Theory of Computation

Theory of Computation (Theory of automata)

→ It is a theoritical branch of computer Science and mathematics which deals with the logic of computation with respect to machines reffered as automata (machines works automatically).

Terminologies :-

→ Alphabet [Σ] : it is a character set, which is a finite non empty set of symbols.

Example:

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

Numerical $\Sigma = \{0 \ldots 9\}$

English $\Sigma = \{a \ldots z\}$

→ String [W] : A string a word is a finite sequence of symbols from the alphabet

Example:

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

$W = \{0, 1, 00, 11, 101, 1011 \ldots\}$

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

$W = \{ab, bc, abc, bac, cab \ldots\}$

→ Length of String [|S|] :

Example: $S = \{01011\}$

$|S| = 5$

→ Empty String [ϵ] Episolon

→ Prefix & Suffix of String:

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

str = {1101011}

→ Reverse of a String [w^R]:

$w = \{01111011\}$

$w^R = \{10111110\}$

Operation And Properties

→ Concatenation (.) $\Sigma = \{a, b\}$ ⇒ ab [AND]

→ Union (+) $\Sigma = \{a, b\}$ ⇒ a+b [OR]

Properties:

i) Kleen closure [L^*] :- A language which is a sequence of strings contains 0 or more concatenations (a language includes a string of length 0)

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

$L^* = \{\epsilon, a, b, aa, ab, bb, aba, bba\}$

ii) Positive closure [L^+] :- This property allows different string of any length except null or Empty string (ϵ)

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

$L^+ = \{a, b, aa, ba, bb, aba, bba\}$

Types of Automata

Abstract Machine / Computer : An Abstract Machine / Computer is a conceptual or theoritical model of computer hardware / software system which really doesn't exist. These machine are not actual machine that's why they called as Abstract machine

i) Finite Automata
ii) Linear bounded Automata
iii) Push down Automata
iv) Turing machine

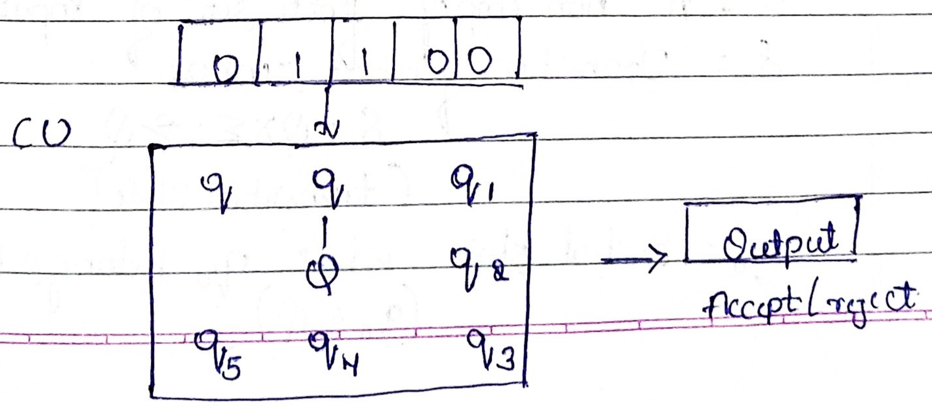
i) **Finite Automata :** Finite automata consist of a finite set of states and a set of transaction from one state to another state that occurs on input symbol taken from alphabet (Σ).

→ Finite Automata consist of following Components are :

- input tape
- Control Unit
- Output

22/1/25

Block diagram : $\Sigma = \{0, 1\}$



Date :
Page No. :

Applications of finite automata:

- → Using in Language processing (NLP)
- → Using in Compiler Constructions
- → Computer Networks
- → Used in Video games
- → Used in design of digital Circuit
- → Used in bio-medical problem solving

Rules:

- → For each input symbols there is exactly one transition out of each state.
- → Initial state denoted as q_0 (start state)
- → Final state denoted by q_F

Mathematical Representation of Finite Automata

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

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

here each term is a one records

where,

Q = A non-empty finite set of states

Σ = A non-empty finite set of inputs symbol

δ = Transition function, where

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

(transition state)

q_0 = Initial state, where q_0 belongs to Q

$(q_0 \in Q)$

q_F - A non empty, finite set of states called as acceptence state (can be one or more states)

$$q_F \in Q$$

Transition Diagram :-

1. Initial state

$$\rightarrow q_0$$

2. Final State

$$q_F$$

3. Loop

$$q_1 \circlearrowleft 0$$

4. From one state to another

$$q_1 \xrightarrow{0} q_2$$

5.

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

→ Here in 4, A machine in state q_1 takes input symbol 0 & moves to q_2

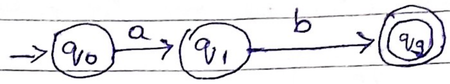
→ Here in 5, A machine in state q_1 takes input symbol 0 or 1 & moves to q_2.

