part-A

1. Define the terms symbol, string and language.

A:- **Symbol:** Symbol is a single object and it is a building block.

Ex: A to Z, a to z, 0 to ∞, *, &,------etc.

String: It is a finite sequence of symbols choosen from some alphabet.

(or)

A String is a finite sequence of symbols over alphabet.

→ It is denoted by 'W' or 'S'

Ex: $W = \{aa, ab, ba, bb\}$.

Language: A set of strings over some alphabet where alphabet $\Sigma = \{a, b\}$.

Ex: set of all strings of length 3

$L = \{aaa, aba, bab, bbb ------\}$.

2. Explain the difference between DFA and NFA.

A:-

DFA	NFA
1) DFA can move to only one state if it is provided with an input character.	1) NFA can move to any number of states with an single input character.
2) In DFA null move is not allowed i.e. DFA cannot change its state without an input character	2) In NFA null move is allowed i.e. it can move from one state to other state without any input symbols
3) In DFA transition function is defined as $\delta : Q \times \Sigma \rightarrow Q$	3) In NFA transition function is defined as $Q \times (\Sigma \cup \epsilon) \longrightarrow 2^Q$.

3. Define the formal definition of Finite Automata?

A:- A Finite Automata contains 5 tuples they are

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

where Q = finite set of all strings

Σ = finite set of input symbols

δ = Transition function

$Q \times \Sigma \rightarrow Q$

q_0 = Initial State

F = Final State

4. Define non-deterministic finite automata with example?

A:- The Finite Automata are called non-deterministic finite automata when their exist many paths for specific input from the current state to next state

→ In NFA null move is allowed

Formal Definition of NFA:

It contains 5 tuples. they are

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

Q = set of states

Σ = finite set of input symbols

δ = Transition function

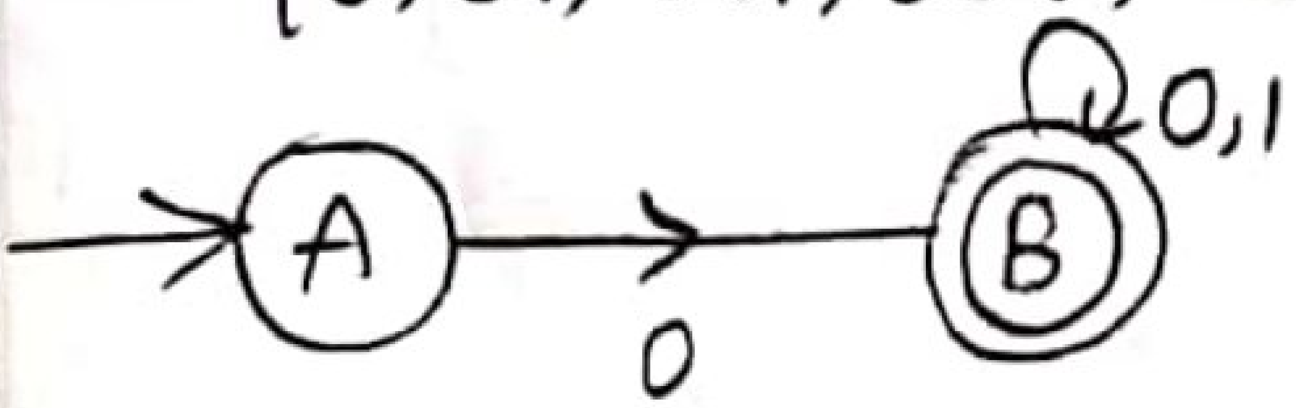
$Q \times \Sigma \rightarrow 2^Q$

q_0 = Initial State

F = Final State

Example:- NFA which accepts set of all strings over $\{0,1\}$ starts with '0'.

$L = \{0, 01, 011, 000, ----\}$



$Q = \{A, B\}$

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

$q_0 = \{A\}$

$F = \{B\}$

δ : $\delta(A,0) = B$

$\delta(A,1) = \phi$

$\delta(B,0) = B$

$\delta(B,1) = B$

$Q \backslash \Sigma$	0	1
A	B	ϕ
B	B	B

5 Define deterministic finite automata with example

A: The Finite automata are called deterministic finite automata if the machine is read an input stream one symbol at a time.

→ In DFA there is only one path for specific input from the current state to next state

Formal definition of DFA:

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

Q = finite set of all strings

Σ = finite set of input symbols

δ = Transition Function

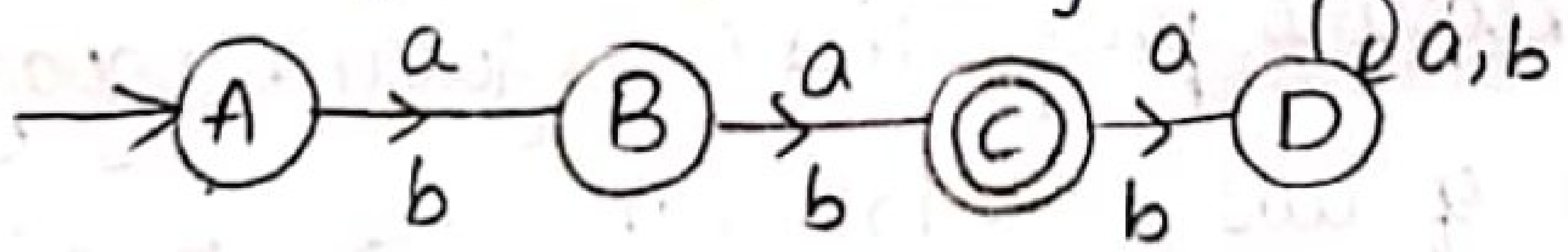
$$Q \times \Sigma \rightarrow Q$$

q_0 = Initial state

F = Final state

Example: DFA that accepts set of all strings over $\Sigma = \{a, b\}$ of length 2

$$L = \{aa, ab, ba, bb\}$$



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

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

$q_0 = \{A\}$

$F = \{C\}$

$\delta(A,a) = B$

$\delta(A,b) = B$

$\delta(B,a) = C$

$\delta(B,b) = C$

$\delta(C,a) = D$

$\delta(C,b) = D$

$\delta(D,a) = D$

$\delta(D,b) = D$

$Q \backslash \Sigma$	a	b
A	B	B
B	C	C
C	D	D
D	D	D

6 List the applications of finite Automata?

A: Applications of finite automata:

→ It plays an important role in compiler design

→ In Switching theory and design and analysis of digital circuits automata theory is applied.

→ Design and analysis of complex hardware and software systems

→ To prove the correctness of the program automata theory is used.

→ To Design finite state machines such as moore machine, mealy machine.

→ It is the base for the formal languages and these formal languages are useful for the programming languages

7. Define ε-NFA with Example

A. In ε-NFA by taking 'ε' as an input symbol it will move from one state to another state

Formal Definition of ε-NFA:

