Page No.	
Date	

Module-1

(Introduction of Finite Automata

L → Language
A → ~~Alphabet~~ Automata
G → Grammar.

Language ↓

- Symbol: {a, b, c, d, 0, 1, 2, 3, ... }
- Alphabet: Σ (a, b). (finite set of Symbol).
- String: (Collection of Alphabets)
  or
  (Sequence of Alphabets)

Eg a, ab, abb, ...
(Length of String 2)
{aa, ab, ba, bb}.

$$\boxed{\Sigma^0 = \varepsilon}$$

- Language ⇒ Collection of Strings.

Eg: Σ (a, b)
(i) L_1 = strings of length 3.

Ans: {aaa, aab, aba, abb, baa, bab, bba, bbb}.

(1) Automata (Model, Machine).

Infinite Languge

Σ = {a, b}.
L_1 = length 2.
} Finite Languge.

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

$\Sigma(a, b)$

L_2 = atleast one A = $\{a, aa, aaa, aaaa, \ldots$
$ab, abab, \ldots$
$ba, baba, \ldots$

= $\{abba\}$.

(2) Power of Σ $\quad \Sigma = \{a, b\}$.

Σ^0 = set of all strings with length '0' $= \epsilon, \lambda$ (2^0)
Σ^1 = " " '1' a, b (2^1)
Σ^2 = " " '2' aa, ab, ba, bb (2^2)
$\vdots$ $\quad 2^3$
$\vdots$

Σ^* (Kleen Closure) = set of all strings of all length possible.

$\boxed{\text{Infinite Language}}$

Also
Σ^+ (Positive closure).

$$\boxed{\begin{aligned} \Sigma^* &= \Sigma^+ + \Sigma^0 \\ \Sigma^* - \epsilon &= \Sigma^+ \end{aligned}}$$

$\Sigma^0 = \epsilon$

Cardinality :- number of elements in a set.

$\Sigma^n = 2^n$

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots$

Date

Finite State Machine

↓

Finite Automata

- FA with output
  - Moore Machine
  - Mealy Machine.
- FA without output
  - DFA (limited memory
  - NFA
  - ϵ-NFA

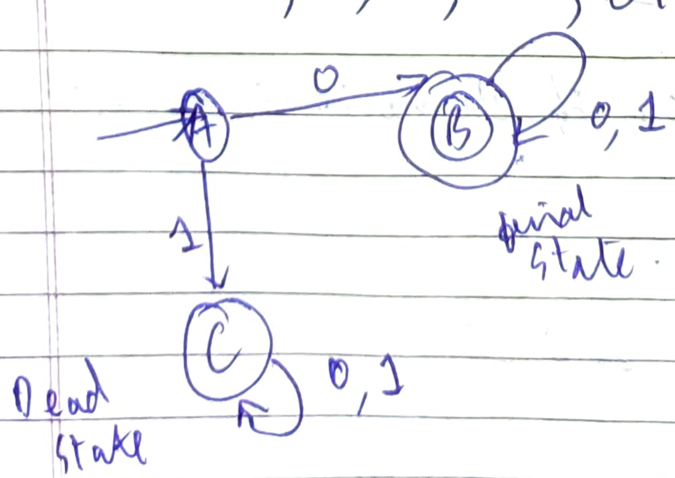
$(Q, \Sigma, q_0, F, \delta)$

Q = set of all states.
Σ = inputs.
q_0 = start state / initial state
F = set of final states
δ = transition func^n from $Q \times \Sigma \rightarrow Q$.

1.2 DFA (Examples)

(1) L_1 = { set of all strings that start with '0' }

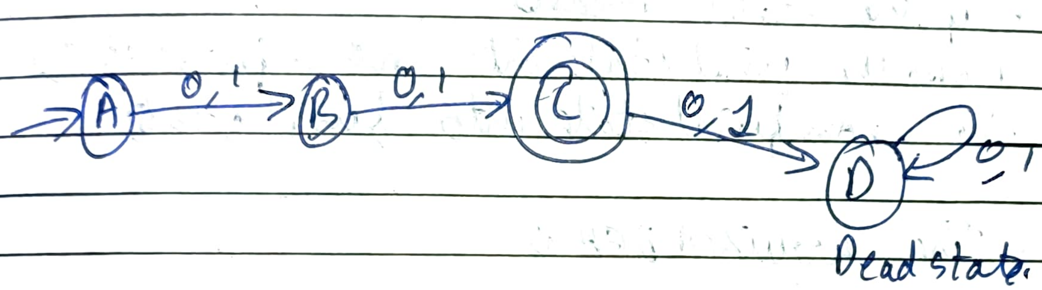
= { 0, 00, 000, 001, 010, ... }.



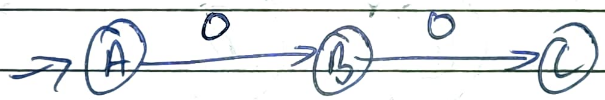
(2) L = { Set of all strings over $\Sigma(0,1)$ of length 2 }.

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

$L = \{00, 01, 10, 11\}$.

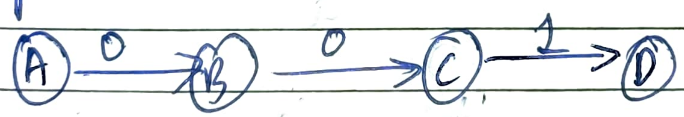


✓ eg: 0 0



Final state.

× eg: 001



Not final state.

(3) L = { Set of all strings that does not contain the string aabb in it.

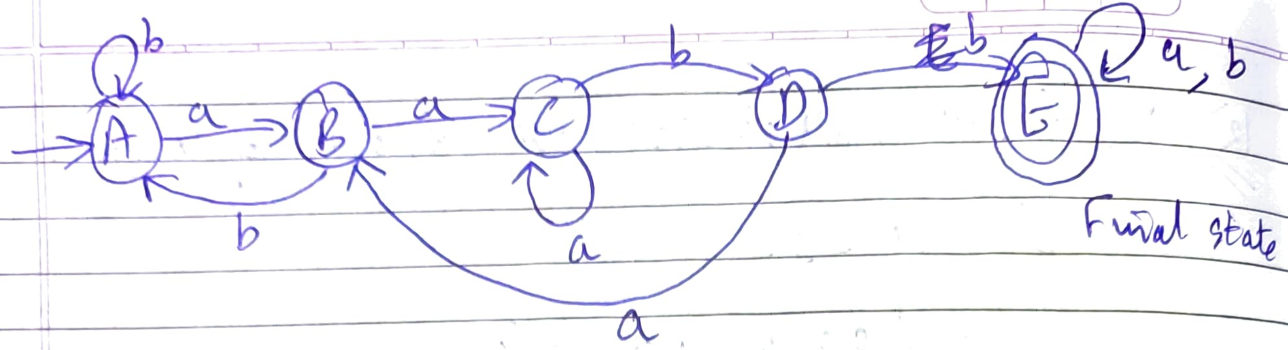
$\Sigma = \{a, b\}$

Make a design which is simpler

(i) First make DFA that contains string aabb in it.

