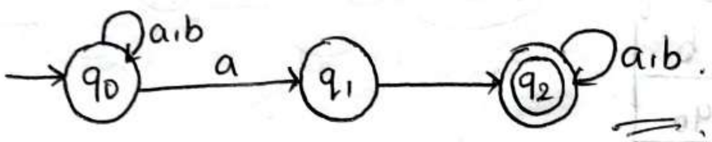
2) To accept all strings that begins with 'a'.

Soln: $L = \{a, ab, aaa, aabb, \dots\}$



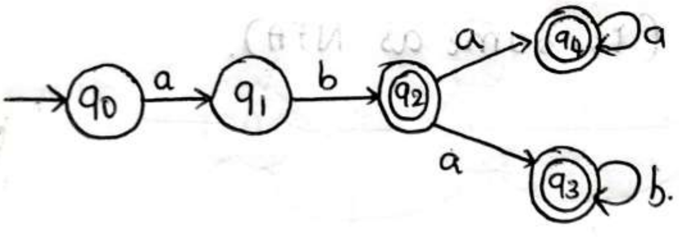
3) To accept all string that contains 'ab'.

Soln: $L = \{ab, aab, abab, abbbb, \dots\}$.

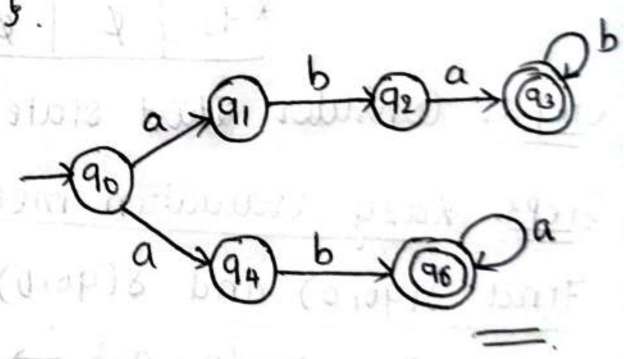


4) To accept string $abab^n$ (or) aba^n with $n \geq 0$.

Soln: $L = \{ab, aba, aba, abab, abaa, \dots\}$.

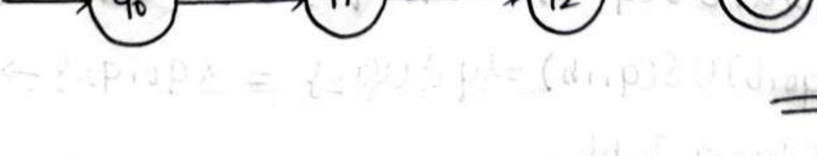


(or)



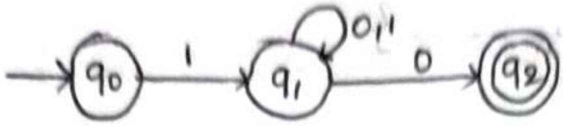
5) To accept string in which 3rd symbol from right is a.

Soln:



6) Strings that begin with 1 and ends with 0.

Soln:- $L = \{10, 1100, 1010, \ldots\}$.



Conversion of NFA into DFA:-

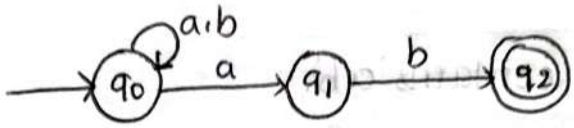
1) Lazy evaluation method

2) Subset construction method.

---

1) Lazy evalution method:-

1. Given,



Soln:- Step1:- Transition table for NFA.

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
$*q_2$	ϕ	ϕ

Step2: Consider Start state of DFA (ie, same as NFA).

Step3: Lazy evaluation method:

Find $\delta(q_0, a)$ and $\delta(q_0, b)$

$\delta(q_0, a) = \{q_0, q_1\} \rightarrow$ new state.

$\delta(q_0, b) = \{q_0\}$.

Find $\delta(\{q_0, q_1\}, a)$ & $\delta(\{q_0, q_1\}, b)$.

$\delta(\{q_0, q_1\}, a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$.

$\delta(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\} \rightarrow$ new state.

Find $\delta(\{q_0, q_2\}, a)$ & $\delta(\{q_0, q_2\}, b)$.

$\delta(\{q_0, q_2\}, a) = \delta(q_0, a) \cup \delta(q_2, a) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$.

$\delta(\{q_0, q_2\}, b) = \delta(q_0, b) \cup \delta(q_2, b) = \{q_0\} \cup \phi = \{q_0\}$.

If there is no new state, further simplification not required.

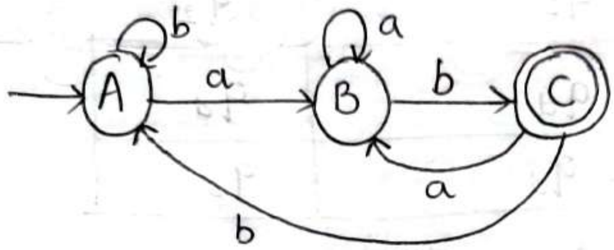
Renaming:-

$\{q_0\} \rightarrow A$; $\{q_0, q_1\} \rightarrow B$, $\{q_0, q_2\} \rightarrow C$.

Transition table for DFA:

	a	b
$\rightarrow A$	B	A
B	B	C
$*C$	B	A.

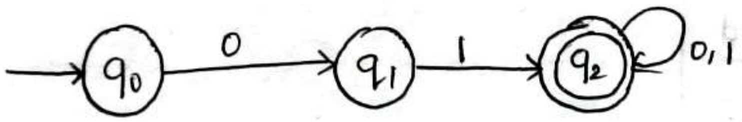
$\Rightarrow$ required DFA. is



08/10/2024

2). Convert given NFA to DFA using lazy evaluation method.

Given,



Soln:- Step1:- Transition table for NFA.

	0	1
$\rightarrow q_0$	q_1	ϕ
q_1	ϕ	q_2
$* q_2$	q_2	q_2.

Step2: start state for DFA is the start state of NFA, ie., q_0

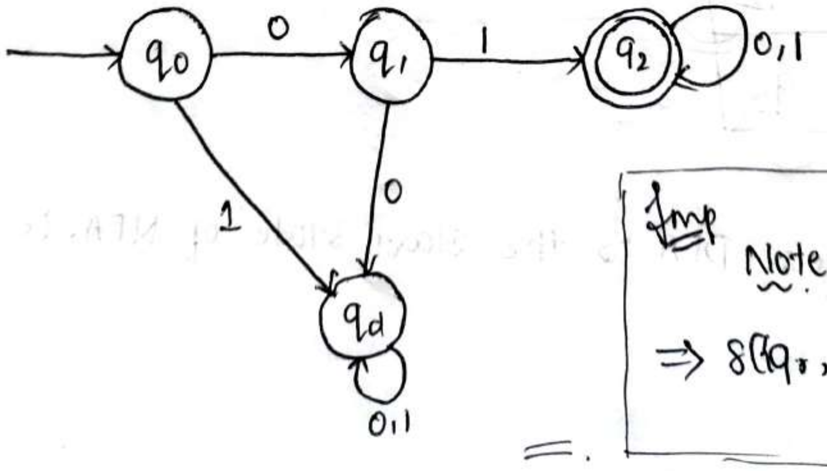
Q_{DFA}	Σ	Transition function	Result
q_0	0	$\delta(q_0, 0) = q_1$	q_1
	1	$\delta(q_0, 1) = \phi = q_d$	q_d
q_1	0	$\delta(q_1, 0) = \phi = q_d$	q_d
	1	$\delta(q_1, 1) = q_2$	q_2
q_2	0	$\delta(q_2, 0) = q_2$	q_2
	1	$\delta(q_2, 1) = q_2$	q_2
q_d	0	$\delta(q_d, 0) = q_d$	q_d
	1	$\delta(q_d, 1) = q_d$	q_d

