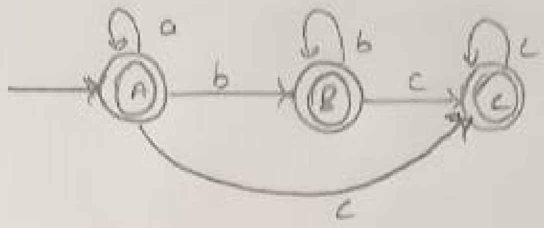
Consider the State E (q_3)

on input a $\delta(q_3, a) = \phi$

on input b $\delta(q_3, b) = \phi$

on input c $\delta(q_3, c)$ = Eclose (q_3) = {q_3} (C).

δ	a	b	c
*A	A	B	C
*B	ϕ	B	C
*C	ϕ	ϕ	C



→ The ε- closure of q denoted by Eclose (q) is the set of all state which are reachable from q on ε-transition only it is recursively defined as. state of is in Enclose (q) i.e., Eclose (q) = q* if Eclose (q) continue pond if there is a transition from state p to state r labled ε, then state is also in Eclose (q)

$\epsilon close(q_0) = \{q_0, q_1, q_2\}$ .. A

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

on input 0

$\delta(\{q_0, q_1, q_2\}, 0) = \epsilon close(\{q_0\})$.

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

On input 1

$\delta(\{q_0, q_1, q_2\}, 1) = \epsilon close(q_1)$

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

on input 2

$\delta(\{q_0, q_1, q_2\}, 2) = \epsilon close(\{q_2\})$.

$= \{q_2\}$ (C)

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

on input 0

$\delta(q_1, q_2) 0) = \epsilon close(q_1, q_2) = \phi$

on input 0

$\delta(q_1, q_2), 0\} = \epsilon close((q_1, q_2), 0) = \phi$

on input 1

$\delta(\{q_1, q_2), 1) = \epsilon close(\{q_1\})$.

$= \{q_1, q_2\}$.

on input 2 $= \delta(\{q_1, q_2, 2) = \epsilon close(\{q_2\})$

$= \{q_2\}$ (C).

(C.G) (D.G) (B.D)

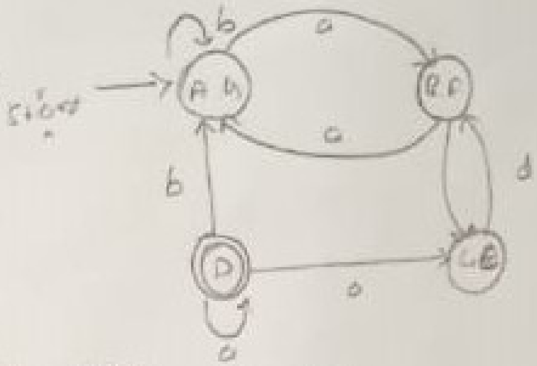
(E.F) (D.G) ~~(B~~(F.E)

(F.G) (D.F) (F.G)

(E.H) (G.F) (E.G)

(F.G) (G.G) (E.D)

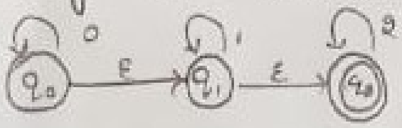
(G.H) (F.G) (D.G)



11) Give the Difference b/w DFA, NFA & ε NFA

DFA	NFA	ε-NFA
1) The DFA is 5-tuple $m = (Q, \Sigma, \delta, q_0, F)$ where Q is finite state, Σ is Alphabet set of input, $\delta : Q \times \Sigma$ to Q, q_0 : Start state, F : final state	1) An NFA is 5-tuple $(\Sigma, Q, \delta, q_0, F) = m$ where Q infinite state, Σ is alphabet set of input, $\delta : Q \times \Sigma$ to 2^Q, q_0 : start state	1) An ε-NFA is 5-tuple $m = (Q, \Sigma, \delta, q_0, F)$ where Q is finite set, Σ is Alphabet set of input, $\delta = Q \times (\Sigma \cup \epsilon) \to 2^Q$, q_0 = start state
2) less no of transition	2) ~~Relatively~~ less no of transition	2) Relatively more transition when compared with NFA
3) Difficult to construct	3) Easy to construct	3) Easy to construct using regular expression.

