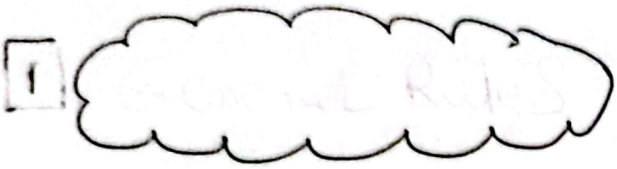
Finite Automata

- Way to describe {Reg. Exp}
- NFA
- DFA
  - Way to Recognize Token



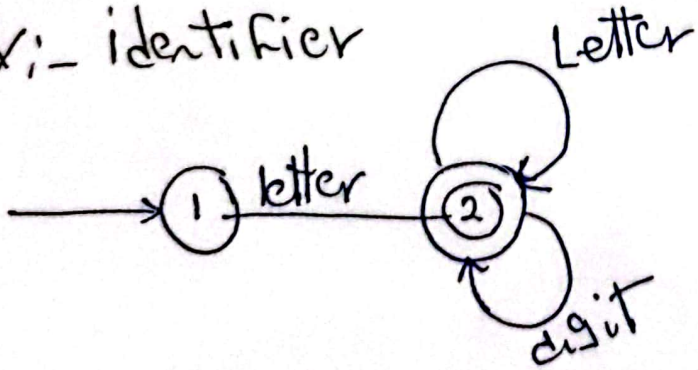
1. only one start state
2. one or more acceptance state
3. one or more transitions

→ start state ◯

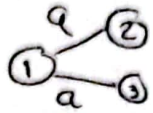
→ acceptance state ◎

Reg. Exp
↓
NFA
↓
DFA
↓
code

Ex:- identifier



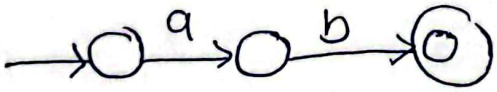
* NFA:- Non-deterministic finite Automata

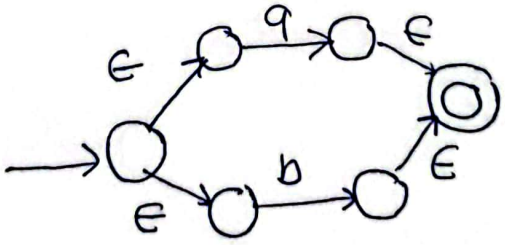
1. ∈ - Transition occur without consuming an character
2. The same input can make transition to different states 

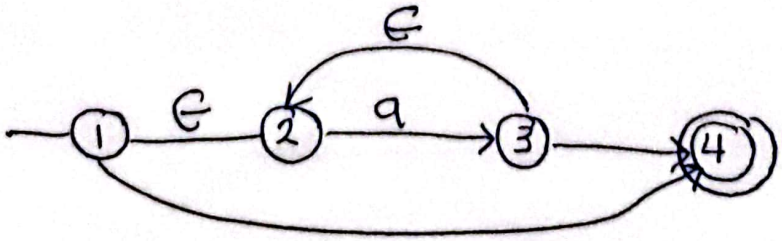
3. only two arrows can go out from one state and must be ∈ transition 