Step 3:

Transition table for DFA:-

	0	1
$\rightarrow q_0$	q_1	q_d
q_1	q_d	q_2
$*q_2$	q_2	q_2
q_d	q_d	q_d

DFA:

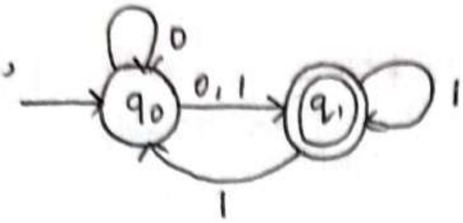


Imp Note: If $\delta(q_s, 0) = \{q_r, q_t\}$

$\Rightarrow \delta(\{q_r, q_t\}, 0) = \delta(q_r, 0) \cup \delta(q_t, 0)$

3) Convert NFA to DFA.

given,



Soln: S1: Transition table for NFA.

	0	1
q_0	$\{q_0, q_1\}$	q_1
q_1	ϕ	$\{q_0, q_1\}$

S2:

Q_{DFA}	Σ	Transition Function	Result
q_0	0	$\delta(q_0, 0) = \{q_0, q_1\}$	$\{q_0, q_1\}$
	1	$\delta(q_0, 1) = q_1$	q_1
q_1	0	$\delta(q_1, 0) = \phi = q_d$	q_d
	1	$\delta(q_1, 1) = \{q_0, q_1\}$	$\{q_0, q_1\}$
$\{q_0, q_1\}$	0	$\delta(\{q_0, q_1\}, 0) = \delta(q_0, 0) \cup \delta(q_1, 0)$ $\{q_0, q_1\} \cup \{q_d\}$	$\{q_0, q_1\}$
	1	$\delta(\{q_0, q_1\}, 1) = \delta(q_0, 1) \cup \delta(q_1, 1)$ $q_1 \cup \{q_0, q_1\}$	$\{q_0, q_1\}$
q_d	0	$\delta(q_d, 0) = q_d$	q_d
	1	$\delta(q_d, 1) = q_d$	q_d

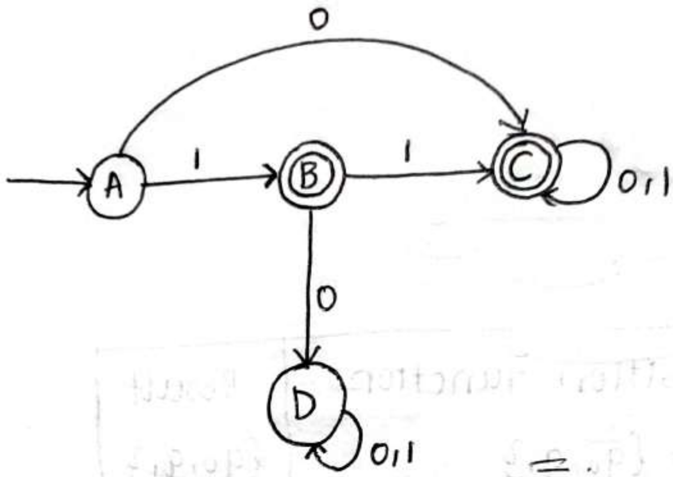
Renaming:

$q_0 : A$, $q_1 : B$, $\{q_0, q_1\} : C$, $q_d : D$

S3:- Transition table for DFA.

	0	1
→A	C	B
B*	D	C
C*	C	C
D	D	D

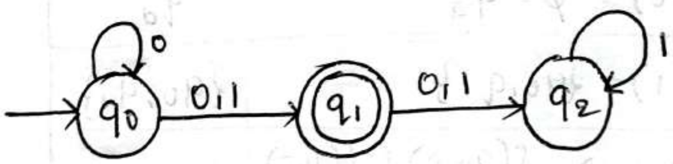
DFA ⇒.



Imp **.

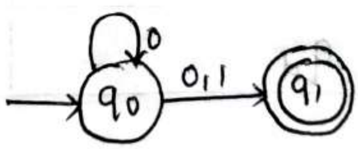
4) Convert NFA to DFA.

given,



Soln:- whenever a dead state is given in NFA, we simplify it and eliminate that dead state.

∴ Given ⇒



Transition table for NFA:

	0	1
q_0	$\{q_0, q_1\}$	q_1
q_1	ϕ	ϕ

Now,

Q_{DFA}	Σ	Transition function	Result
q_0	0	$\delta(q_0, 0) = \{q_0, q_1\}$	$\{q_0, q_1\}$
	1	$\delta(q_0, 1) = q_1$	q_1
q_1	0	$\delta(q_1, 0) = \phi = q_d$	q_d
	1	$\delta(q_1, 1) = \phi = q_d$	q_d
$\{q_0, q_1\}$	0	$\delta(\{q_0, q_1\}, 0) = \{q_0, q_1\} \cup \phi$	$\{q_0, q_1\}$
	1	$\delta(\{q_0, q_1\}, 1) = q_1 \cup \phi$	q_1.
q_d	0	$\delta(q_d, 0) = q_d$	q_d
	1	$\delta(q_d, 1) = q_d$	q_d

Transition table for DFA:

	0	1
→A	C	B
B*	D	D
C*	C	B
D	D	D

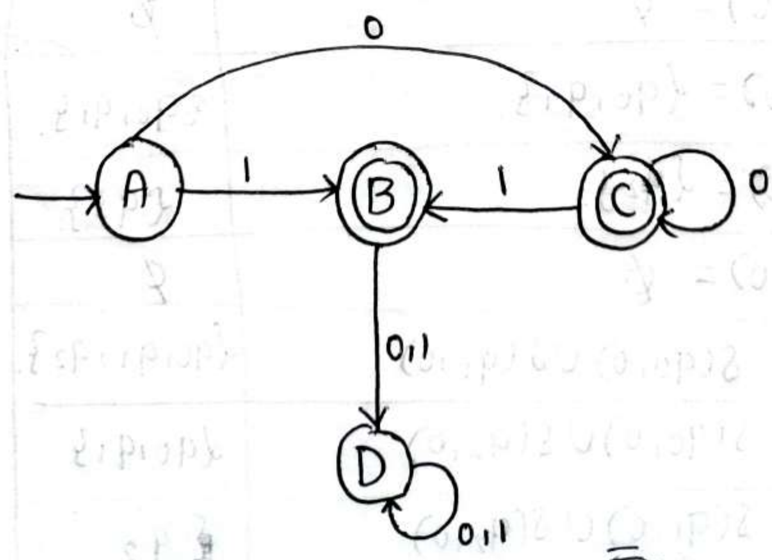
here A: q_0

B: q_1

C = $\{q_0, q_1\}$

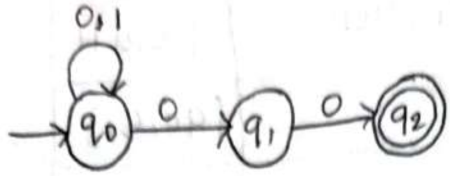
D = q_d.

