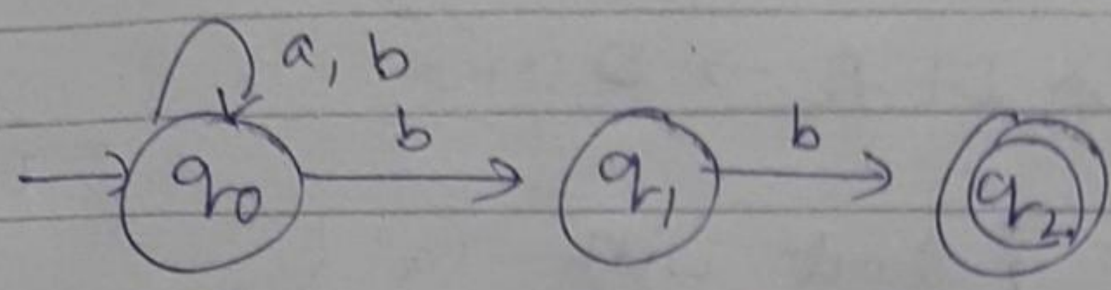
Conversion of NFA to DFA

DATE: / /
PAGE

$\delta : Q \times \Sigma \to 2^Q$ $\delta : Q \times \Sigma \to Q$



⇒ NFA

1. Construct NFA ~~Table~~ Transition Table

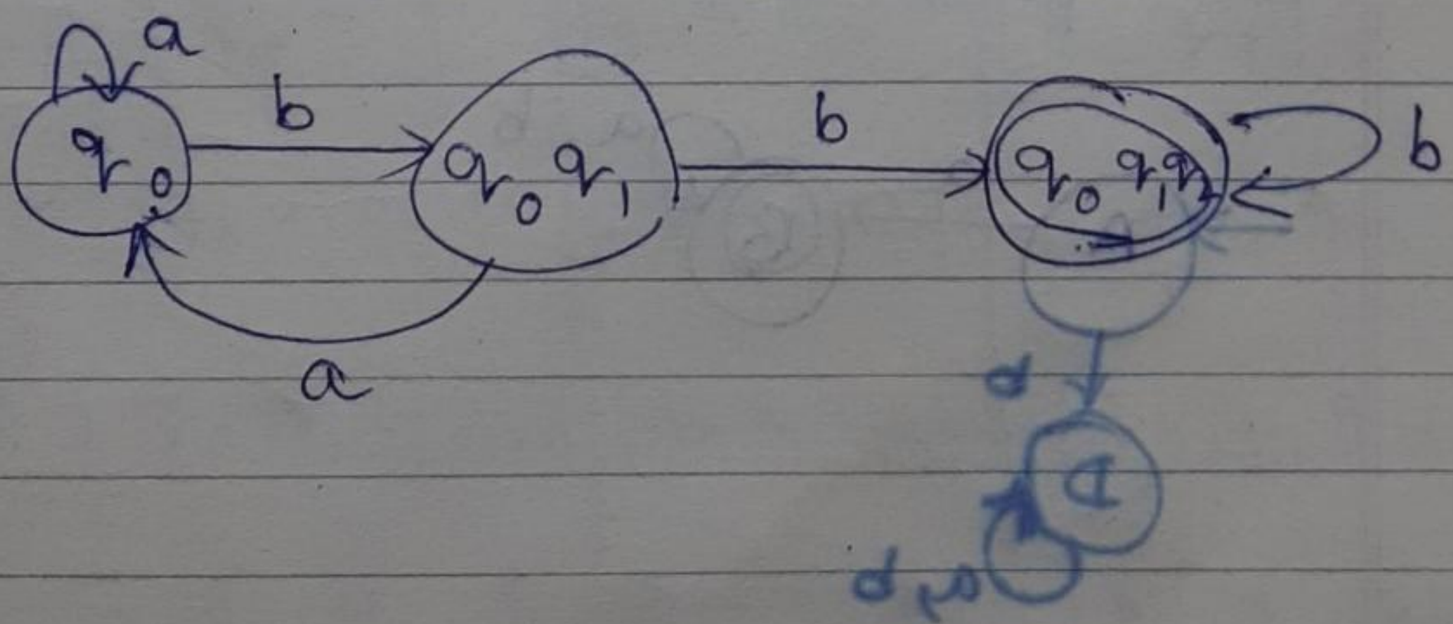
	a	b
→ q_0	q_0	$\{q_0, q_1\}$
q_1	$\in$ Not in	q_2
q_2	—	—

