12/7/18

Formal language & Automatic Theory is called Theory of Automata

↳ before the Computers

4 languages :-

→ Regular language (RL)

→ context free language (CFL)

→ context sensitive language (CSL)

→ unrestricted language (UL)

⇒ In 1930's we got Turing Machine

1940 – 1950's → Finite Automata

late 1950's → N.chomsky

↳ given a hierarchy of all 4 languages

↳ He studied not only about machines but also about the Grammer

1st language → $\boxed{\text{Type : 3 Regular language}}$

↳ Finite Automata (basic Automata work on some extent)

$\boxed{\text{Type 2 : CFL}}$ used for Push down Automata (PDA) Machine

$\boxed{\text{Type 1 CSL}}$ → linear bounded Automata (LBA)

$\boxed{\text{Type 0 : UL Recursively Ennumerable language}}$ Turing machine

N. Chomsky Hierarchy

Type 0 : UL / REL	Turing machine
Type 1 : CSL	more than one stack Lenior bound Automata (LBA)
Type 2 : CFL	(stack) push down Automata (PDA) (compiler)
Type 3 : RL	Finite Automata (FA) doesn't have memory

(non empty)

FA :- A machine which is having a finite no of states which consist of input & output or accepts a string.

(Mathematical representation of a machine)

$Q \rightarrow$ set no. of states $\longrightarrow$ {ON/OFF}

$\Sigma \rightarrow$ set of inputs alphabet set {electricity}

$\delta \rightarrow$ transition fn $\longrightarrow$ (OFF) $\xrightarrow{\text{electricity}}$ ON

$\delta(\text{OFF}, \text{ele}) = \text{ON}$ $\quad \delta(q_0, \text{inp}) =$ not stable

{off} $q_0 \longrightarrow$ initial state

↳ from one state of i/p signal what is the next state

$F \longrightarrow$ final state {ON}

$$M = (\underbrace{Q, \Sigma, \delta, q_0, F}_{\text{Tuple}}) \rightarrow 5 \text{ Tuples}$$

no. of tuples for different Automata may vary

Alphabet → symbols

↳ represented by 'Σ'

→ uppercase Alphabet letters cannot be used as States

→ lowercase are taken as input

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

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

$\Sigma = \{+, -, *, /\}$.

Alphabet :- A finite non empty set of symbols

String :- combinations of letters ↘ set of Alphabets

→ powerset

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

Σ^0 = empty string (ϵ) ↗ epsilon

↳ string length should be '0'

$\Sigma^1 = \{a, b, c\}$ (Alphabet set)

$\Sigma^2 = \{ab, ac, aa$
$ba \quad bc \quad bb$
$ca \quad cb \cdot cc\}$

if any restriction is given then the string becomes finite

string are represented by 'w'

length of string : $|w|$

concadation : $x = aa$
$y = bb$
$xy = aabb$.

Language :- represented by 'L'

↳ set of strings.

L^* (star closure)

L^+ (positive closure)

The language which is represented by the input Σ including nullset is represented by L^*
→ the language which is generated the string of length '0'

$\boxed{L^* = L^0 \cup L^1 \cup L^2 \cup \ldots}$ (Infinite set)

which doesnot contain nullset is positive.

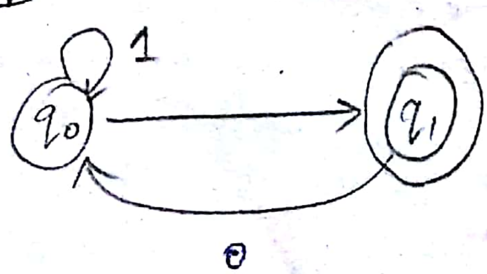
$\boxed{L^+ = L^1 \cup L^2 \cup L^3 \ldots}$

ϕ = empty language
ϵ = empty string
$\phi = \{\epsilon\}$

→ Finitestate machine
→ " expression
→ " Table

{It should contain 1 initial state and can have more finite states}

example :-



[path is there no input is ϵ]

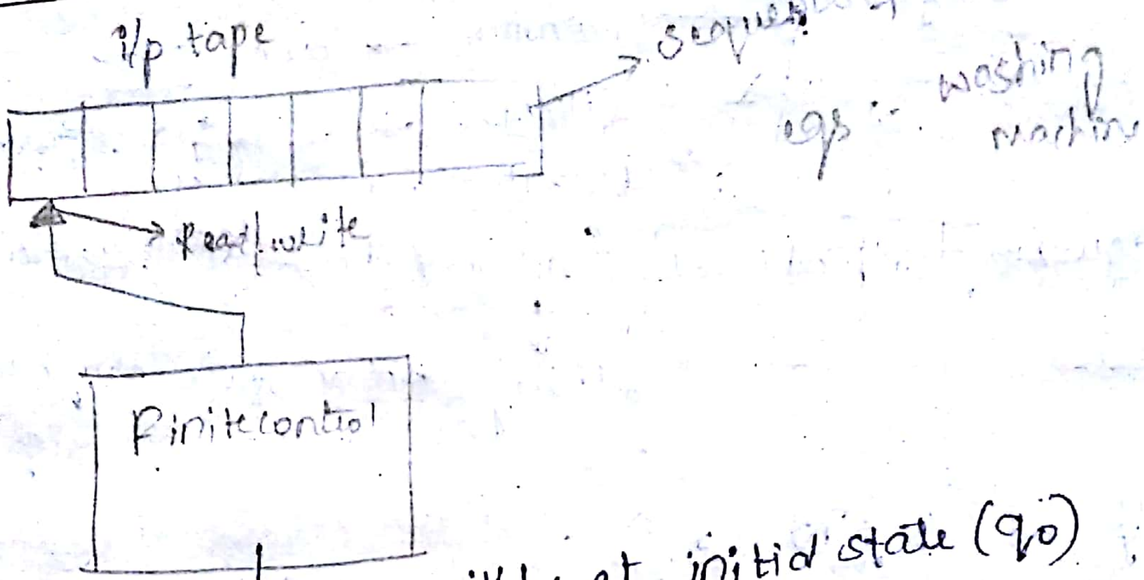
	q_0	q_1
0	ϕ	q_0
1	q_0	ϕ
ϵ	q_1	ϕ

3/07/18

TOA :- which is a machine it consits of a no. of states state of input and outputs and also the Transition function is called TOA. which contain output / string

Language is a collection/set of strings

Finite :-



it will be at initial state (q_0)

with this (q_0) what will be the next state

Finite Control: from that state with the i/p signal what will be the next state

FA (Finite Automata)

- without o/p
  - Deterministic finite automata
  - Non-D FA / NDFA / NFA
- with o/p
  - Mealy M/c
  - Moore M/c

DFA :-

It can be determined (represented) by Transition Diagram & Transition table. (graphical representation of automata)

Transition Diagram
- → state → O
- → Transition → arcs ⇄
- → initialstate → $\rightarrow q_0$ halfcut the incoming arrow towards the state
- → Finalstate → (F)

incoming halfcut arrow (initialstate)
↓
always initial state may not be q_0

eg: $M(Q, \Sigma, \delta, q_3, q_5)$
q_3 ↓ initial, q_5 ↓ final.

→ Construct a finite Automata which accepts the string ending with "aa".

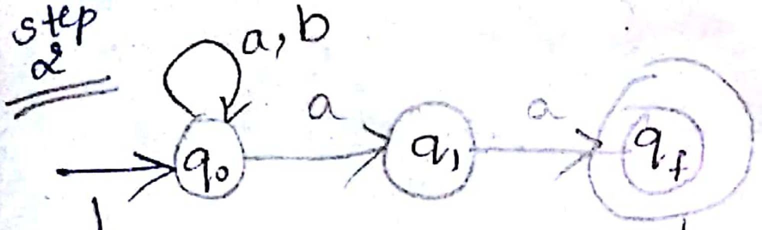
$M = (Q, \{a,b\}, \delta, q_0, q_f)$

→ no. of inputs $\{a,b\} = 2 = \Sigma$

→ w {(strings)}

Step 1:- $L = \{aa, aaa, baa \cdots\}$

step 2



length of substring = 2
it requires 3 states

initialstate
(halfand arrow)

finalstate
two circles.