12) Define ε-closure and also find the ε-closure for the following & convert it to DFA,



10b obtain the distinguishable table for automation and then minimise the state of following DFA.

δ	a	b
A	B	A
B	A	C
C	D	B
D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

	A	B	C	D	E	F	G
B							
C							
D	*	*	*				
E				×			
F				×			
G				×			
H				×			

δ	a	b
(A.B)	(A.B)	(A.C)
(A.C)	(B.D)	(A.B)
(A.E)	(B.D)	(A.F)
(A.F)	(B.G)	(A.E)
(A.G)	(B.F)	(A.G)
(A.H)	(B.G)	(A.D)
(B.C)	(A.D)	(C.F)
(B.F)	(A.G)	(C.E)
(B.G)	(A.F)	(C.G)
(B.H)	(A.G)	(C.E)
(B.G)	(A.F)	(C.E).
(B.H)	(A.G)	(C.E).
(C.E)	(D.D)	(B.F)
(C.F)	(D.G)	(B.E)

	A	B	C	D	E	F	G	H
B								
C	*	*						
D	*	*	*					
E	*	*		*				
F			*	*	*			
G			*	*	*			
H	*	*	*	*	*	*	*	*

δ	0	1
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	q_2	q_2

→ q_0 is the start state of NFA $[q_0]$ is the state of DFA

→ $\Sigma = \{0, 1\}$

for state $\{q_0\}$ I.S = 0

$\delta_D(\{q_0\}, 0) = \{q_0, q_1\}$

$I_1 = 1$

$\delta_D(\{q_0\}, 1) = \{q_0\}.$

for state $\{q_0, q_1\}$

I.S = 0

$\delta_D(\{q_0, q_1\}, 0) = \delta_N(\{q_0, q_1\}, 0)$

$= \delta_N(\{q_0\}, 0) \cup (\{q_1\}, 0)$

$= \{q_0, q_1\} \cup \{\phi\}$

$= \{q_0, q_1\}.$

For state $\{q_0, q_1, q_2\}$: $I_S = 0$

$\delta_D(\{q_0, q_1, q_2\}, 0) = \delta_N(\{q_0, q_1, q_2\}, 0)$

$= \delta_N(\{q_0\}, 0) \cup \delta_N(\{q_1\}, 0) \cup \delta_N(\{q_2\}, 0)$

$= \{q_0, q_1\} \cup \{\phi\} \cup \{q_2\}.$

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

$I_S = 1$

$\delta_D(\{q_0, q_1, q_2\}, 1) = \delta_N(\{q_0, q_1, q_2\}, 1)$

$= \delta_N(\{q_0\}, 0) \cup \delta_N(\{q_1\}, 1) \cup \delta_N(\{q_2\}, 1)$

$= \{q_0\} \cup \{q_2\} \cup \{q_2\}.$

$= \{q_0, q_2\}.$

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

$I = \{0, 1\}$

q_0 = start state

$F_S = \{(q_0, q_1), q_2\}.$