* Thompson's construction

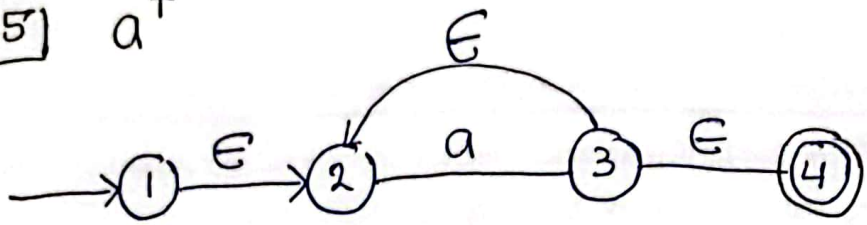
1. a 

2. ab 

3. a|b 

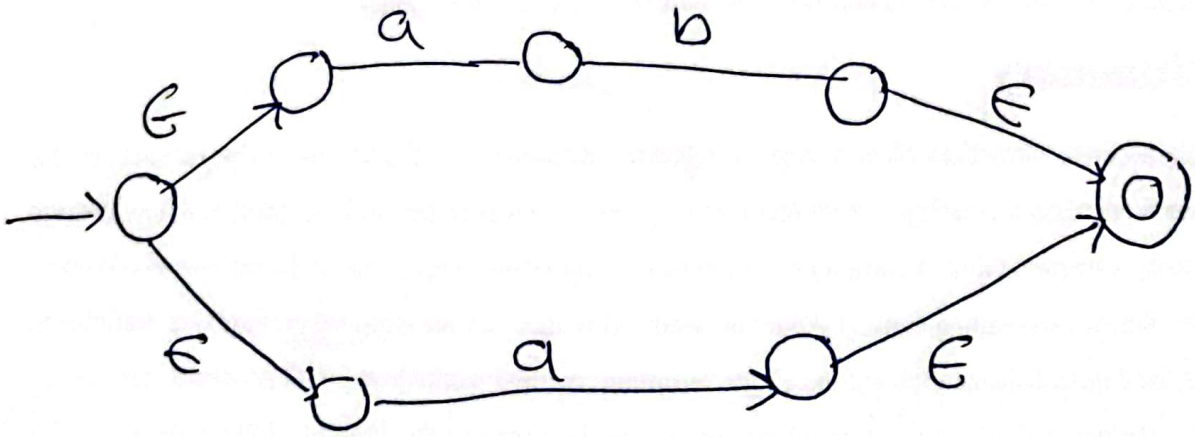
4. a* 

5 a^+

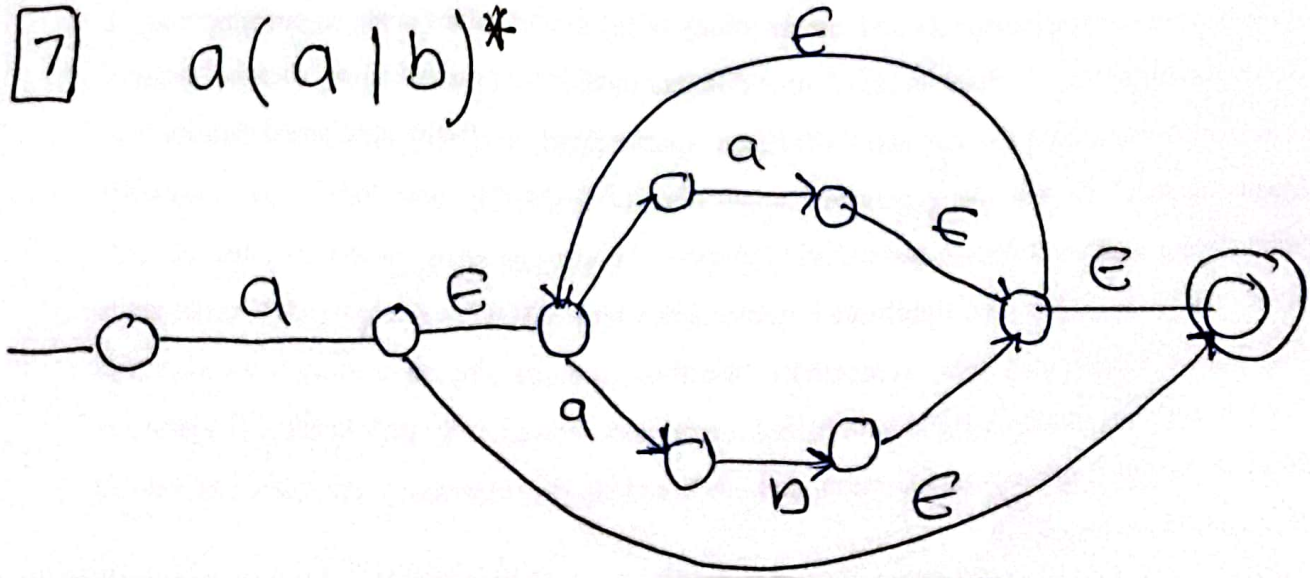


6 $a\underbrace{b}_{A}|\underbrace{a}_{B}$

Sol

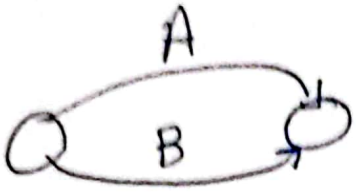


7 $a(a|b)^*$

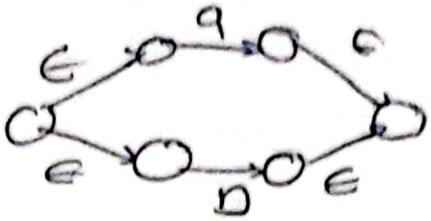


regular expression (NFA)

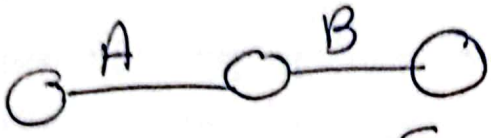
① A|B



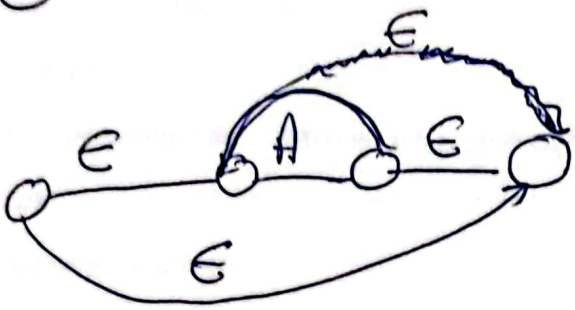
or



② AB



③ A*



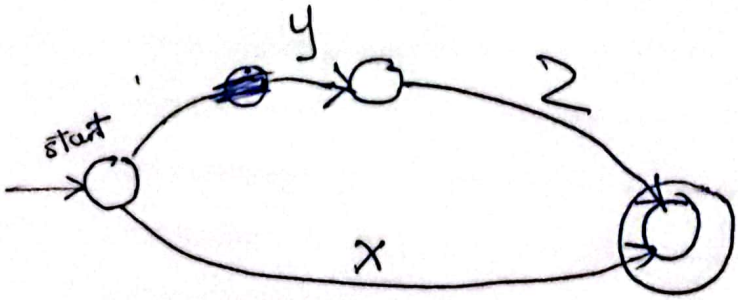
الأولويات

* ①

conc ②

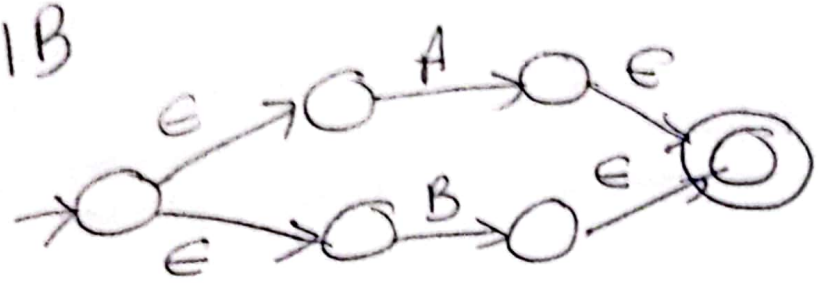
| ③

Ex:- X | (YZ)