These are not undirected graph. $w = abbcabaa$
They should be directed

For 'n' strings we will be having $(n+1)$ no. of states

$\boxed{\text{Extended Transition fn. is } \hat{\delta}(q_0, w)}$

⇒ construct FA which accepts the string with substring as "01"

01 → inputs (no. of input) = 2

1:- $W = \{01, 001, 011, \ldots\}$

we need to take only one i/p state

$\delta(q_0, a) = (q_0)$

$\delta(q_0, b) = (q_0)$

$\delta(q_0, b) = (q_0)$

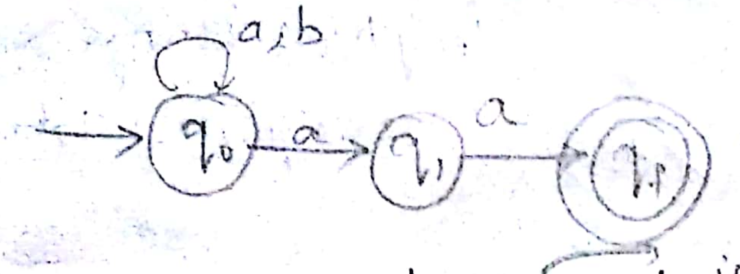
$\delta(q_0, a) = (q_0)$

$\delta(q_0, b) : (q_0)$

$\delta(q_0, a) = (q_0) \rightarrow (q_1)$

$\overset{q_1}{\delta}(q_0, a) = (q_0)\ (q_f)$ ✓ F.S.

but does not reached to F.S.



→ Transition diagram

row (states)

	a	b
q_0	$\{q_0, q_1\}$	q_0
q_1	q_f	ϕ
q_f	ϕ	ϕ

inputs (columns)

Transition table

$\boxed{\delta : Q \times \Sigma \to Q}$

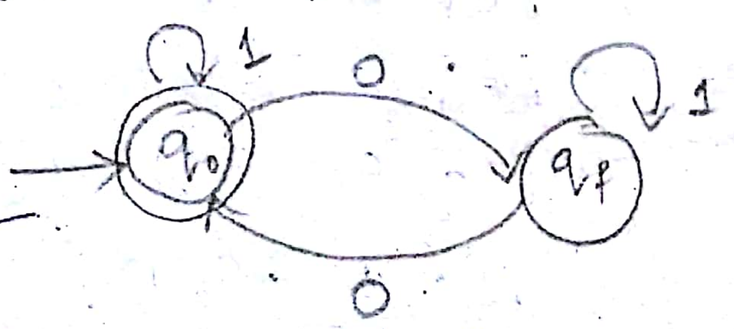
DFA :- the finite A from one state with one (symbol) i/p signal it has to go only one next state (It should unique) $\delta(q_0, a) = \{q_0\ q_1\}$

NFA :- one state has a many next states (more than 1)

$\boxed{\delta : Q \times \Sigma \to 2^Q}$ → powerstate of NFA

→ even no. of zero's in it It should be DFA

least even no of zero's = 2



$|w| = 1001$

$w = 1000110$

\$ (ending of string) represented by \$

it can be also called ϵ because at initial state only it reached to final.

→ even no of zero's + even no of One's

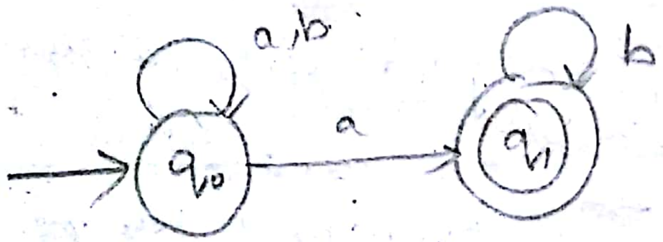
4/07/18:-

Transition functions :-

↳ Direct → $\delta(q, a)$ → i/p is a symbol

→ Indirect → $\hat{\delta}(q, w)$ → i/p is set of symbols (or) words.

→ For Transition Table



columns

rows

	a	b
q_0	$\{q_0, q_1\}$	q_0
q_1	$\emptyset$	q_1

Transition fn :-

The state with i/p signal what is the next State

Direct.

It does not contain previous/next. It contains only particular state. represented by 'δ' also called as One step transition

Indirect.

It consists of initial state and it is going to take the entire string (Alphabet/character / symbols) is taken in one transition what is the next state is called extended transition fn
↳ from initial state to end of string
represented by $\hat{\delta}$ (or) δ' (or) $\bar{\delta}$

problem:

compute the extended Transition fn $\hat{\delta}(q_0, w)$ when $w = 011101$ & given with 5 states

	0	1
→ q_0	q_1	q_4
q_1	q_4	q_2
∧+* q_2	q_3	q_3
q_3	q_2	q_2
q_4	q_4	q_4

↳ it is DFA since it has only 1 state

☆ final state can be represented as a * or a ○ or + , ∧

It should contain only one initial state
[Final state can be more than one]

Sol:- $\hat{\delta}(q_0, 011101)$ → we need to take the string seperately

$\hat{\delta}(q_0, \epsilon)$ = state itself (q_0) [initially the string should be empty]

→ $\hat{\delta}(q_0, 0) =$

$$\delta(\underbrace{\hat{\delta}(q_0, \epsilon)}_{q_0}, 0) = \delta(q_0, 0) = q_1$$

→ $\hat{\delta}(q_0, 01) = \delta(\underbrace{\hat{\delta}(q_0, 0)}, 1) = \delta(q_1, 1) = q_2$

→ $\hat{\delta}(q_0, 011) = \delta(\underbrace{\hat{\delta}(q_0, 01)}, 1) = \delta(q_2, 1) = q_3$

→ $\hat{\delta}(q_0, 0111) = \delta(\underbrace{\hat{\delta}(q_0, 011)}, 1) = \delta(q_3, 1) = q_2$

→ $\hat{\delta}(q_0, 01110) = \delta(\underbrace{\hat{\delta}(q_0, 0111)}, 0) = \delta(q_2, 0) = q_3$

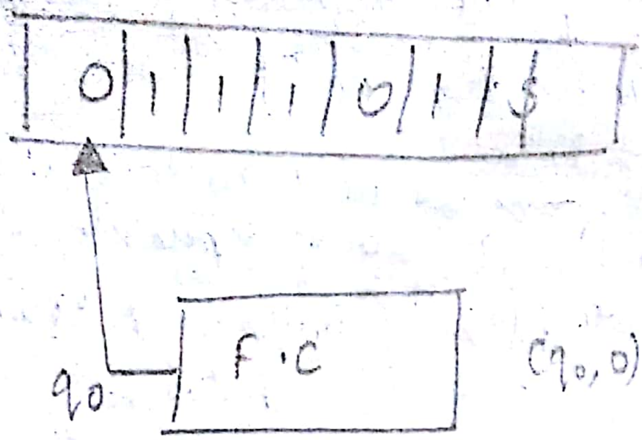
→ $\hat{\delta}(q_0, 011101) = \delta(\underbrace{\hat{\delta}(q_0, 01110)}, 1) = \delta(q_3, 1) = q_2$ → final state