It contains 5 tuples those are

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

Q = finite set of all states

Σ = finite set of input symbols

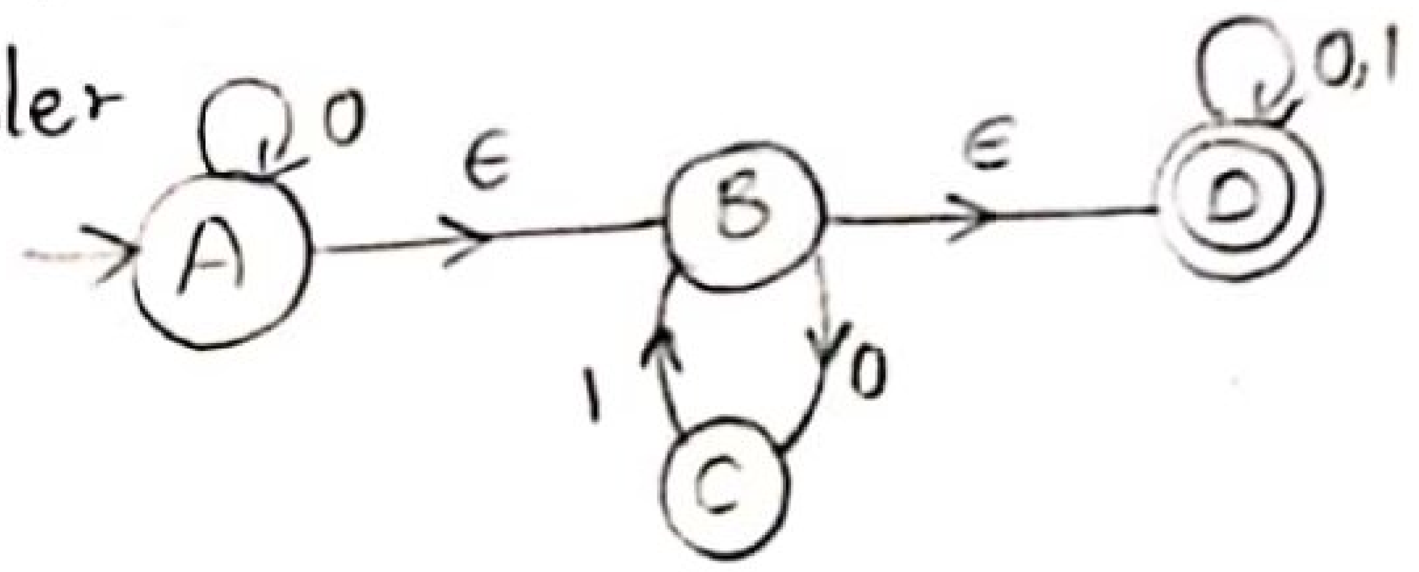
δ = Transition function

$Q \times \Sigma \cup \epsilon \longrightarrow 2^Q$

q_0 = Initial state

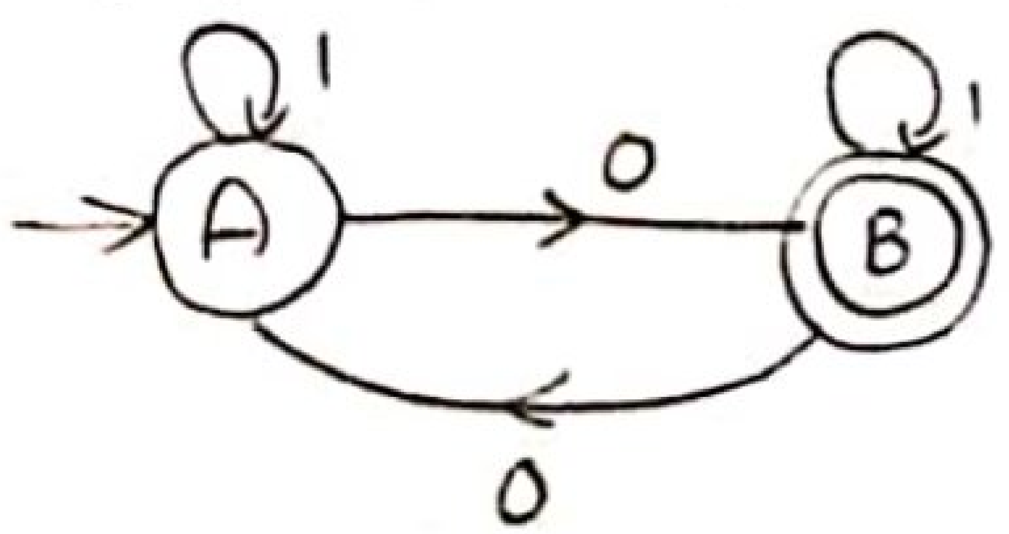
F = Final state

Example:



8. Construct DFA for a string accepting odd number of 0's

A. $L = \{0, 1, 000, 10000, \ldots\}$



$Q = \{A, B\}$

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

$q_0 = \{A\}$

$F = \{B\}$

Scanned with CamScanner

$\delta =$

Q \ Σ	0	1
A	B	A
B	A	B

Scanned with CamScanner

9. Define ϵ-closure.

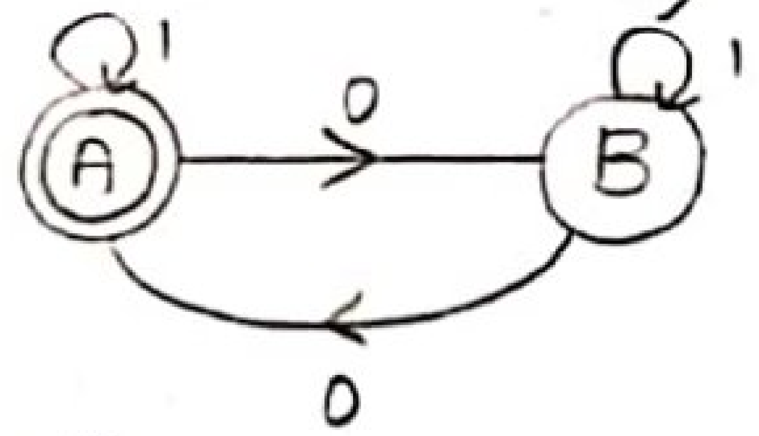
A: Epsilon means present state can goto other state without any input. This can happen only if the present state have epsilon transition to other state. Epsilon closure is finding all the states which can be reached from the present state on one or more epsilon transitions.

→ Every state on 'ϵ' goes to itself.

10 Construct DFA for a string accepting even number of 0's.

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

$L = \{00, 001, \ldots\ldots\}$



$Q = \{A, B\}$

$\delta \Rightarrow \delta(A, 0) = B$

$\delta(A, 1) = A$

$\delta(B, 0) = A$

$\delta(B, 1) = B$

Transition table:

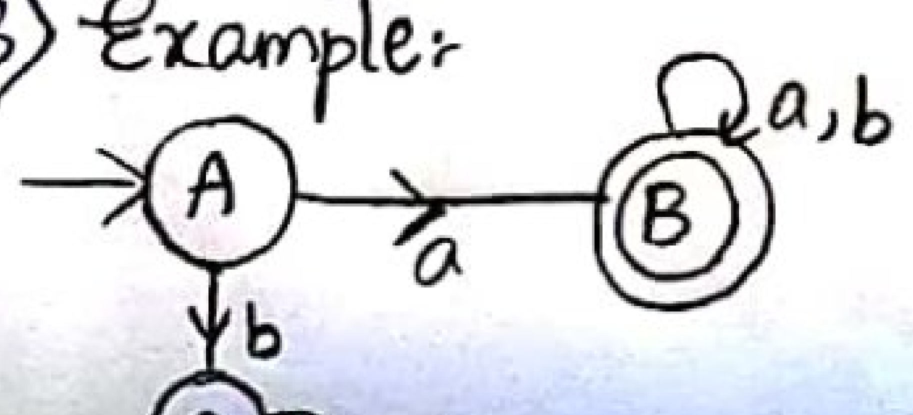
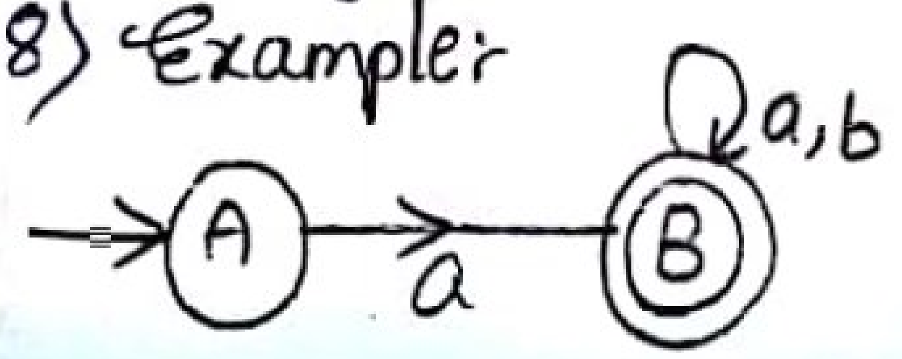
Q \ Σ	0	1
A	B	A
B	A	B

Scanned with CamScanner

PART-B (LONG ANSWERS)

11\. a) Distinguish between DFA and NFA

A:

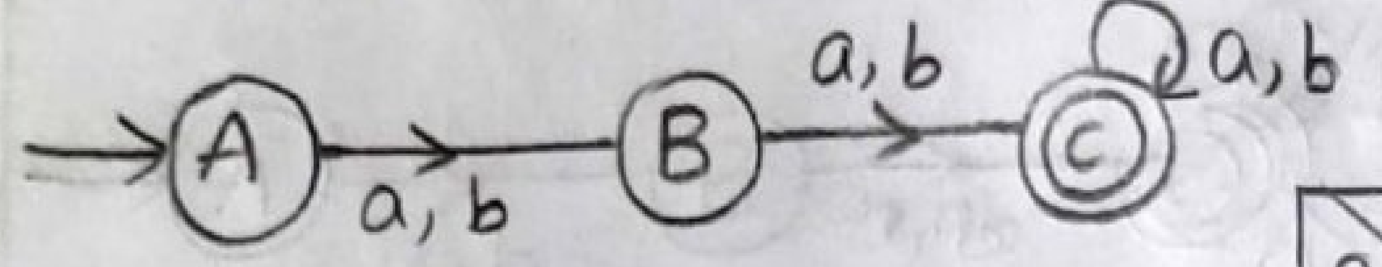
DFA	NFA
1) DFA stands for Deterministic Finite Automata	1) NFA stands for Non-Deterministic Finite Automata
2) DFA can move to only one state if it is provided with an input character	2) NFA can move to any number of states with an single input character
3) In DFA null move is not allowed i.e. DFA cannot change its state without an input character	3) In NFA null move is allowed i.e. it can move frome one state to another state without any input symbols
4) In DFA transition function is defined as $\delta : Q \times \Sigma \rightarrow Q$	4) In NFA transition function is defined as $Q \times (\Sigma \cup \epsilon) \longrightarrow 2^Q$
5) Backtracking is allowed	5) Backtracking is not allowed
6) Consumes more memory space	6) Consumes less memory space
7) Every DFA is not NFA	7) Every NFA is DFA
8) Example:- 	8) Example:- 

b) Design DFA for the following over $\{a, b\}$

i) All strings that has atleast length is 2

$|W| \geq 2$

$L = \{aa, ab, ba, bb, aaa, aba, \text{------}\}$



Q \ Σ	a	b
A	B	B
B	C	C
C	C	C

$Q = \{A, B, C\}$

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

$\delta: \delta(A, a) = B$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(B, b) = C$

$\delta(C, a) = C$

$\delta(C, b) = C$

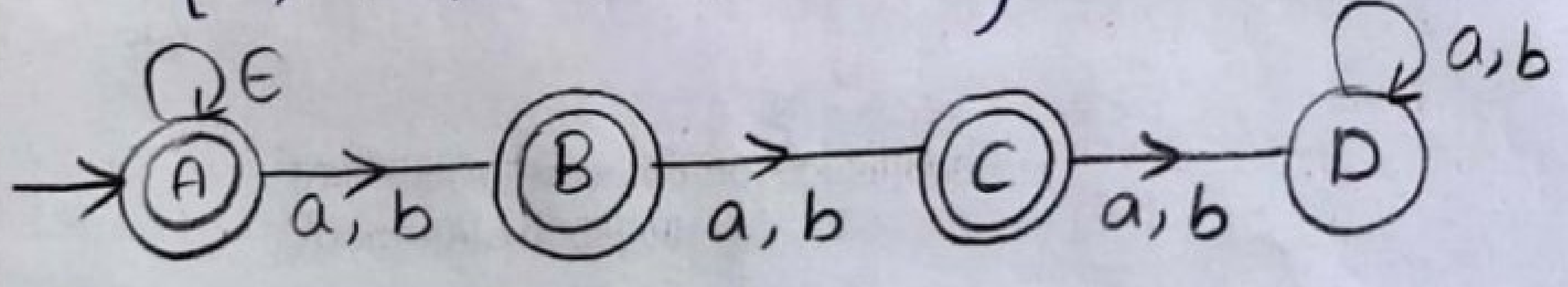
$q_0 = \{A\}$

$F = \{C\}$

ii) All Strings that has almost length is 2

$|W| \leq 2$

$L = \{\epsilon, a, b, aa, ab, ba, bb\}$



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

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

$q_0 = \{A\}$

$F = \{A, B, C\}$

δ: $\delta(A, \epsilon) = A$

$\delta(A, a) = B$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(B, b) = C$

$\delta(C, a) = D$

$\delta(C, b) = D$

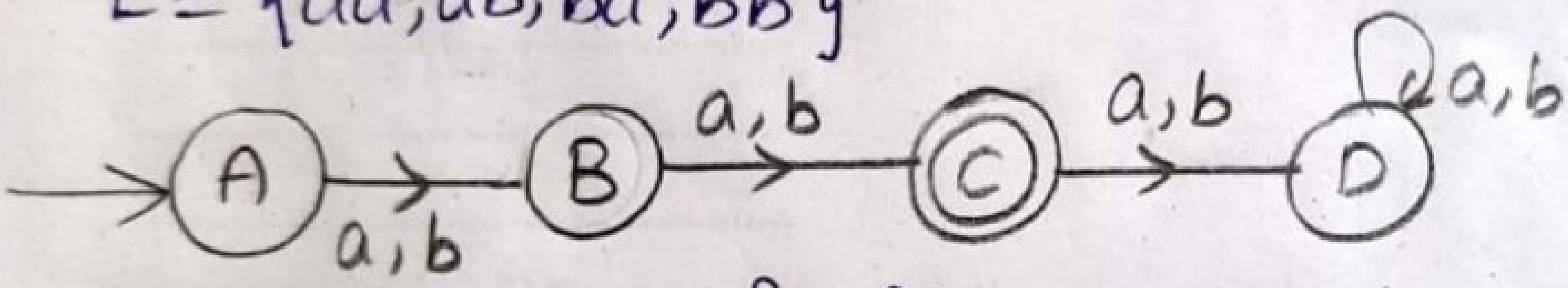
$\delta(D, a) = D$

$\delta(D, b) = D$

Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	D	D

iii) All strings that has exactly length is 2

$L = \{aa, ab, ba, bb\}$



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

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

$q_0 = \{A\}$

$F = \{C\}$

δ: $\delta(A, a) = B$

$\delta(A, b) = B$

$\delta(B, a) = C$

$\delta(B, b) = C$

$\delta(C, a) = D$

$\delta(C, b) = D$

$\delta(D, a) = D$

$\delta(D, b) = D$

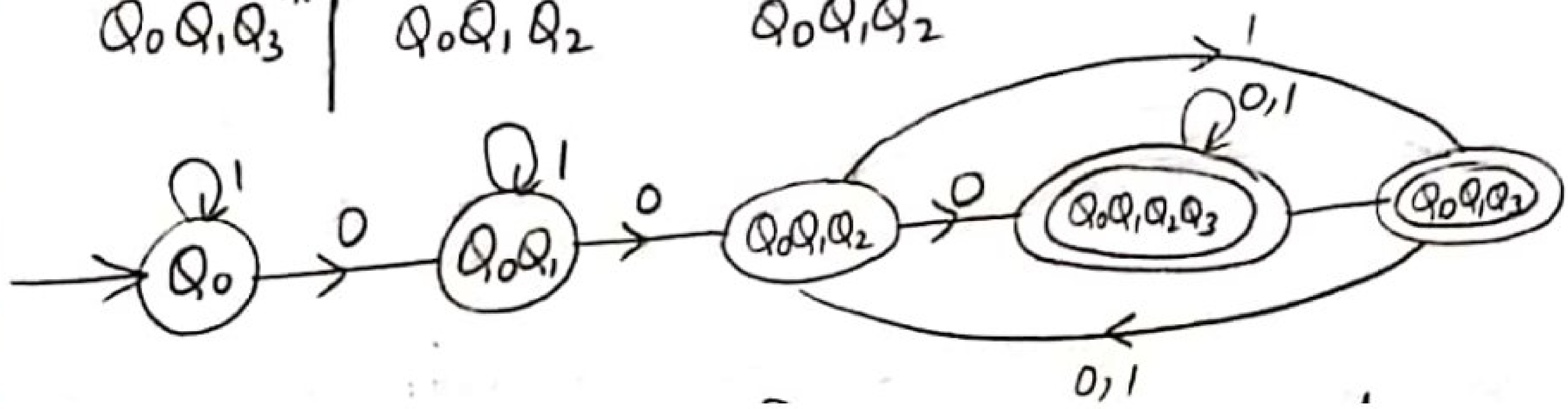
Q \ Σ	a	b
A	B	B
B	C	C
C	D	D
D	D	D

12 Construct a DFA equivalent to the following NFA

State/Σ	0	1
Q_0	$\{Q_0, Q_1\}$	Q_0
Q_1	Q_2	Q_1
Q_2	Q_3	Q_3
Q_3	ϕ	Q_2

A:

Q \ Σ	0	1
$\rightarrow Q_0$	Q_0Q_1	Q_0
Q_0Q_1	$Q_0Q_1Q_2$	Q_0Q_1
$Q_0Q_1Q_2$	$Q_0Q_1Q_2Q_3$	$Q_0Q_1Q_3$
$Q_0Q_1Q_2Q_3^*$	$Q_0Q_1Q_2Q_3$	$Q_0Q_1Q_2Q_3$
$Q_0Q_1Q_3^*$	$Q_0Q_1Q_2$	$Q_0Q_1Q_2$



13 Define DFA. Construct a minimal DFA over $\{a,b\}$ where language $L = \{a^n b^m / n,m >= 1\}$

A: The Finite automata are called DFA if the machine is read an input stream one symbol at a time

→ In DFA there is only one path for specific input from the current state to next state

Formal Definition of DFA:

It contains 5 tuples. They are

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

Q = finite set of all strings

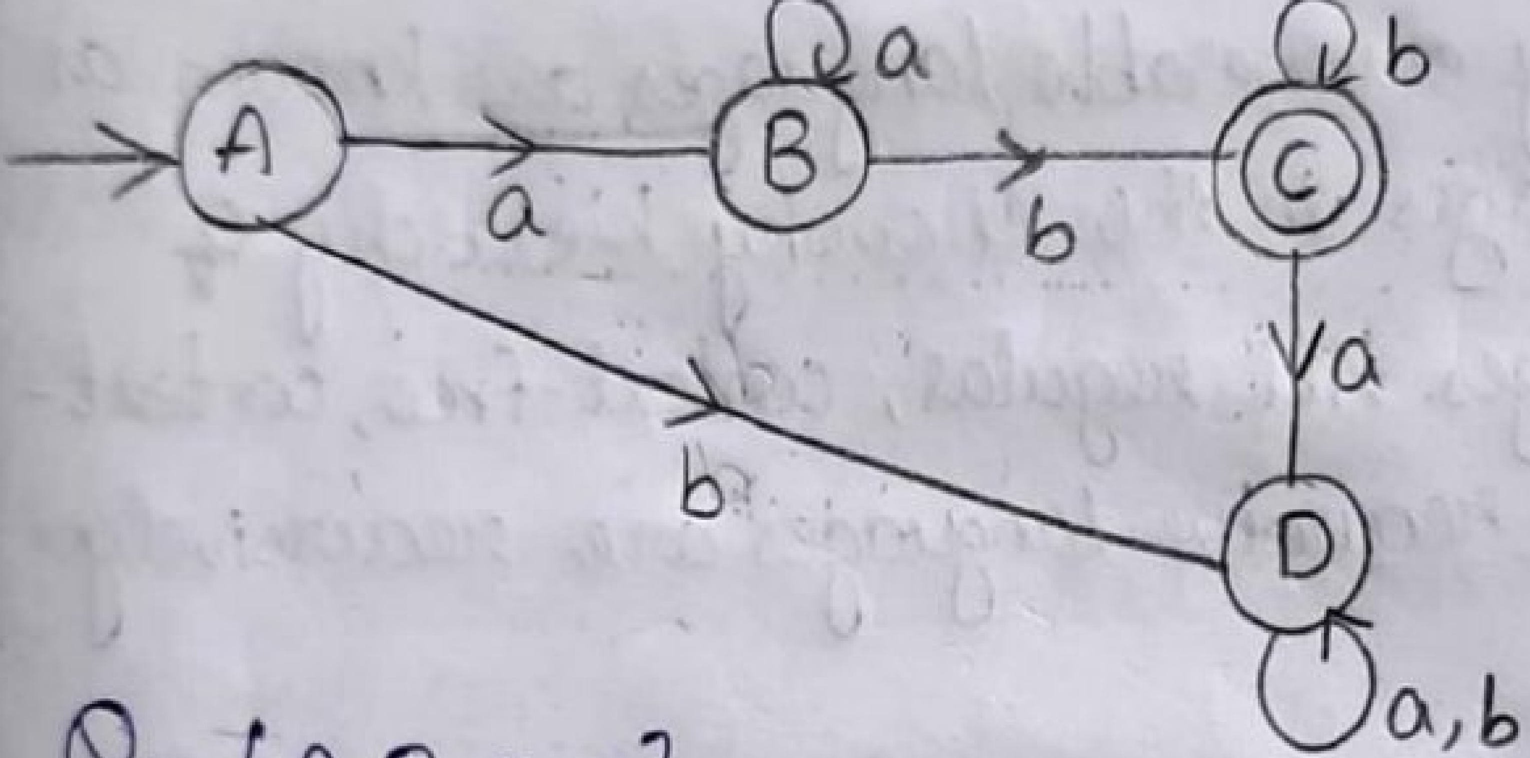
Σ = finite set of input symbols

δ = Transition function

$Q \times \Sigma \rightarrow Q$

q_0 = initial state
F = final state
Given, $L = \{a^n b^m / n, m >= 1\}$

$L = \{ab, aab, abbb, aaabbbb, \ldots\}$

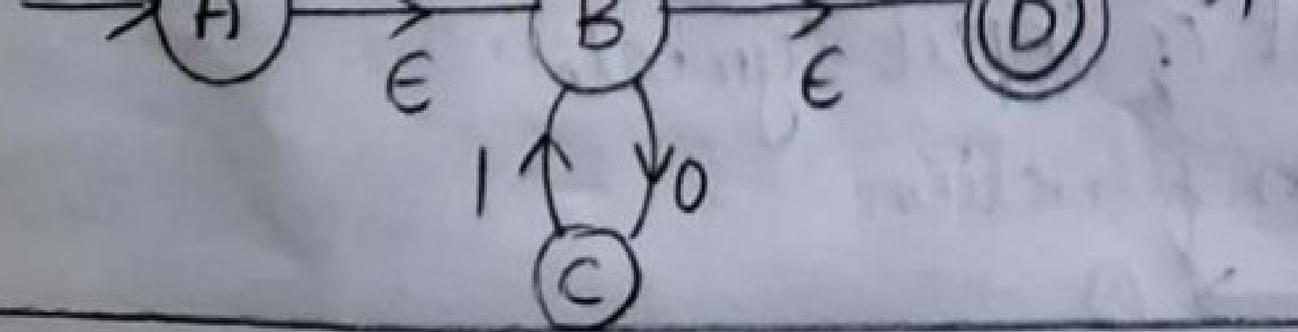


$Q = \{A, B, C, D\}$
$\Sigma = \{a, b\}$
$q_0 = \{A\}$
$F = \{C\}$

δ: $\delta(A,a) = B$
$\delta(A,b) = D$
$\delta(B,a) = B$
$\delta(B,b) = C$
$\delta(C,a) = D$
$\delta(C,b) = C$
$\delta(D,a) = D$
$\delta(D,b) = D$

Q \ Σ	a	b
A	B	D
B	B	C
C	D	C
D	D	D

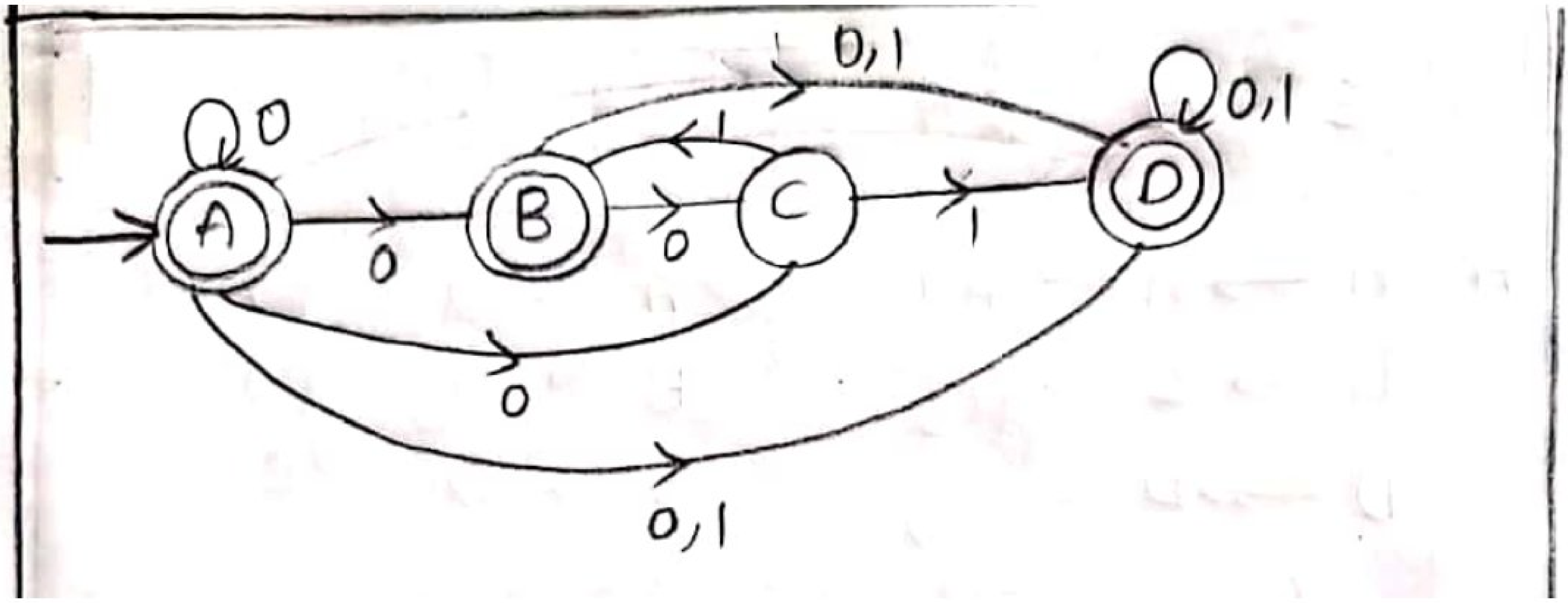
14 Design NFA from the given ε-NFA



A:

State	ϵ^* 0 ϵ^*	ϵ^* 1 ϵ^*
A	$A \xrightarrow{0} A - \{A,B,D\}$ $B \xrightarrow{0} C - \{C\}$ $D \xrightarrow{0} D - \{D\}$ $\{A,B,C,D\}$	$A \xrightarrow{1} \phi - \{\}$ $B \xrightarrow{1} \phi - \{\}$ $D \xrightarrow{1} D - \{D\}$ $\{D\}$
B	$B \xrightarrow{0} C - \{C\}$ $D \xrightarrow{0} D - \{D\}$ $\{C,D\}$	$B \xrightarrow{1} \phi - \{\}$ $D \xrightarrow{1} D - \{D\}$ $\{D\}$
C	$C \xrightarrow{0} \phi - \phi$ $\{\}$	$C \xrightarrow{1} B - \{B,D\}$ $\{B,D\}$
D	$D \xrightarrow{0} D - \{D\}$ $\{D\}$	$D \xrightarrow{1} D - \{D\}$ $\{D\}$

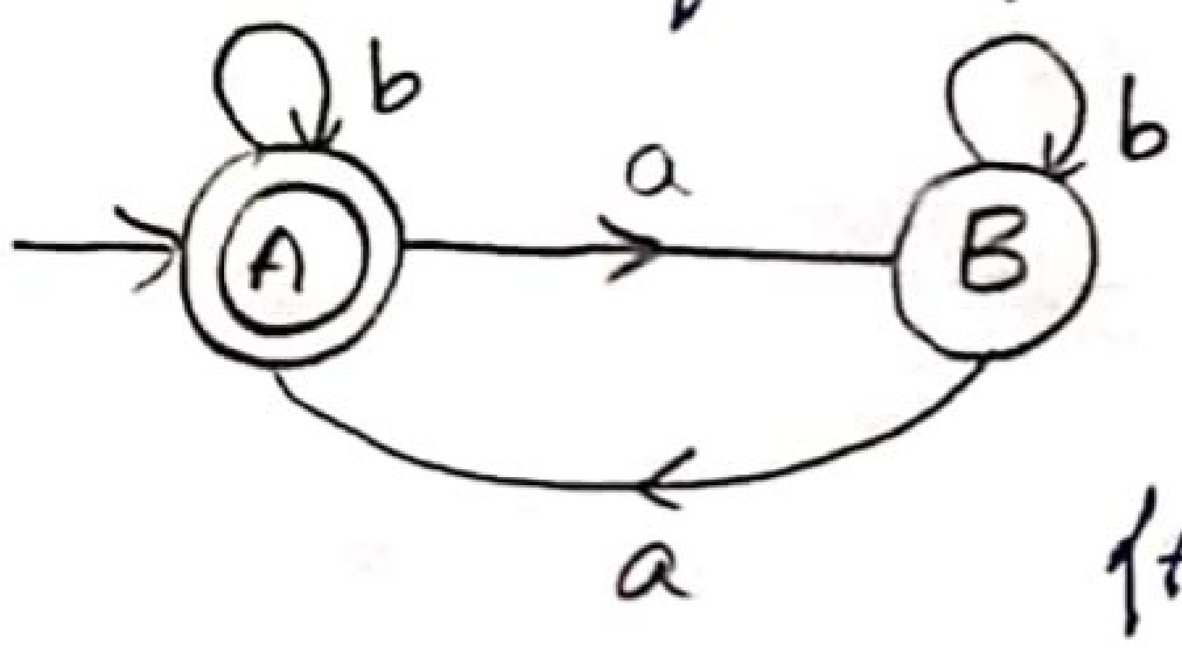
States/Σ	0	1
$\rightarrow A^*$	$\{A,B,C,D\}$	$\{D\}$
B^*	$\{C,D\}$	$\{D\}$
C	ϕ	$\{B,D\}$
D^*	$\{D\}$	$\{D\}$


A
B
C
D
0
0
0
0
1
1
0,1
0,1
0,1

15 Construct DFA to accept the language of all strings of even enumbers of a's & numbers of b's over alphabet $\Sigma = \{a, b\}$

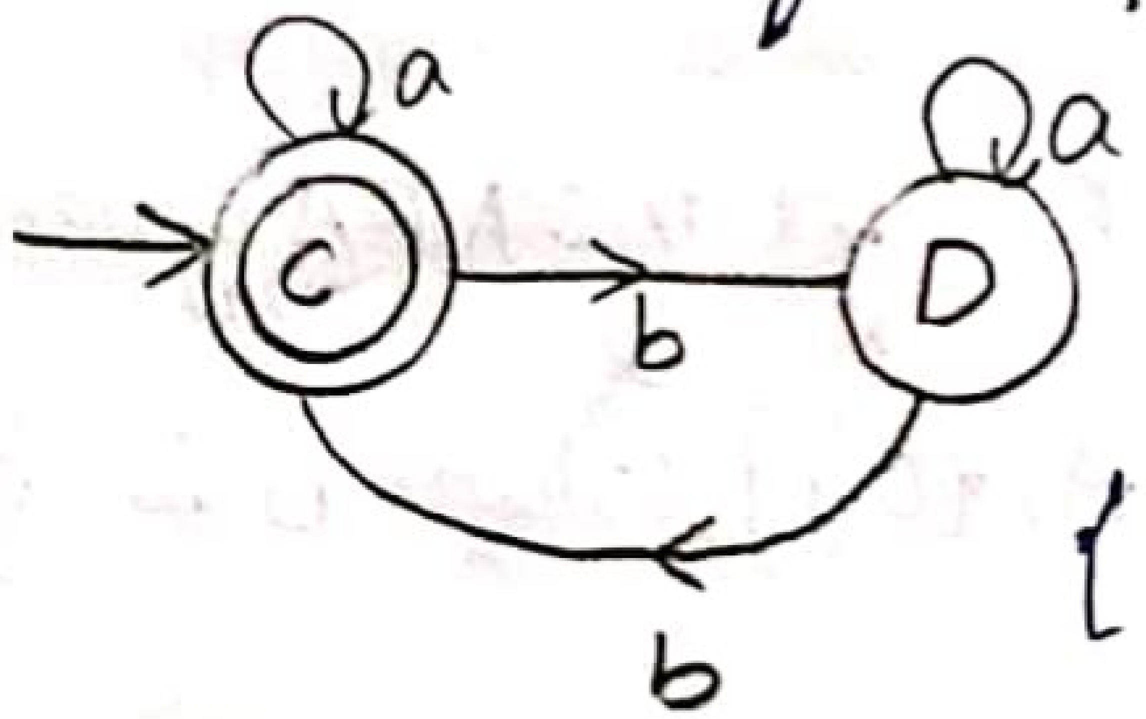
A: $\Sigma = \{a, b\}$

even number of a's :



$\{A, B\}$

even number of b's :