$= \phi \cup q_2$

$= \{q_0, q_2\}$

for state $\{q_0, q_2\}$

$I.S = 0$

$\delta_D(\{q_0, q_2\}, 1) = \delta_N(\{q_0, q_2\}, 1)$

$= \delta_N(\{q_0\}, 1) \cup \delta_N(\{q_2\}, 1)$

$= \{q_0\} \cup \{q_2\}$

$= \{q_0, q_2\}$

for state $\{q_1, q_2\}$

$I.S = 0$

$\delta_D(\{q_1, q_2\}, 0) = \delta_N(\{q_1, q_2\}, 0)$

$= \delta_N(\{q_1\}, 0) \cup \delta_N(\{q_2\}, 0)$

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

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

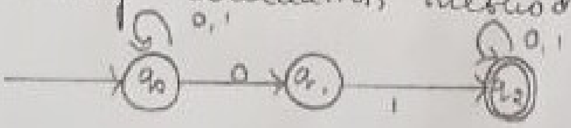
$I.S = 1$

$\delta_D(\{q_1, q_2\}, 1) = \delta_N(\{q_1, q_2\}, 1)$

$= \delta_N(\{q_1\}, 1) \cup \delta_N$

δ	0	1
ϕ	ϕ	ϕ
q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{\phi\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_2\}$
(q_0, q_1)	$\{q_0, q_1\}$	$\{q_0, q_2\}$
(q_0, q_2)	$\{q_0, q_1, q_2\}$	(q_0, q_2)
(q_1, q_2)	$\{q_2\}$	$\{q_2\}$
(q_0, q_1, q_2)	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$.

iii) Lazy evaluation method



9) convert the following NFA to DFA
i) by subset construction.
ii) by Lazy evaluation method.

i) subset construction.

δ	0	1
q_0	$\{q_0, q_1\}$	q_0
q_1	ϕ	q_2
q_2	q_2	q_2

Step 1: q_0 is start state.
Step 2: $\Sigma = \{0, b\}$
Step 3: $Q_N = \{q_0, q_1, q_2\}$

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

Step 4:- final state from above subset.

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

For state ϕ.

Input symbol = 0   Input symbol = 1

$\delta_D(\phi, 0) = \phi$ $\delta_N(\phi, 1) = \phi$

For state q_0

Input symbol = 0   Input = 1

for state $\{q_0, q_1\}$ $\delta_D{}''(q_0, 1) = \{q_0\}$

$\delta_D(\{q_0, q_1\}, 0) = \delta_N(\{q_0, q_1\}, 0)$

$= \delta_N(\{q_0\}, 0) \cup \delta_N(\{q_1\}, 0)$

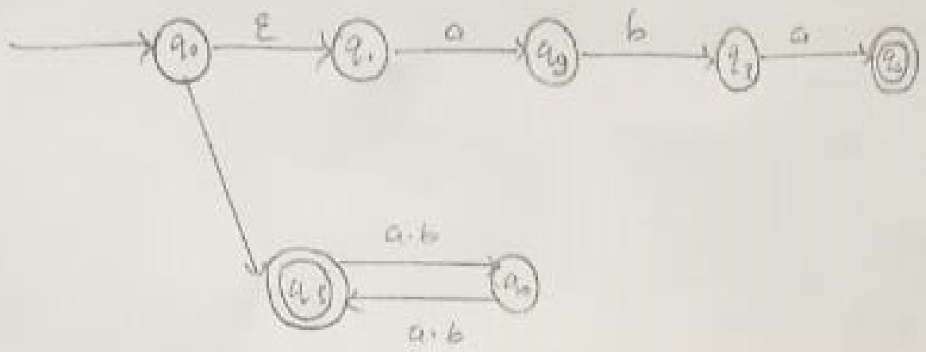
$= \{q_0, q_1\} \cup \phi$

$= \{q_0, q_1\}$.

