1.a) Define automata:-

The term "Automata" is derived from the greek word, which means "Self-Acting". Automata is an abstract Self-propelled computing device which follows a predetermined sequence of operations. Which follows a predetermined sequence of operations automatically.

An automation with finite numbers of state is called — Finite Automata or FA.

* Automata can be defined by (5) touples —

(I) Q - finite set of states.

(II) Σ - alphabets

(III) δ - Tramsisation function.

(IV) q_0 - initial state.

(V) F - final state.

1. b) Types of automata:-

There are 2 types — (I) finite Automata
(II) Infinite Automata.

Now; finite automata
- Deterministic FA (DFA)
- Non-deterministic FA (NFA)

1. (c) NDFA : ~~[illegible]~~ In automata theory, a finite state machine is called DFA if – (i) each of its transisations is uniquely determined by its source state & input symbols.

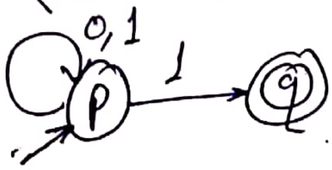
(ii) Reading an input symbol requires a state transisations.

→ But NFA, does not bound to that restrictions. In particularly DFA is an NFA.

(*) example :- Let M bean NFA, determines if input ends with a '1'.

by formal notation :

$M = (\{p, q\}, \{0, 1\}, \delta, p, \{q\})$



	0	1
p	p	p, q
q	ϕ	ϕ

Thus, Regular expression will be

↳ $(0|1)^* 1$

ϕ = dead / trap state.

1.(d) Define RE: Let Σ be an alphabet set. We define regular expression (RE) of alphabet Σ, as follows — if $\Sigma = \{a, b\}$
then a, b, Σ all are RE.
So, it is a formal language that can be expressed using regular expression. Alternatively a regular language expressions & finite automata recongnizes the RE, also known as Kleene's theorem.

(*) ex:- All finite languages are regular, particularly empty string language $(\Sigma) = \phi^*$ is a regular.

$$\boxed{L(r) = \phi^*}$$

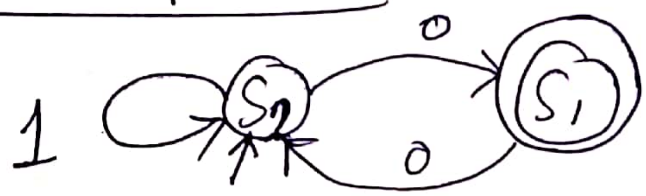
1. (e) define Grammar:- A phase-structure grammar is (V_N, Σ, P, S) where it means —

(i) V_N = finite nonempty set, variables.
(ii) Σ = finite nonempty set, terminals.
(iii) $V_N \cap \Sigma = \phi$
(iv) S = Special variable, starting symbol.
(v) P = production.

BRAINWARE GROUP OF INSTITUTIONS
398 Ramkrishnapur Road, Barasat, Kolkata 700124

1. (f) Transition table:- (b) In Automata theory transition table is a table showing what state a finite state machine will moved to, based on the current state & other inputs. It is essentially a truthtable in which some of the inputs are current state, & output is equal to next state.

Example:-



	1	0
S_1	S_1	S_2
S_2	S_2	S_1

State Transition table for NFA:-

	1	0
S_1	S_1	$\{S_2, S_3\}$
S_2	S_2	S_1
S_3	S_2	S_1

Fig:- Truth table.

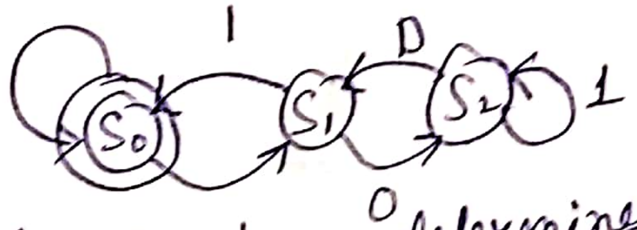
1. (e) define DFA:

In Automata Theory, DFA is a finite state machine that accepts or rejects strings of symbols and produces unique

(8)

Unique Computation. Deterministic comes from uniqueness of the Computation. produces unique output.

④ example:-



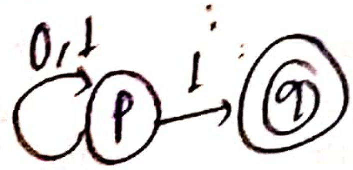
④ Touples:- It can be determines by ⑤

touples →
Q = finite state sets.
Σ = finite set of i/p symbols.
δ = $Q \times \Sigma \rightarrow Q$ [transition table]
q_0 = initial state
F = final state

1. (h) dead state / trap state:-

It NFA, when we don't have any output regading of a input symbol, then the output state is called — dead state.

ex:-



	0	1
p	{p}	{p, q}
q	ϕ	ϕ

∴ ϕ = dead state.