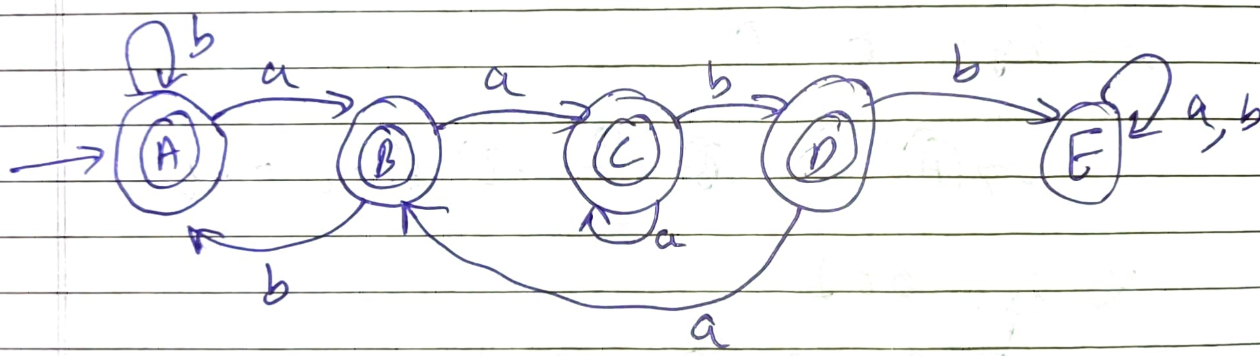
uaba

Page No.
Date

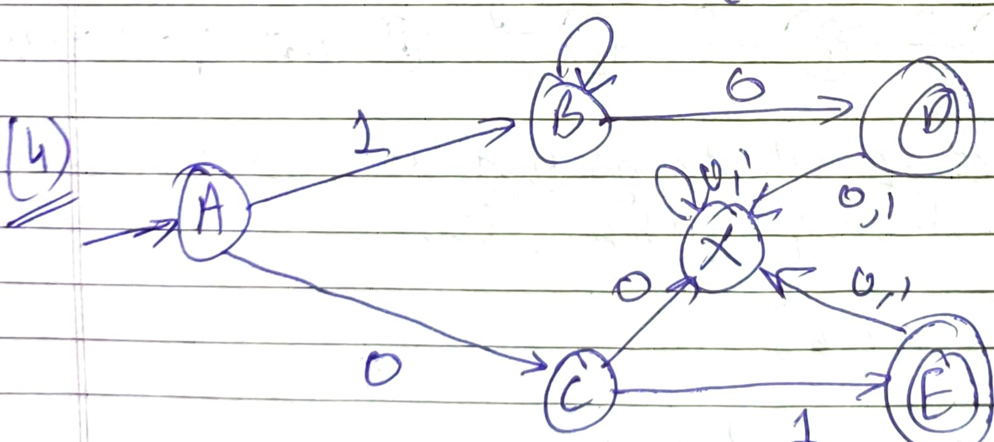


(iii) - Flip the states means
- Make the final state into non-final states & make the non final state into final state.

∴ The required DFA is,



4

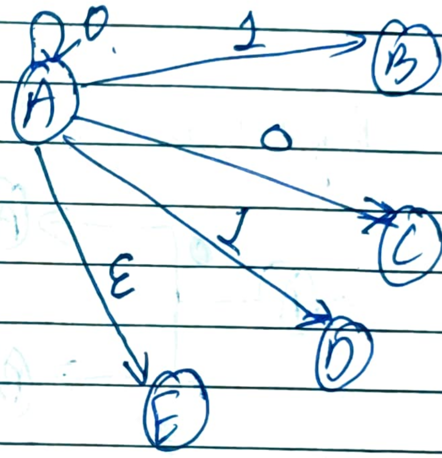


L = {Accepts the string 01 or a string of atleast one '1' followed by a '0'}

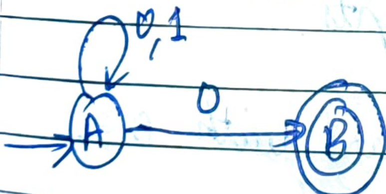
Eg: 001, 010, 011, 1101, 1100

1.3 Non-Deterministic Finite Automata (NFA)

~~Det~~ DFA	NFA
• simple & easy to design • Given current state we know what the next state will be.	• There could be multiple ne[xt] states. • All next states may be chos[en] in parallel.



Formal definition (NFA)



L = { set of all strings that end with 0 }.

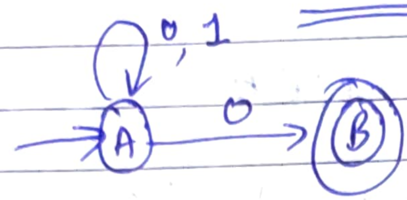
(Q, Σ, q_0, F, δ)

Q = set of all states - { A, B }.
Σ = set of inputs - { 0, 1 }.
q_0 = initial state - A
F = Set of Final states - B
δ = $Q \times \Sigma \to 2^Q$ (Transition function)

$A \times 0 \to A$
$A \times 0 \to B$
~~B~~ $A \times 1 \to A$
$B \times 1 \to \phi$
$B \times 0 \to \phi$

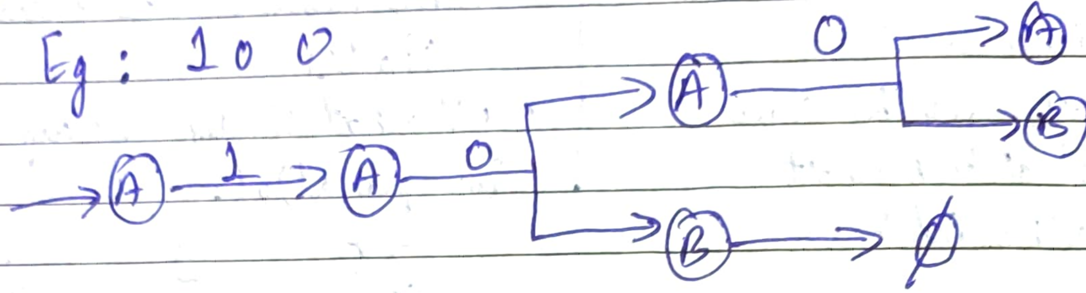
$A \to A, B, AB, \phi$

Examples

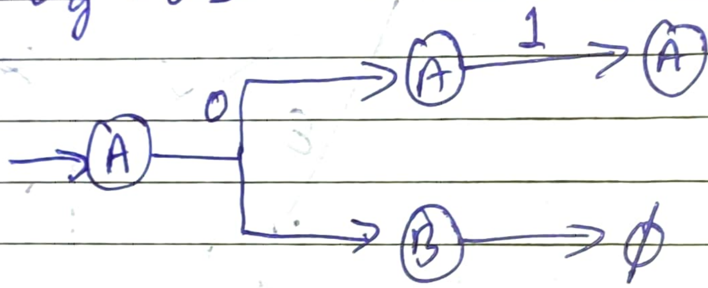
1) 

L = { set of all strings that end with 0 }

Eg : 1 0 0



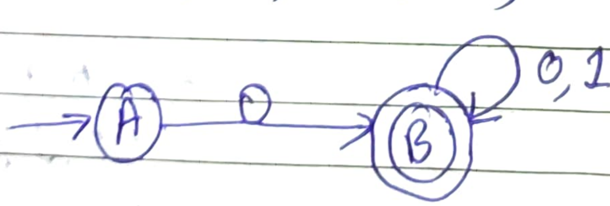
Eg : 0 1



If there is any way to run machine that ends in any set of states out of which atleast one state is final state, then the NFA accepts.

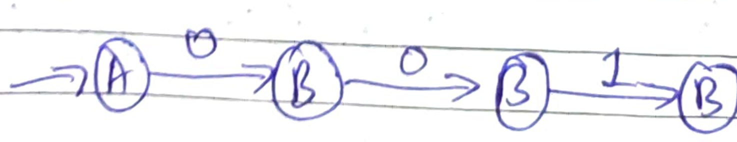
2) L = { set of all strings that start with 0 }.