$\therefore$ $I_S = 1$

$\delta_D(\{q_0, q_1\}, 1) = \delta_N(\{q_0, q_1\}, 1)$

$= \delta_N(\{q_0\}, 1) \cup \delta_N(\{q_1\}, 1)$



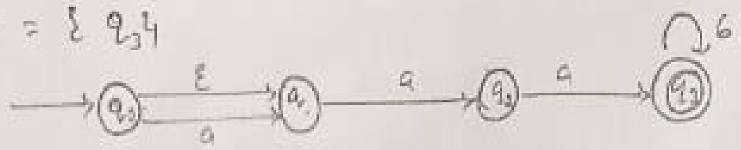
δ	a	b	ϵ
q_0	ϕ	ϕ	$\{q_0, q_5\}$
q_1	q_2	ϕ	ϕ
q_2	ϕ	q_3	ϕ
q_3	q_4	ϕ	ϕ
q_4	ϕ	ϕ	ϕ
q_5	q_6	q_6	ϕ
q_6	ϕ	ϕ	ϕ

ii) $L = \{w \in \{a, b\}^*$: w is made up of an ~~optional~~ optional of a followed by zero 'a' & more 'b's.

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

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

$F = \{q_3\}$



δ	a	b	ϵ
q_0	q_1	ϕ	q_1
q_1	q_2	ϕ	ϕ
q_2	q_3	ϕ	ϕ
q_3	q_3	q_3	ϕ

Start state = q_0 final state = q_1.

x) $L = \{ w : w \in \{a..z\}^*$ all five vowels a, e, i, o, u, across in alphabet order $\}$

→ $\Sigma = \{a\}$' means all the elements of Σ accept a' & let label 'Σ' mean all elements of Σ.

$Q = \{S, A, E, I, O, U, y_u\}$

S.t $= \{S\}$

final state $= \{y_u\}$

$\Sigma = \{a, e, i, o, u\}$

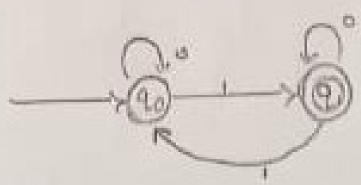
δ	a	e	i	o	u
S	A	φ	φ	φ	φ
A	φ	e	φ	φ	φ
e	φ	φ	i	φ	φ
i	φ	φ	φ	o	φ
o	φ	φ	φ	φ	u
y_u	A	φ	φ	φ	φ

xi) $L = \{ w \in \{a, b\}^* : w$ has odd parity $\}$

$\Sigma = \{0, 1\}$ $Q = \{q_0, q_1\}$

F.S $= \{q_1\}$

S.S $= \{q_0\}$



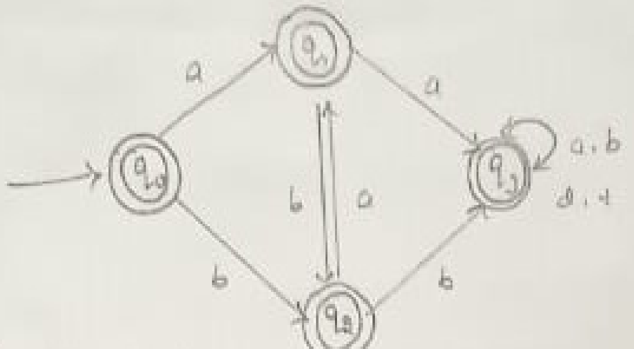
δ	0	1
q_0	q_0	q_1
q_1	q_1	q_0

8) Design the ε-non deterministic finite automata for the following $L = \{ w \in \{a, b\}^* : w = aba$ or $[w]$ is even$\}$

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

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

$F = \{q_4, q_6\}$



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

Start state $= q_0$

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

d. s $\{q_3\}$.

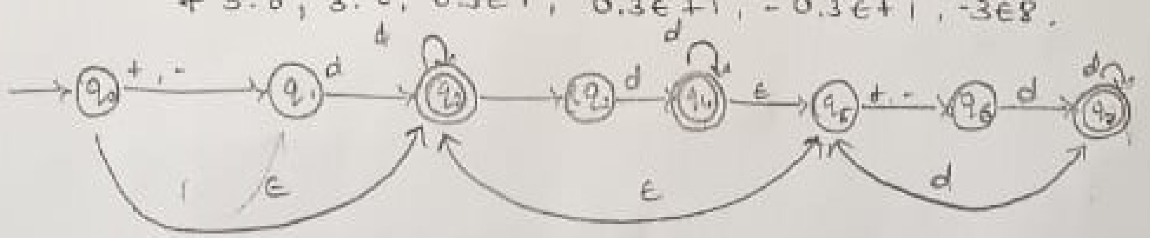
δ	a	b
q_0	q_1	q_2
q_1	q_3	q_2
q_2	q_1	q_3
q_3	q_3	q_3