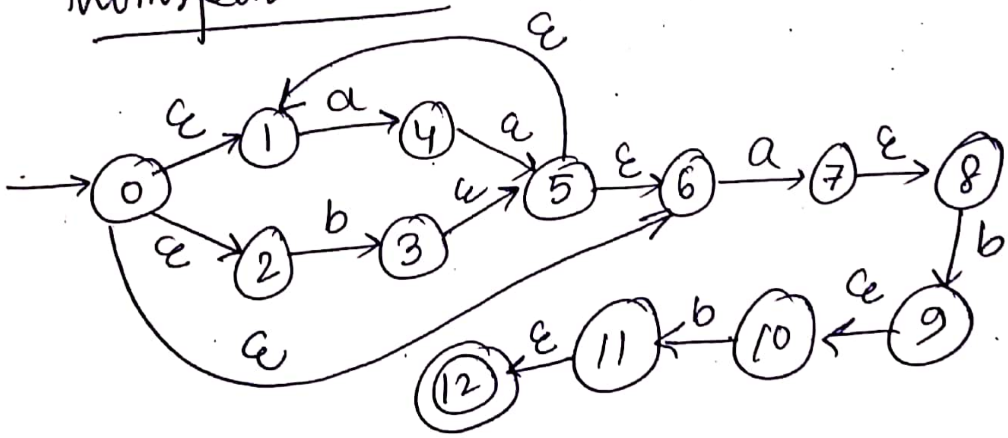
3. $S = \{0,1\}$ ending with '011'.

$L(R)$ = Set of all strings over.

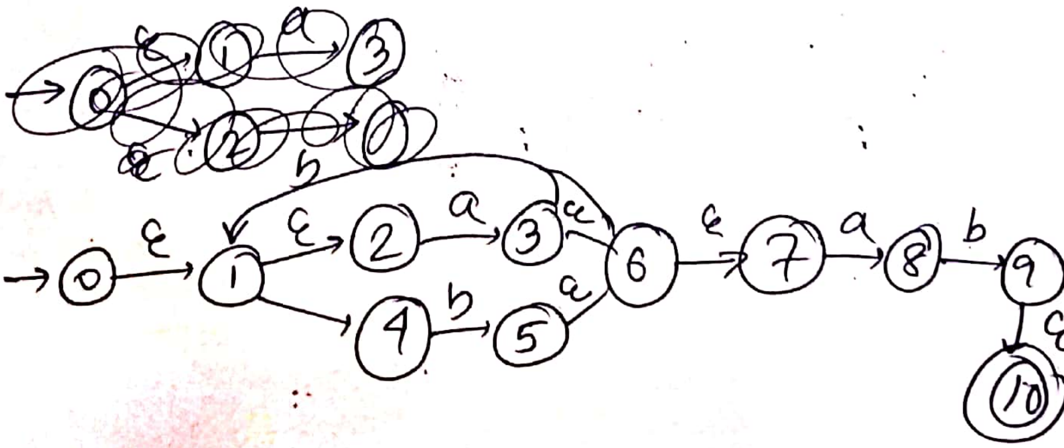
$\therefore L(R) = \{011, 0011, 1011, 1001011, \ldots\}$

$$R = (0+1)^{*}011$$

(2) Regular Expression → $(a|b)^{*}abb$ by Thomspon Rule. NFA.



(*) Compress the NFA if possible:-



$\therefore$ as ϵ is given here, we have to ϵ-closure

$\therefore$ (I) $\epsilon\text{-Closure}(0) \rightarrow \{0,1,7,2,4\} = \{0,1,2,4,7\} = A$

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

$$\text{Move}_{DFA}(A,a) = \epsilon.\text{closure}\left(\text{Move}_{NFA}(A,a)\right)$$
$$= \epsilon.\text{closure}(\{3,8\})$$
$$= \{3,6,1,2,4,8\}$$
$$= \{1,2,3,4,6,7,8\} = B$$

$\therefore \rightarrow A \xrightarrow{a} B \quad \leftarrow$ DFA

(II) $\text{Move}_{DFA}(A,b) = \epsilon.\text{Closure}\left(\text{Move}_{NFA}(A,b)\right)$
$$= \epsilon.\text{closure}(\{5\})$$
$$= \{5,6,7,1,2,4\}$$
$$= \{1,2,4,5,6,7\} = C$$

$$\therefore \begin{tikzcd} & B \\ \rightarrow A \arrow[ur, "a"] \arrow[dr, "b"'] & \\ & C \end{tikzcd}$$

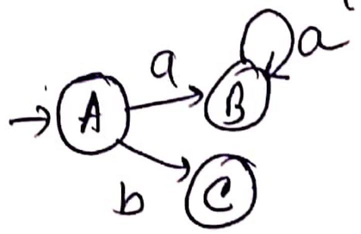
(III) $Move_{DFA}(B, a) = \epsilon.closure(Move_{NFA}(B, a))$

$= \epsilon.closure(\{3, 8\})$

$= \{3, 6, 7, 1, 2, 4, 8\}$

$= \{1, 2, 3, 4, 6, 7, 8\} = B$

DFA.