DFA ⇒



Conversion of NFA to DFA using sub-set construction method:-

1). Given,



Soln:- Transition table of NFA.

	0	1
$\rightarrow q_0$	$\{q_0,q_1\}$	q_0
q_1	q_2	ϕ
q_2^*	ϕ	ϕ

Conversion process:-

S1:- Find all possible subset

ϕ, $\{q_0\}$, $\{q_1\}$, $\{q_2\}$, $\{q_0,q_1\}$, $\{q_0,q_2\}$, $\{q_1,q_2\}$, $\{q_0,q_1,q_2\}$.

S2:- Identify the start state.

$q_{0_{DFA}} = q_0$.

S3:- Transition for '0' input:-

Q_{DFA}	Transition function	Result
ϕ.	$\delta(\phi,0) = \phi$	ϕ
$\rightarrow \{q_0\}$	$\delta(q_0,0) = \{q_0,q_1\}$.	$\{q_0,q_1\}$.
$\{q_1\}$	$\delta(q_1,0) = \{q_2\}$.	$\{q_2\}$.
* $\{q_2\}$	$\delta(q_2,0) = \phi$	ϕ
$\{q_0,q_1\}$	$\delta(\{q_0,q_1\},0) = \delta(q_0,0) \cup \delta(q_1,0)$	$\{q_0,q_1,q_2\}$.
* $\{q_0,q_2\}$	$\delta(\{q_0,q_2\},0) = \delta(q_0,0) \cup \delta(q_2,0)$	$\{q_0,q_1\}$
* $\{q_1,q_2\}$	$\delta(\{q_1,q_2\},0) = \delta(q_1,0) \cup \delta(q_2,0)$	q_2
* $\{q_0,q_1,q_2\}$	$\delta(\{q_0,q_1,q_2\},0) = \delta(q_0,0) \cup \delta(q_1,0) \cup \delta(q_2,0)$	$\{q_0,q_1,q_2\}$

Transition for '1' inputs:-

Q_{DFA}	Transition function	Result
ϕ	$\delta(\phi,1)=\phi$	ϕ
$\rightarrow\{q_0\}$	$\delta(\{q_0\},1)= q_0$	q_0
$\{q_1\}$	$\delta(\{q_1\},1)=\phi$	ϕ
$*\{q_2\}$	$\delta(\{q_2\},1)=\phi$	ϕ
$\{q_0,q_1\}$	$\delta(\{q_0,q_1\},1) = \delta(q_0,1) \cup \delta(q_1,1)$	$\{q_0\}$
$*\{q_0,q_2\}$	$\delta(\{q_0,q_2\},1) = \delta(q_0,1) \cup \delta(q_2,1)$	$\{q_0\}$
$*\{q_1,q_2\}$	$\delta(\{q_1,q_2\},1\} = \delta(q_1,1) \cup \delta(q_2,1)$	ϕ
$*\{q_0,q_1,q_2\}$	$\delta(\{q_0,q_1,q_2\},1\} = \delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1)$	$\{q_0\}$

*Step 4:- $\boxed{\text{All reachable states.}}$

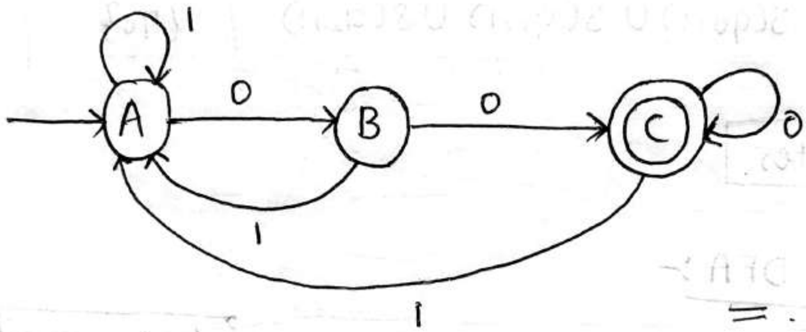
Transition table of DFA :-

Q_{DFA}	Transition Function.	Result
$\rightarrow\{q_0\}$		
$\{q_0,q_1\}$		

	0	1
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$.
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$
$\{q_0, q_1, q_2\}^*$	$\{q_0, q_1, q_2\}$	$\{q_0\}$.

required DFA.

Let. $\{q_0\} \rightarrow A$, $\{q_0, q_1\} \rightarrow B$, $\{q_0, q_1, q_2\} \rightarrow C$.



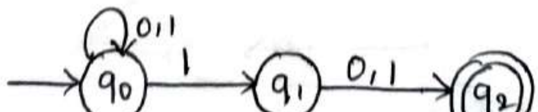
Imp

Note:- If we get ϕ in the selected state, then only we have to consider dead state in the final DFA (automata)

* If we have to convert NFA to DFA with 4 (or) more states given, then we go for lazy evaluation method (or) < 4, then only use subset construction method.

2) Convert NFA to DFA.

given,



Soln: Transition table of NFA:-

	0	1
$\rightarrow q_0$	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
$*q_2$	ϕ	ϕ.

Transition for '0' input:-

Q_{DFA}.	Transition function	Result
ϕ.	$\delta(\phi, 0) = q_0$	ϕ
$\rightarrow \{q_0\}$	$\delta(\{q_0\}, 0) = q_0$	q_0
$\{q_1\}$	$\delta(\{q_1\}, 0) = q_2$	q_2
$\{q_2\}^*$	$\delta(\{q_2\}, 0) = \phi$	ϕ
$\{q_0, q_1\}$	$\delta(q_0, 0) \cup \delta(q_1, 0)$	$\{q_0, q_2\}$
$\{q_0, q_2\}^*$	$\delta(q_0, 0) \cup \delta(q_2, 0)$	$\{q_0\}$.
$\{q_1, q_2\}^*$	$\delta(q_1, 0) \cup \delta(q_2, 0)$	$\{q_2\}$
$\{q_0, q_1, q_2\}^*$	$\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)$	$\{q_0, q_2\}$

Transition for '1' input:-

Q_{DFA}	Transition function	Result
ϕ	$\delta(\phi,1) = \phi$	ϕ
$\rightarrow \{q_0\}$	$\delta(q_0,1) = \{q_0,q_1\}$	$\{q_0,q_1\}$
$\{q_1\}$	$\delta(q_1,1) = q_2$	$\{q_2\}$
$\{q_2\}$ *	$\delta(q_2,1) = \phi$	ϕ
$\{q_0,q_1\}$	$\delta(q_0,1) \cup \delta(q_1,1)$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$ *	$\delta(q_0,1) \cup \delta(q_2,1)$	$\cancel{\phi}$ $\{q_0,q_1\}$
$\{q_1,q_2\}$ *	$\delta(q_1,1) \cup \delta(q_2,1)$	$\{q_2\}$
$\{q_0,q_1,q_2\}$ *	$\delta(q_0,1) \cup \delta(q_1,1) \cup \delta(q_2,1)$	$\{q_0,q_1,q_2\}$

- Transition for DFA:

	0	1
$\rightarrow \{q_0\}$	$\{q_0\}$	$\{q_0,q_1\}$
$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0\}$	$\{q_0,q_1\}$

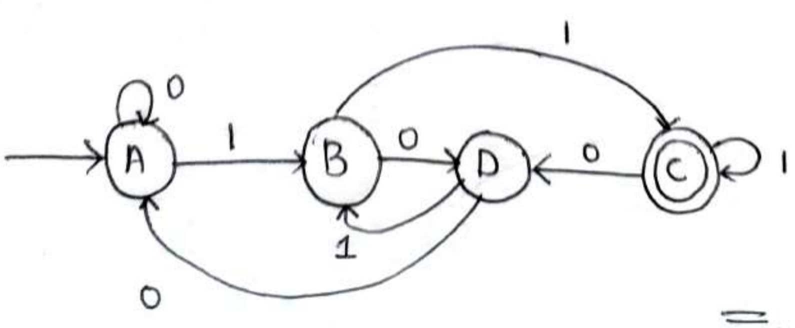
Let $\{q_0\}$: A

$\{q_0,q_1\}$: B

$\{q_0,q_1,q_2\}$: D

$\{q_0,q_2\}$: C

required DFA.