= { 0, 00, 000, 01, 001, ...



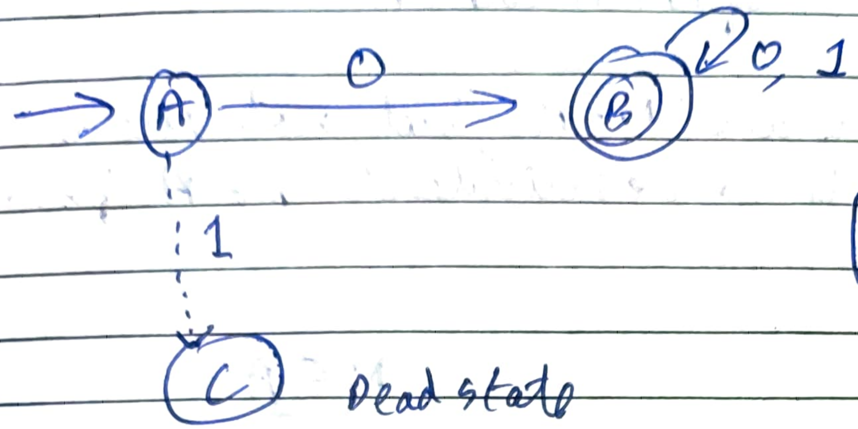
[NFA].

Eg : 0 0 1.



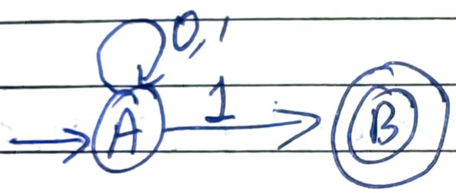
Eg : 1 0 1.

$\rightarrow A \xrightarrow{1} \phi$ (Dead configuration).

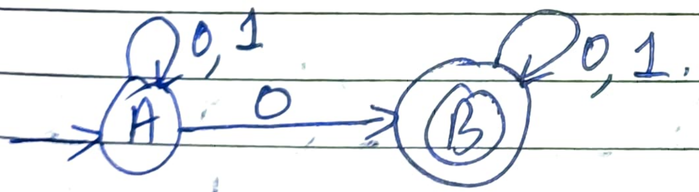


[DFA]

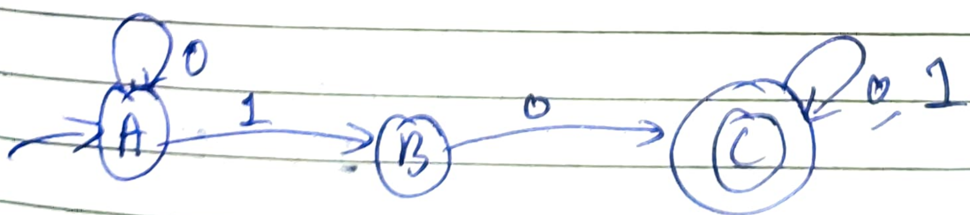
3) L_1 = { set of all strings that ends with '1' }.



L_2 = { set of all strings that contain '0' }.



L_3 = { Set of all strings that start with '1 0' }.



1.4 Equivalence of DFAs and NFAs, DFA Minimization

Conversion of NFA to DFA

Every DFA is an NFA, but not vice versa. But there is an equivalent DFA for every NFA.

DFA

Transition function $\delta = Q \times \Sigma \rightarrow Q$

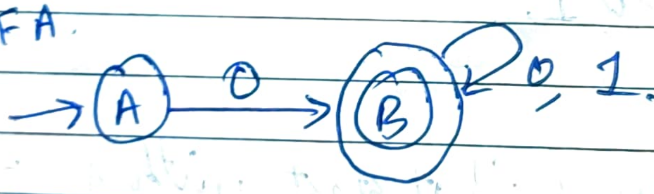
NFA

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

Eg: L = {set of all strings over (0, 1) that starts with '0'}.

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

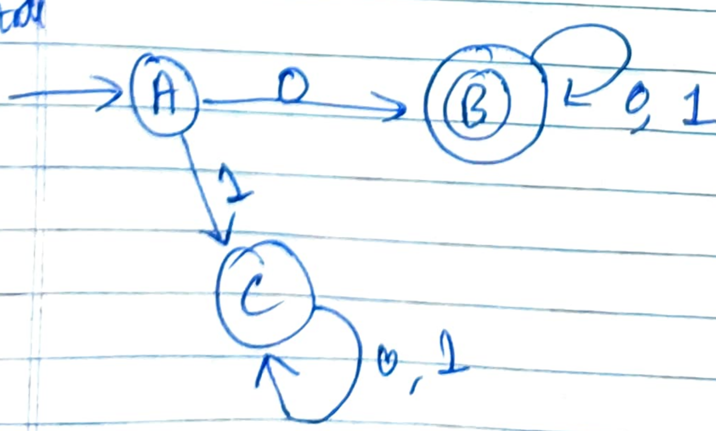
NFA



$Q \backslash \Sigma$	0	1
A	B	$\emptyset$
B	B	B

DFA

→ Dead state or trap state



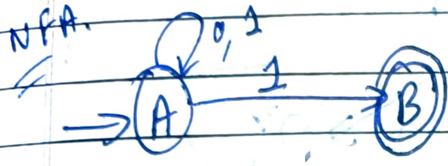
$Q \backslash \Sigma$	0	1
A	B	C
B	B	B
C	C	C

Examples

1) L = { set of all strings over (0, 1) that ends with '1' }.

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

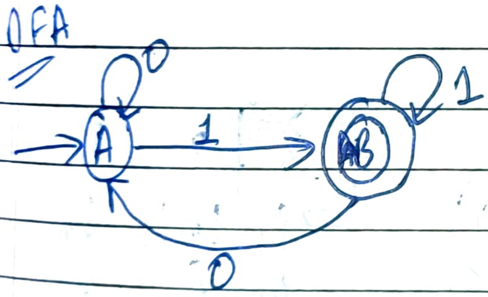
NFA.



Q \ Σ	0	1
A	{A}	{A, B}
B	ϕ	ϕ.

Method used Subset construction method

DFA



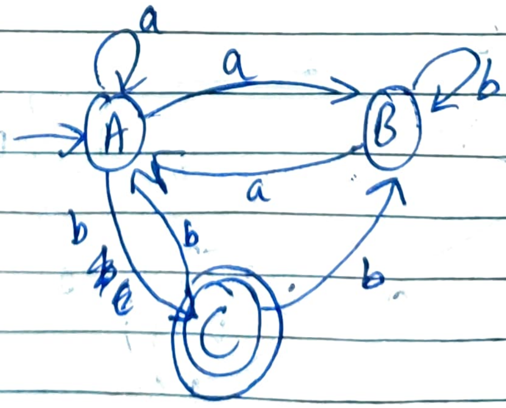
Q \ Σ	0	1
A	{A}	{A, B}.
AB	{A}	{AB}.

AB - single state.

2) Find equivalent DFA for NFA by $M = [\{A, B, C\}, (a, b), \delta, A, \{C\}]$.