In i/p tape

0	1	1	1	0	1	$	

Last should be separated by $

Since FA doesnot known the string is completed



$(q_0, 0) \cdot q_1$

q_0

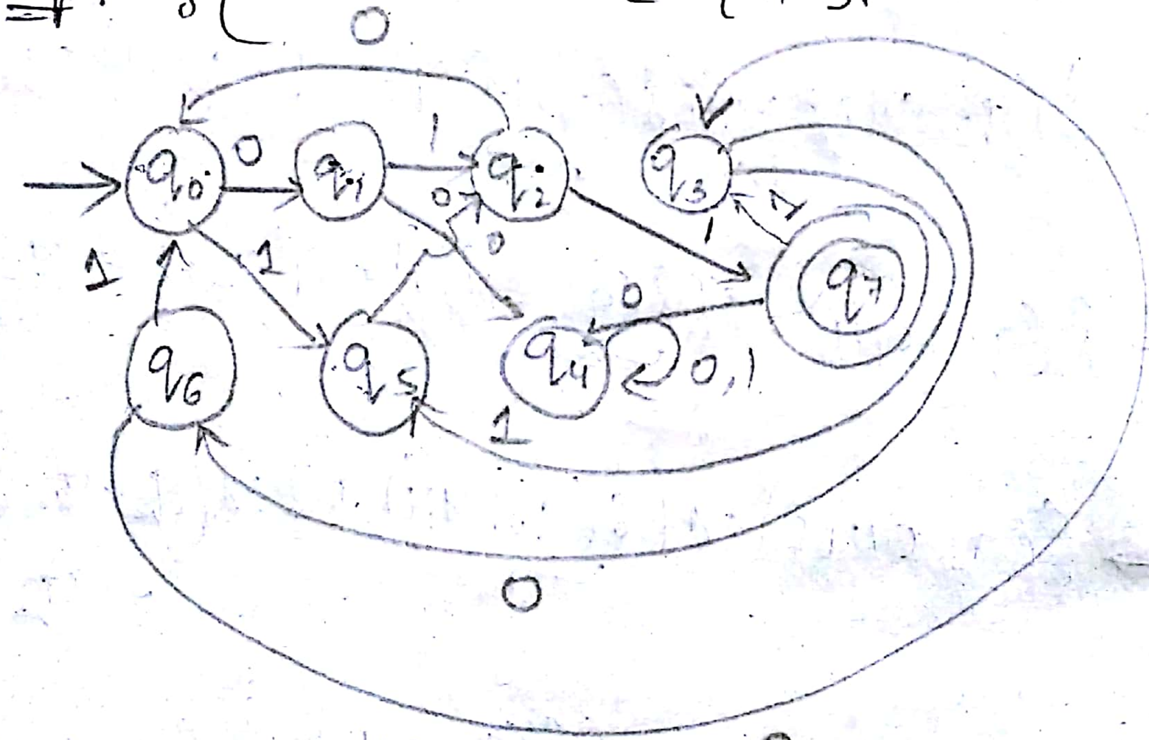
q_1

q_2

$q_3 q_2$

$q_3 q_2$

eg :- $\hat{\delta}($ $\Sigma = \{0, 1\}$. inputs

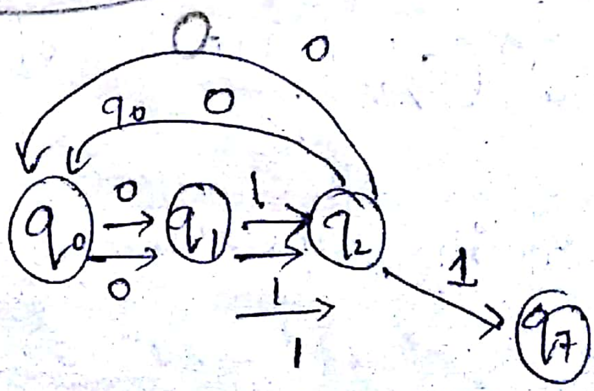


step 1

$|x| = 011$

$x = 010010011$

$(q_0, 0) = q_1$



	0	1
q_0	q_1	q_5
q_1	q_4	q_2
q_2	q_0	q_7
q_3	q_6	q_5
q_4	q_4	q_4
q_5	q_2	ϕ
q_6	q_3	q_0
q_7	q_4	q_3

$w = 0100100011$

$\Rightarrow \hat{\delta}(q_0, \epsilon) = q_0$ (state itself).

$\Rightarrow \hat{\delta}(q_0, 0) = \delta(\underbrace{\hat{\delta}(q_0, \epsilon)}, 0) = \delta(q_0, 0) = q_1$

$\Rightarrow \hat{\delta}(q_0, 01) = \delta[\underbrace{\hat{\delta}(q_0, 0)}_{q_1}, 1) = \delta(q_1, 1) = q_2$

$\Rightarrow \hat{\delta}(q_0, 010) = \delta(\underbrace{\hat{\delta}(q_0, 01)}, 0) = \delta(q_2, 0) = q_0$

$\Rightarrow \hat{\delta}(q_0, 0100) = \delta(\underbrace{\hat{\delta}(q_0, 010)}, 0) = \delta(q_0, 0) = q_1$

$$\hat{\delta}(q_0, 01001) = \delta(\hat{\delta}\underbrace{(q_0, 0100)}, 1) = \delta(q_1, 1)$$
$$= q_2$$

$$\hat{\delta}(q_0, 010010) = \delta(\hat{\delta}\underbrace{(q_0, 01001)}, 0) = \delta(q_2, 0)$$
$$= q_0$$

$$\hat{\delta}(q_0, 0100100) = \delta(\hat{\delta}(q_0, \underbrace{010010}), 0) =$$
$$\delta(q_0, 0)$$
$$= q_1$$

$$\hat{\delta}(q_0, 01001001) = \delta(\hat{\delta}\underbrace{(q_0, 0100100)}_{q_1}, 1)$$
$$= \delta(q_1, 1)$$
$$= q_2$$

$$\hat{\delta}(q_0, 010010011) = \delta(\hat{\delta}(q_0, \underbrace{01001001}), 1)$$
$$= \delta(q_2, 1)$$
$$= \underline{\underline{q_7}}$$
$$\downarrow$$
final state

9/7/18 :-

⇒ D.F.A has a Regular language

→ The language of M is set of strings w that take start state of q_0 to one of accepting state then we say L is a regular language. If L is L(M) for some DFA(M)

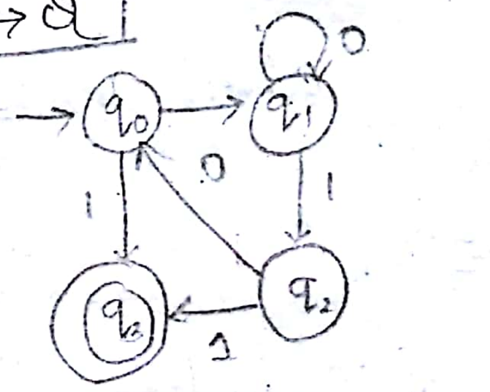
$$L(M) = \{ w \mid \hat{\delta}(q_0, w) = q_f \text{ where } q_f \in F \}$$

where F is the set of inputs

⇒ Deterministic $\delta : Q \times \Sigma \to Q$

Eg:- no. of inputs

	0	1
q_0	q_1	q_3
q_1	q_1	q_2
q_2	q_0	q_3
q_3	ϕ	ϕ



Transition diagram

This is example of DFA since it has only one next state.

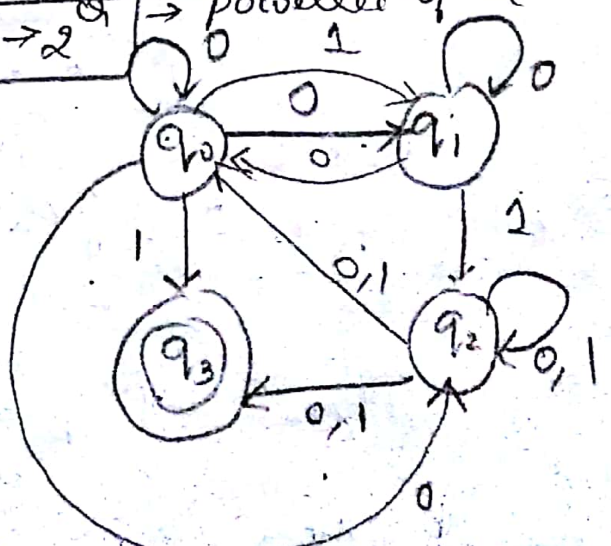
$$\boxed{L = \{ W \mid \delta(q_0, w) = 2^Q \text{ where } 2^Q \subset F}$$