DFA

δ	a	b
→ q_0	q_0	$[q_0 q_1]$
$[q_0 q_1]$	$[q_0]$	$[q_0 q_1 q_2]$
$[q_0 q_1 q_2]$	$[q_0]$	$[q_0 q_1 q_2]$

⇒ single state

Taking the Union



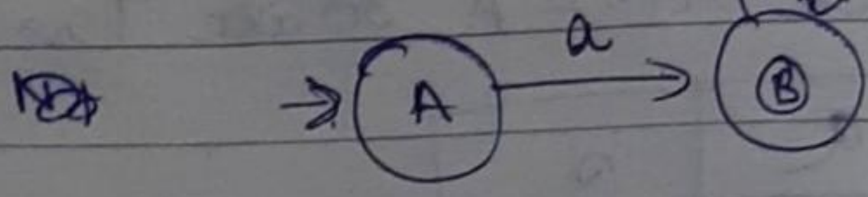
DATE: / /
PAGE:

NFA → DFA

X DFA → NFA

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

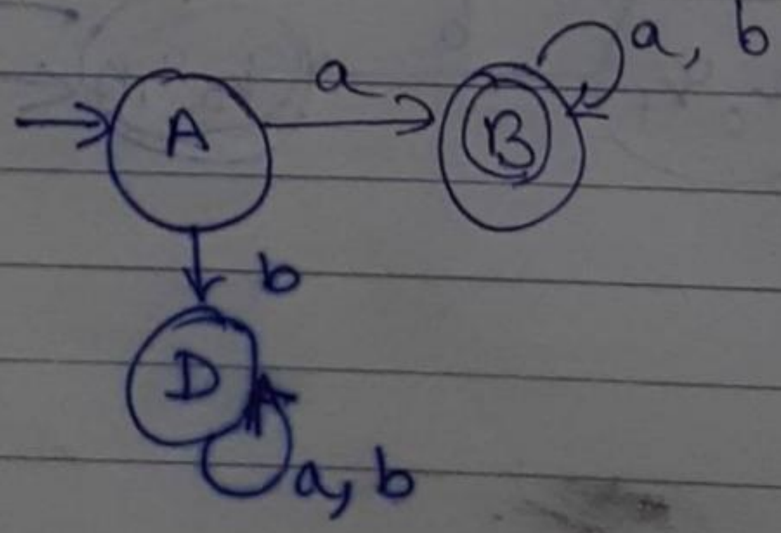
L1 = {start with 'a'}



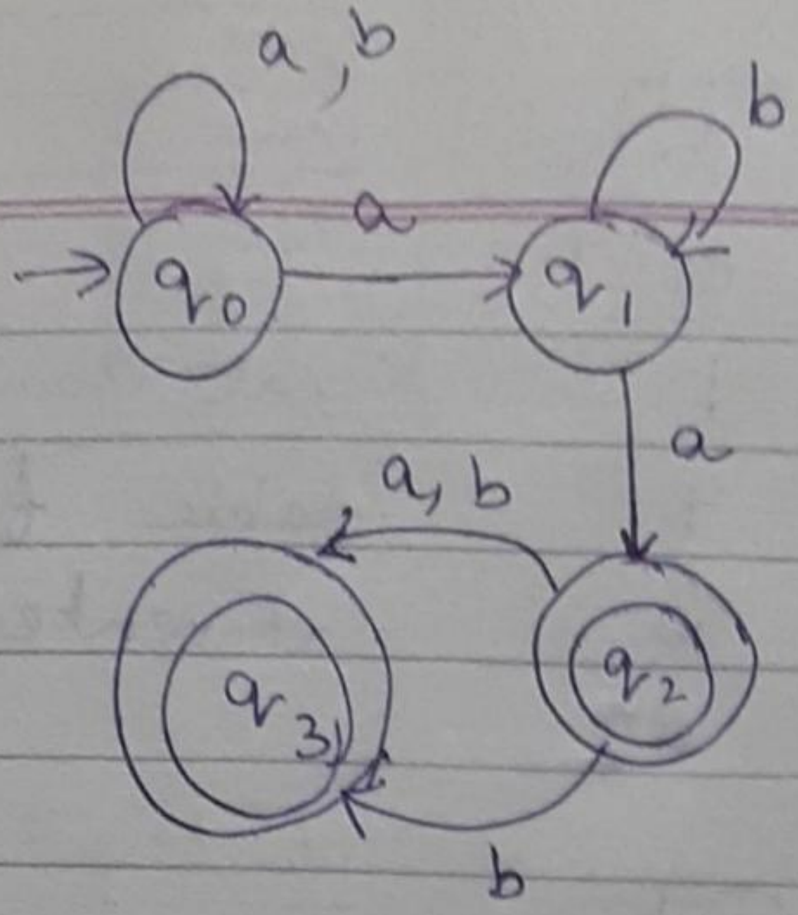
Subset Construction :—

NFA	a	b
A	B	ϕ → Dead Configuration
B	B	B

DFA	a	b
A	B	D Deadstate
*B	B	B
D	D	D



Ex



DATE: / /
PAGE:

NFA

Present state	Next states	
	a	b
→ q_0	$\{q_0, q_1\}$	q_0
q_1	q_2	q_1
* q_2	q_3	q_3
* q_3	—	q_2