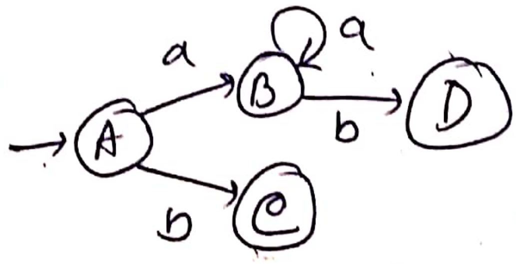
(IV) $Move_{DFA}(B, b) = \epsilon.closure(Move_{NFA}(B, b))$

$= \epsilon.closure(\{5, 9\})$

$= \{5, 6, 7, 1, 2, 4, 9\}$

$= \{1, 2, 4, 5, 6, 7, 9\}$ D

DFA.



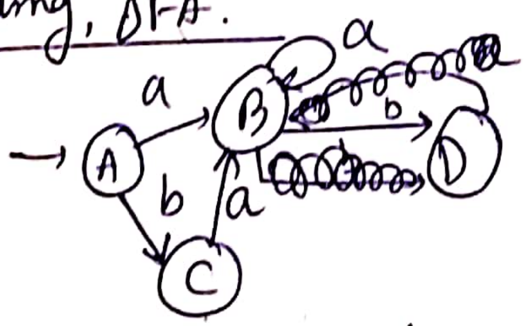
(V) $Move_{DFA}(C, a) = \epsilon.closure(Move_{NFA}(C, a))$

$= \epsilon.closure(\{3, 8\})$

$= \{3, 6, 7, 1, 2, 4, 8\}$

$= \{1, 2, 3, 4, 6, 7, 8\}$ - B

Corresponding DFA.



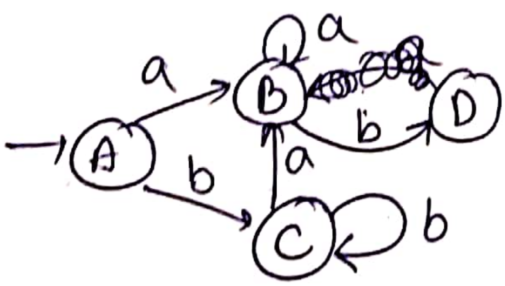
(V) $Move_{DFA}(C, b) = \epsilon.closure(Move_{NFA}(C, b))$

$= \epsilon.closure(\{5\})$

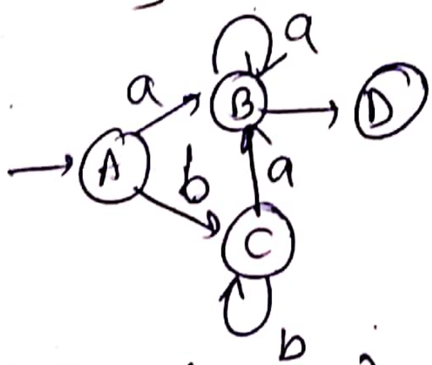
$= \{5, 6, 7, 1, 2, 4\}$

$= \{1, 2, 4, 5, 6, 7\}$ — C

DFA.



→



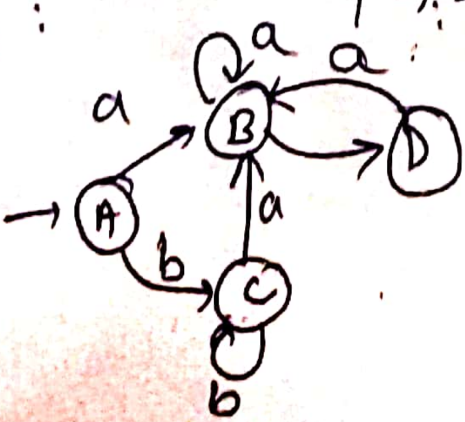
(VI) $Move_{DFA}(D, a) = \epsilon.closure(Move_{NFA}(D, a))$

$= \epsilon.closure(\{3, 8\})$

$= \{3, 6, 7, 1, 2, 4, 8\}$

$= \{1, 2, 3, 4, 6, 7, 8\}$ — B

DFA :

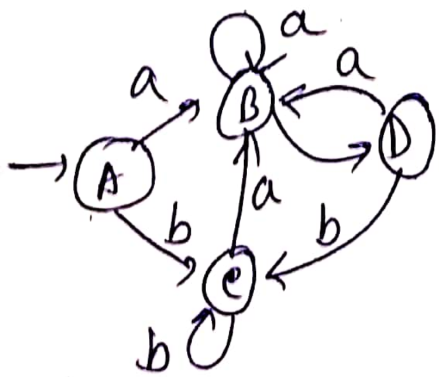


(VII) $\text{Move}_{DFA}(D, b) = \epsilon\text{-closure}(\text{Move}_{NFA}(D, b))$

$= \epsilon\text{-closure}(\{5\})$

$= \{5, 6, 7, 1, 2, 4\} = \{1, 2, 4, 5, 6, 7\}$ (C)

DFA.



∴ DFA will be (optimised)