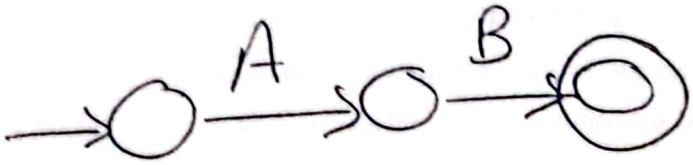


$r = (z \mid w \mid x)^* \longrightarrow A^*$

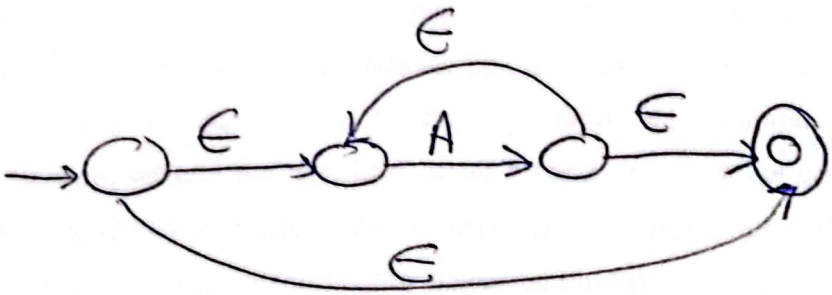
① $A \mid B$



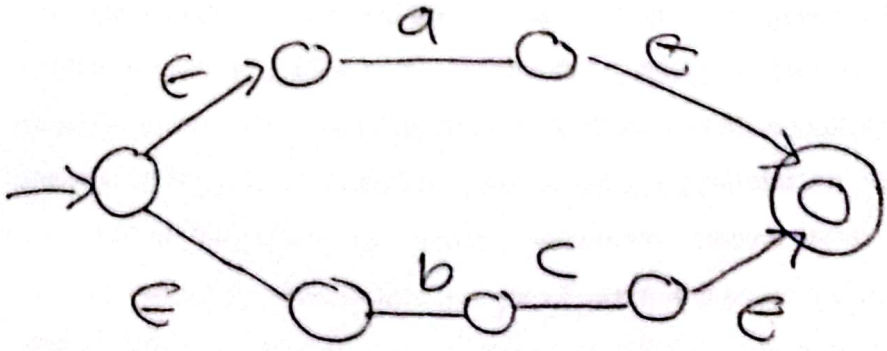
② AB



③ A^*

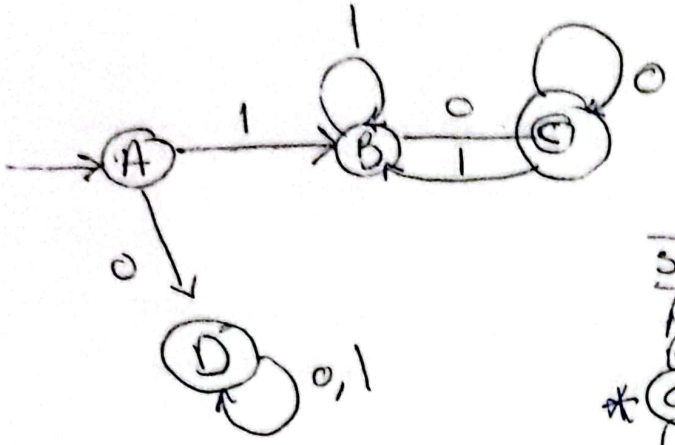


Ex:- $(a \mid bc) \longrightarrow A \mid B$



(1) Start with one and end zero

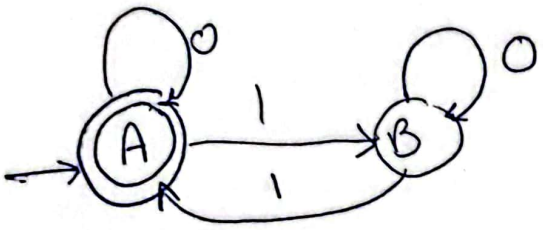
10, 100, 110, 1110, 1010 ----



S	0	1
A	D	B
B	C	B
* C	C	B
D	D	D

(2) even no of ones

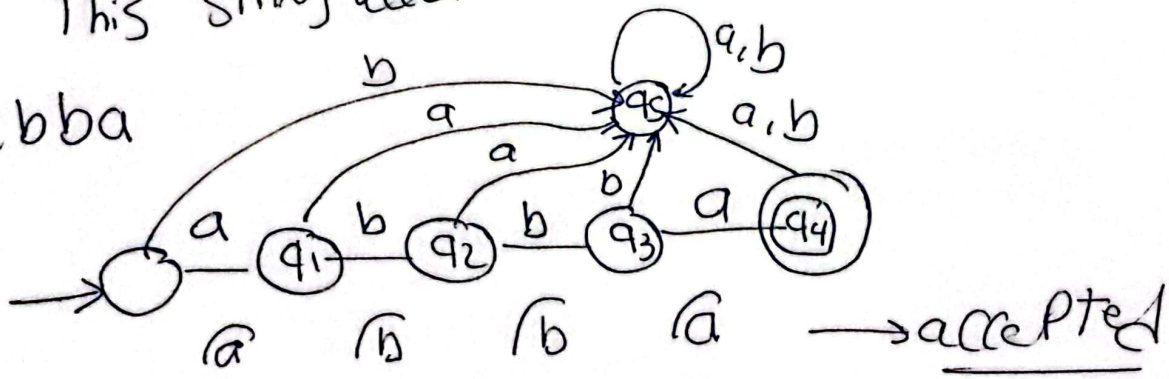
0, 11, 101, 011, 0101 ----



S	0	1
* A	A	B
B	B	A

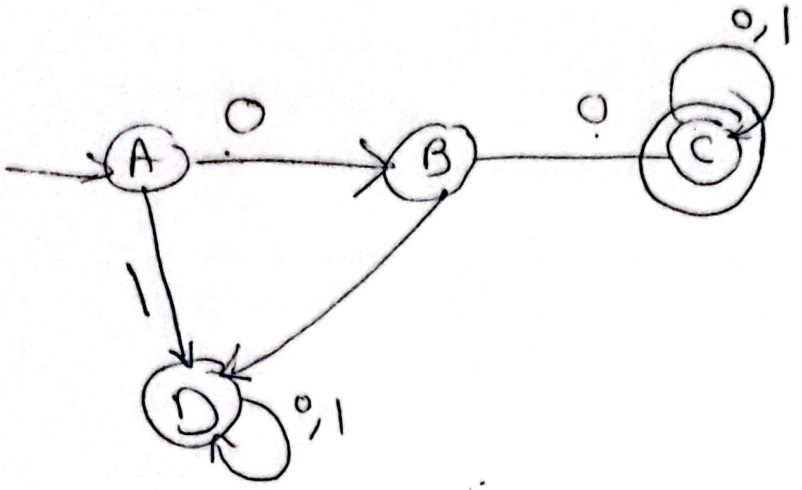
Ex:- This string accepted or Reject

abba



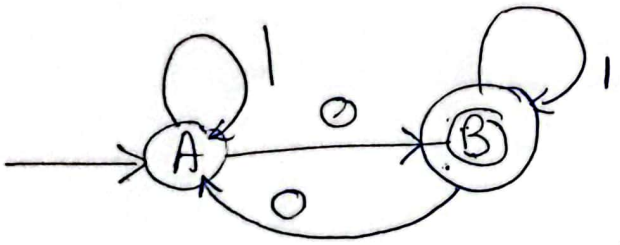
→ accepted

(1) 00 (011)*



(2) odd no of Zero's

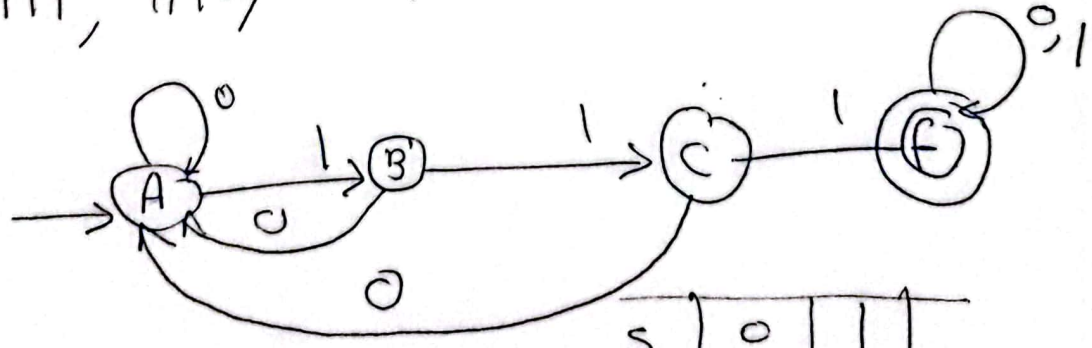
0, 000, 0100, 01, 10, 110, ...