Ex) $L = \{w : w$ is the string representing a floating point number$\}$

* A floating point number is an optional sign, followed by a decimal number followed by an optional imponent.
* A decimal number may be of the form x or x.y where x & y are non-empty string of decimal digits.
* An integer is a non-empty string of decimal digits

so. for e.g these strings represents floating point numbers

+3.0, 3.0, 0.3E1, 0.3E+1, -0.3E+1, -3E8.



$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}$ $\Sigma = \{+, -, d, E\}$

In this we have used the short hand ~~to~~ d to stand for any -one of the decimal digits (0,-9) & we have omitted the dead state to avoid arrows crossing each other.

δ	a	b
q_0	q_1	q_3
q_1	q_2	q_1
q_2	q_2	q_1
q_3	q_3	q_3

ds

VIII DFA to accept string of a's & b's having odd number of a's and even no of b's

a_a = odd no of a's

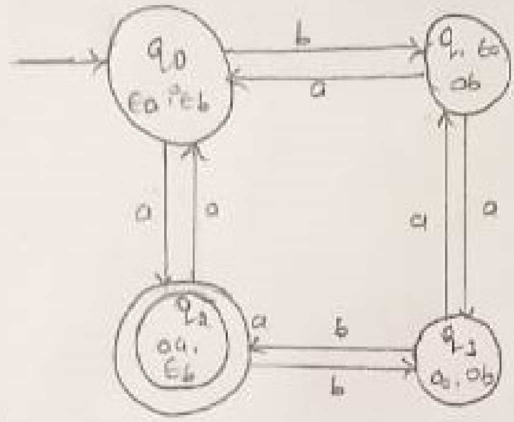
eb = even no of b's

O_b = odd no of b's

O_a = Even no of a's

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

$Q = \{(ea, eb), (ea, Ob), (O_a, Eb)$
(O_a, Ob)



$F = \{O_a, E_b\}$

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

VIII: $L = \{w + \{a, b\}$: no two consecutive characters are same

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

ab, ba.

v) $L = \{w : |w| \bmod 4 \neq 0\}$ where $\Sigma = \{a, b\}$.

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

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

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

$\delta(q_0, a) = q_1$ $\quad \delta(q_1, (a.b)) = q_2$ $\quad \delta(q_3 (a.b)) = q_0$

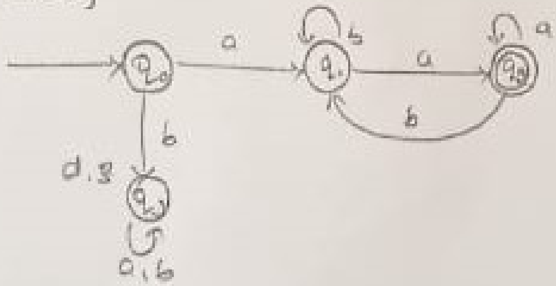
$\delta(q_0, b) = q_1$ $\quad \delta(q_2 (a.b)) = q_3$.

δ	a	b
q_0	q_1	q_1
q_1	q_2	q_2
q_2	q_3	q_3
q_3	q_0	q_0

vi) $L = \{awa \mid w \in (a+b)^*\}$

$S = \{aa, aaa, aba, abbababa...\}$

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



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

$F = \{q_2\}$.

$\delta(q_0, a) = q_1$ $\quad \delta(q_1, a) = q_2$ $\quad \delta(q_2, a) = q_2$

$\delta(q_0, b) = q_3$ $\quad \delta(q_1, b) = q_1$ $\quad \delta(q_2, b) = q_1$

$\delta(q_3, a) = q_3$

$\delta(q_3, b) = q_3$.