⇒ Non-Deterministic Finite Automata

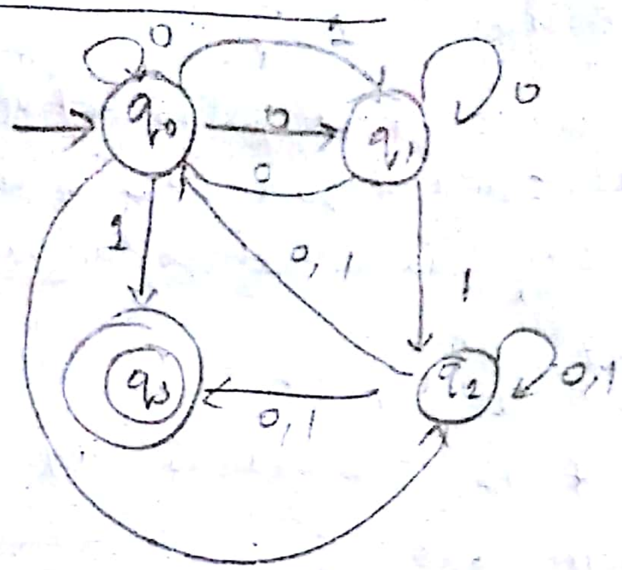
$\delta : Q \times \Sigma \to 2^Q$ → powerset of Q

eg

	0	1
q_0	q_0, q_1, q_2	q_1, q_3
q_1	q_0, q_1	q_2
q_2	q_0, q_2, q_3	q_0, q_2, q_3
q_3	ϕ	ϕ



Extended Transtition function



Given the string is 001001

Sol:- $\hat{\delta}(q_0, \epsilon) = q_0$ (initial state)

$\hat{\delta}(q_0, 001001) \Rightarrow \delta(q_0, 0) = \{q_0, q_1, q_2\}$

$\Rightarrow \hat{\delta}(q_0, 00) \Rightarrow \delta(\underbrace{\hat{\delta}(q_0, 0)}), 0)$

$= \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) =$

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

$\{q_0\ q_2\ q_3\}$

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

$\Rightarrow \hat{\delta}(q_0, 001) = \delta(\hat{\delta}(q_0, 00), 1)$

$= \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1) \cup \delta(q_3, 1)$

$= \{q_1 q_3\} \cup \{q_0, q_1\} \cup \{q_0, q_2, q_3\} \cup \{\phi\}$

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

$\Rightarrow \hat{\delta}(q_0, 0010) = \delta(\underbrace{\hat{\delta}(q_0, 001)}_{\{q_0, q_1, q_2, q_3\}}, 0)$

$= \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)$

$= \{q_0 q_1 q_2 q_3\}$

$\Rightarrow \hat{\delta}(q_0, 00100) = \delta(\underbrace{\hat{\delta}(q_0, 0010)}_{\{q_0, q_1, q_2, q_3\}}, 0)$

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

$\Rightarrow \hat{\delta}(q_0, 001001) = \delta(\underbrace{\hat{\delta}(q_0, 00100)}_{\{q_0, q_1, q_2, q_3\}}, 1)$

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

0 $\Rightarrow$ it consists of set of state s. ~~of/8~~

eg :-

δ	0	1
q_0	q_0	$q_0 q_1$
q_1	q_2	q_2
* q_2	ϕ	ϕ

$W = 010101 \; 0$ checkout whether NFA is accepted / Not

Scanned by CamScanner

Sol :-

$\Rightarrow \hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 0101010)$

$\delta(\underbrace{\hat{\delta}(q_0, \epsilon)}_{q_0}, 0)$

$\Rightarrow \hat{\delta}(q_0, 0) = q_0$

$\Rightarrow \hat{\delta}(q_0, 01) = \delta(\underbrace{\hat{\delta}(q_0, 0)}, 1) =$

$\delta(q_0, 1) = q_0 q_1$

$\Rightarrow \hat{\delta}(q_0, 010) = \delta(\underbrace{\hat{\delta}(q_0, 01)}_{q_0 q_1}, 0)$

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

$\{q_0, q_2\}$

$\Rightarrow \hat{\delta}(q_0, 0101) = \delta(\underbrace{\hat{\delta}(q_0, 010)}_{\{q_0, q_2\}}, 1)$

$= \delta(q_0, 1) \cup \delta(q_2, 1) = \{q_0 q_1\} \cup \{\phi\}$

$= \{q_0, q_1\}$

$\Rightarrow \hat{\delta}(q_0, 01010) = \delta(\underbrace{\hat{\delta}(q_0, 0101)}_{\{q_0, q_1\}}, 0)$

$\Rightarrow \delta(q_0, 0) \cup \delta(q_1, 0)$

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

$\Rightarrow \hat{\delta}(q_0, 010101) = \delta(\underbrace{\hat{\delta}(q_0, 01010)}, 1)$

$= \{q_0, q_2\}$

$\Rightarrow \delta(q_0, 1) \cup \delta(q_2, 1)$

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

$= \{q_0, q_1\}$

$\rightarrow \hat{\delta}(q_0, 0101010) = \delta(\hat{\delta}(q_0, 010101), 0)$

$= \{q_0, q_2\}$

Final state it contains

eg string $\boxed{w = 00100101}$ → not a NDFA

$\hat{\delta}(q_0, E) = q_0$

$\hat{\delta}(q_0, 0) = \delta(\underbrace{\hat{\delta}(q_0, E)}_{q_0}, 0) = \delta(q_0, 0) = q_0$

$\hat{\delta}(q_0, 00) = \delta(\underbrace{\hat{\delta}(q_0, 0)}, 0) = \delta(q_0, 0) = q_0$

$\hat{\delta}(q_0, 001) = \delta(\underbrace{\hat{\delta}(q_0, 00)}, 1) = \delta(q_0, 1) = q_0, q_1$

$\hat{\delta}(q_0, 0010) = \delta(\underbrace{\hat{\delta}(q_0, 001)}_{q_0q_1}, 0)$ 00100101

$\Rightarrow \delta(q_0, 0) \cup \delta(q_1, 0)$

$= q_0 \cup q_2 = \{q_0q_2\}$

$\hat{\delta}(q_0, 00100) = \delta(\underbrace{\hat{\delta}(q_0, 0010)}_{q_0q_2}, 0)$

$= \delta(q_0, 0) \cup \delta(q_2, 0) \Rightarrow q_0 \cup \phi = q_0$

$\hat{\delta}(q_0, 001001) = \delta(\underbrace{\hat{\delta}(q_0, 00100)}_{q_0}, 1) \Rightarrow \delta(q_0, 1) = q_0q_1$

$\hat{\delta}(q_0, 0010010) = \delta(\hat{\delta}(q_0, 001001), 0) \Rightarrow \delta(q_0, 0) \cup \delta(q_1, 0)$

$= q_0 \cup q_2 = q_0q_2$

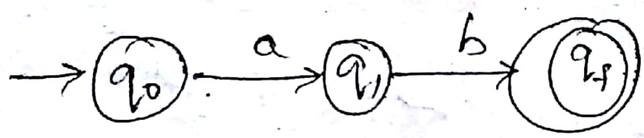
$\hat{\delta}(q_0, 00100101) = \delta(\underbrace{\hat{\delta}(q_0, 0010010)}_{q_0q_2}, 1) = \delta(q_0, 1) \cup \delta(q_2, 1)$

$= q_0q_1 \cup \phi$

Acceptability of strings

if the transition fn of string does not reach the final state then it is not acceptable.

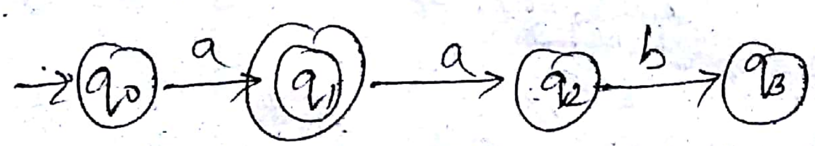
eg:



$\omega = ab$

it is acceptable

(ii)



$\omega = aab$

it is not acceptable since the string is aab E_i the final state is at q_1 only so it cannot reach the string of aab

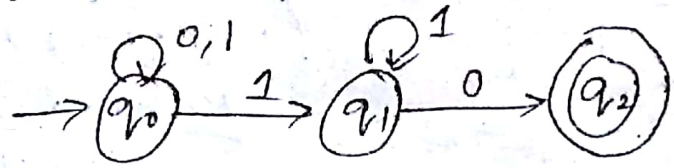
10/7/18

Equivalence of NFA & DFA

All DFA's are NFA's

⇒ convert NFA to DFA

eg



	0	1
q_0	q_0	$q_0 q_1$
q_1	q_2	q_1
q_2	ϕ	ϕ

Step-1 keep the state transitions as it is (initial state transitions)

Step 2:- the states are represented in a [ ] divided by ',' when there are more than one state.

