Types of FA (Finite Automaton) :-

There use two types of FA -

(i) DFA : Deterministic finite Automata

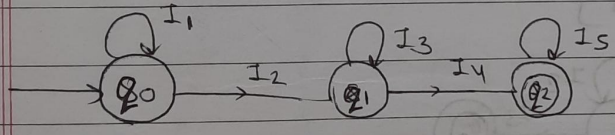
(ii) NDFA or NFA : Non - deterministic FA.

DFA :- It is a special case of FSM, where from each state there will be unique transition for any input symbol.

Technical Representation of DFA :-

$$\delta(Q \times \Sigma) \longrightarrow Q$$

For deterministic automaton (DFA), the outcome is a state, i.e. an element of Q ;



Transition Table :-

δ	I_1	I_2	I_3	I_4	I_5
$\rightarrow q_0$	q_0	q_1	ϕ	ϕ	ϕ ← Phi
q_1	ϕ	ϕ	q_1	q_2	ϕ
q_2	ϕ	ϕ	ϕ	ϕ	q_2

NDFA / NFA :- It is a special case of FSM, where exist some state from which two or more transition are possible for some input symbol. which is called as NFA / NDFA.

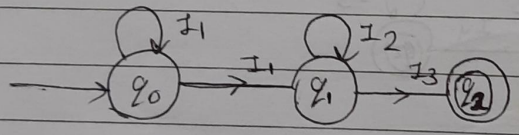
Technical Representation of NDFA :-

$$\delta (Q \times \Sigma) \longmapsto 2^Q$$

2^Q which is the power set of Q, the set of all subsets of Q;

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

$2^Q = \{ \{\phi\}, \{q_0\} \{q_1\} \{q_2\} \{q_0 q_1\} \{q_1 q_2\} \{q_2 q_0\} \{q_0 q_1 q_2\} \}$



Transition Table :-

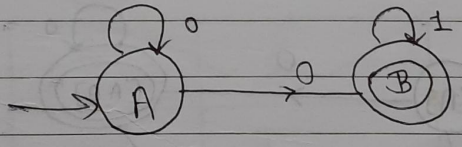
δ	I_1	I_2	I_3
→ q_0	$q_0 q_1$	ϕ	ϕ
q_1	ϕ	q_1	q_2
(*) q_2	ϕ	ϕ	ϕ

Equivalence of DFA and NDFA :-
NFA TO DFA CONVERSION :-

Rules :-

(1) Copy initial state transition as it is into new DFA table.
(2) All multiple states are written in square braket [].
(3) All newly generated o/p states must be consider for making new transition once.
(4) Repeat step (3) till all newly generated o/p states are diverted into transition column.
(5) For final state we always check possible combination of available final state.
(6) If any states does not give any o/p then we mark a dash (-) in o/p.
(7) Define the DFA & draw transition diagram.

1. Q :-

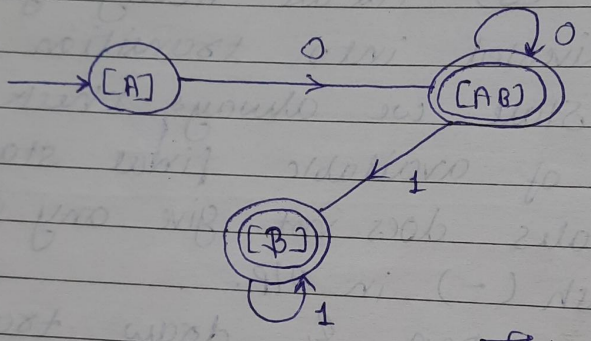


Sol :- **Transition Table of NFA :-**

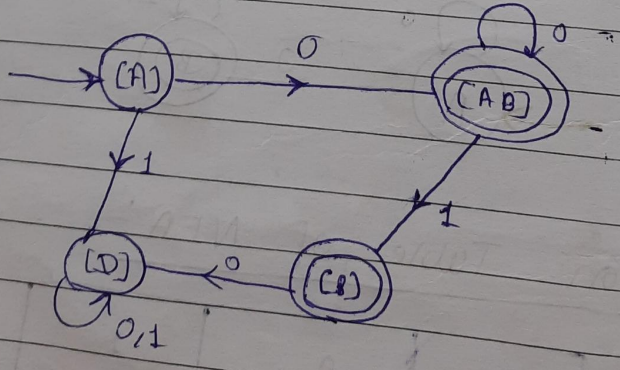
S	0	1
→ A	AB	∅
(B)	∅	B

Transition Table of DFA :-

δ	0	1
→[A]	[AB]	-
*[AB]	[AB]	[B]
*[B]	-	[B]



This is required DFA.



Here, D is Dummy state.

——x——