$\{C, D\}$

$\{A,B\} \times \{C,D\} = \{AC, AD, BC, BD\}$

$L = \{aabb, aaaabbbb, \ldots\ldots\}$

$AC \xrightarrow{a} BC$

$AC \xrightarrow{b} AD$

$AD \xrightarrow{a} BD$

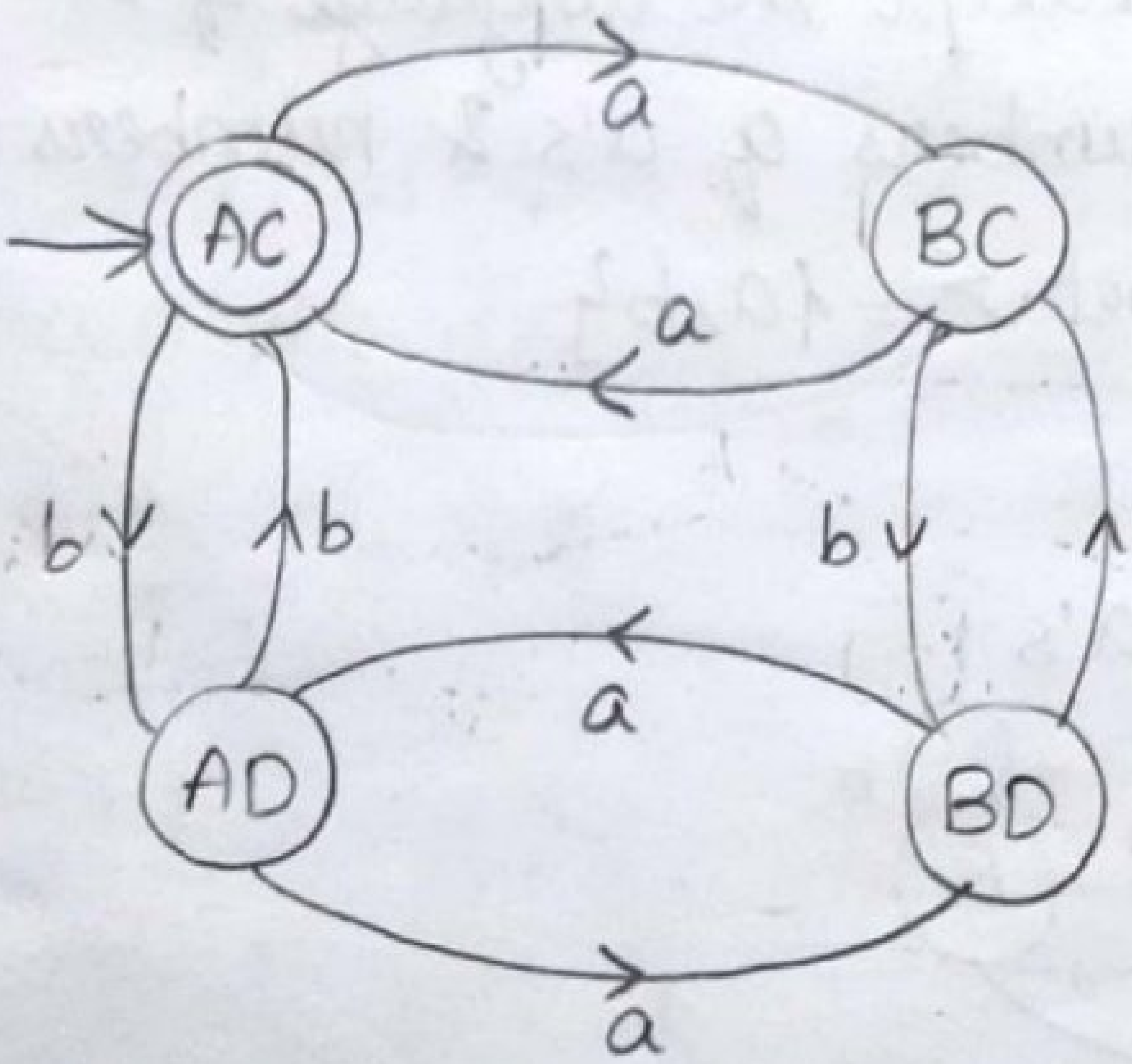
$AD \xrightarrow{b} AC$

$BC \xrightarrow{a} AC$

$BC \xrightarrow{b} BD$

$BD \xrightarrow{a} AD$

$BD \xrightarrow{b} BC$



$q_0 = \{AC\}$

$F = \{AC\}$

16. Find the equivalent DFA for the NFA given by $M = [\{A,B,C\}, \{a,b\}, \delta, A, \{C\}]$ where δ is given by

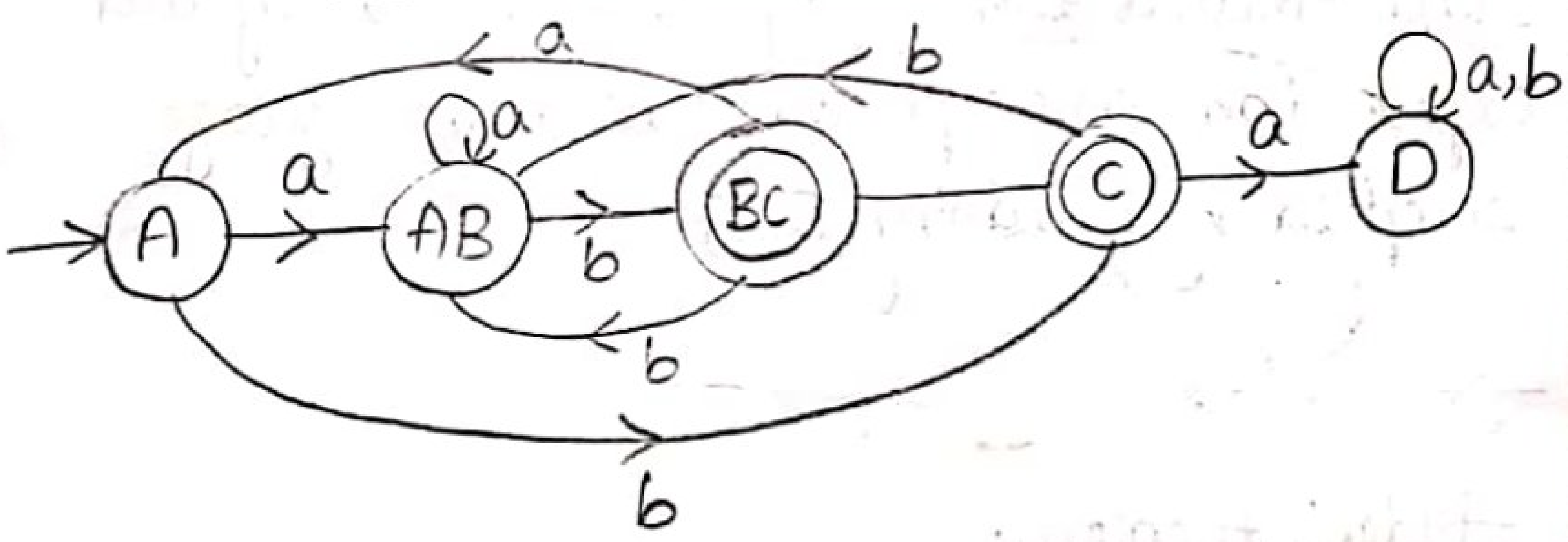
Scanned by CamScanner

Scanned with CamScanner

Q \ Σ	a	b
→A	{A,B}	C
B	A	B
C*	ϕ	{A,B}

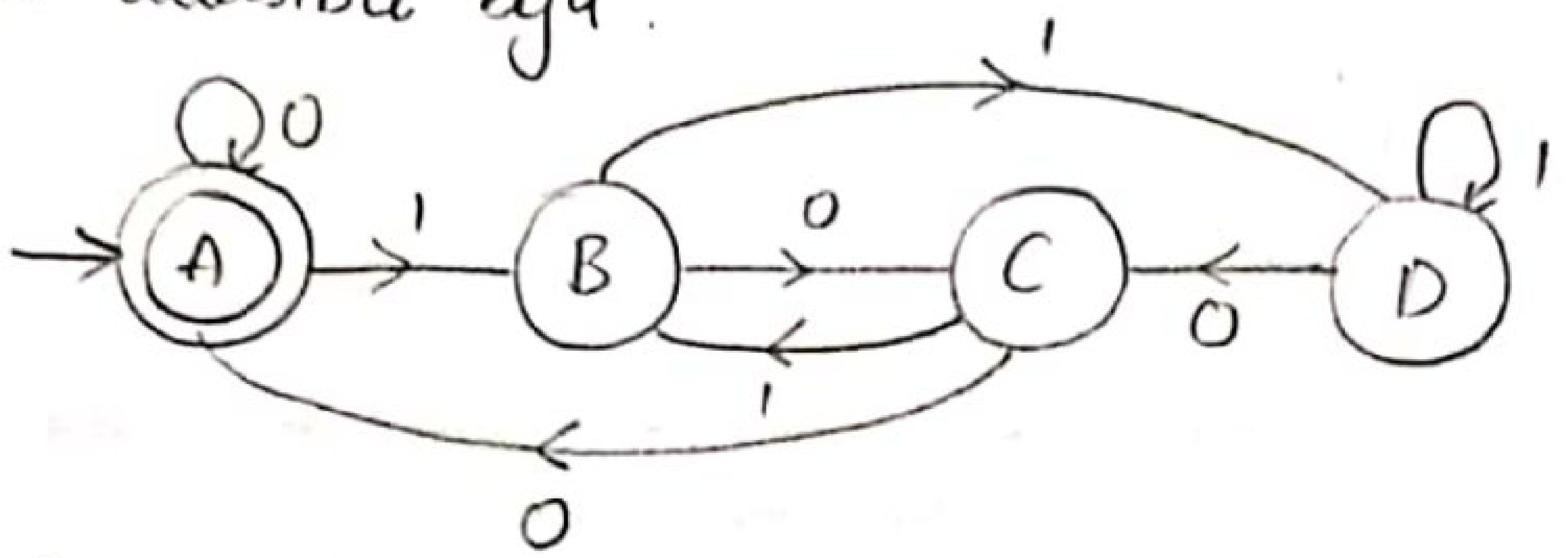
A:

Q \ Σ	a	b
→A	AB	C
AB	AB	BC
BC*	A	AB
C*	D	AB
D	D	D



17 Construct a minimal DFA which accepts set of all strings over {0, 1} which when interpreted as binary number is divisible by 4.