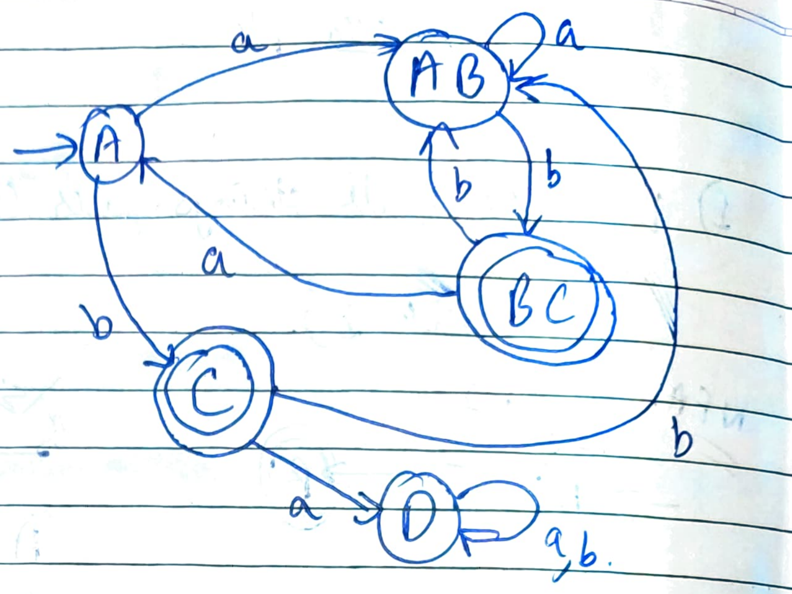
where δ is

Q \ Σ	a	b
→ A	A, B	C
B	A	B
(C)	–	A, B



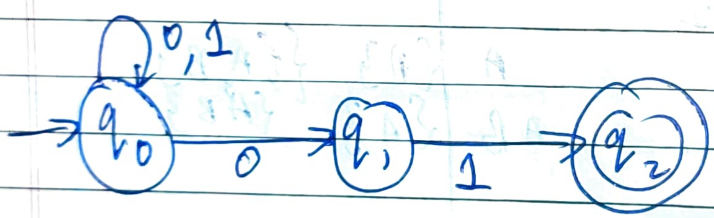
In DFA ~~both~~ States containing C is final states.

$Q \backslash \Sigma$	a	b
→ A	AB	C
AB	AB	BC
BC	A	AB
C	D	AB
D	D	D



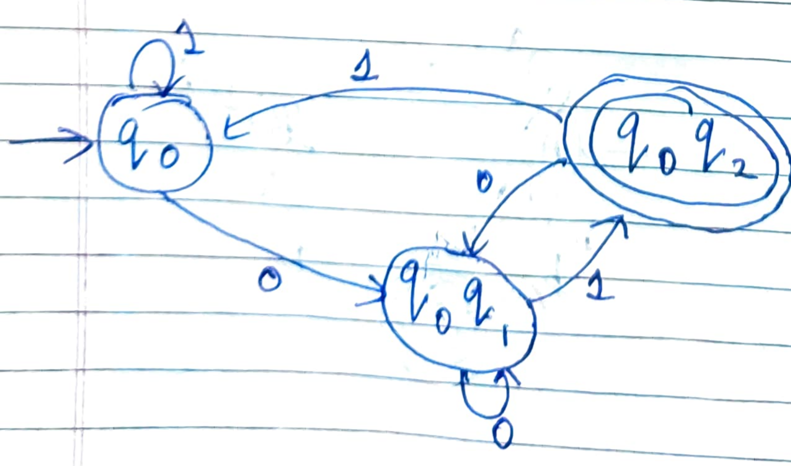
3) Given below NFA language.

L = {set of all strings over (0, 1) that ends with '01'}.



$Q \backslash \Sigma$	0	1
→ q_0	q_0, q_1	q_0
q_1	ϕ	q_2
q_2	ϕ	ϕ

DFA



$Q \backslash \Sigma$	0	1
→ q_0	$q_0 q_1$	q_0
$q_0 q_1$	$q_0 q_1$	$q_0 q_2$
$q_0 q_2$	$q_0 q_1$	q_0

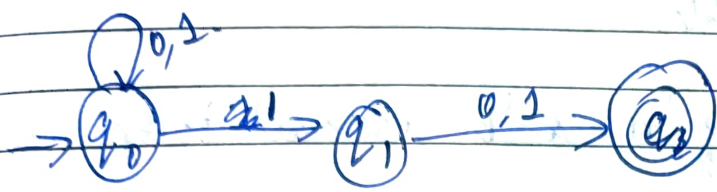
1101 ✓

1110 X

10101000 X

Design NFA.

4) $L = \{$ set of all strings over $\{0, 1\}$ in which second last symbol is always '1' $\}$.



✓ 10<u>1</u>0

✓ 11100<u>1</u>0

✓ 11<u>1</u>0

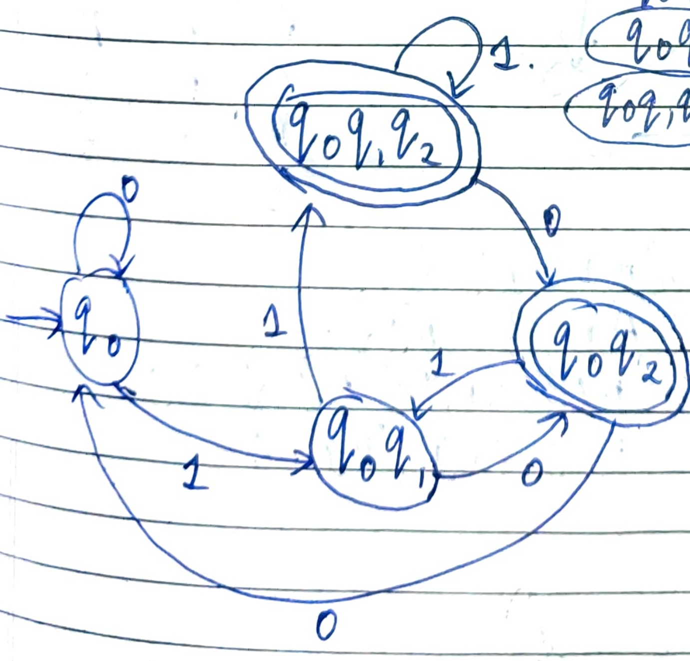
✓ 00<u>1</u>0.

✗ 1101

$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_0	q_0, q_1
q_1	q_2	q_2
q_2	ϕ	ϕ.

DFA

$Q \backslash \Sigma$	0	1
$\rightarrow q_0$	q_0	$q_0 q_1$
$q_0 q_1$	$q_0 q_2$	$q_0 q_1 q_2$
$q_0 q_2$	q_0	$q_0 q_1$
$q_0 q_1 q_2$	$q_0 q_2$.	$q_0 q_1 q_2$



DFA Minimization

Minimization of DFA is required to obtain the minimal version of any DFA which consists of min number of states possible.