S	0	1
A	B	A
*B	A	B

(3) Three consective ones

111, 1110, 0111 ---

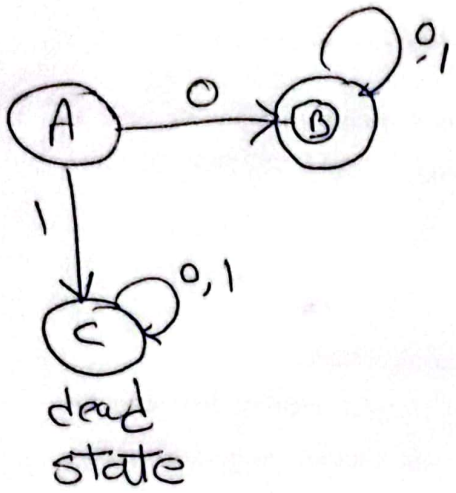


S	0	1
A	A	B
B	C	A
C	F	A
*F	F	f

Ex:- DFA

L_1 : set of all strings that start with 0.

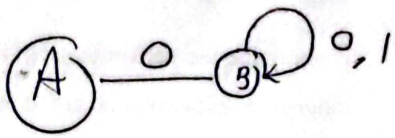
$= \{0, 00, 01, 000, 010, 011, 0000, \ldots\}$

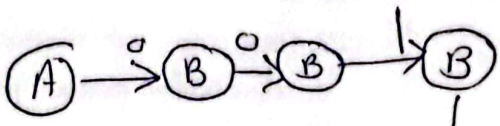


check

e.g 001

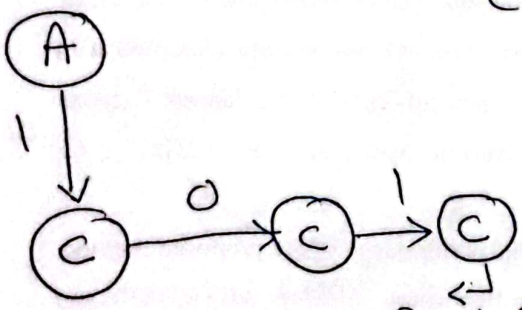
initial state





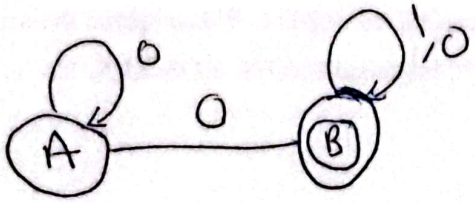
(accepted) (final) end state

Eg 101



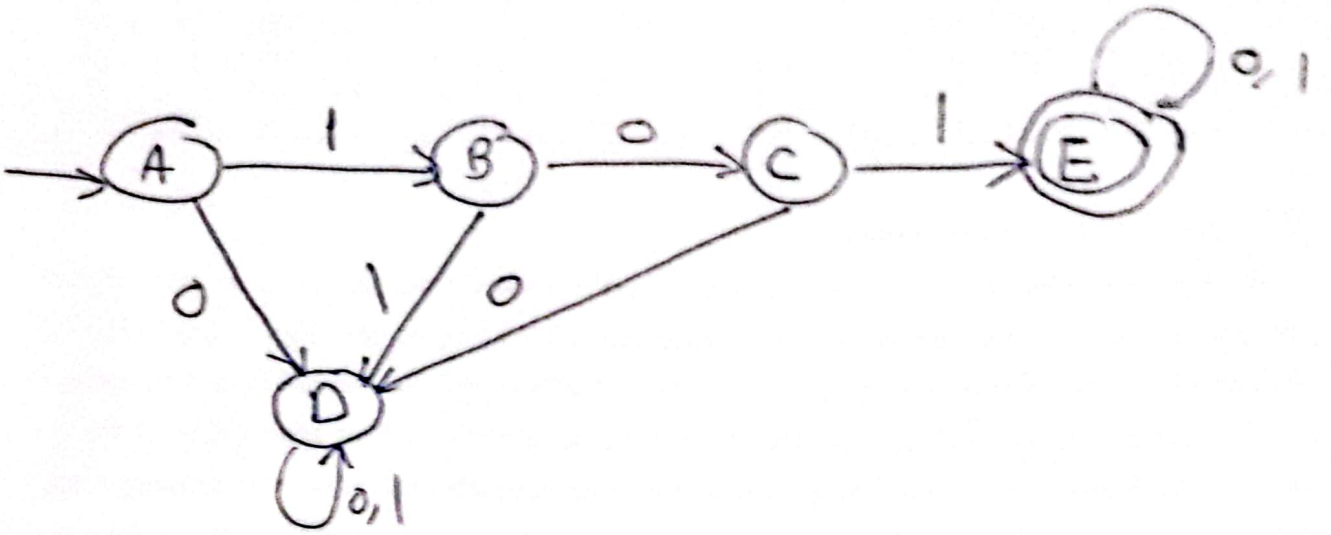
dead state (reject)