* Minimum string $\delta = \{ab^* \text{ or } ba\}$.

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

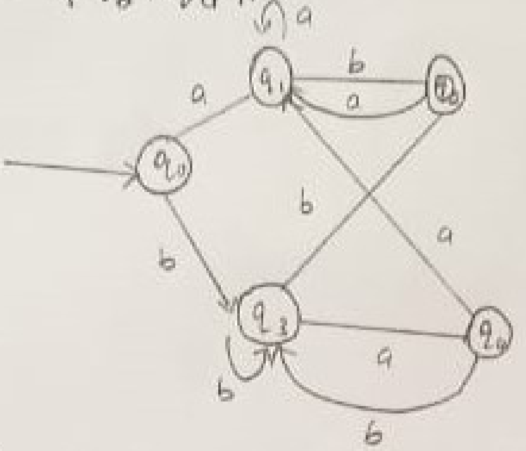
$*J = \{Q.\Sigma, \delta, q_0, F\}$

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

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

q_0 = strate late

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



δ	a	b
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_1	q_3
q_2	q_4	q_3
q_4	q_1	q_5

iv) $L = \{w : Na(w) \leq 3, w \in \{a.b\}^*\}$

$\Sigma = (a.b)$.

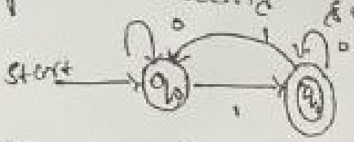
$Q = \{q_0, q_1, q_2, q_3, q_4\}$

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

q_0 = starting state.

δ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_4	q_3
2.5 q_4	q_4	q_4

For e.g:- it means sense to implement parity checking FSM.



odd parity

* An FSM can be simulated by general purpose interpreter.
* An FSM can be used as a specification for can be implemen -ted in ~~shown~~ software just as any specification might be and the correctness of the implementation can be shown by verify -ng that the implementation satisfies the specification.

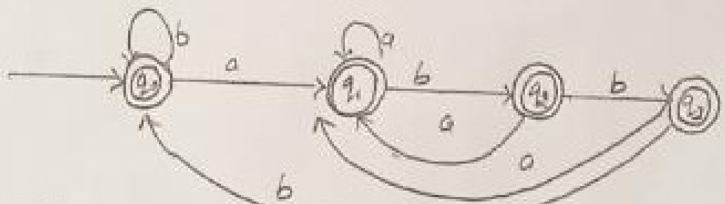
7) Design a DFA for the following.

i) DFA to accept string of a's and b's not ending with string abb

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

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

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



$\delta(q_0, a) = q_1$
$\delta(q_0, b) = q_0$
$\delta(q_3, a) = q_1$
$\delta(q_3, b) = q_0$

$\delta(q_1, a) = q_1$
$\delta(q_1, b) = q_2$

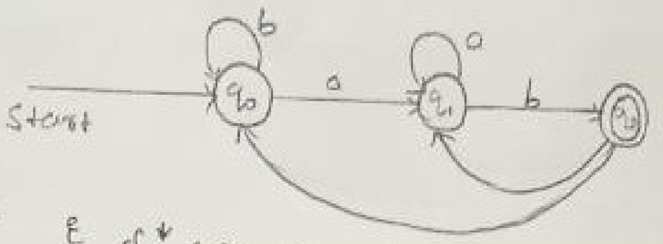
$\delta(q_2, a) = q_1$
$\delta(q_2, q_0) = q_3$.

δ	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

ii) $L = \{w(ab+ba) \; w \in \{a, b\}^*\}$

iii) DFA to accept strings of a's and b's ends with ab or ba.