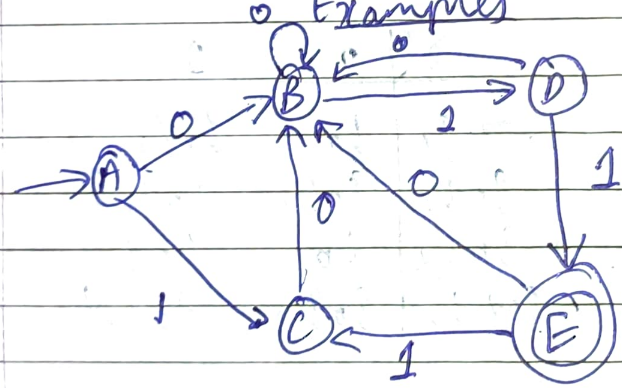
Two state ~~B~~ ~~q~~ 'A' & 'B' are said to be equivalent if

$\delta(A, x) \rightarrow F$ and $\delta(B, x) \rightarrow F$ OR $\delta(A, x) \not\rightarrow F$ and $\delta(B, x) \not\rightarrow F$

where 'x' is any input string

Examples

1)



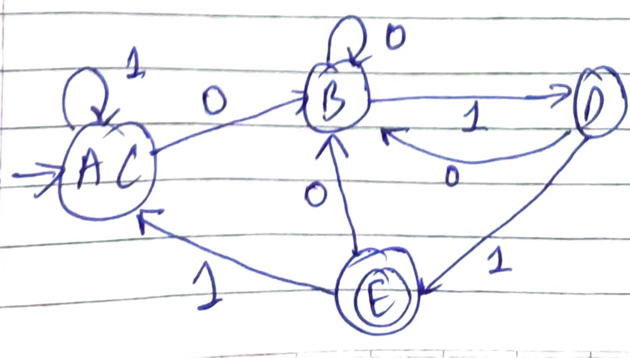
Q \ Σ	0	1
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C

0 Equivalence : {A,B,C,D} {E}

1 " " : {A B C} {D} {E}

2 " " : {A,C} {B} {D} {E}

3 " " : {A,C} {B} {D} {E}



Minimal version of DFA (More efficient).

Page No.	
Date	

2) Construct min DFA equivalent to DFA described.

	0	1
$\rightarrow q_0$	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_7	q_6
q_6	q_6	q_4
q_7	q_6	q_2

0 "Equivalence" $\{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}$ $\{q_2\}$

~~1 "~~
~~2 "~~
~~3 "~~

~~1 Equivalence.~~
~~$\{q_0, q_4\}$~~
~~$\{q_1, q_2\}$~~
~~$\{q_3\}$~~
~~$\{$~~

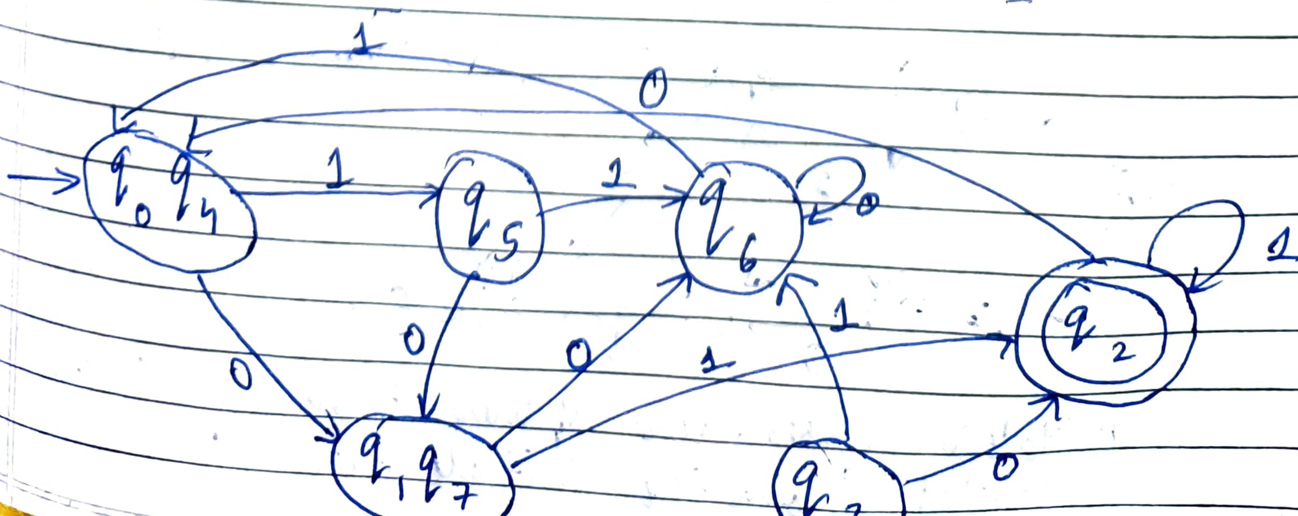
1 equivalence.
$\{q_0, q_4, q_5, q_6\}$ $\{q_2\}$.
$\{q_1, q_7\}$.
$\{q_3\}$
$\{q_7$

2 equivalence.

$\{q_0, q_4, q_5\}$ $\{q_6\}$ $\{q_2\}$ $\{q_1, q_7\}$ $\{q_3\}$.

3 equivalence.
$\{q_0, q_4\}$ $\{q_5\}$ $\{q_6\}$ $\{q_2\}$ $\{q_1, q_7\}$ $\{q_3\}$.

Q \ Σ	0	1
$\rightarrow q_0q_4$	q_1q_7	q_5
q_1q_7	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_5	q_7	q_6
q_6	q_6	q_4



Page No.
Date

When there are more than one final states involved

	0	1
→A	B	C
B	A	D
(C)	E	F
(D)	E	F
(E)	E	F
F	F	F

0 Equivalence –
{A, B, F} {C, D, E}.

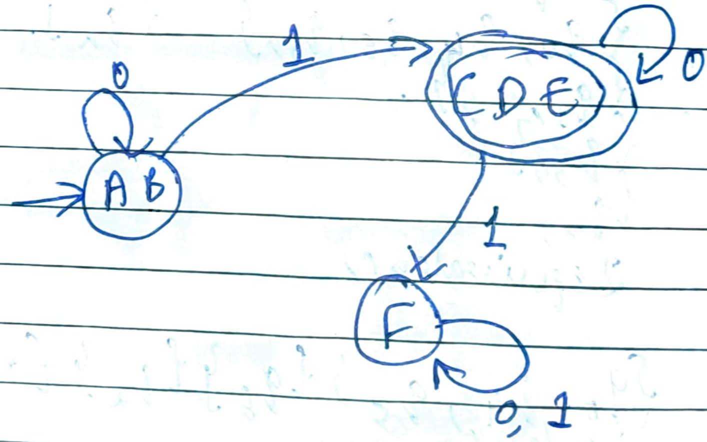
1 Equivalence.

{A, B} {F} {C, D, E}.

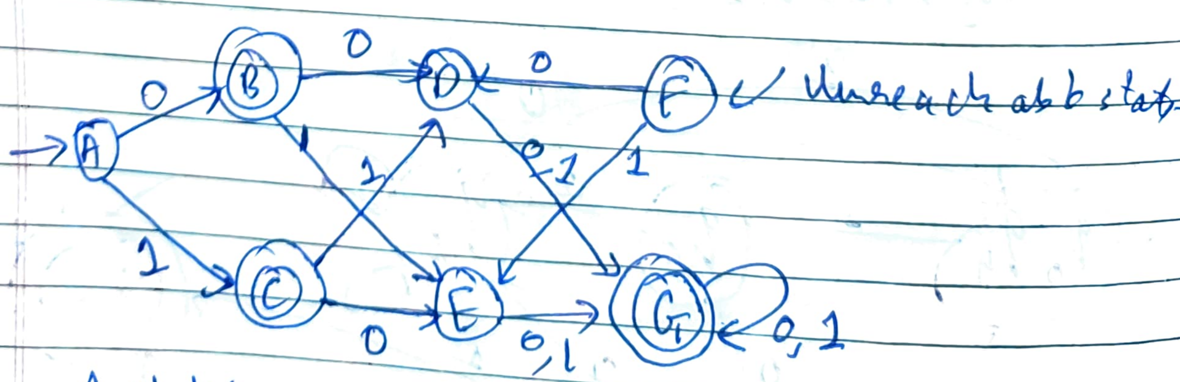
2 Equivalence.