DFA

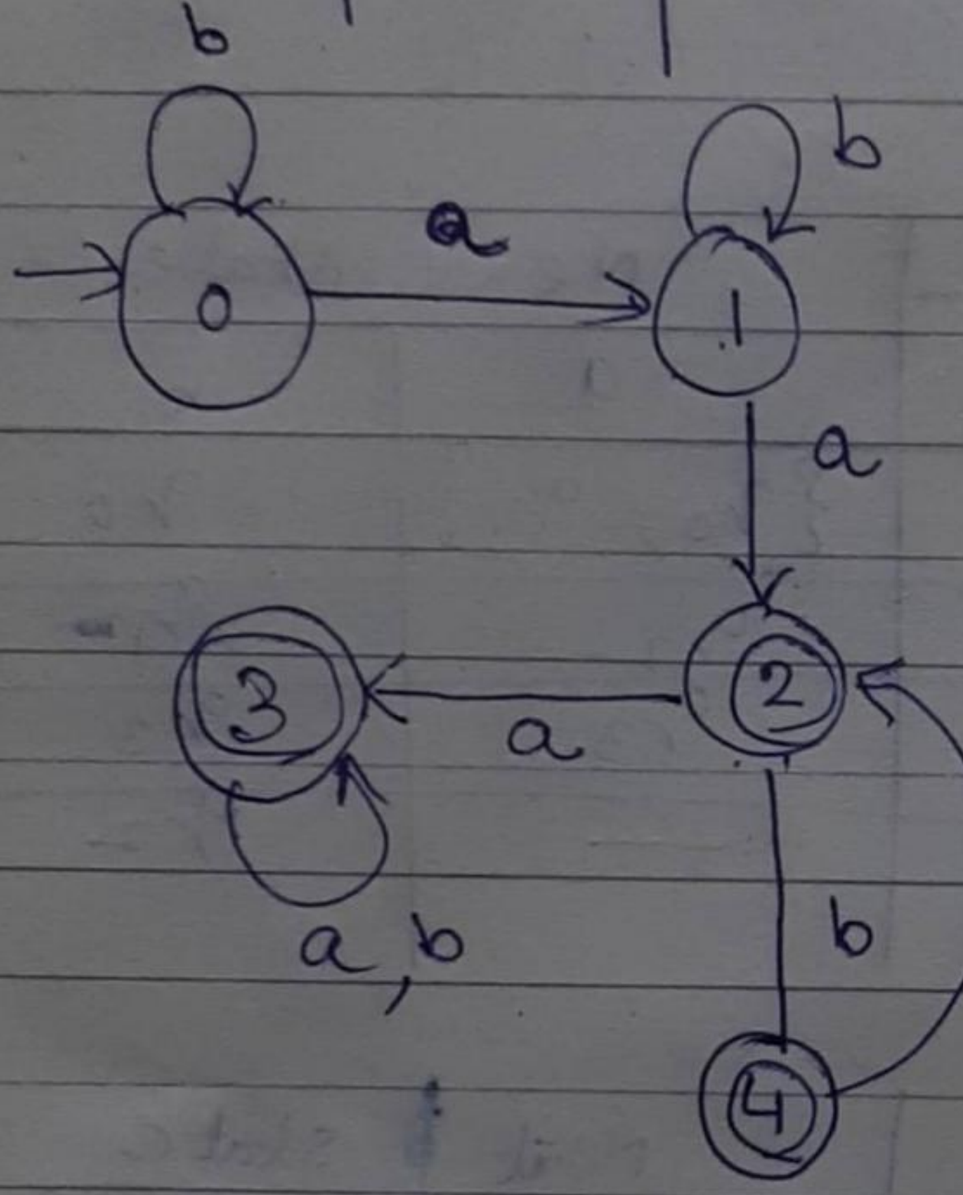
Rename

	Present	Next state a	Next state b
0	→ q_0	$[q_0 q_1]$	q_0
1	$[q_0 q_1]$	$[q_0 q_1 q_2]$	$[q_0 q_1]$
2	* $[q_0 q_1 q_2]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_3]$
3	* $[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_3]$
4	* $[q_0 q_1 q_3]$	$[q_0 q_1 q_2 q_3]$	$[q_0 q_1 q_3]$

Either q_2 or q_3 would be final state