Then $\delta'(q, w) = \delta^*(q, xa) = \{r_1, r_2, r_3 \dots r_k\}$



prefix ϵ $\delta^*(q_0, \epsilon) = \{q_0\}$

prefix a $\delta^*(q_0, a) = \delta(\delta^*(q_0, \epsilon), a)$
$= \delta(q_0, a)$
$= q_1$

prefix ab $\delta^*(q_0, ab) = \delta(\delta^*(q_0, a), b)$
$= \delta(q_1, b)$
$= q_2$

Prefix aba $\delta^*(q_0, aba) = \delta(\delta^*(q_0, ab), a)$
$= \delta(q_2, a)$
$= q_1$

prefix abaa $\delta^*(q_0, abaa) = \delta(\delta^*(q_0, aba), a)$
$= \delta(q_1, a)$
$= q_1$

Prefix abaab $\delta^*(q_0, abaab) = \delta(\delta^*(q_0, abaa), b)$
$= \delta(q_1, b)$
$= q_2$.

$\therefore$ The final transition.

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2$

6) catoegorize How FSM's are used in operational system?.

=> An FSM is an abstraction.

* FSMs for real problem can be formed into operational systems in any of number ways.
* An FSM can be translated into a circuit design and implemented directly in hardware.

→ ε.NFA: A transition which an empty string is called an ε-transition i.e., if there is a transition from one state to another state without any input is called ε-transition.

ε.NFA is 5-tuple

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

$\delta = Q \times (\Sigma \cup \varepsilon)$ to 2^Q i.e., δ is the transition function which is a mapping from $Q \times (\Sigma \cup \varepsilon)$ to 2^Q. Based on the current state there can be a transition to another states with a without any input symbols.

5> Define extended transition function for DFA and what are the transition made for string "aboab"?.

=> The extended transition function δ describe what happens to a state of machine when the input is a string.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be an DFA. The extended transition $\delta^* = Q \times \Sigma^*$ to 2^Q is defined recursively as shown below.

Basis: $\delta^*(q, \varepsilon) = \{q\}$ This indicates that if the machine is in state q and read no input then the machine is state q.

Induction: Let $w = xa$ when a is last symbol of w and x is the remaining string of w. Let q is the current state and x is the string to be processed & after consuming the string.

x, - Let the state of machine is $\{P_1, P_2, P_3 \ldots P_n\}$

i.e., $\delta^*(c, x) = \{P_1, P_2, P_3 \ldots P_n\}$.

Let the transition from $\{P_1, P_2, P_3 \ldots P_n\}$ on input symbol a is

$\delta(\{P_1, P_2, P_3 \ldots P_n\}, a) = \{r_1, r_2, r_3 \ldots r_k\}$.

i.e., $\bigcup_{i=1} \delta(P, a) = \{r_1, r_2, \ldots r_k\}$

→ Deterministic finite Automata.

finite state machine is computational device where input is a string and where output is one of two values that we can call accept or reject.

~~DFA~~: DFA:

The DFA in a Deterministic finite Automata is 5 triple or Quintiple indicating five components.

$M = (Q, \Sigma, \delta, q_0, F)$ where

M is the name of the machine it can be also be (5 tripeo) called by any name.

Q is non-empty finite set of state.

Σ is non-empty finite set of input alphabets. ~~alphates~~.

$\delta : Q \times \Sigma$ to Q i.e, δ is transition function which is a mapping from $Q \times \Sigma$ to Q Based on the component state and input symbol, the machine enters into another state

$q_0 \in Q$ - is the start state.

$F \subseteq Q$ - is set of accepting or final state.

A configuration of DFSM of M is ~~KxA~~ $K \times \Sigma^*$

NDFA :-

it is defined as 5 triple a Quintple indicating five components $M = (Q, \Sigma, \delta, q_0, F)$

where M is the name of machine

Q is non-empty, finite set of state

Σ is non-empty, finite set of input alphabet.