{A, B} {F} {C, D, E}.

Σ	0	1
→{A, B}	{A, B}	{C, D, E}.
F	F	F
({C, D, E})	{C, D, E}	F



When there are unreachable states



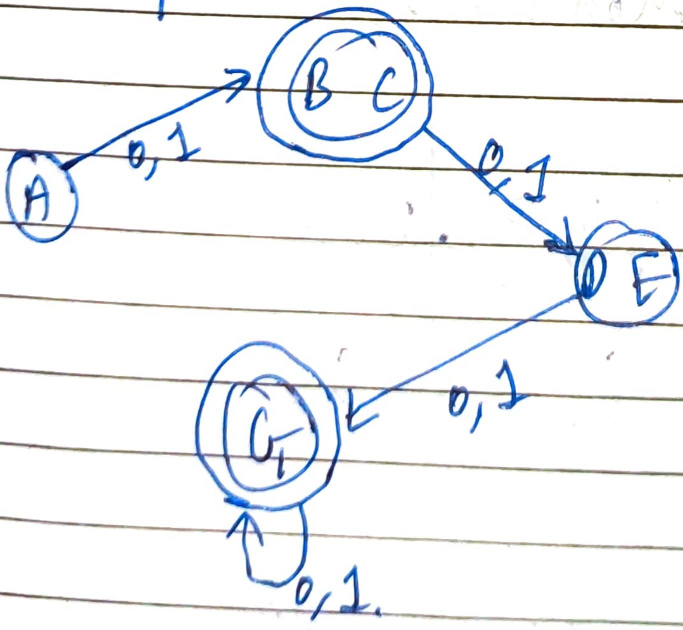
A state is unreachable if there is no way it can be reached from the initial state.

Page No.
Date

	0	1
→A	B	C
B	D	E
C	E	D
D	G	G
E	G	G
~~F~~	~~D~~	~~E~~
G	G	G.

0 Equivalence : {A, D, E} {B, C, G}.
1 Equivalence : {A, D, E} {B, C} {G}.
2 Equivalence : {A} {D, E} {B, C} {G}.
3 Equivalence : {A} {D, E} {B, C} {G}.

Q \ Σ	0	1
→{A}	{B, C}	{B, C}
{D, E}	{G}	{G}.
{B, C}	{D, E}	{D, E}.
{G}	{G}	{G}



Page No. Date

Examples

Design a FSM to check if the given numbers (decimal) is ~~of~~ divisible by 3

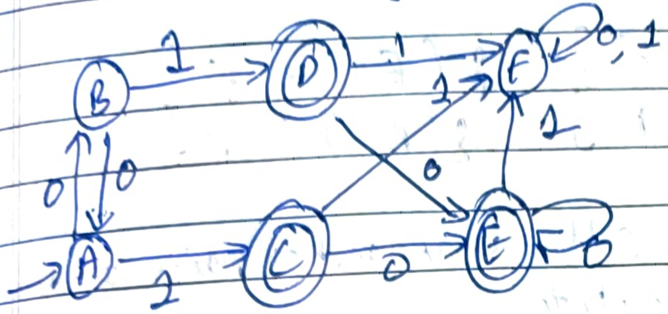
Ans: Q = set of all states = $\{q_0, q_1, q_2\}$.
Σ = set of input states = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
δ = Transition function
q_0 = Start state
F = Final states = $\{q_0\}$.

q_0 – $n \bmod 3 = 0$
q_1 – $n \bmod 3 = 1$
q_2 – $n \bmod 3 = 2$

$Q \backslash \Sigma$	0, 3, 6, 9	1, 4, 7	2, 5, 8
rem 0 $\rightarrow q_0$	q_0	q_1	q_2
rem 1 $\rightarrow q_1$	q_1	q_2	q_0
rem 2 $\rightarrow q_2$	q_2	q_0	q_1

DFA Minimization - Table Filling Method

(Myhill - Nerode theorem).



	A	B	C	D	E	F
A						
B						
C	✓	✓				
D	✓	✓				
E	✓					
F						

	A	B	C	D	E	F
A						
B						
C	✓	✓				
D	✓	✓				
E	✓	✓				
F	✓	✓	✓	✓	✓	

F → Final state.

- Mark all pairs where $P \in F$ and $Q \notin F$

- If there are unmarked pairs ~~and make~~ (P, Q) such that $[\delta(P, x), \delta(Q, x)]$ is marked, then mark [P, Q] where 'x' is an input symbol. Repeat this until no more markings can be made.

$(B, A) - \left.\begin{array}{l}\delta(B, 0) = A \\ \delta(A, 0) = B\end{array}\right\}\quad \left.\begin{array}{l}\delta(B, 1) = D \\ \delta(A, 1) = C\end{array}\right\}$

$(D, C) - \left.\begin{array}{l}\delta(D, 0) = E \\ \delta(C, 0) = E\end{array}\right\}\quad \left.\begin{array}{l}\delta(D, 1) = F \\ \delta(C, 1) = F\end{array}\right\}$.

$(E, C) - \left.\begin{array}{l}\delta(E, 0) = E \\ \delta(C, 0) = E\end{array}\right\}\quad \left.\begin{array}{l}\delta(E, 1) = F \\ \delta(C, 1) = F\end{array}\right\}$.

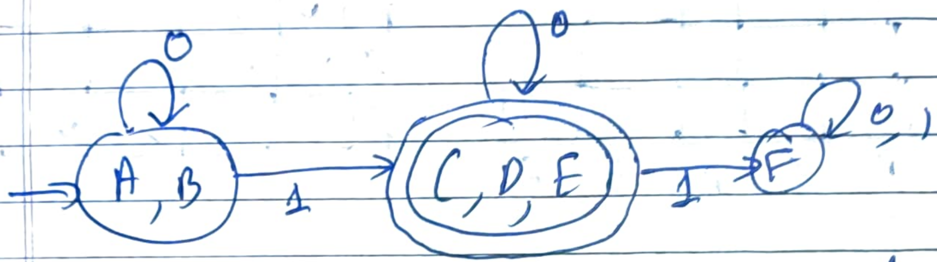
$(E, D) - \left.\begin{array}{l}\delta(E, 0) = E \\ \delta(D, 0) = E\end{array}\right\}\quad \left.\begin{array}{l}\delta(E, 1) = F \\ \delta(D, 1) = F\end{array}\right\}$

(mark it) (F,A) – δ(F,0) = F } δ(F,1) = F
δ(A,0) = B } δ(A,1) = C

(F,B) – δ(F,0) = F } δ(F,1) = F }
δ(B,0) = A } δ(B,1) = D }

- Finally combine all unmarked pairs and make them single state.

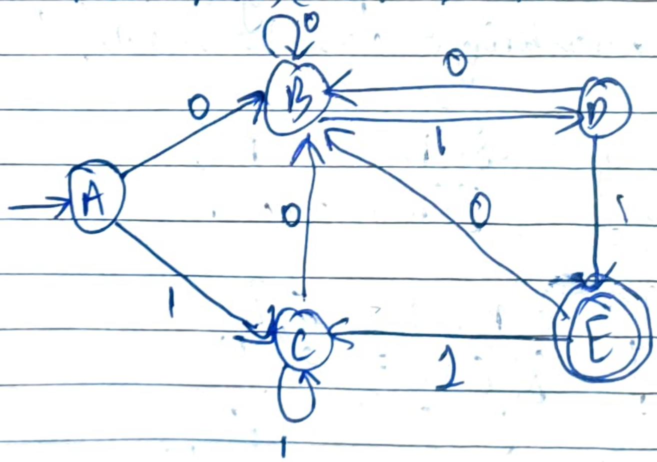
(A,B) (C,D) (E,C) (E,D)



Minimised DFA (Only 3 states)

Example.