3Q).

	0	1
$\rightarrow p$	$\{p,q\}$	$\{p\}$
q	ϕ	$\{r\}$
$*r$	$\{p,r\}$	$\{q\}$

Soln: Transition for '0' input:-

Q_{DFA}	Transition function	Result
ϕ	$\delta(\phi,0) = \phi$	ϕ
$\rightarrow\{p\}$	$\delta(p,0) = \{p,q\}$	$\{p,q\}$
$\{q\}$	$\delta(q,0) = \phi$	ϕ
$\{p,q\}$	$\delta(p,0) \cup \delta(q,0) = \{p,q\} \cup \phi$	$\{p,q\}$
$\{p,r\}*$	$\delta(p,0) \cup \delta(r,0) = \{p,q\} \cup \{p,r\}$	$\{p,q,r\}$
$\{r\}*$	$\delta(r,0) = \{p,r\}$	$\{p,r\}$
$\{q,r\}*$	$\delta(q,0) \cup \delta(r,0) = \{p,r\} \cup \phi$	$\{p,r\}$
$\{p,q,r\}*$	$\delta(p,0) \cup \delta(q,0) \cup \delta(r,0)$	$\{p,q,r\}$

Transition for '1' input:-

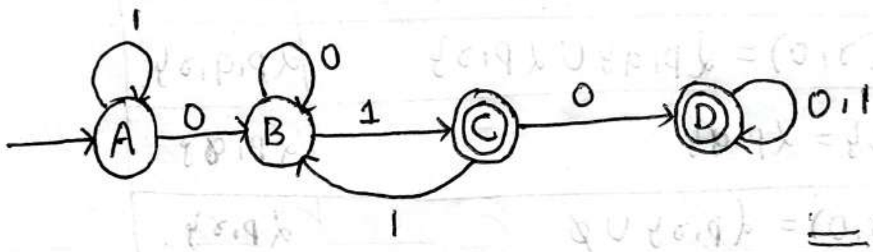
Q_{DFA}	Transition Function	Result
ϕ	$\delta(\phi,1) = \phi$	ϕ
$\rightarrow\{p\}$	$\delta(p,1) = \{p\}$	$\{p\}$
$\{q\}$	$\delta(q,1) = \{r\}$	$\{r\}$
$\{r\}*$	$\delta(r,1) = \{q\}$	$\{q\}$
$\{p,q\}*$	$\delta(p,1) \cup \delta(q,1)$	$\{p,r\}$
$\{p,r\}*$	$\delta(p,1) \cup \delta(r,1)$	$\{p,q\}$
$\{q,r\}*$	$\delta(q,1) \cup \delta(r,1)$	$\{r,q\}$
$\{p,q,r\}*$	$\delta(p,1) \cup \delta(q,1) \cup \delta(r,1)$	$\{p,r,q\}$

Transition of DFA:-

	0	1
→{p}	{p,q}	{p}
{p,q}	{p,q}	{p,r}
* {p,r}	{p,q,r}	{p,q}
* {p,q,r}	{p,q,r}	{p,q,r}

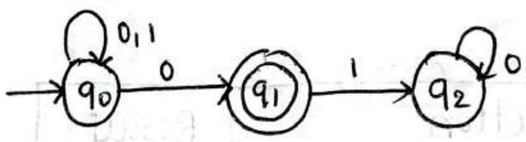
Renaming:-

{p} : A, {p,q} : B , {p,r} : C , {p,q,r} : D.



09/10/2024

4). given,



convert to DFA.

{ no need of taking q_2, as it is a dead state }.

Soln:- Transition table for NFA.

	0	1
→ q_0	$\{q_0,q_1\}$	q_0
* q_1	ϕ	q_2
* q_2	q_2	ϕ

{solution wrong}.

Transition of '0' input:-

ϕ	$\delta(\phi,0)$	ϕ.
$\rightarrow\{q_0\}$	$\delta(q_0,0)=\{q_0,q_1\}$	$\{q_0,q_1\}$.
$*\{q_1\}$	$\delta(q_1,0)=\phi$	ϕ
$\{q_2\}$	$\delta(q_2,0)=\{q_2\}$	$\{q_2\}$.
$*\{q_0,q_1\}$	$\delta(q_0,0)\cup\delta(q_1,0)$	$\{q_0,q_1\}$.
$\{q_0,q_2\}$	$\delta(q_0,0)\cup\delta(q_2,0)$	$\{q_0,q_1,q_2\}$.
$*\{q_1,q_2\}$	$\delta(q_1,0)\cup\delta(q_2,0)$	$\{q_2\}$.
$*\{q_0,q_1,q_2\}$	$\delta(q_0,0)\cup\delta(q_1,0)\cup\delta(q_2,0)$	$\{q_0,q_1,q_2\}$.

Transition for '1' input:

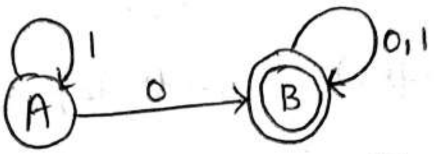
ϕ	$\delta(\phi,1)$	ϕ.
$\rightarrow\{q_0\}$	$\delta(q_0,1)=q_0$	$\{q_0\}$.
$*\{q_1\}$	$\delta(q_1,1)=q_2$.	$\{q_2\}$.
$\{q_2\}$.	$\delta(q_2,1)=\phi$.	ϕ.
$*\{q_0,q_1\}$	$\delta(q_0,1)\cup\delta(q_1,1)$	$\{q_0,q_1\}$.
$\{q_0,q_2\}$	$\delta(q_0,1)\cup\delta(q_2,1)$	$\{q_0\}$.
$*\{q_1,q_2\}$.	$\delta(q_1,1)\cup\delta(q_2,1)$	$\{q_1\}$.
$*\{q_0,q_1,q_2\}$	$\delta(q_0,1)\cup\delta(q_1,1)\cup\delta(q_2,1)$	$\{q_0,q_1\}$.

Transition table of DFA :-

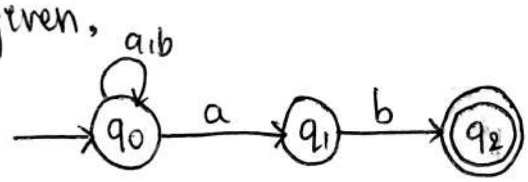
	0	1
→ q_0	$\{q_0,q_1\}$	q_0.
* $\{q_0,q_1\}$	$\{q_0,q_1\}$	$\{q_0,q_1\}$

renaming, q_0 : A , $\{q_0,q_1\}$: B.

DFA :-



5). given,



convert to DFA.