6.

$$\underset{\substack{\downarrow \\ \text{Initial State}}}{\rightarrow q_0} \; \underbrace{- - - - - - - -}_{\text{Intermediate State}} \; \underset{\substack{\downarrow \\ \text{Final State}}}{q_F}$$

* Transition table:

Ex:

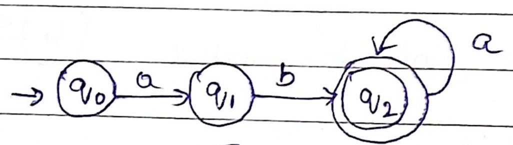


δ	a	b
$\rightarrow q_0$	q_1	-
q_1	-	q_3
$* q_2$	-	-

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

$M = \{\{q_0, q_1, q_2\}, \{a, b\}, \delta, \{q_0\}, \{q_2\}\}$

Ex:



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

$M = \{\{q_0, q_1, q_2\}, \{a, b\}, \delta, \{q_0\}, \{q_2\}\}$

Imp Questions :-

i) Define finite automata & mention its application.

ii) Define FA work components & mathematical representation.

Date :
Page No. :

Types of Finite Automata (FA):

1) DFA (Deterministic Finite Automata)
a) Pattern Machine Problems
i) string ends with ii) string starting with
iii) string having substring
b) Divide by 'K' problems
c) Modulo 'K' Counter Problems.
2) NFA [Non Deterministic FA] 3) E-NFA [Epsolon)
DFA problem with string ending with

1) Design a DFA to accept the string ends with 10 where $\Sigma = \{0,1\}$

* Note: Language = combination / set of valid string starting from minimum length to the maximum length using Σ.

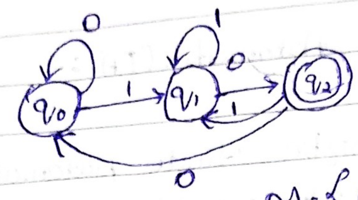
Step 1: $\vec{q_0} \rightarrow$ Initial State
$q_1 \rightarrow$ string ends with 1
$q_2 \rightarrow$ string ends with 10

Step 2: Transition table

	δ	0	1
ϵ	$\rightarrow q_0$	$\epsilon 0$	$\epsilon 1$
1	q_1	10	11
10	$* q_2$	100	101

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

Step 3:



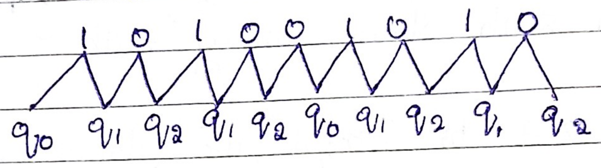
$M = \{Q, \Sigma, \delta, q_0, q_F\}$

$M = \{\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\}\}$

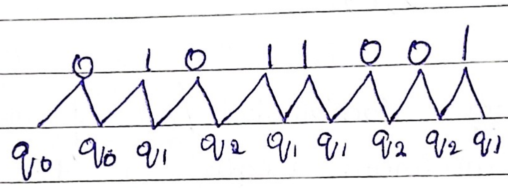
Verification:

1010010101 0

10 10 010 10



Accepted



Rejected

Q) Design a DFA to accept the string ends with 01 where $\Sigma = \{0,1\}$

→

Step 1:

→ $q_0 = \epsilon$ (initial state)

$q_1 = 0$ (Ends with zero(0))

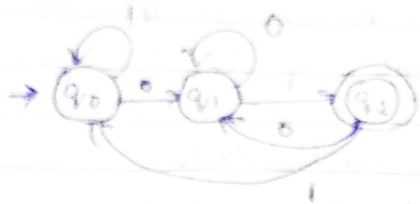
$q_2 = 01$ (ends with 01)

Step 2:

	δ	0	1
ϵ	$\rightarrow q_0$	$\cancel{\epsilon}0$	$\cancel{\epsilon}1$
0	q_1	$\cancel{0}0$	01
01*	q_2	$\cancel{0}10$	$\cancel{0}11$

	δ	0	1
ϵ	$\rightarrow q_0$	q_1	q_0
0	q_1	q_1	q_2
01	q_2	q_1	q_0

Step 3:

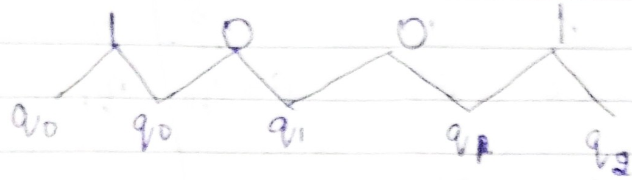


Transition Diagram

$M = \{\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\}\}$

Verification

1001



Accepted

Q11) Design a DFA to accept the string ends with 11 where $\Sigma = \{0,1\}$.