Ans



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

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

$q_0 = \{A\}$

$F = \{A\}$

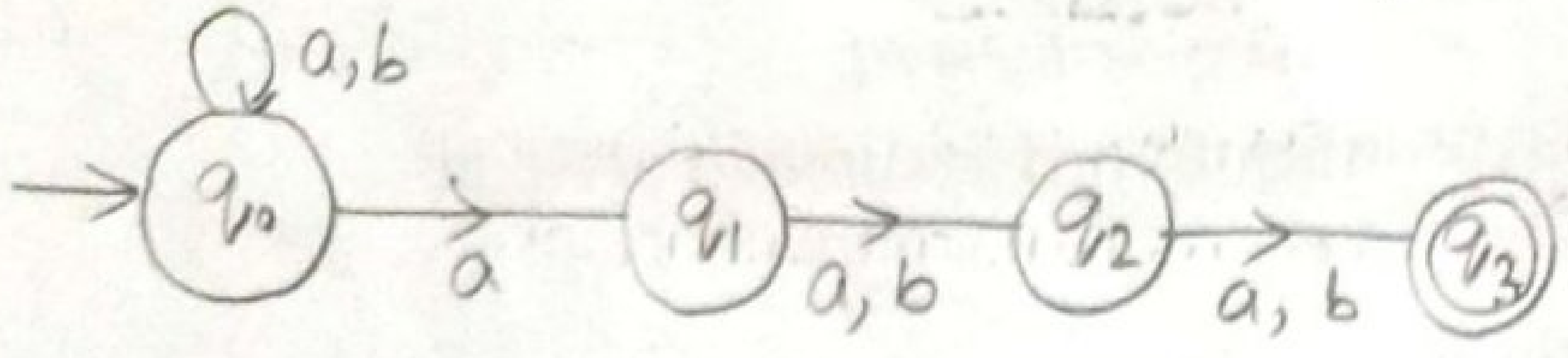
$\delta \Rightarrow$

Q \ Σ	0	1
→A*	A	B
B	C	D
C	A	B
D	C	D

18 a) Define NFA. Construct NFA which accepts set of all strings over {a, b} where third symbol from R.H.S is 'a'

Ans NFA: The Finite automata are called NFA when there exist many paths for specific input from the current state to next state.

$L = \{abb, baaa, aaa, \ldots\ldots\}$



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

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

$q_0 = \{q_0\}$

$F = \{q_3\}$

$\delta \Rightarrow$

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

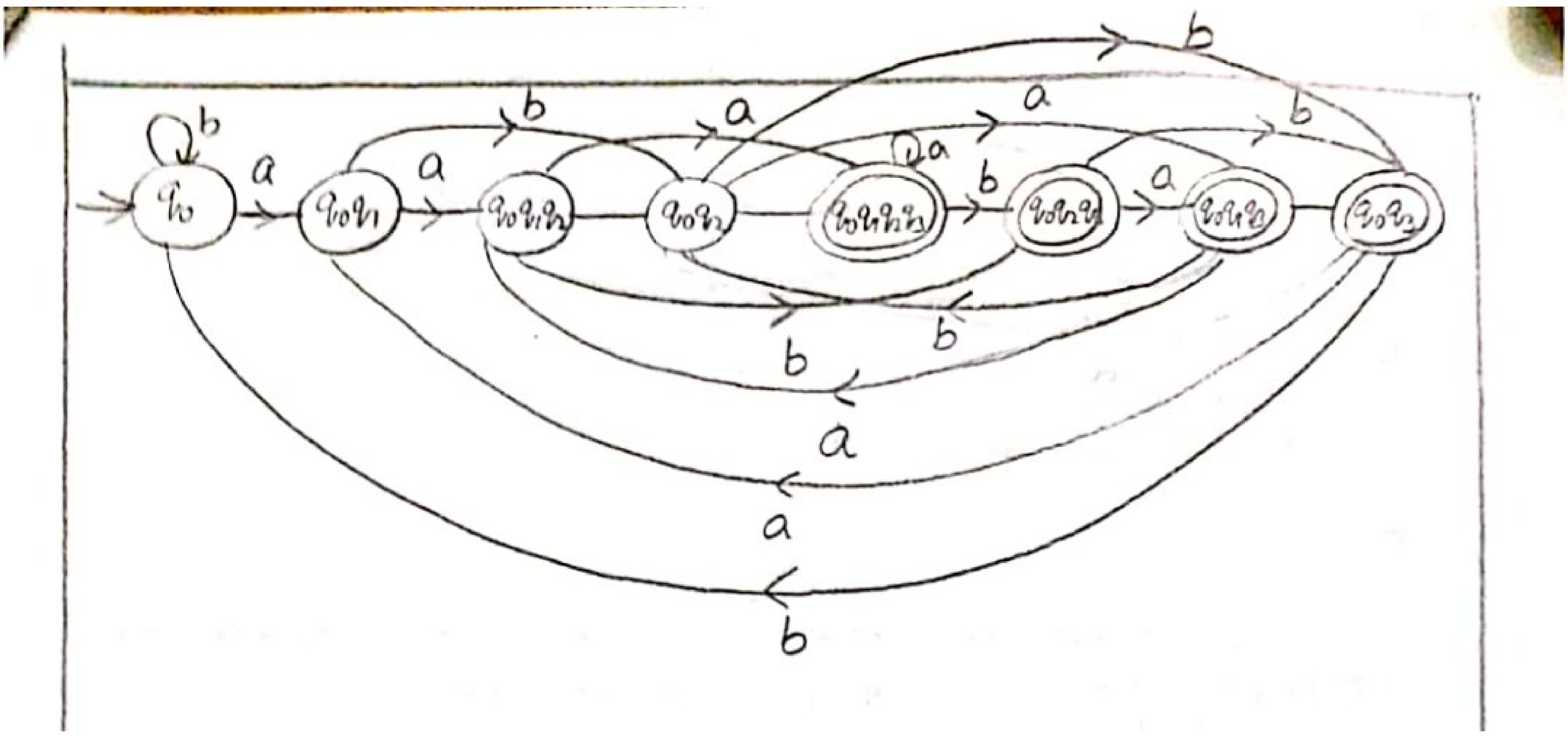
b) Convert that corresponding NFA to DFA

Ans

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

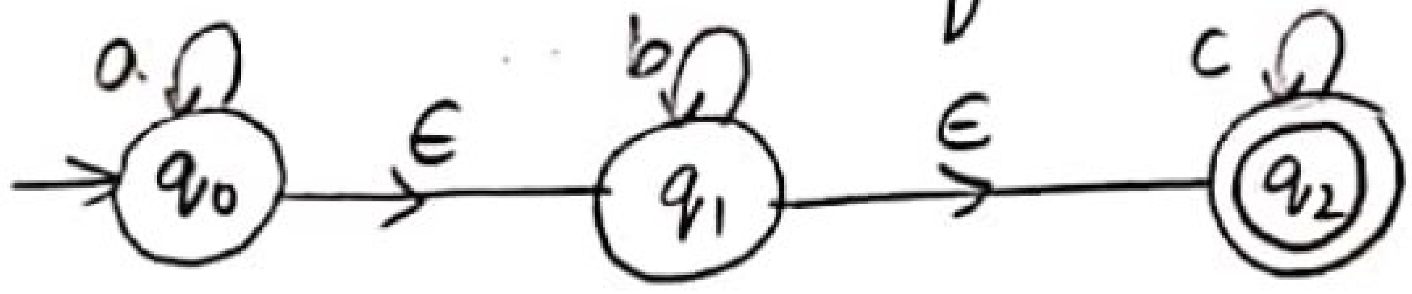
$\Rightarrow$

Q \ Σ	a	b
$\rightarrow q_0$	$[q_0 q_1]$	$[q_0]$
$q_0 q_1$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
$q_0 q_1 q_2$	$[q_0 q_1 q_2 q_3]$	$(q_0 q_2 q_3)$
$q_0 q_2$	$[q_0 q_1 q_3]$	$[q_0 q_3]$
$q_0 q_1 q_2 q_3$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_2 q_3]$
$q_0 q_2 q_3$	$[q_0 q_1 q_3]$	$[q_0 q_3]$
$q_0 q_1 q_3$	$[q_0 q_1 q_2]$	$[q_0 q_2]$
$q_0 q_3$	$[q_0 q_1]$	$[q_0]$