NFA



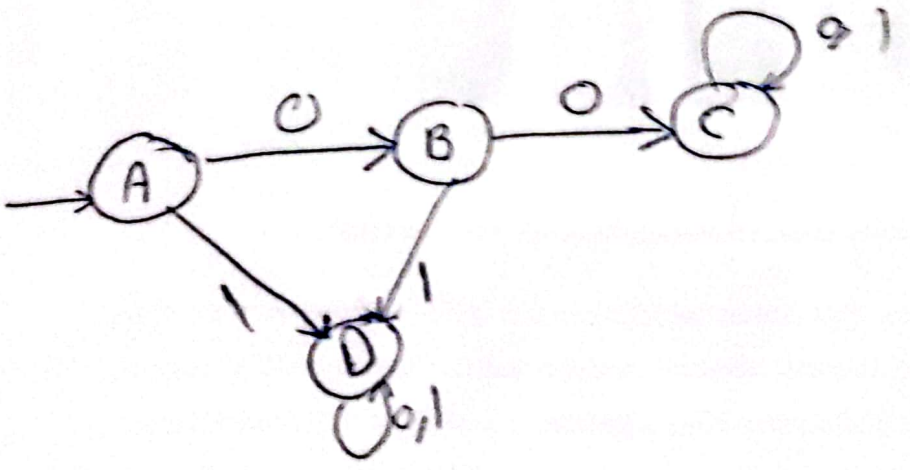
Ex:- $101(0|1)^*$

$\{101, 1010, 1011, 10101, \cdots\}$



Ex: $00(0+1)^*$

$\{00, 000, 001, 0001, \cdots\}$

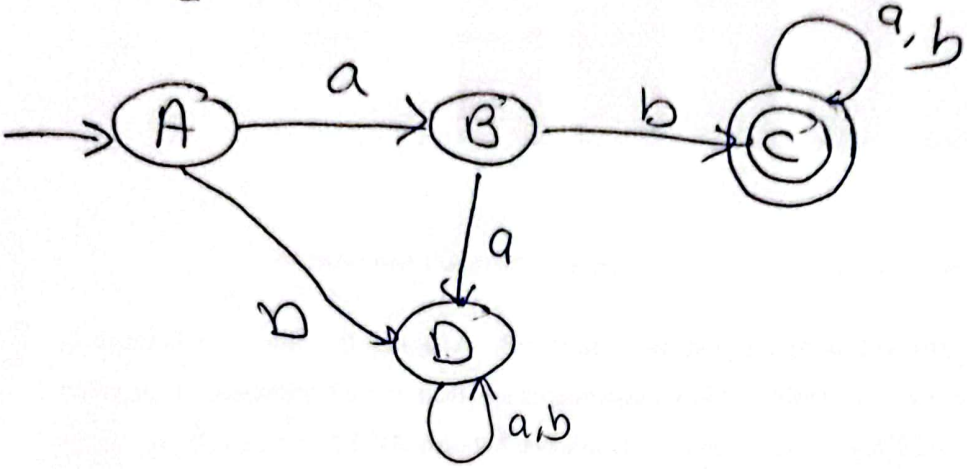


Ex:- Give regular expression

$ab(a|b)^*$

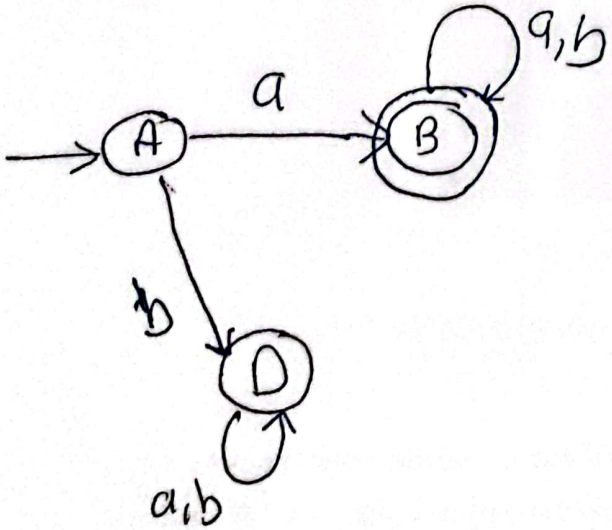
Construct DFA

{ ab, aba, abb, abaa, abab ... }



Ex:- $a(a|b)^*$

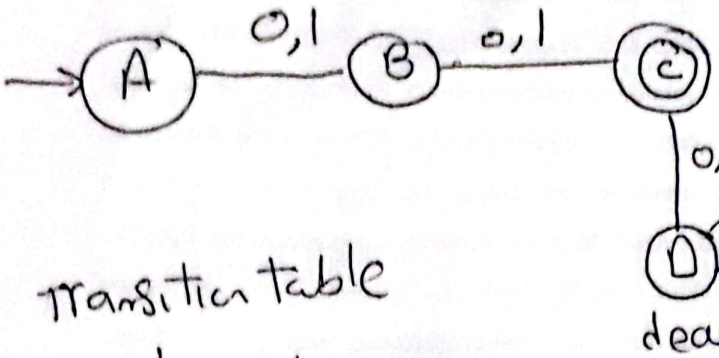
{ a, aa, ab, aab, aba, ... }



Eg 2:- sets of all strings over {0,1} of length 2

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

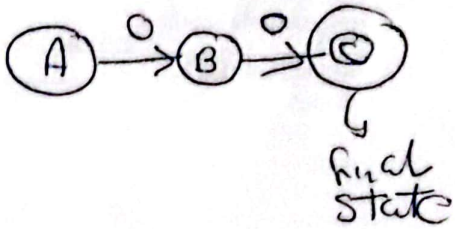
$L = \{00, 01, 10, 11\}$



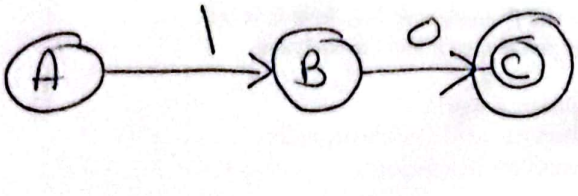
Transition table

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

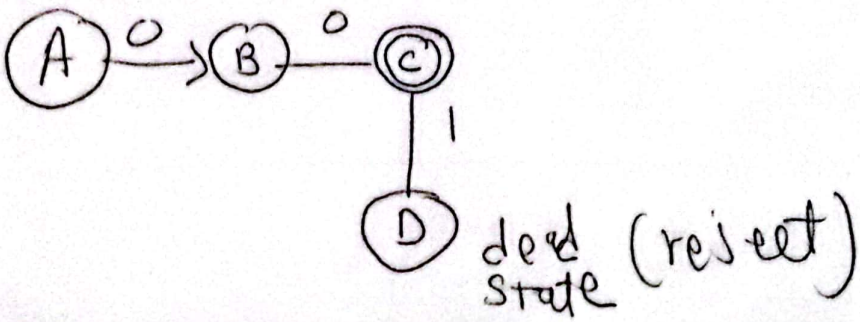
E.g:- 00



E.g:- 10



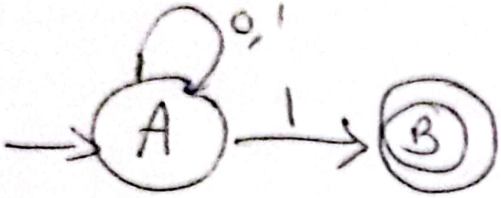
E.g:- 001



Ex-

L = Set of all strings over (0, 1) that ends with (1)