Sol:- Transition table for NFA.

	a	b
q_0	$\{q_0,q_1\}$	q_0
q_1	ϕ.	q_2
q_2	ϕ	ϕ.

Transition on 'a' input:-

ϕ.	$\delta(\phi, a) = \phi$.	ϕ.
$\{q_0\}$	$\delta(q_0, a) = \{q_0, q_1\}$.	$\{q_0, q_1\}$.
$\{q_1\}$	$\delta(q_1, a) = \phi$	ϕ
$\{q_2\}$	$\delta(q_2, a) = \phi$	ϕ
$\{q_0, q_1\}$	$\delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\}$.	$\{q_0, q_1\}$
$\{q_0, q_2\}$	$\delta(q_0, a) \cup \delta(q_2, a) = \{q_0, q_1\}$	$\{q_0, q_1\}$
$\{q_1, q_2\}$	$\delta(q_1, a) \cup \delta(q_2, a) = \phi$	ϕ
$\{q_1, q_2, q_3\}$	$\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a) = \{q_0, q_1\}$	$\{q_0, q_1\}$

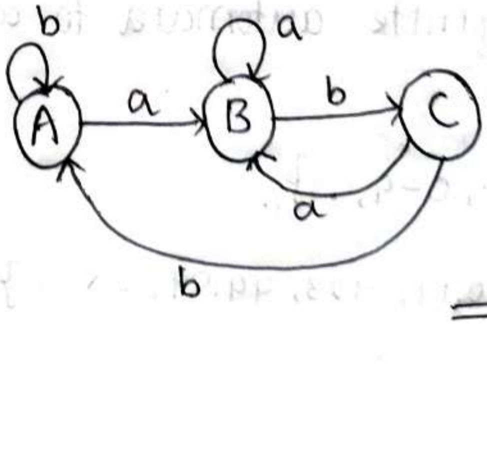
Transition on 'b' input:

ϕ.	$\delta(\phi, b) = \phi$	ϕ
$\{q_0\}$	$\delta(q_0, b) = \{q_0\}$	$\{q_0\}$
$\{q_1\}$	$\delta(q_1, b) = \{q_2\}$	$\{q_2\}$
$\{q_2\}$	$\delta(q_2, b) = \phi$	ϕ
$\{q_0, q_1\}$	$\delta(q_0, b) \cup \delta(q_1, b) = \{q_0, q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\delta(q_0, b) \cup \delta(q_2, b) = \{q_0\}$	$\{q_0\}$
$\{q_1, q_2\}$	$\delta(q_1, b) \cup \delta(q_2, b) = \{q_2\}$	$\{q_2\}$.
$\{q_0, q_1, q_2\}$	$\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b) = \{q_0, q_2\}$	$\{q_0, q_2\}$.

Transition table of DFA:-

	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	q_0.
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$

required DFA.



renaming: $\{q_0\} \rightarrow A$

$\{q_0, q_1\} \rightarrow B$

$\{q_0, q_2\} \rightarrow C$

~~$\{q_0, q_1, q_2\} \rightarrow D$~~.

Epsilon NFA (ε-NFA):-

* It is an extended version of NFA.

* Formal definition:-

$$M_{\varepsilon\text{-NFA}} : (Q, \Sigma, q_0, F, \delta)$$

where Q: finite set of all states.

Σ: Alphabet set $\cup\ \varepsilon$.

q_0: Start state

F: finite set of all final states.

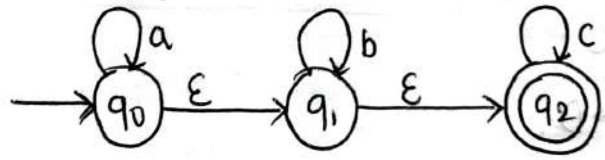
δ: transition function

$$Q \times (\Sigma \cup \varepsilon) \rightarrow 2^Q$$

---

1). construct a finite automata which accepts 0 (or) more no of a's followed by 0 (or) more no of b's followed by 0 (or) more no of c's.

Soln:- $L = \{\varepsilon, abc, aaa, bbb, ccc, aaa\_\_bbb\_\_ccc, ----\}$

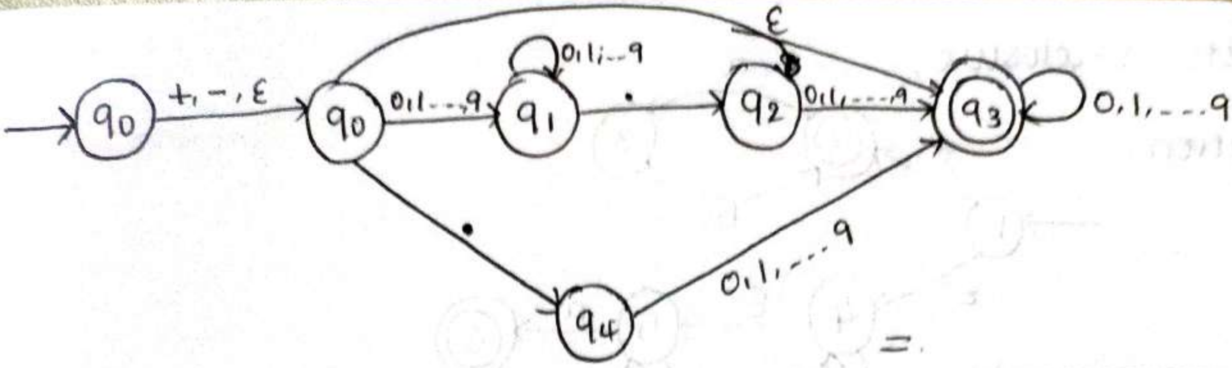


---

2) Construct a finite automata to accept signed (or) unsigned number.

Soln:- $\Sigma = \{+, -, 0\text{-}9, .\}$.

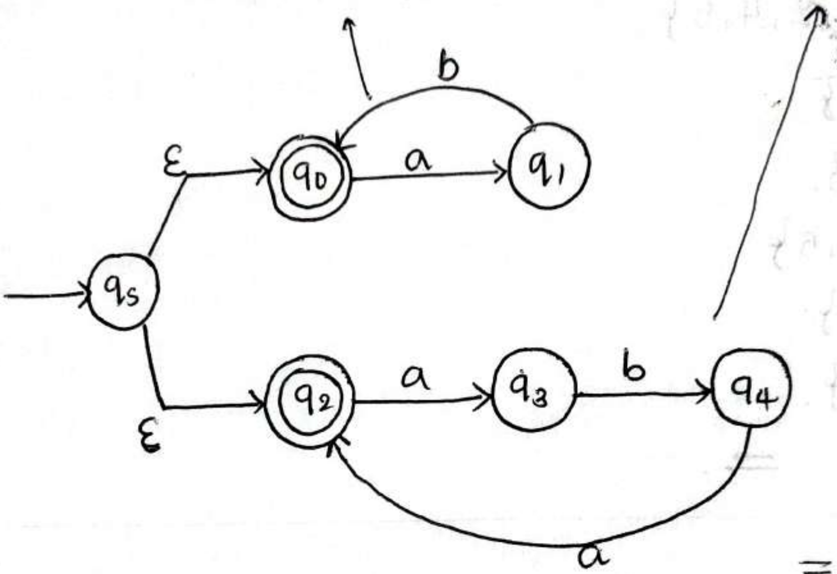
$L = \{0.23, 0.17, -123, 99.99, ----\}$.