δ :- $Q \times \Sigma$ to 2^Q i.e, δ is transition function which is mapping from $Q \times \Sigma$ to 2^Q

$q_0 \in Q$ is the start state.

F is set of accepting or final state.

$\Sigma^+ = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$

$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001 \ldots\}$

$\{0.1\}^* = \{\varepsilon, 0, 1, 00, 10, 11, 000, 001 \ldots\}$ which is set of string of 0's is of any length.

3) Define Automata with an example?.

→ Theory of Automata is a theoritical branch of computer science and mathamatical it is the study of abstract machine and the computation problem that can be solved using these machanics.

computability: understanding the internal processing of the machine Abstract machine. is a conceptual model or theoritical model of a computer ~~hardworker~~ hardware are not actual machine & ~~th~~ hence. they are called hypothetical computer.

e.g:- $\Sigma = \{a, b\}$

$\Sigma = \{A, B, C, D\}$

$\Sigma = \{0, 1, 2\}$

$\Sigma = \{0, 1 \ldots 5\}$

These machines are studied with following methods.

1). finite Automata or finite state machanics.

2). Linear bounded Automata.

3). push down Automata.

4). Turning machine.

4). Define ~~Determinatio~~ Deterministic finite Automata Non Deterministic finite automata and ε-NFA?.

language:- A language can be defined as set of strings obtained from Σ^n where Σ is set of alphabets of a particular language. formally, a language 'L' over Σ is subset of Σ which is denoted by $L \subseteq \Sigma^*$

E.g:- $L = \{ \varepsilon, 01, 0011, 000111, \dots \}$

2) Define klene plus and kleene closure with an Examples.

→ kleene plus:- Σ^+

the kleene plus is a variation of kleene star operator. the kleene plus denoted by Σ^+ is defined as follows.

$$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

which is the set of words of any length except the null string i.e., $\varepsilon (\Sigma^n)$ is not part of Σ^+ and hence $\varepsilon \notin \Sigma^+$

for E.g:- let $\Sigma = \{0, 1\}$. Then it is shown below.

$\Sigma^+ = \{0, 1, 00, 01, 10, 11, 000, 001, \dots\}$

The above set is defined as of strings of 0's and 1's of any length except the null string.

Note $\Sigma^* = \Sigma^+ \cup \varepsilon$. this can be written as $\Sigma^+ \cdot \Sigma^+ = \Sigma^* - \varepsilon$.

kleene closure (or kleene star / star operator): Σ^n

The kleene closure it is defined as follows:

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup$ which is the set of words of any length Each string is made up of symbols only Σ.

E.g:- Let $\Sigma = \{0, 1\}$ then Σ^* is balanced

$\Sigma^0 = \{\varepsilon\}$ set of word's of length 0.

$\Sigma^1 = \{0, 1\}$ set of word's of length 1.

$\Sigma^2 = \{00, 01, 10, 11\}$ set of words of length 2.

$\Sigma^3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$ set of words of length.

Sub: Automata theory and compatibility
USN: 1DB18IS098
Sem: 5th
Sec: B

Assignment - I

1) Define Alphabet, String, power set, set, language with an example for each

→ Alphabet:- A latte language consists of various symbol from which the word and statement can be obtained. These symbols are called alphabets.

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

$\Sigma = \{a, b, \ldots, z, A, B, \ldots, Z, 0, 1 \ldots 9 \ldots\}$

String:- A string is defined as a finite sequence of symbols from an alphabet Σ.

* The shortest string is empty string & it is denoted by symbol ε

* The set of all possible string over an alphabet Σ is written as Σ^{*} for e.g:- $\Sigma = \{0, 1\}$ is set of binary op. alphabets

power set:- Let A be the set, the set of all subsets of set A is called power set of A and is denoted by P(A).

E.g:- Let $A = \{1, 2, 3\}$.

The subsets of set A are shown below.

$\{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

The set of above subset is called and is denoted by 2^A

$2^n = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \{\}\}$.