Scanned with CamScanner

19) Construct the DFA from ϵ-NFA



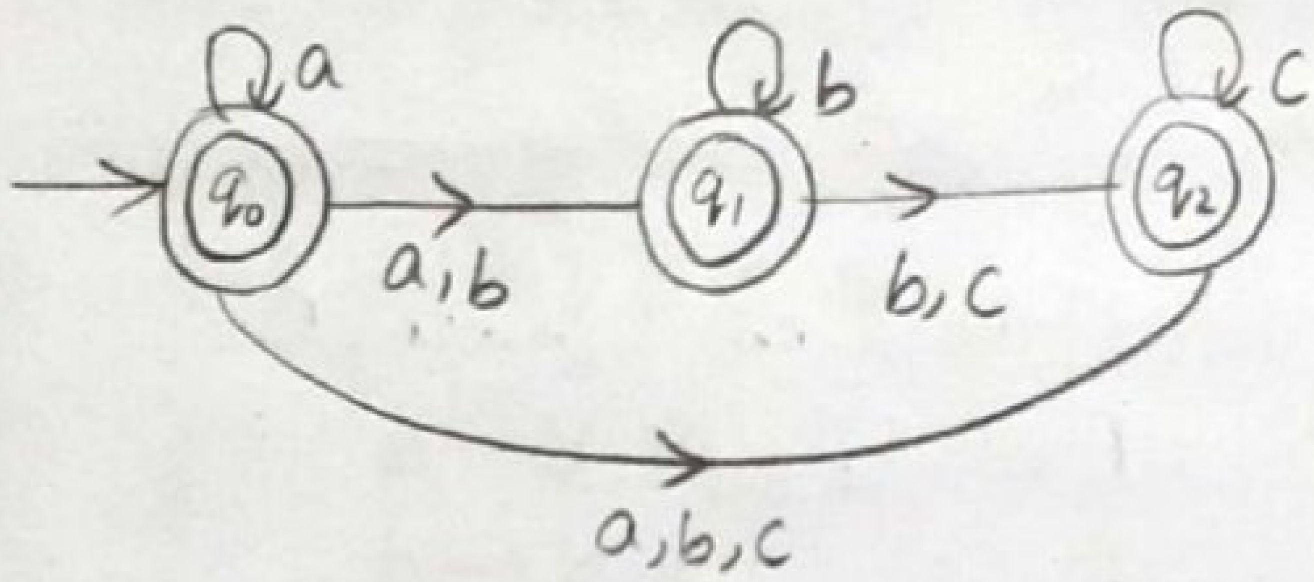
Ar

$\overset{q_0}{\cancel{A}} \Rightarrow \epsilon^* \quad a \quad \epsilon^*$ $q_0 \xrightarrow{a} q_0 \left\langle \begin{matrix} q_0 \\ q_1 \\ q_2 \end{matrix} \right.$ $q_1 \xrightarrow{a} \phi$ $q_2 \xrightarrow{a} \phi$ $\{q_0, q_1, q_2\}$	$\epsilon^* \quad b \quad \epsilon^*$ $q_0 \xrightarrow{b} \phi$ $q_1 \xrightarrow{b} q_1 \left\langle \begin{matrix} q_2 \\ q_1 \end{matrix} \right.$ $q_2 \xrightarrow{b} \phi$ $\{q_1, q_2\}$	$\epsilon^* \quad c \quad \epsilon^*$ $q_0 \xrightarrow{c} \phi$ $q_1 \xrightarrow{c} \phi$ $q_2 \xrightarrow{c} q_2 - q_2$ $\{q_2\}$
$q_1 \Rightarrow \epsilon^* \quad a \quad \epsilon^*$ $q_1 \xrightarrow{a} \phi$ $q_2 \xrightarrow{a} \phi$ $\{ \}$	$\epsilon^* \quad b \quad \epsilon^*$ $q_1 \xrightarrow{b} q_1 \left\langle \begin{matrix} q_1 \\ q_2 \end{matrix} \right.$ $q_2 \xrightarrow{b} \phi$ $\{q_1, q_2\}$	$\epsilon^* \quad c \quad \epsilon^*$ $q_1 \xrightarrow{c} \phi$ $q_2 \xrightarrow{c} q_2 - q_2$ $\{q_2\}$

$q_2 \Rightarrow \epsilon^*\ a\ \epsilon^*$	$\epsilon^*\ b\ \epsilon^*$	$\epsilon^*\ c\ \epsilon^*$
$q_2 \xrightarrow{a} \phi$	$q_2 \xrightarrow{b} \phi$	$q_2 \xrightarrow{c} q_2 - q_2$
$\{\}$	$\{\}$	$\{q_2\}$

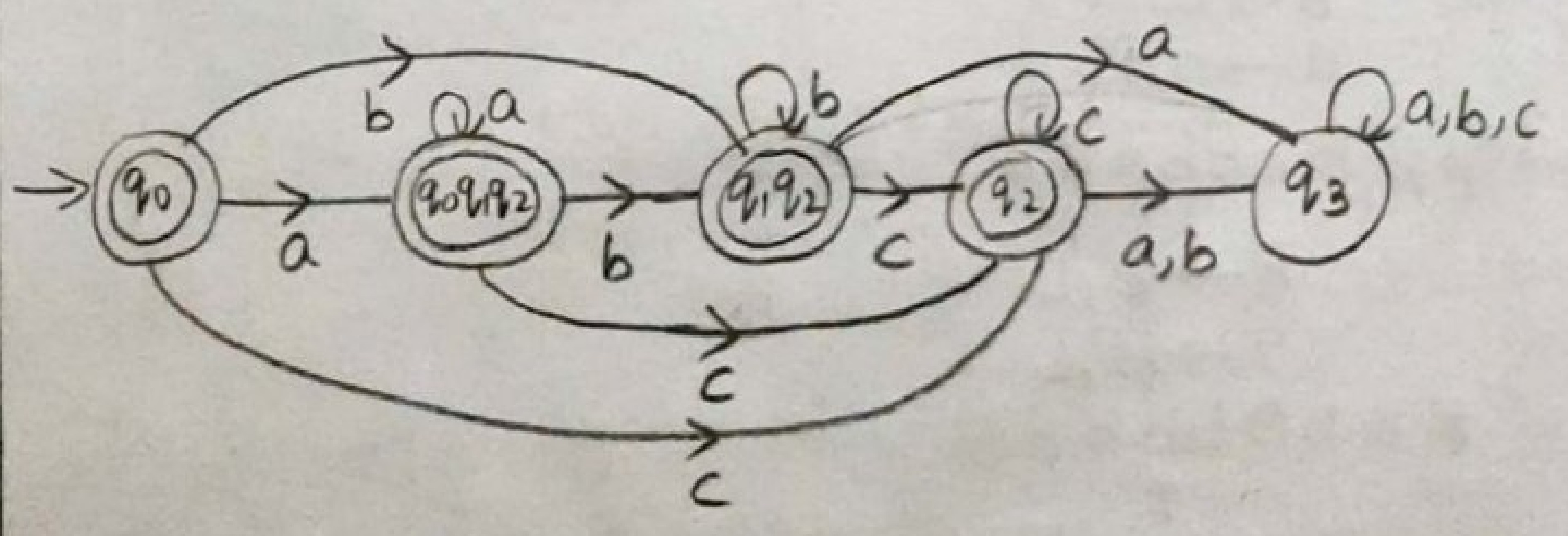
NFA:

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



DFA:

	a	b	c
$\rightarrow q_0^*$	$[q_0q_1q_2]$	$[q_1q_2]$	$[q_2]$
$q_0q_1q_2^*$	$[q_0q_1q_2]$	$[q_1q_2]$	$[q_2]$
$q_1q_2^*$	$[q_3]$	$[q_1q_2]$	$[q_2]$
q_2^*	$[q_3]$	$[q_3]$	$[q_2]$
q_3	$[q_3]$	$[q_3]$	$[q_3]$



20 Construct a minimal DFA which accepts the language over {a, b, c} where $L = \{a^n b^m c^l / n, m, l >= 0\}$

A:- $L = \{\epsilon, a, aa \ldots, b, bb, \ldots, c, cc, \ldots, abc, aabc, abbc, aaaabbc, \ldots\}$