1) Minimise the DFA.



	A	B	C	D	E
A					
B	✓				
C		✓			
D	✓	✓	✓		
E	✓	✓	✓	✓	

$(B, A) - \left.\begin{array}{l}\delta(B, 0) = B \\ \delta(A, 0) = \cancel{A}B\end{array}\right\} \quad \left.\begin{array}{l}\delta(B, 1) = D \\ \delta(A, 1) = C\end{array}\right\}.$

$(C, A) - \left.\begin{array}{l}\delta(\cancel{B}C, 0) = B \\ \delta(A, 0) = B\end{array}\right\} \quad \left.\begin{array}{l}\delta(C, 1) = C \\ \delta(A, 1) = C\end{array}\right\}$

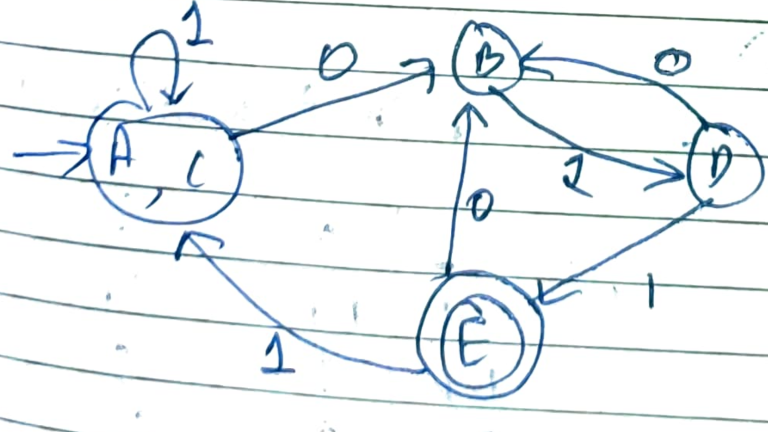
$(C, B) - \left.\begin{array}{l}\delta(C, 0) = B \\ \delta(B, 0) = B\end{array}\right\} \quad \left.\begin{array}{l}\delta(C, 1) = C \\ \delta(B, 1) = D\end{array}\right\}.$

Mark it $(D, A) - \left.\begin{array}{l}\delta(D, 0) = B \\ \delta(\cancel{D}A, 0) = B\end{array}\right\} \quad \left.\begin{array}{l}\delta(D, 1) = E \\ \delta(A, 1) = C\end{array}\right\}$

Mark it $(D, B) - \left.\begin{array}{l}\delta(D, 0) = B \\ \delta(B, 0) = B\end{array}\right\} \quad \left.\begin{array}{l}\delta(D, 1) = E \\ \delta(B, 1) = D\end{array}\right\}.$

Mark it $(D, C) - \left.\begin{array}{l}\delta(D, 0) = B \\ \delta(C, 0) = B\end{array}\right\} \quad \left.\begin{array}{l}\delta(D, 1) = \cancel{B}E \\ \delta(C, 1) = C\end{array}\right\}$

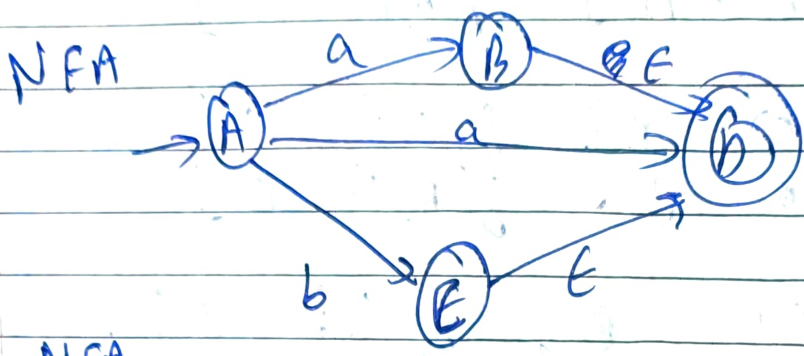
(A, C) B D E



Minimised DFA.

2) Convert the given into equivalent DFA and minimise the same.

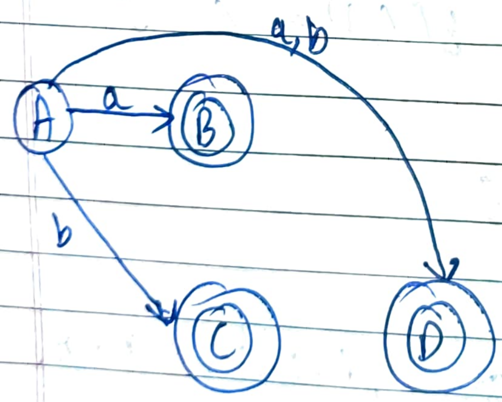
NFA



NFA

Q \ Σ	a	b
→ A	{B,D}	{E,D}
B	ϕ	ϕ
C	ϕ	ϕ
D	ϕ	ϕ

	ε*	a	ε*		ε*	b	ε*
A	A	B D	B,D D	A	A	E	E,D
B	B D	ϕ ϕ	ϕ ϕ	B	B D	ϕ ϕ	ϕ ϕ
C	C D	ϕ ϕ	ϕ ϕ	C	C D	ϕ ϕ	ϕ ϕ
D	D	ϕ	ϕ	D	ϕ	ϕ	ϕ



DFA transition table

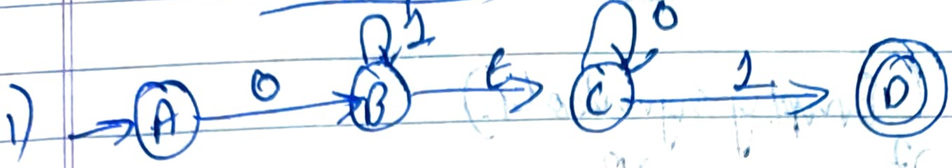
	a	b
→ A	B D	C D
B D	ϕ	ϕ
C D	ϕ	ϕ
ϕ	ϕ	ϕ

ϕ → qd.

PAGE NO.
DATE: / /

Examples

1)