=

3) Construct finite automata to accept (ab, aba)*

Soln: $L = \{\varepsilon, ab, abab, ababab, ----, aba, abaaba, abaabaaba-----\}$

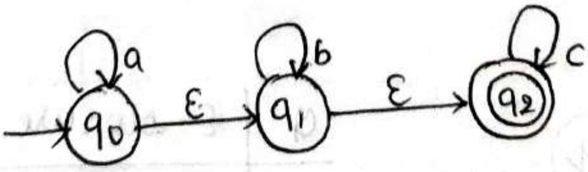


=

ε-Closure:-

ε-closure is the set of all states that are reachable from a particular state on an epsilon input either directly (or) indirectly

Ex: given,



here $q_i \in Q$.

Note: (i) ε-closure $(q_i) = \{q_i\}$

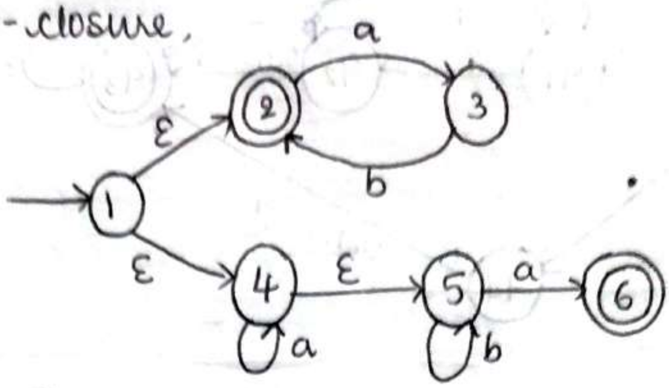
(ii) $\delta(q_i, \varepsilon) = q_j$, then ε-closure $(q_i) = \{q_i, q_j, \varepsilon\text{closure}(q_i)\}$.

Soln:

Q	ε-closure(Q)
q_0	$\{q_0, q_1, q_2\}$
q_1	$\{q_1, q_2\}$
q_2	$\{q_2\}$.

4) Write ε-closure.

given



Soln:-

Q	ε-closure(Q)
1	{1, 2, 4, 5}.
2.	{2}
3	{3}.
4	{4, 5}
5	{5}
6	{6}.

=.

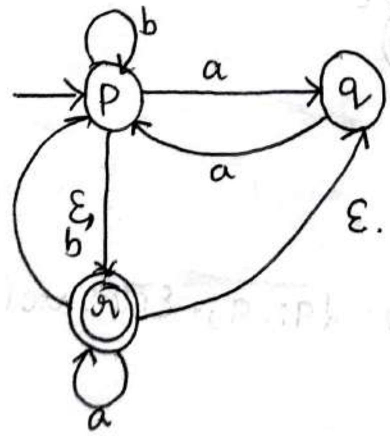
---

5). given,

δ	ε	a	b
→ p	{r}	{q}	{p, r}
q	∅	{p}	∅
* r	{p, q}	{r}	{p}

Soln:-

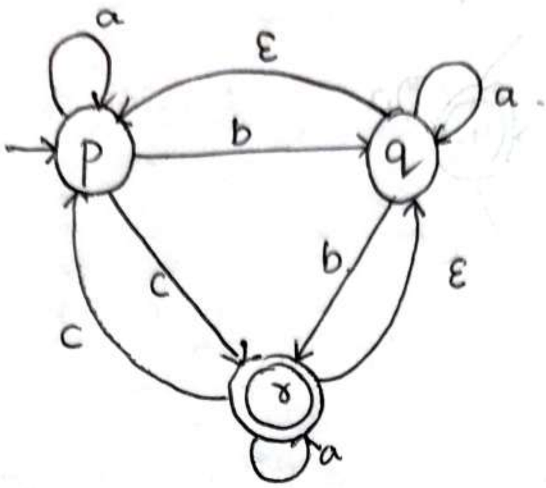
(If required)



Q	ε-closure (Q)
p	{p, r, q}.
q	{q}.
r	{r, p, q}.

=.

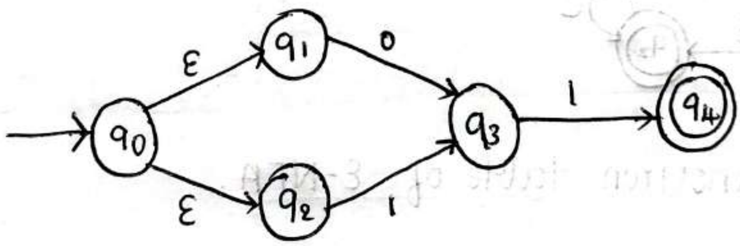
6) given,



Soln:

Q	ε-closure
p	{p}
q	{q, p}
r	{r, q, p}

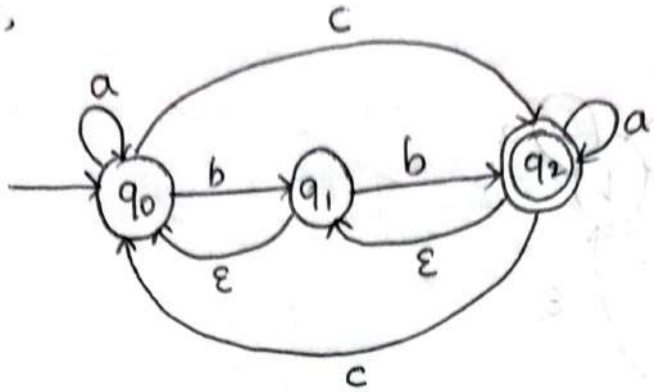
7) given,



Soln:

Q	ε-closure
q_0	$\{q_0, q_1, q_2\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3\}$
q_4	$\{q_4\}$

8) given,

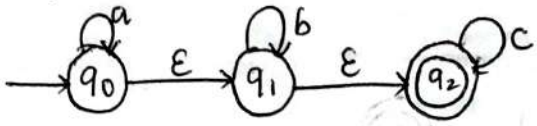


<u>Soln:</u>

Q	ε-closure
q_0	$\{q_0\}$.
q_1	$\{q_1, q_0\}$.
q_2	$\{q_2, q_1, q_0\}$.

Conversion of ε-NFA to DFA:-

1). given,



Soln:- Step 1:- write transition table of ε-NFA.

	ε	a	b	c
→ q_0	q_1	q_0	ϕ	ϕ
q_1	q_2	ϕ.	q_1	ϕ.
* q_2	ϕ	ϕ	ϕ	q_2

Step 2:- Find ε-closure of all states.

Q	ε-closure
q_0	$\{q_0, q_1, q_2\}$
q_1	$\{q_1, q_2\}$
q_2	$\{q_2\}$

Step 3: Identify the $q_{0\,DFA}$.

$$q_{0_{DFA}} = \varepsilon\text{-closure}(q_{0_{\varepsilon\text{-NFA}}}).$$

$$= \varepsilon\text{-closure}(q_0).$$

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

$$= A.$$

Consider A,

$$\delta(A, a) = \varepsilon\text{-closure}(\delta(A, a))$$

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

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

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

$$= \varepsilon\text{-closure}(q_0).$$

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

$$\delta(A, b) = \varepsilon\text{-closure}(\delta(A, b)).$$

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

$$= \varepsilon\text{-closure}(\phi \cup q_1 \cup \phi).$$

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

$$\delta(A, c) = \varepsilon\text{-closure}(\delta(A, c)).$$

$$= \varepsilon\text{-closure}(\delta(\{q_0, q_1, q_2\}, c)).$$

$$= \varepsilon\text{-closure}(\phi \cup \phi \cup q_2\}$$

$$= q_2 = C.$$

Consider B,

$\delta(B,a) = \varepsilon\text{-closure}(\delta(B,a))$.