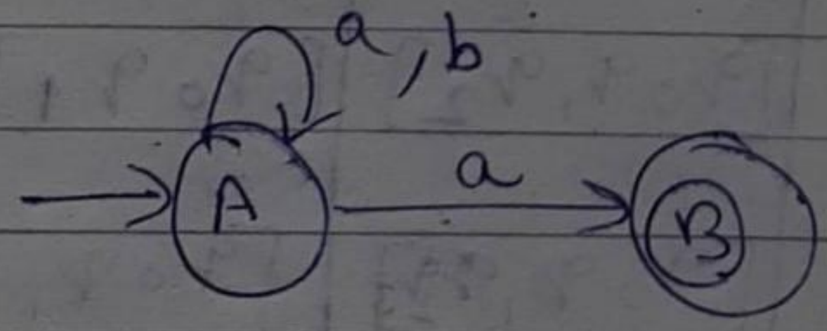
	a	b
→ 0	1	0
1	2	1
2	3	4
3	3	3
4	2	2

Final Transition table for converted DFA



Ex.

L_2 = { ends with 'a' }



NFA

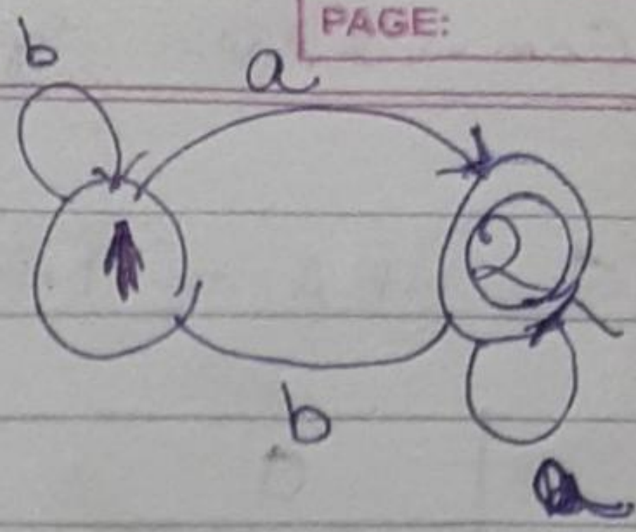
	a	b
→ A	{A, B}	{ A}
*B	{ }	{ }

1 → A

2 *AB

	a	b
[AB]	[AB]	[A]
[AB]	[AB]	[A]

	a	b
1	2	1
*2	2	1

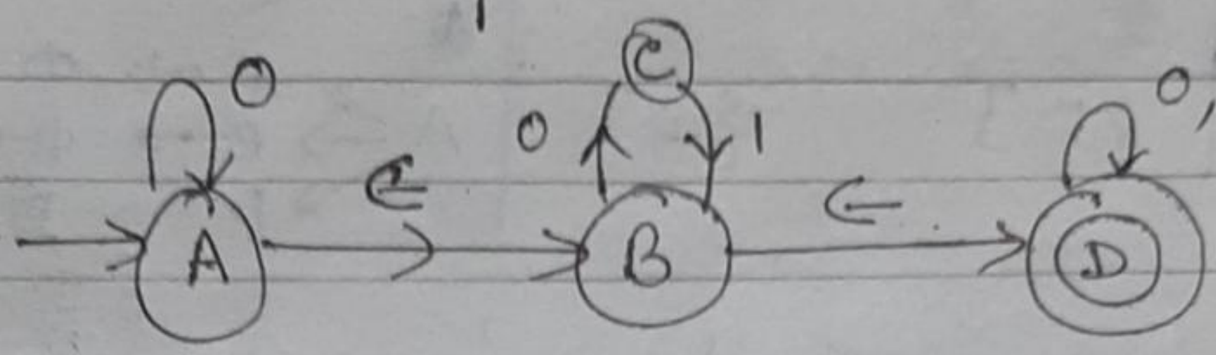


DATE: / /

PAGE:

** Epsilon NFA. ∈-NFA

It is special kind of NFA.



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

$\delta : Q \times \Sigma\{\in\} \rightarrow 2^Q$

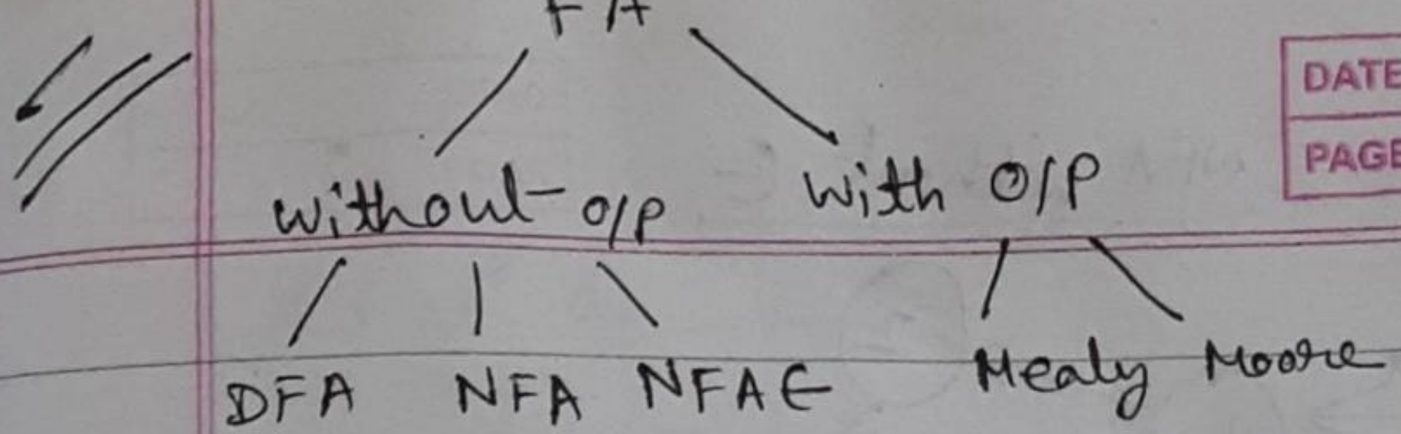
If ∈ is removed then it is a NFA

without giving anything it can goes to one state.

∈ - closure (A) = { A , B, D }

$A \xrightarrow{\in} A$

$A \xrightarrow{\in} B \xrightarrow{\in} D$



DATE: / /
PAGE:

NFA with ∈

↓

Empty symbol

↓

Char.

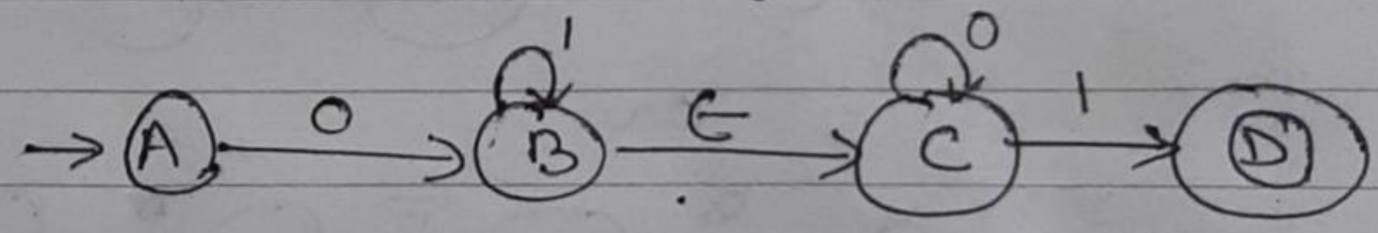
⇒ Regular NFA with 5 tuples $\{Q, \Sigma, q_0, F, \delta\}$

$\delta : Q \times \Sigma \to 2^Q$

→ ∈-NFA with 5 tuples $\{Q, \Sigma, q_0, F, \delta\}$

where $\delta : Q \times \Sigma \cup \in \to 2^Q$

eg.