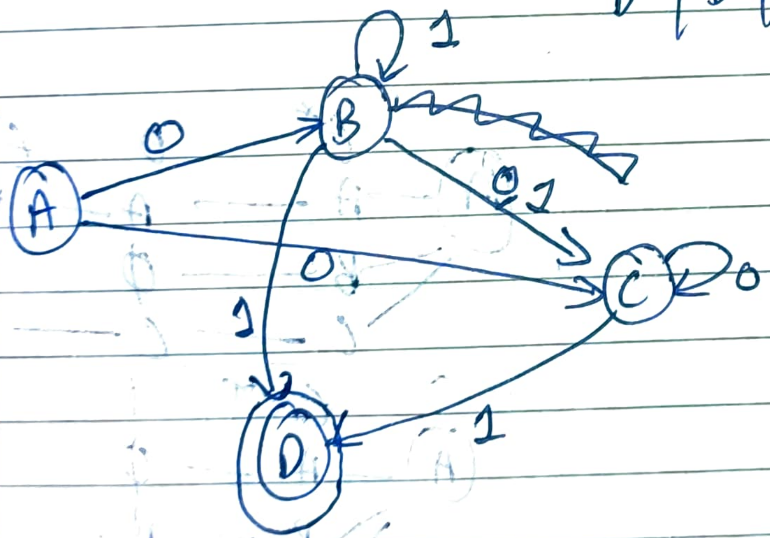


NFA

	0	1
→ A	{B, C}	ϕ
B	{C}	{B, C, D}
C	{C}	{D}
D	ϕ	ϕ

	ε*	0	ε*
A	A	B	B, C
B	B, C	ϕ, C	ϕ, C
C	C	C	C
D	D	ϕ	ϕ

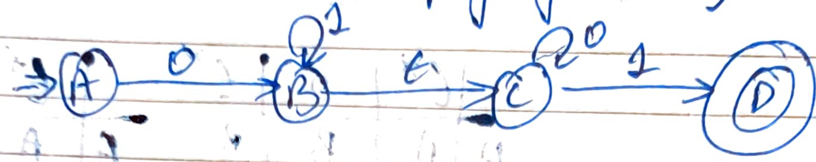
	ε*	1	ε*
A	A	ϕ	ϕ
B	B, C	B, D	B, C, D
C	C	D	D
D	D	ϕ	ϕ



NFA derived from ε – NFA.

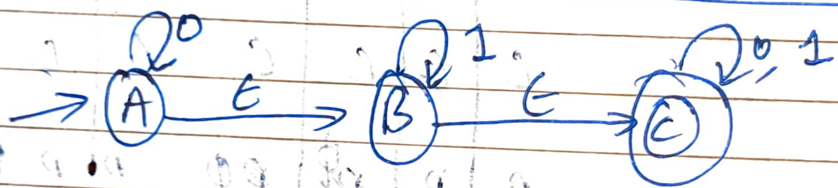
Epsilon NFA (ε)

ε-NFA (Empty symbol.)



Every state on epsilon ε goes to itself.

Conversion of Epsilon-NFA to NFA



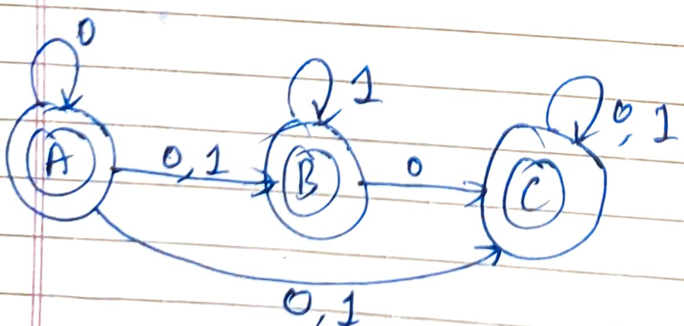
Transition of NFA

	0	1
→A	{A,B,C}.	{B,C}
B	{C}	{B,C}
C	{C}	{C}.

A: ε* — 0 — ε*
A — A — A, B, C
B — φ
C — C — C

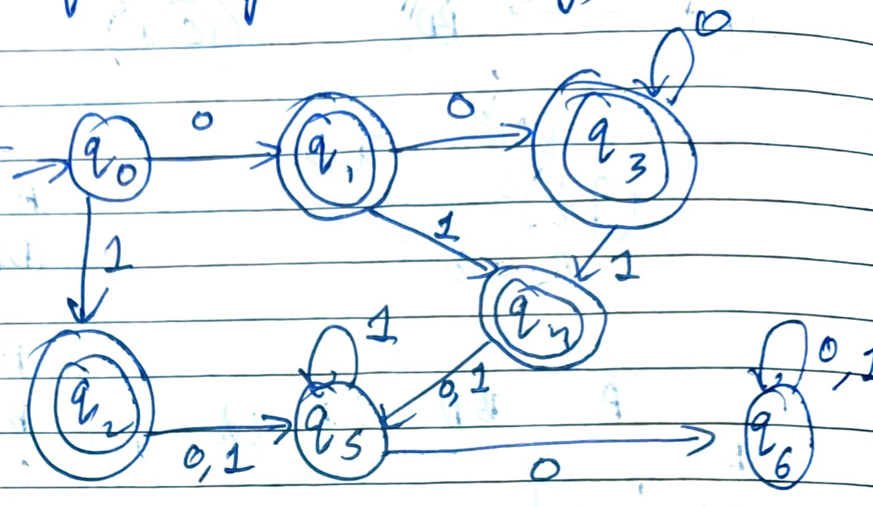
ε* — 1 — ε*
A — A — φ
B — B — B, C.
C — C — C

B — ε B — φ —
C — C — C.



Q. Minimise the following DFA (Using)

ii) ~~Table filler~~



i) Using Table filling method.

	q_0	q_1	q_2	q_3	q_4	q_5	q_6
q_0							
q_1	✓						
q_2	✓	✓					
q_3	✓	✓	✓				
q_4	✓	✓		✓			
q_5	✓	✓	✓	✓	✓		
q_6	✓	✓	✓	✓	✓		

$(q_1, q_2) - \left.\begin{matrix} \delta(q_1, 0) = q_3 \\ \delta(q_2, 0) = q_5 \end{matrix}\right\}$ $\left.\begin{matrix} \delta(q_1, 1) = q_4 \\ \delta(q_2, 1) = q_5 \end{matrix}\right\}$.

$(q_1, q_3) - \left.\begin{matrix} \delta(q_1, 0) = q_3 \\ \delta(q_3, 0) = q_3 \end{matrix}\right\}$ $\left.\begin{matrix} \delta(q_1, 1) = q_4 \\ \delta(q_3, 1) = q_4 \end{matrix}\right\}$.

$(q_3, q_2) - \left.\begin{matrix} \delta(q_3, 0) = q_3 \\ \delta(q_2, 0) = q_5 \end{matrix}\right\}$ $\left.\begin{matrix} \delta(q_3, 1) = q_4 \\ \delta(q_2, 1) = q_5 \end{matrix}\right\}$

(Mark it) $(q_4, q_1) - \left.\begin{matrix} \delta(q_4, 0) = q_5 \\ \delta(q_1, 0) = q_3 \end{matrix}\right\}$ ~~$\delta(q$~~ ~~$\delta($~~

Page No.
Date

$(q_4, q_2) \Rightarrow \delta(q_4, 0) = q_5 \}$ $\delta(q_4, 1) = q_5 \}$
$\delta(q_2, 0) = q_5 \}$ $\delta(q_2, 1) = q_5 \}$

(Mark it) $(q_5, q_0) \Rightarrow \delta(q_5, 0) = q_6 \}$
$\delta(q_0, 0) = q_1 \}$

Mark it $(q_6, q_0) \Rightarrow \delta(q_6, 0) = q_6 \}$
$\delta(q_0, 0) = q_1 \}$

$(q_6, q_5) \Rightarrow \delta(q_6, 0) = q_6 \}$ $\delta(q_6, 1) = q_6 \}$
$\delta(q_5, 0) = q_6 \}$ $\delta(q_5, 1) = q_5 \}$

$(q_4, q_3) \Rightarrow \delta(q_4, 0) = q_5$ $\delta(q_4, 1) = q_5$
$\delta(q_3, 0) = q_3$ $\delta(q_3, 1) = q_4$

Now combine states which are unmarked.

(q_1, q_2) ; (q_3, q_1); (q_3, q_2) ; $\cancel{(q_4, q_2)}$
; $\cancel{(q_3, q_4)}$

$\cancel{(q_6, q_5)}$

(q_0) (q_1, q_3) (q_2, q_4) (q_5, q_6)