Step 3 The second state will be the combined state

$\delta([q_0, q_1], 0) = \delta(q_0, 0) \cup \delta(q_1, 0) = \{q_0\} \cup \{q_2\}$
$= [q_0, q_2]$

$\delta([q_0, q_1), 1) = \delta(q_0, 1) \cup \delta(q_1, 1) = [q_0, q_1]$

Step 4: If any new state is generated consider

$\delta([q_0, q_2], 0) = \delta(q_0, 0) \cup \delta(q_2, 0) = q_0$

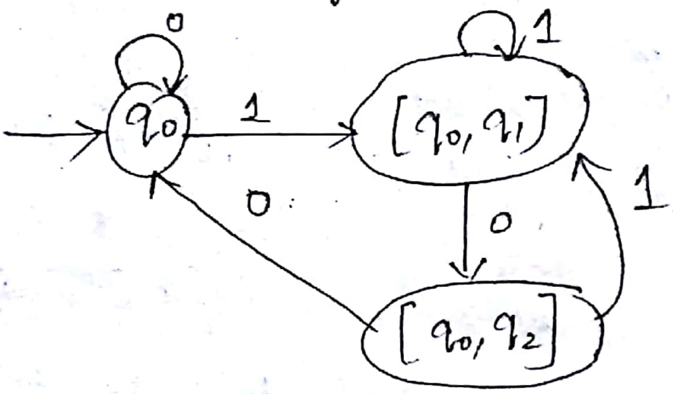
$\delta([q_0 q_2], 1) = \delta(q_0, 1) \cup \delta(q_2, 1) = [q_0, q_1]$

Scanned by CamScanner

	0	1
q_0	q_0	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_2]$	$[q_0, q_1]$
$[q_0, q_2]$	q_0	$[q_0, q_1]$

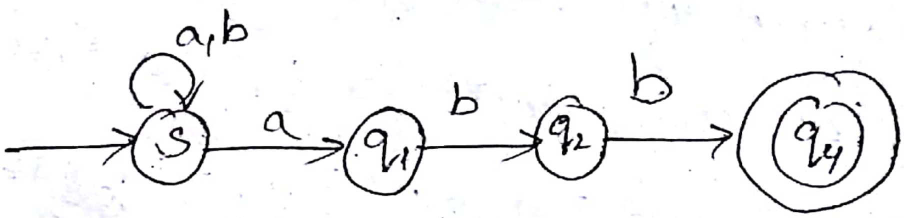
(no new state is generated ⇒ transition is stopped)

⇒ In how many groups the final state of NFA is present all those groups are final states of DFA

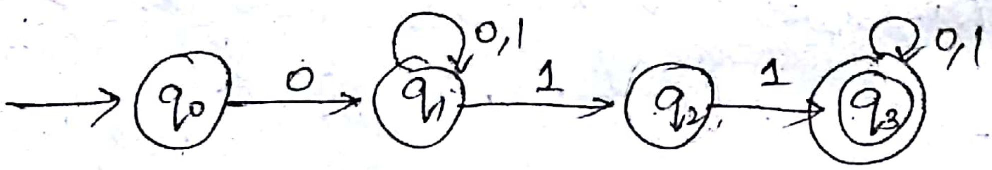


eg:-
⇒

(i)



(ii)



(i) NFA

	a	b	DFA	a	b
S	{S, q_1}	S	S	[S, q_1]	S
q_1	ϕ	q_2 ⇒	[S, q_1]	[S, q_1]	[S, q_2]
q_2	ϕ	q_4	[S, q_2]	[S, q_1]	[S, q_4]
q_4	ϕ	ϕ	* [S, q_4]	[S, q_1]	S

$\delta([s, q_1], a) = \delta(s, a) \cup \delta(q_1, a) = [s, q_1]$

$\delta([s, q_1], b) = \delta(s, b) \cup \delta(q_1, b) = [s, q_2]$

$\delta([s, q_2], a) = \delta(s, a) \cup \delta(q_2, a) = [s, q_1]$

$\delta([s, q_2], b) = \delta(s, b) \cup \delta(q_2, b) = [s, q_4]$

$\delta([s, q_4], a) = \delta(s, a) \cup \delta(q_4, a) = [s, q_1]$

$\delta([s, q_4], b) = \delta(s, b) \cup \delta(q_4, b) = s$

(ii)

NFA	0	1
q_0	q_1	ϕ
q_1	q_1	$\{q_1, q_2\}$
q_2	ϕ	q_3
q_3	q_3	q_3

$\Rightarrow$

DFA

	0	1
q_0	q_1	ϕ
q_1	q_1	$[q_1, q_2]$
$[q_1, q_2]$	q_1	$[q_1, q_2, q_3]$
$[q_1, q_2, q_3]$	$[q_1, q_3]$	$[q_1, q_2, q_3]$
$[q_1, q_3]$	$[q_1, q_3]$	$[q_1, q_2, q_3]$

$\delta([q_1, q_2], 0) = q_1$

$\delta([q_1, q_2], 1) = [q_1, q_2, q_3]$

$\delta([q_1, q_2, q_3], 0) = [q_1, q_3]$

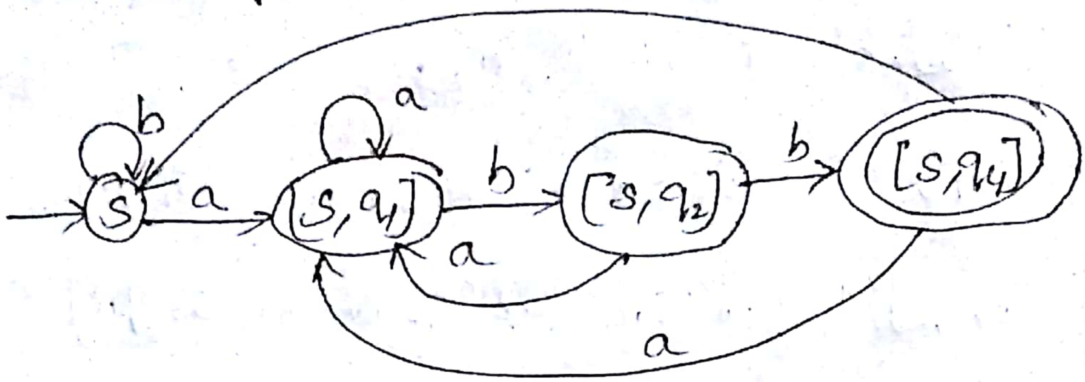
$\delta([q_1, q_2, q_3], 1) = [q_1, q_2, q_3]$

$\delta([q_1, q_3], 0) = [q_1, q_3]$

$\delta([q_1, q_3], 1) = [q_1, q_2, q_3]$

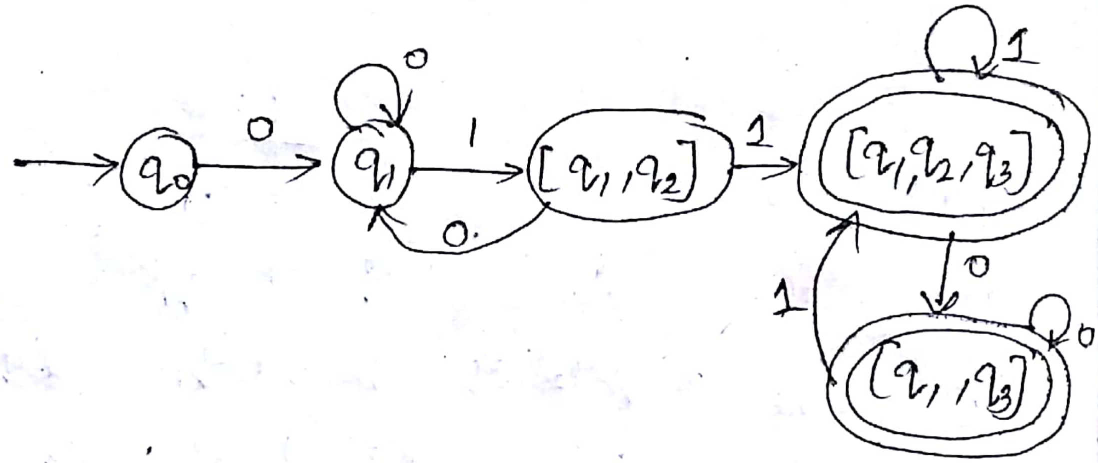
For (i) diagram

(i)