$= \varepsilon\text{-closure}(\delta\{q_1,q_2\},a)$

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

$= \phi = D$.

$\delta(B,b) = \varepsilon\text{-closure}(\delta(B,b))$.

$= \varepsilon\text{-closure}(q_1 \cup \phi)$

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

$\delta(B,c) = \varepsilon\text{-closure}(\delta(B,c))$.

$= \varepsilon\text{-closure}(\phi \cup q_2)$.

$= q_2 = E$.

Consider C,

$\delta(C,a) = \varepsilon\text{-closure}(\delta(C,a))$.

$= \phi = D$.

$\delta(C,b) = \varepsilon\text{-closure}(\delta(C,b))$

$= \phi = D$

$\delta(C,c) = \varepsilon\text{-closure}(\delta(C,c))$.

$= \varepsilon\text{-closure}(\delta(q_2, c))$.

$= q_2 = C$

Consider D,

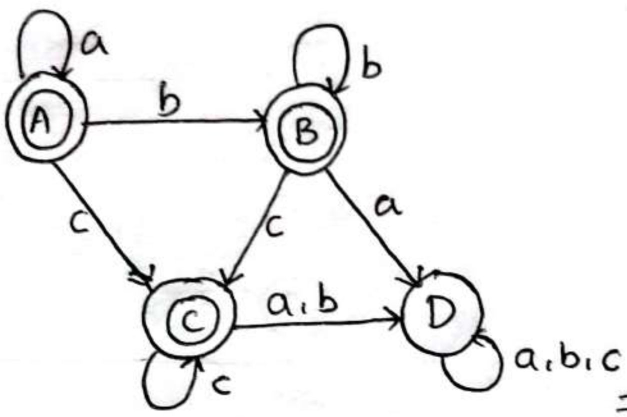
$\delta(D,a) = D$.

$\delta(D,b) = D$

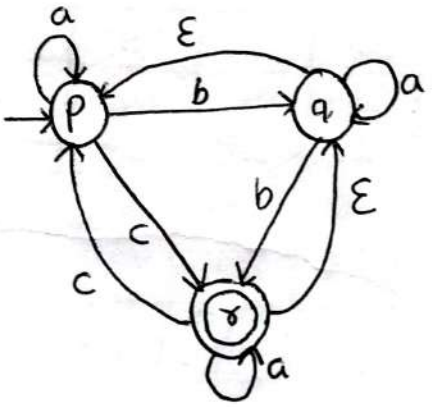
$\delta(D,c) = D$.

Transition table for DFA.

	a	b	c
→A*	A	B	C
*B	D	B	C
*C	D	D	C
D	D	D	D.



2) given,



Soln:- transition table of ε-NFA.

	ε	a	b	c
→p	∅.	p	q	r
q	p	q	r	∅
*r	q	r	∅	p

ε-closure.

q	ε-closure
→ p	$\{p\}$
q	$\{q, p\}$
r	$\{r, q, p\}$

$$q_{0_{DFA}} = \varepsilon\text{-closure}(q_{0_{\varepsilon\text{-NFA}}}).$$

$$= \{p\} = A.$$

Consider A,

$\delta(A, a) = p = A.$

$\delta(A, b) = q = B = \{q, p\}.$

$\delta(A, c) = r = C = \{r, q, p\}.$

Consider B,

$\delta(B, a) = \varepsilon\text{-closure}(\delta(\{q, p\}, a))$

$= \varepsilon\text{-closure}(q \cup p\}.$

$= \{q, p\} = B$

$\delta(B, b) = C.$

$\delta(B, c) = \{r, p, q\} = C.$

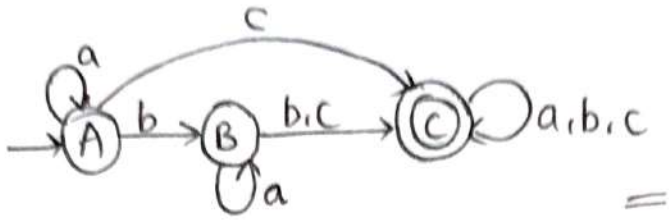
Consider C,

$\delta(C, a) = \{p, q, r\} = C.$

$\delta(C, b) = \{q, r, p\} = C$

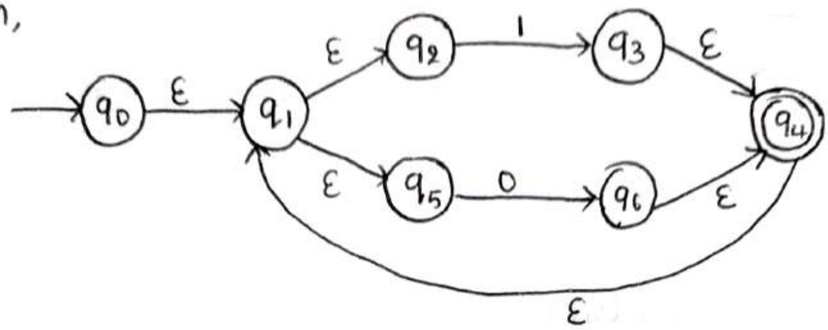
$\delta(C, c) = \{p, q, r\} = C.$

	a	b	c
→ A	A	B	C
B	B	C	C
C	C	C	C



10/10/2024

3) Given,



Soln:- Transition table for ε-NFA:-

	ε	0	1
→ q_0	q_1	ϕ	ϕ
q_1	$\{q_2, q_5\}$	ϕ	ϕ
q_2	ϕ	ϕ	q_3
q_3	q_4	ϕ	ϕ
* q_4	q_1	ϕ	ϕ
q_5	ϕ	q_6	ϕ
q_6	q_4	ϕ	ϕ

ε-closure:

Q	ε-closure
q_0	$\{q_0, q_1, q_2, q_5\}$
q_1	$\{q_1, q_2, q_5\}$
q_2	$\{q_2\}$
q_3	$\{q_3, q_4, q_1, q_2, q_5\}$
q_4	$\{q_4, q_1, q_2, q_5\}$
q_5	$\{q_5\}$
q_6	$\{q_6, q_4, q_1, q_2, q_5\}$

Start state:

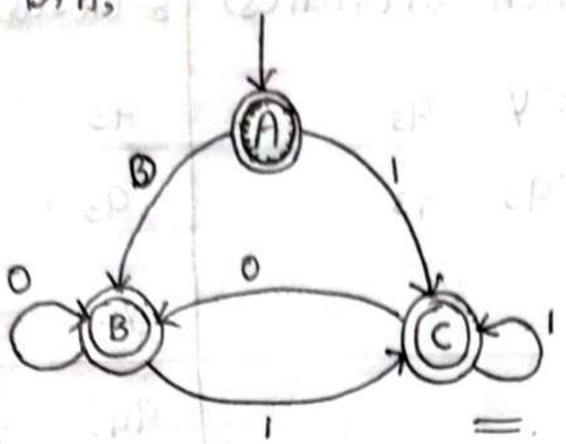
$\varepsilon\text{-closure}(q_{0_{\varepsilon\text{-NFA}}}) = \varepsilon\text{-closure}(q_0)$.

$= \{q_0, q_1, q_2, q_5\}$

$= A$.