Σ = 0, 1



NFA

Transition Table

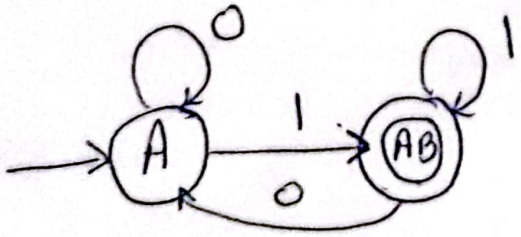
	0	1
A	{A}	{A,B}
B	ϕ	ϕ

* To convert NFA to DFA

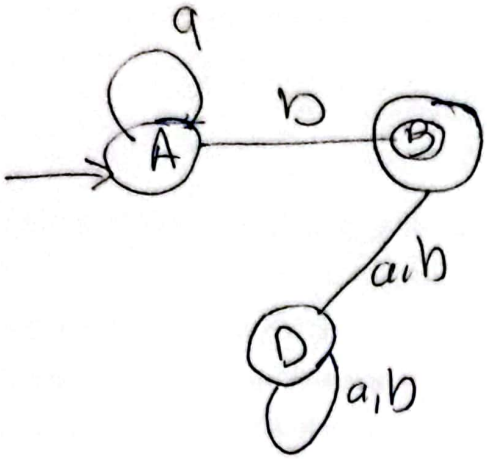
transition table of DFA

	0	1
A	{A}	{AB}
{AB}	{A}	{AB}

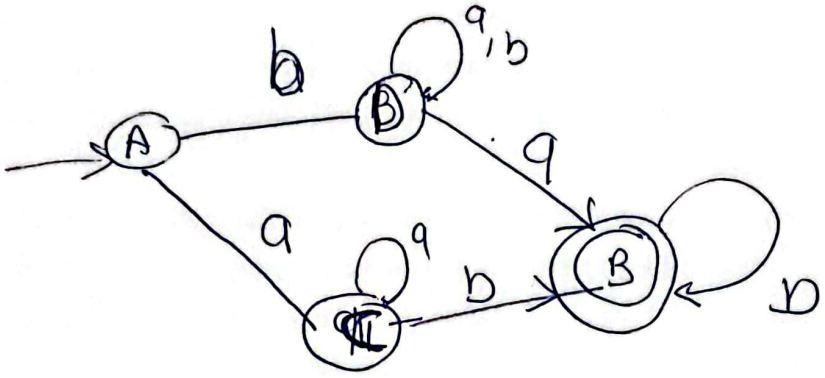
AB is a single state



1 design DFA a^*b $\{a^m b : m \geq 0\}$

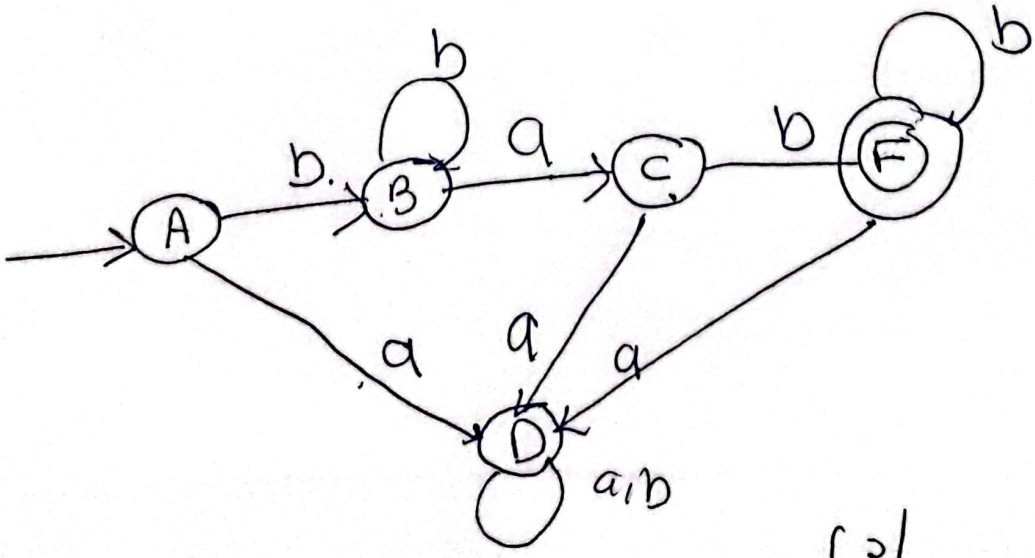


2 $a^m b^n$ $m,n > 0$ $L = \{a^m b^n : m,n > 0$



3 $b^m a b^n$ $m,n > 0$ $L = \{b^m a b^n : m,n > 0\}$

$b^+ a b^+$

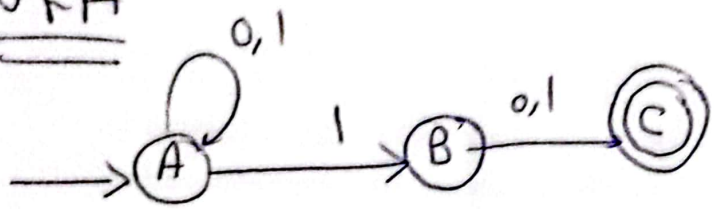


(3)

Ex.4 design NFA for language that accepts all strings over {0,1} in which the second last symbole is 1 then convert it to equivalent DFA

Ex:- 1010
110
1101010

① NFA

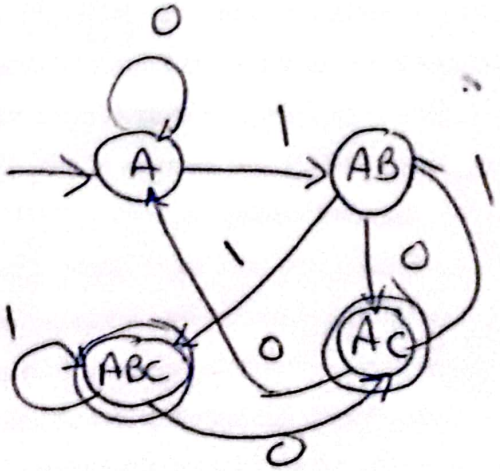


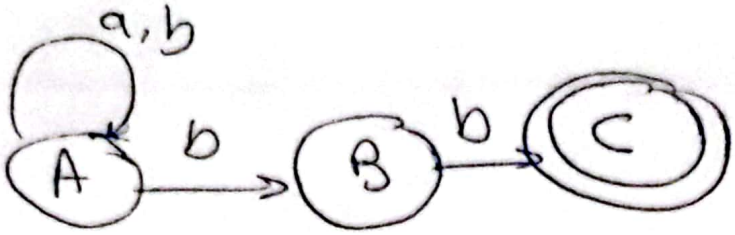
	0	1
A	A	A,B
B	C	C
(C)	ϕ	ϕ

(NFA Transition Table)