For (ii) diagram

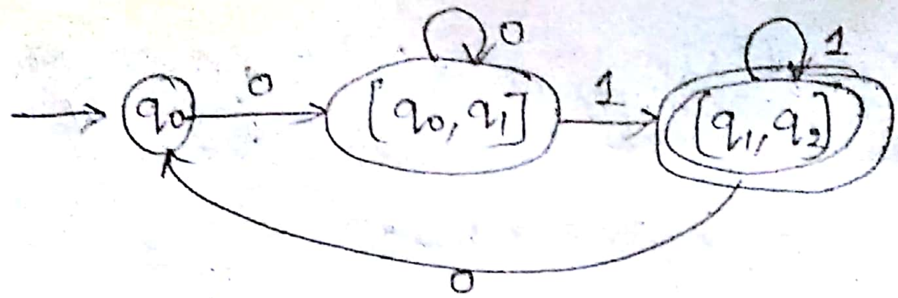
(i)



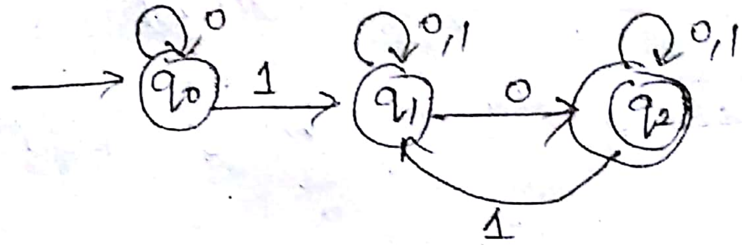
(iii) NFA

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

A	0	1
q_0	$[q_0, q_1]$	ϕ
$[q_0, q_1]$	$[q_0, q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	q_0	$[q_1, q_2]$



iv)



$\omega = 11001$

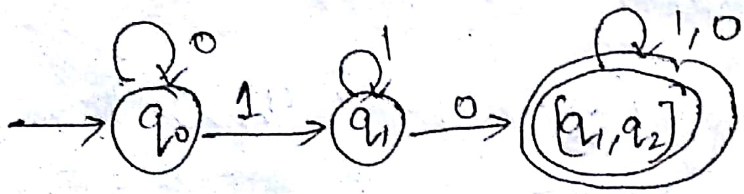
NFA

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

$\Rightarrow$

DFA

	0	1
q_0	q_0	q_1
q_1	$[q_1, q_2]$	q_1
$[q_1, q_2]$	$[q_1, q_2]$	$[q_1, q_2]$



For NFA $\omega = 11001$

$\hat{\delta}(q_0, E) = q_0$

$\hat{\delta}(q_0, 11001) = \hat{\delta}(q_0, 1) = \delta(\hat{\delta}(q_0, E), 1) = \delta(q_0, 1)$

$= q_1$

$\hat{\delta}(q_0, 11) = \delta(\underbrace{\hat{\delta}(q_0, 1)}, 1) = \delta(q_1, 1) = q_1$

$\hat{\delta}(q_0, 110) = \delta(q_1, 0) = \{q_1, q_2\}$

$\hat{\delta}(q_0, 1100) = \delta(\underbrace{\hat{\delta}(q_0, 110)}, 0) = \delta(\{q_1, q_2\}, 0)$

$= \delta(q_1, 0) \cup \delta(q_2, 0)$

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

$= \{q_1, q_2\}$

$\hat{\delta}(q_0, 11001) = \delta(q_1, 1) \cup \delta(q_2, 1) = \{q_1, q_2\} \Rightarrow$ Accepted

For DFA $\quad w = 11001$

$\hat{\delta}(q_0, \epsilon) = q_0$

$\hat{\delta}(q_0, 1) = \delta(\underbrace{\hat{\delta}(q_0, \epsilon)}_{q_0}, 1) = q_1$

$\hat{\delta}(q_0, 11) = \delta(q_1, 1) = q_1$

$\hat{\delta}(q_0, 110) = \delta(q_1, 0) = [q_1, q_2]$

$\hat{\delta}(q_0, 1100) = \delta(\underbrace{\hat{\delta}(q_0, 110)}, 0) = \delta([q_1, q_2], 0)$

$= \delta(q_1, 0) \cup \delta(q_2, 0)$

$= [q_1, q_2] \cup \phi$

$= [q_1, q_2]$

$\hat{\delta}(q_0, 11001) = \delta(\underbrace{\hat{\delta}(q_0, 1100)}_{[q_1, q_2]}, 1)$

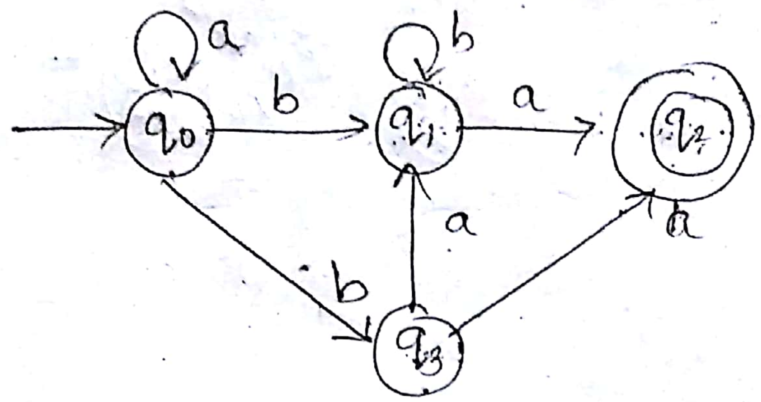
$= \delta(q_1, 1) \cup \delta(q_2, 1)$

$= [q_1, q_2] \cup \phi$

$= [q_1, q_2] \Rightarrow$ Accepted.

---

v)



NFA

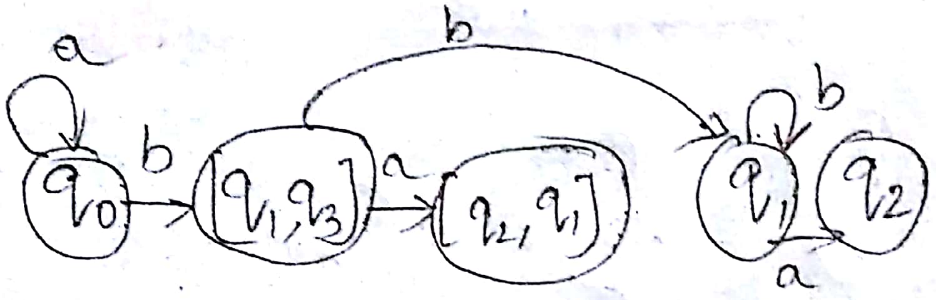
	a	b
$\rightarrow q_0$	q_0	$\{q_1, q_3\}$
q_1	q_2	q_1
$* q_2$	ϕ	ϕ
q_3	q_1	ϕ

DFA

	a	b
q_0	q_0	$[q_1, q_3]$
$[q_1, q_3]$	$[q_2, q_1]$	q_1
$[q_2, q_1]$	ϕ	ϕ

	a	b
q_1	q_2	q_1
q_2	ϕ	ϕ

Diagram

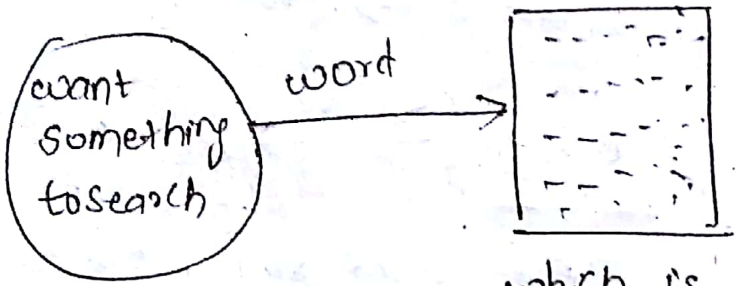


11/07/18:-

Text Search:-

It uses Inverted Indexes.