→ Step 1:

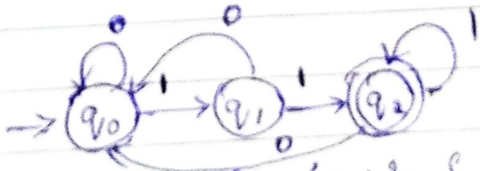
$\rightarrow q_0 = \epsilon$ (initial state)
$q_1 = 1$ (ends with 0)
$q_2 = 11$ (ends with 11)

Step 2:

	δ	0	1
ϵ	$\rightarrow q_0$	$\epsilon 0$	$\epsilon 1$
1	q_1	10	11
11	$*q_2$	110	111

	δ	0	1
ϵ	$\rightarrow q_0$	q_0	q_1
1	q_1	q_0	q_2
11	q_2	q_0	q_2

Step 3:



$M = \{\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\}\}$

9) Design a DFA to accept the strings ending with 00 where $\Sigma = \{0,1\}$

→

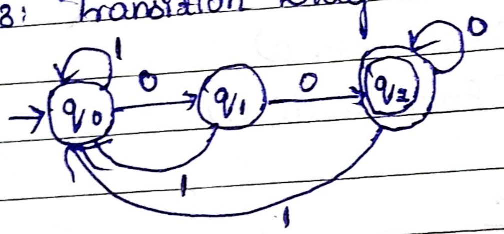
Step 1: → $q_0 = \epsilon$ (initial state)
$q_1 = 0$ (ends with 0)
$q_2 = 0$ (ends with 00)

Step 2:-

	δ	0	1
ϵ	$\rightarrow q_0$	$\epsilon 0$	$\epsilon 1$
0	q_1	00	01
00	q_2*	000	001

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

Step 3: Transition Diagram



$M = \{\{q_0, q_1, q_2\}, \{0,1\}, \delta, \{q_0\}, \{q_2\}\}$

→ Design a DFA to accept the strings of a's & b's ending with abb

→

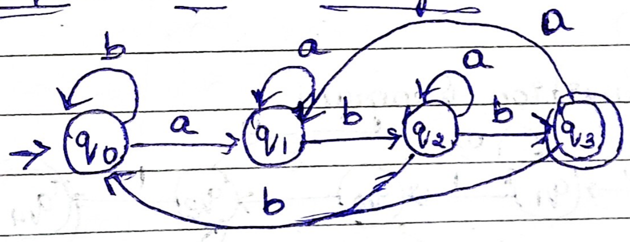
Step 1: → q_0 = initial state
q_1 = Ends with a
q_2 = Ends with ab
q_3 = Ends with abb

Step 2:

	δ	a	b
→ ε	q_0	εa	εb
a	q_1	εa	ab
ab	q_2	εab	abb
abb*	q_3	abba	abbb

δ	a	b
→ q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_2	q_3
q_3*	q_1	q_0

Step 3: Transition Diagram



$M = \{\{q_0, q_1, q_2, q_3\}, \{a, b\}, \delta, \{q_0\}, \{q_3\}\}$

Date :
Page No. :

vii) 0101

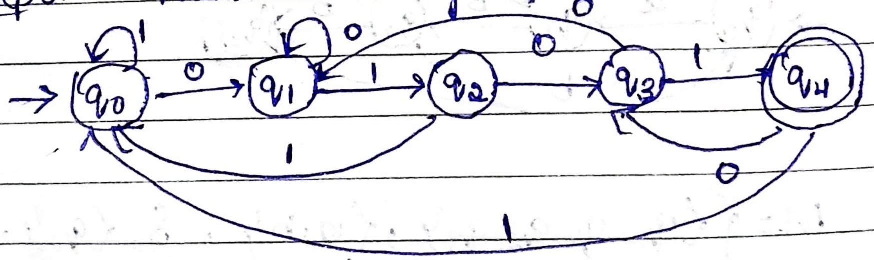
Step 1: → q_0 = Initial state
q_1 = Ends with 0
q_2 = Ends with 01
q_3 = Ends with 010
q_4 = Ends with 0101

Step 2:

	δ	Σ 0	1
→ε	q_0	Ø0	Ø1
0	q_1	Ø0	01
01	q_2	Ø10	ØX1
010	q_3	ØXØ0	Ø101
0101 *	q_4	ØX010	ØXØ11

δ	Σ 0	1
→ q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_3	q_0
q_3	q_1	q_4
q_4 *	q_3	q_0

Step 3: Transition Diagram



$M = \{\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \delta, \{q_0\}, \{q_4\}\}$