② DFA

	0	1
A	A	AB
AB	AC	ABC
(AC)	A	AB
(ABC)	AC	ABC



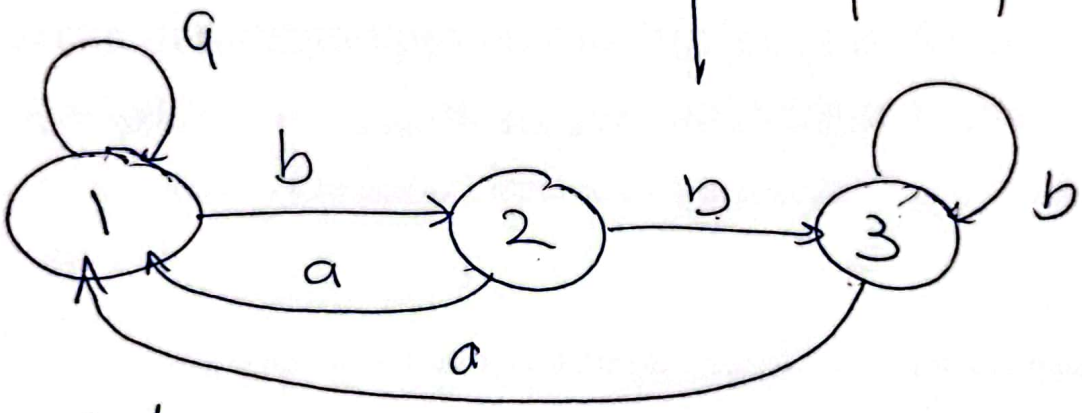


NFA

	a	b
A	A	A,B
B	ϕ	C
* C	ϕ	ϕ

DFA

	State	a	b
1	A	A	AB
2	AB	A	ABC
* 3	ABC	A	ABC



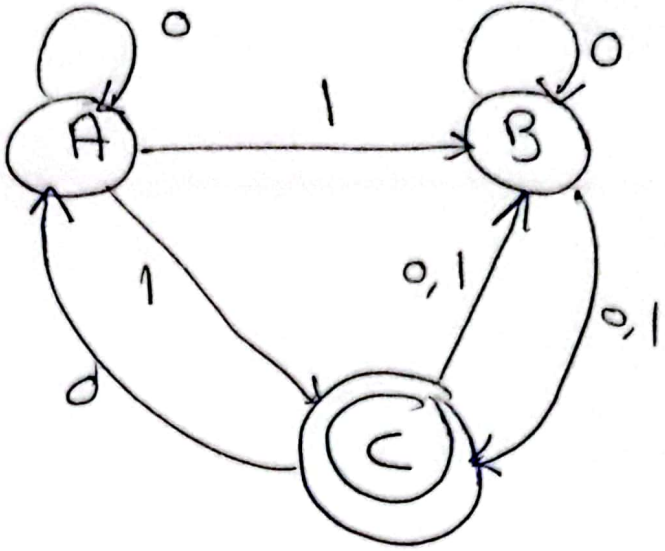
minimization

Π_1 :- { A 1, 2 } { 3 }

{ A 1 } { 2 } { 3 }

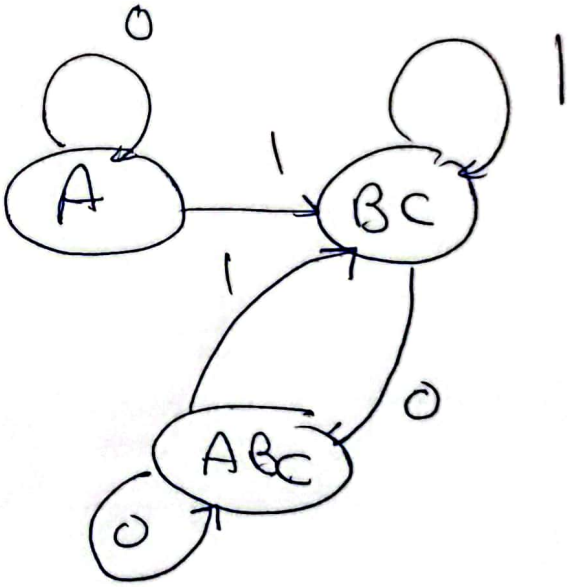
NFA

S	0	1
A	A	{B, C}
B	BC	C
C	AB	B



DFA

	0	1
A	A	BC
* BC	ABC	BC
* ABC	ABC	BC

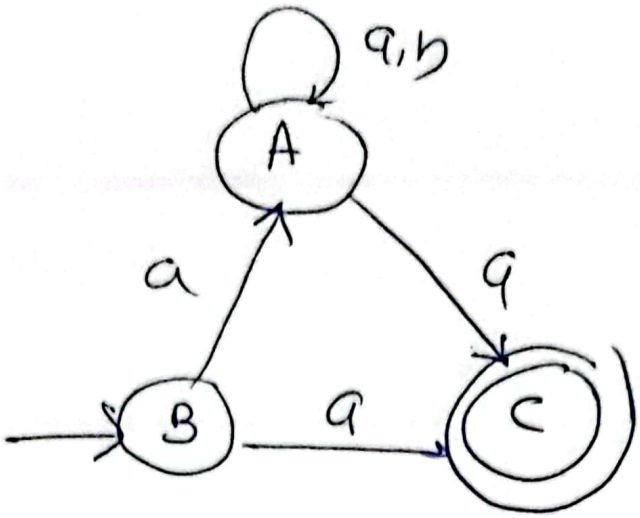


minimization

i:- {A, ~~BC~~} {BC, ABC}

{A} {BC, ABC} ✓

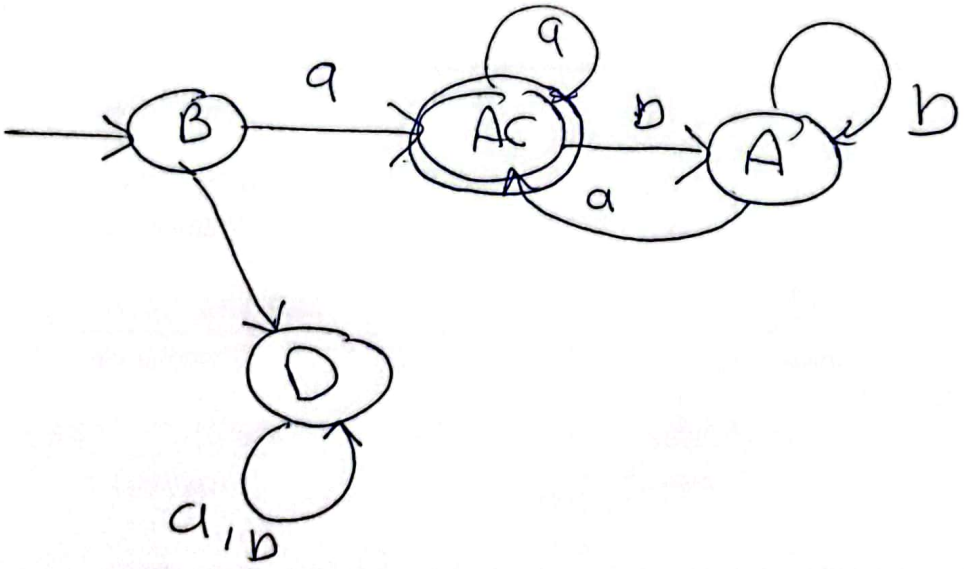
3



NfA

	a	b
B	A,C	ϕ
A	AC	A
C	ϕ	ϕ

	a	b
B	AC	ϕ
AC	AC	A
A	AC	A
ϕ	ϕ	ϕ



minization

① Remark deadnode

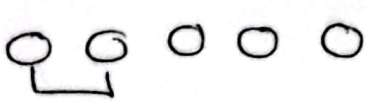
L1- {A,B} {AC}

{A} {B} {AC}

Minimization of DFA

1) is To obtain the minimal version of any DFA which consistor the minimum number of states possible