NFA has text Search:-



which is used to search the data

NFA with Epsilon Transactions:-

↳ we can also used ε as an i/p signal

All DFA's are NFA's

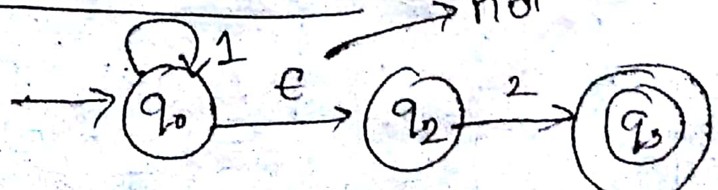
The DFA won't accept ε transition

→ it may accept one state with one i/p signal it can have a one State

1) Transition Notation.

→ not basic i/p element for automata

eg:-



no of i/p $\Sigma = \{1, 2\}$

ϵ is used one of the extra symbol used for transition of one state to another state.

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

DFA

$Q \times \Sigma \rightarrow Q$

No 'ϵ' transition will be present

NFA

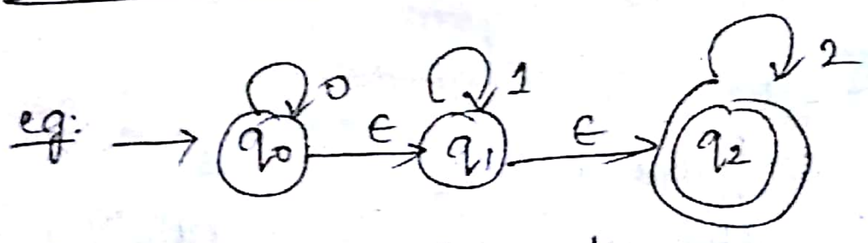
$Q \times \Sigma \rightarrow 2^Q$

$\Rightarrow \boxed{\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow 2^Q}$

we need to not write 'ϵ' in the i/p set.

2) Usage of ϵ-Transition

eg:



Transition diagram

Transition table

States	0	1	2	ϵ
$\rightarrow q_0$	q_0	ϕ	ϕ	q_1
q_1	ϕ	q_1	ϕ	q_2
q_2	ϕ	ϕ	q_2	ϕ

no: of inputs $\Rightarrow$ $\Sigma = \{0, 1, 2\}$

Given string $w = 002$ (not accepted)

$$\rightarrow q_0 \xrightarrow{0} q_0 \xrightarrow{0} q_0 \xrightarrow{\epsilon} q_1 \xrightarrow{\epsilon} q_2 \xrightarrow{2} q_2$$

↓

i/p signal 0

(accepted)

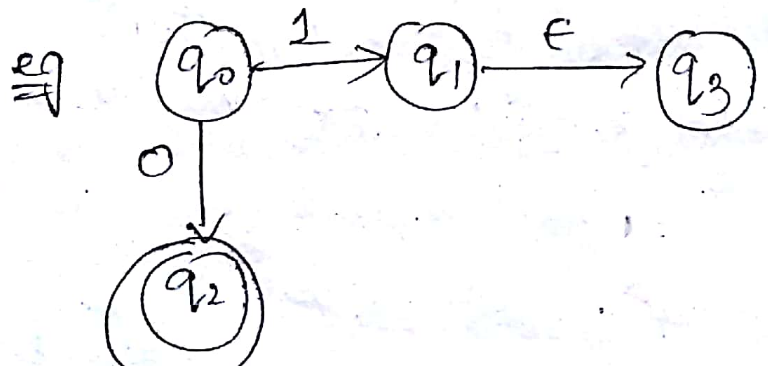
In transition we don't represent them as

$\Rightarrow$ $00\epsilon\epsilon2$

→ So we only write the string as 002

→ we need to not include ϵ symbol whenever we require

eg



Given

$w = 1$

↓

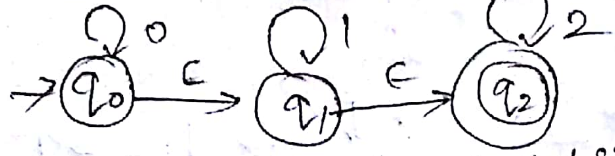
not accepted

when we don't have path we need to not do a path for eg: from q_3 to q_2 writing 'ϵ'

3) Epsilon - closure :-

↳ which can be applied on every state of F.A

→ if we are have n F.A ↓ it should be applied (epsilon closure) to n state

eg 

initial state (state itself)

ϵ-closure $(q_0) = \{q_0, q_1, q_2\}$

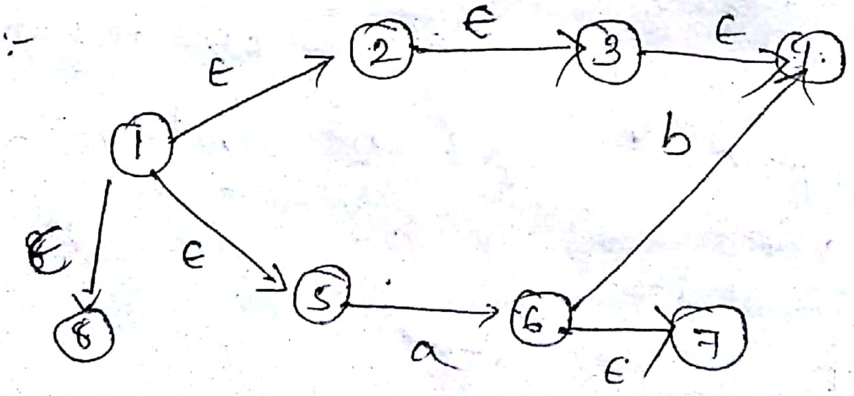
ϵ-closure $(q_1) = \{q_1, q_2\}$

ϵ-closure $(q_2) = \{q_2\}$

→ F.A is a directed way (it should contain directions)

<u>step-1:-</u>
The state itself

<u>step 2:</u>
check whether it is having any ϵ transition again we need to check for another ϵ transition

<u>eg:-</u> 

ϵ-closure $(1) = \{1, 2, 5, 3, 4, 8\}$.

it should check all possible cases

for 1 it has ϵ (2, 5) → it is not like having one path to final state.

ϵ-closure (2)
= {2, 3, 4}

ϵ-closure (3) = {3, 4}.

ϵ-closure (4) = {4, }

ϵ-closure (5) = {5, }

ϵ-closure (6) = {6, 7}

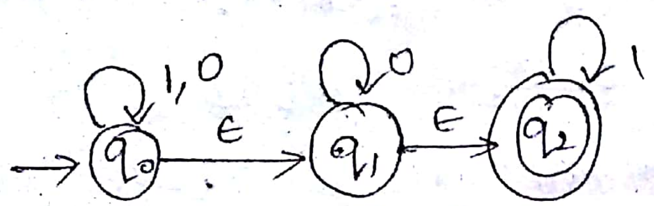
ϵ-closure (7) = {7, }.

→ $\boxed{\epsilon\text{-closure is different from the } \epsilon\text{-transition}}$

	a	b	ϵ
1	ϕ	ϕ	2, 5, 8
2	ϕ	ϕ	3.

→ (this is different from ϵ-closure)

eg



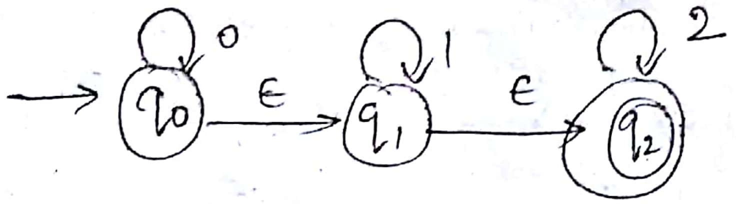
	0	1	ϵ
q_0	q_0	q_0	q_1
q_1	q_1	ϕ	q_2
q_2	ϕ	q_2	ϕ

ϵ-closure$(q_0) = \{q_0, q_1, q_2\}$

ϵ-closure$(q_1) = \{q_1, q_2\}$.