Q_{DFA}	Σ	Transition Function (δ)	ε-closure	Rena
A	0	$\delta(A,0) = \phi \cup \phi \cup \phi \cup q_6 = q_6$	$\{q_6, q_4, q_1, q_2, q_5\}$	B
	1	$\delta(A,1) = \phi \cup \phi \cup q_3 \cup \phi = q_3$	$\{q_3, q_4, q_1, q_2, q_5\}$	C
B	0	$\delta(B,0) = \phi \cup \phi \cup \phi \cup \phi \cup q_6 = q_6$	$\{q_6, q_4, q_1, q_2, q_5\}$	B
	1	$\delta(B,1) = \phi \cup \phi \cup \phi \cup q_3 \cup \phi = q_3$	$\{q_3, q_4, q_1, q_2, q_5\}$	C
C	0	$\delta(C,0) = \phi \cup \phi \cup \phi \cup \phi \cup q_6 = q_6$	$\{q_6, q_4, q_1, q_2, q_5\}$	B
	1	$\delta(C,1) = \phi \cup \phi \cup \phi \cup q_3 \cup \phi = q_3$	$\{q_3, q_4, q_1, q_2, q_5\}$	C

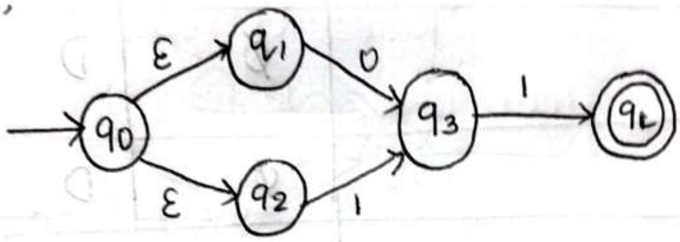
required DFA,



Transition table.

	0	1
A	B	C
B	B	C
C	B	C

4) Given,



Soln: Transition table for ε-NFA:-

	ε	0	1
→ q_0	$\{q_1, q_2\}$	ϕ	ϕ.
q_1	ϕ	q_3	ϕ
q_2	ϕ	ϕ	q_3
q_3	ϕ	ϕ	q_4
* q_4	ϕ	ϕ	ϕ

ε-closure:

Q	ε-closure
q_0	$\{q_0, q_1, q_2\}$
q_1	$\{q_1\}$
$\{q_2\}$	$\{q_2\}$
$\{q_3\}$	$\{q_3\}$
$\{q_4\}$	$\{q_4\}$

Start state:-

ε-closure ($q_{0_{\varepsilon\text{-NFA}}}$) = $\{q_0, q_1, q_2\}$.

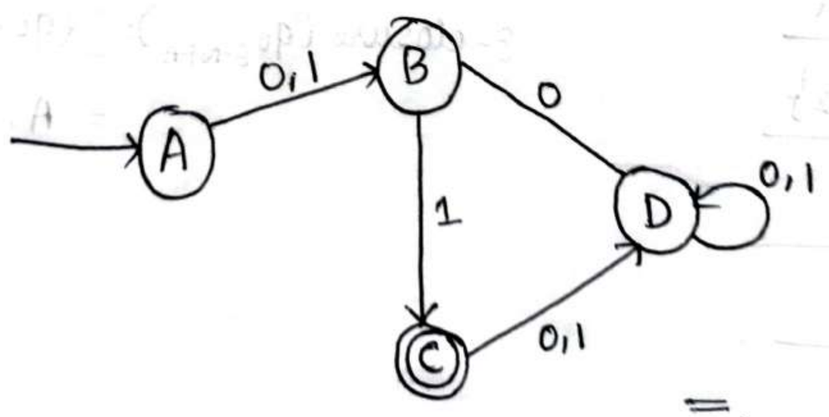
= A.

Q_{DFA}	Σ	Transition Function (δ)	ε-closure	Rename
A.	0	$\delta(A,0) = \phi \cup q_3 \cup \phi = q_3$	q_3	B.
	1	$\delta(A,1) = \phi \cup \phi \cup q_3 = q_3$	q_3	B
B	0.	$\delta(B,0) = \phi$.	ϕ.	D.
	1	$\delta(B,1) = q_4$	q_4.	C
C.	0	$\delta(C,0) = \phi$	ϕ	D
	1	$\delta(C,1) = \phi$	ϕ	D
D	0	$\delta(D,0) = \phi$	ϕ	D
	1	$\delta(D,1) = \phi$	ϕ	D.

Transition table for DFA.

	0	1
$\rightarrow$ A	B	B
B	D	C
* C	D	D
D	D	D

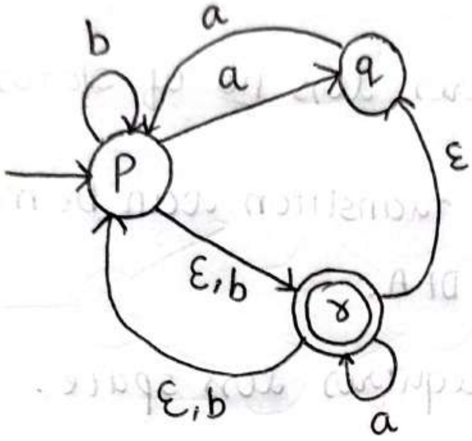
DFA required is



5). Given,

δ	ε	a	b
$\rightarrow p$	$\{r\}$	$\{q\}$	$\{p, r\}$
q	ϕ	$\{p\}$	ϕ
$* r$	$\{p, q\}$	$\{r\}$	$\{p\}$

Soln:-



ε - closure.

p	$\{p, r, q\}$
q	$\{q\}$
r	$\{r, p, q\}$

Start state.

$\varepsilon\text{-closure}(p) = \{p, r, q\} = A$.

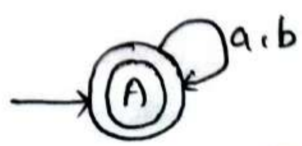
Q_{DFA}	Σ	Transition function (δ)	ε-closure	Rename
A	a	$\delta(A, a) = q \cup p \cup r = \{p, q, r\}$	$\{p, q, r\}$	A.
	b.	$\delta(A, b) = \{p, r\}$	$\{p, q, r\}$	A

Transition table:-

	a	b
$\rightarrow A^*$	A	A

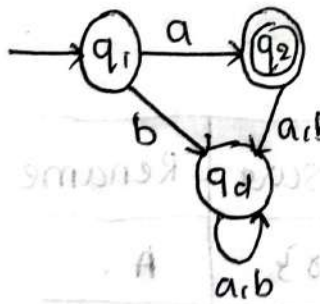
required DFA

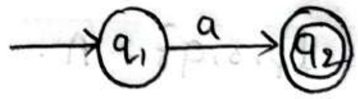
$\Rightarrow$



Difference between DFA and NFA:-

DFA.	NFA.
* Transition ~~DFA~~ is deterministic.	Transition is non-deterministic.
* Difficult to design	Easy to design.
* Requires more no of states	Requires less no of states.
* No of transitions are less.	No of transition can be more than DFA.
* It requires more space	It requires less space.
* Transition on empty string is not possible	Transition on empty string is possible.
* Practical implementation of DFA is possible.	Practical implementation is not possible in NFA.
* Ex: DFA to accept only 'a' over {a,b}.	Ex: NFA to accept only 'a' over {a,b}





Applications:

* Pattern matching
* Natural language processing
* Machine Learning

x