why :- ① by combining two states

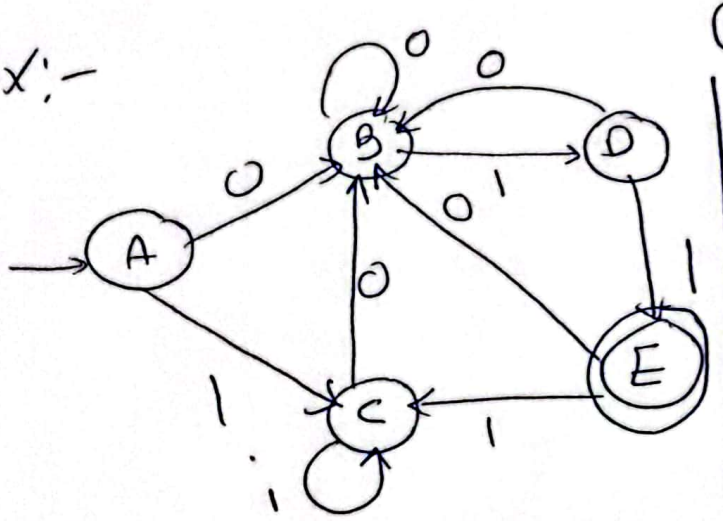


② when two statesare equivalent

$$\begin{cases} \delta(A,x) \rightarrow f \\ \delta(A,x) \rightarrow f \end{cases} \quad \begin{cases} \delta(B,x) \rightarrow f \\ \delta(B,x) \rightarrow f \end{cases}$$

where x is any input string

Ex:-



first draw Transition table

	0	1
→A	B	C
B	B	D
C	B	C
D	B	E
(E)	B	C

① Equivalent

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

② Equivalent

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

A, B ✓
A, C ✓
C, D ✗

③ Eq

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

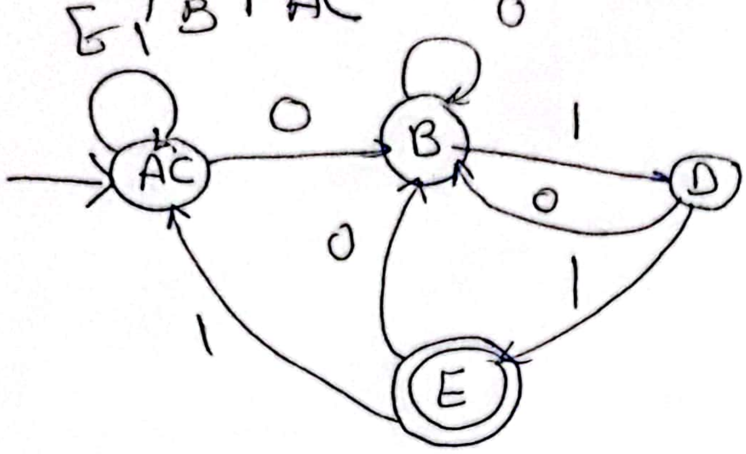
A, B ✗
A, C ✓

④ Eq

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

δ	0	1
AC	B	AC
B	B	D
D	B	E
E	B	AC

* تكون هي نفس قيمة الـ inputs
* دي نفس set



Ex:- 2

Construct a minimum DFA equivalent to The DFA described by

	0	1
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_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

①

0 EQ

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

1 EQ

$\{q_0, q_4, q_6\}$

$\{q_1, q_7\}$

$\{q_3, q_5\}$ $\{q_2\}$

2 EQ

$\{q_0, q_4\}$ $\{q_6\}$ $\{q_1, q_7\}$

$\{q_3, q_5\}$ $\{q_2\}$

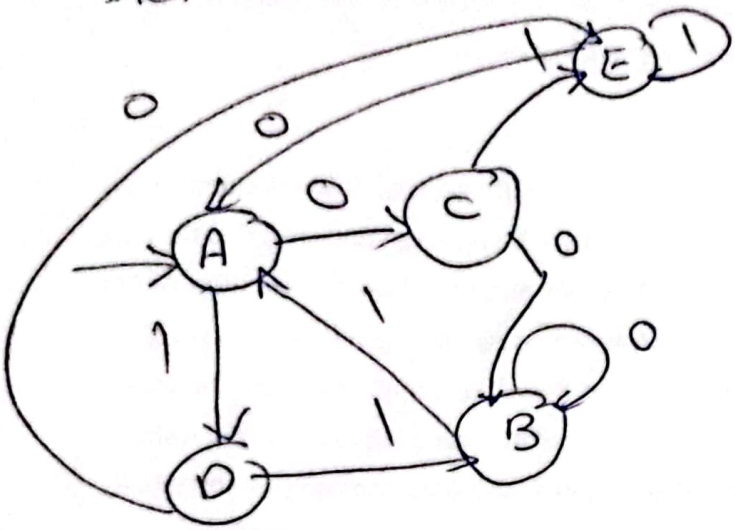
3E9

$\{q_0, q_4\}$ $\{q_6\}$

$\{q_1, q_7\}$ $\{q_3, q_5\}$ $\{q_2\}$

2E9 ≃ 3E9

Then Combin $[q_0, q_4]$, $[q_1, q_7]$ $[q_3, q_5]$

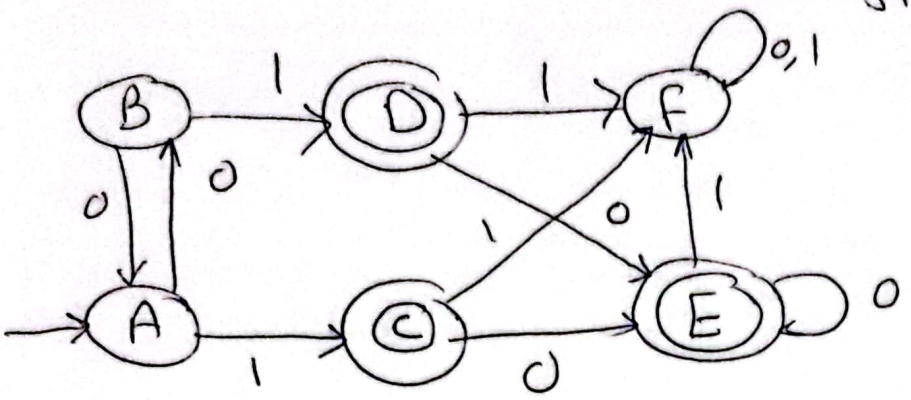


		0	1
A	q_0, q_4	$q_1 q_7$	$q_3 q_5$
B	q_6	q_6	$q_0 q_4$
C	q_1, q_7	q_6	q_2
D	q_3, q_5	q_2	q_6
E	q_2	q_0 / q_4	q_2

	0	1
A	C	D
B	B	A
C	B	E
D	E	B
E	A	E

EX:-2

minimize DFA when more than final state



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

0 E q

{A, B, f} {C, D, E}

1 E q

{A, B} {f} {C, D, E}

=

2 E q

{A, B} {f} {C, D, E}

	0	1
→ AB	AB	CDE
f	f	f
CDE	EDE	f