ϵ-closure$(q_2) = \{q_2\}$

⇒ Extended - Transition function of NFA with ϵ-closure



Given string 01

$$\hat{\delta}(q_0, \epsilon) = \{q_0\}$$

$$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0)$$

$$= \delta(q_0, 0)$$

$$= q_0$$

$$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1)$$

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

$$= \phi$$

it is not accepted since it cannot reach to final state.

NFA with ε-closure

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

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

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

Given string $\underline{01}$

$\Rightarrow \hat{\delta}(q_0, 01) = \delta(\epsilon\text{-closure}(q_0), 0)$

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

$= q_0 \cup \phi \cup \phi$

↓ it is not simple q_0

it is an ε-closure of q_0

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

$\Rightarrow \delta(\epsilon\text{-closure}(q_0, q_1, q_2), 1) = \delta(\{q_0, q_1, q_2\}, 1)$

q_0, q_1, q_2

q_1, q_2

q_2

$\{q_0, q_1, q_2\}$

$= \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1)$

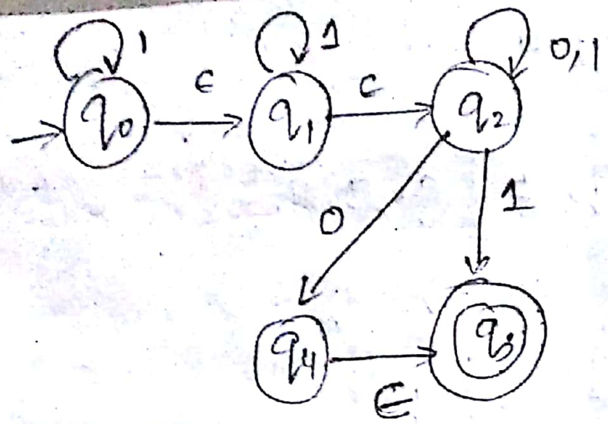
$= \phi \cup q_1 \cup \phi$

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

$= \{q_1, q_2\}$

↳ accepted.

eg:-



$w = 110001$

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

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

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

$\epsilon\text{-closure}(q_3) = \{q_3\}$

$\epsilon\text{-closure}(q_4) = \{q_4, q_3\}$

$\Rightarrow \hat{\delta}(q_0, 110001) = \delta(\epsilon\text{-closure}(q_0), 1)$

$\downarrow$

$\{q_0, q_1, q_2\}$

$= \delta(q_0, 1) \cup \delta(q_1, 1) \cup \delta(q_2, 1)$

$q_0 \cup q_1 \cup q_2 q_3$

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

$= q_0, q_1, q_2, q_3$

$\Rightarrow \delta(\epsilon\text{-closure}\{q_0, q_1, q_2, q_3\}), 1) = \delta(q_0, 1) \cup \delta(q_1, 1) \cup$

$\delta(q_2, 1) \cup$

$\delta(q_3, 1)$

$q_0, q_1, q_2 \quad q_1 q_2$

$q_0 q_1, q_2, q_3$

$= \cancel{q_0 q_1 q_2 q_3}$

$= q_0 q_1, q_2 q_3$

Scanned by CamScanner

$\Rightarrow \delta(\epsilon\text{-closure}(q_0, q_1, q_2, q_3), 0) \Rightarrow \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0) \cup \delta(q_3, 0)$

$\underline{\qquad\qquad\qquad}$

q_0, q_1, q_2, q_3

$\Rightarrow \phi \cup \phi \cup q_2 \phi$

$= q_2$

$\downarrow$

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

$\delta(\underbrace{\epsilon\text{-closure}(q_2)}_{q_2}, 0) = \delta(q_2, 0) = q_2$

$\downarrow$

$\epsilon\text{-closure}(q_2), 0)$

$\delta(q_2, 0) = q_2$

$\downarrow$

$\delta(\underbrace{\epsilon\text{-closure of } q_2}_{q_2}, 1) = \delta(q_2, 1)$

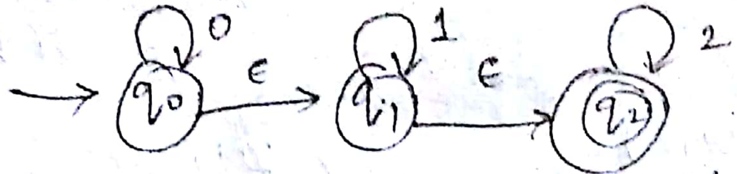
$= \{q_2, q_3\}$

$\downarrow$

accepted

Since it contains the final states it is accepted.

4) Elimination of ϵ-Transitions from NFA



calculate ϵ-closure of all states

Step 1:- $\epsilon\text{-closure}(q_0) = \{q_0, q_1, q_2\}$

$\epsilon\text{-closure}(q_1) = \{q_1, q_2\}$ $\quad \Sigma = i/p\{0,1,2\}$

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

state with first input signal

Step 2:- $\delta(q_0,0) = \epsilon\text{closure}(\delta(\epsilon\text{-closure}(q_0), 0))$

$= \epsilon\text{-closure}(\delta\{q_0, q_1, q_2\}, 0)$

$= \epsilon\text{-closure}(\delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0))$

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

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

$= q_0 q_1 q_2$

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

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

$= \epsilon(q_1) = [q_1, q_2]$

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

$= \epsilon\text{-}(\phi \cup \phi \cup q_2)$

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

$= \{q_2\}$

→ ε-closure of NFA ⟶ without ε of NFA

→ ε-NFA → DFA } two ways.

	0	1	2
q_0	$[q_0, q_1, q_2]$	$[q_1, q_2]$	q_2
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$(q_1\ q_2)$	q_2
$(q_1 q_2)$	ϕ	$[q_1, q_2]$	q_2
q_2	ϕ	ϕ	q_2

$\Rightarrow \delta(q_0, q_1, q_2, 0) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0, q_1, q_2), 0))$

$= \epsilon\text{-closure}(\delta((q_0, q_1, q_2), 0))$

$= \epsilon\text{-closure}(\delta(q_0, 0) \cup (\delta(q_1, 0) \cup \delta(q_2, 0))$

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

$= [q_0, q_1, q_2]$

$\Rightarrow \delta((q_0, q_1, q_2), 1) = \epsilon\text{-closure}(\delta(\epsilon\text{-closure}(q_0, q_1, q_2), 1))$

$= \epsilon\text{-closure}(\delta(q_0, q_1, q_2), 1)$

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

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

$\Rightarrow \delta((q_0, q_1, q_2), 2) = \epsilon(\delta(\underbrace{\epsilon(q_0, q_1, q_2)}), 2)$

$= \epsilon(\delta(q_0, q_1, q_2), 2)$

$\epsilon(q_2)$

$\Rightarrow \{q_2\}$.

$\Rightarrow \delta((q_1, q_2), 0) = \epsilon(\delta(\epsilon(q_1, q_2), 0)$

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

$= \epsilon(\delta(q_1, 0) \cup \delta(q_2, 0))$

$= \epsilon(\phi \cup \phi)$

$= \phi$

$\Rightarrow \delta((q_1, q_2), 1) = \epsilon(\delta(\epsilon(q_1, q_2), 1)$

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

$\delta(q_1, 1) \cup \delta(q_2, 1)$

$= \epsilon(q_1, \phi)$

$= \{q_1, q_2\}$

$\Rightarrow \delta((q_1, q_2), 2) = \epsilon(\delta(\epsilon(q_1, q_2), 2)$

$= \epsilon(\delta(q_1, q_2), 2)$

$(q_1, 2) \cup (q_2, 2)$

$= \epsilon(\phi \cup q_2) = \{q_2\}$

Scanned by CamScanner

$\Rightarrow \delta(q_2, 0) = \epsilon\text{-}(\delta(\underbrace{\epsilon(q_2)}, 0)$

$= \epsilon(q_2, 0)$

$= \phi$

$\Rightarrow \delta(q_2, 1) = \epsilon\text{-}(\delta(\underbrace{\epsilon(q_2)}, 1)$

$= \epsilon(q_2, 1)$

$= \phi$

$\Rightarrow \delta(q_2, 2) = \epsilon\text{-}(\delta(\epsilon(q_2), 2)$

$= \epsilon\text{-}(q_2, 2)$

$= q_2$