⇒ B on seeing nothing it can go to 'C'.

Note Every state on ∈ goes to itself.

Conversion of ∈ NFA to DFA

1st → ∈ - NFA → NFA

		0	1
NFA from the given ∈-NFA	A	{A, B, C, D}	{D}
	B	{C, D}	{D}
	C	{ϕ}	{B, D}
	D	{D}	{D}

∈ -closure (δ (∈ closure (A), 0))

∈* 0 ∈*

A → A → A, B, D

B → C → ϕ

D → D → D

∈* 1 ∈*

A → A → ϕ

A → B → ϕ

D → D → D

∈* 0 ∈*

B → B → C → C

D → D → D

∈* 1 ∈*

B → B → ϕ

D → D → D

∈* 0 ∈*

C → C → ϕ

∈* 1 ∈*

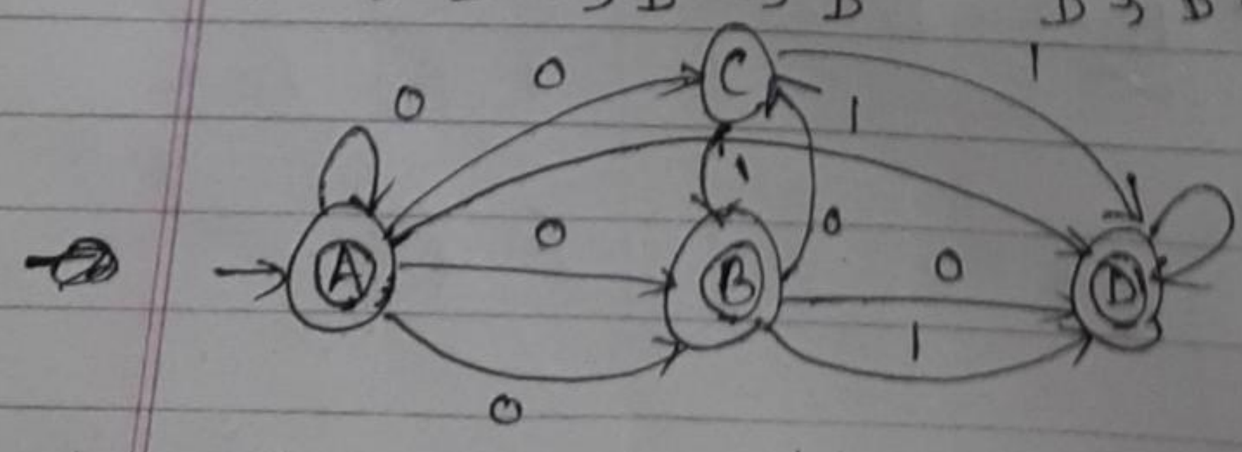
C → C → B → B, D

∈* 0 ∈*

D → D → D → D

∈* 1 ∈*

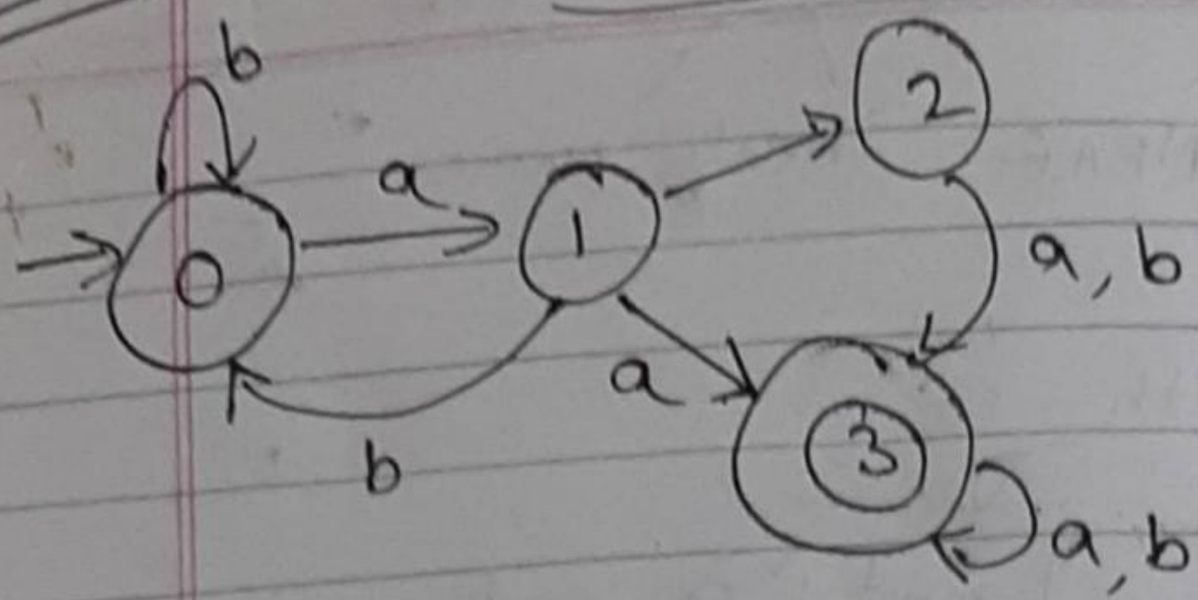
D → D → D → D



A can reach to B only when it sees ∈, thats why B can be a final state.

Eg: 2

NFA without ∈

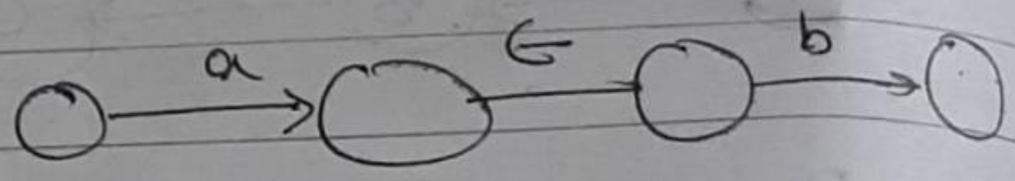


Regular Expression to ∈ - NFA

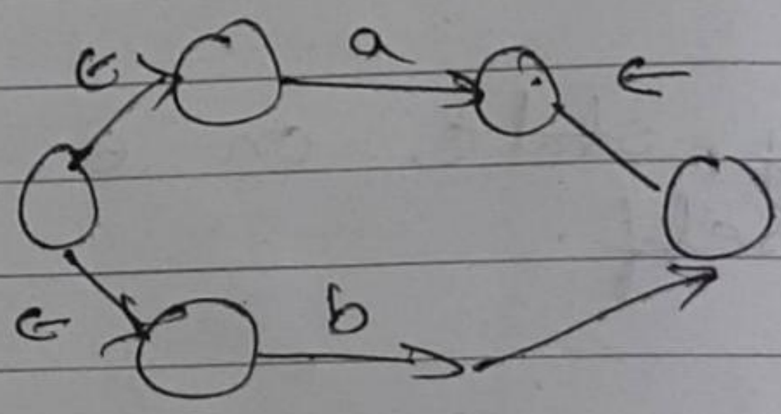
1. ~~Cond~~ concatination

ab

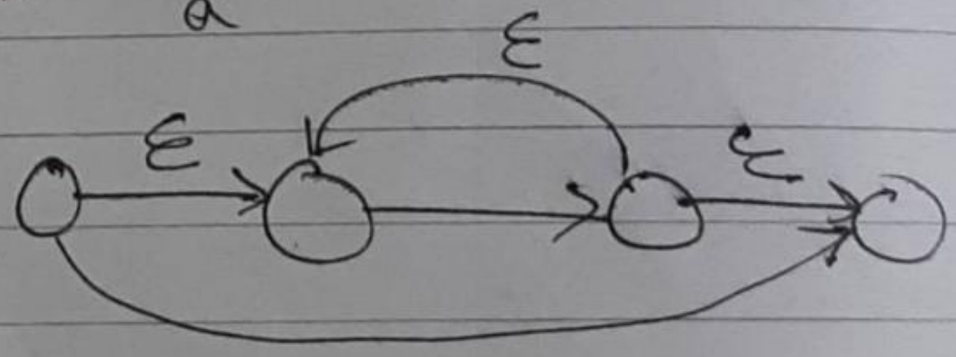
or



2. Union a+b



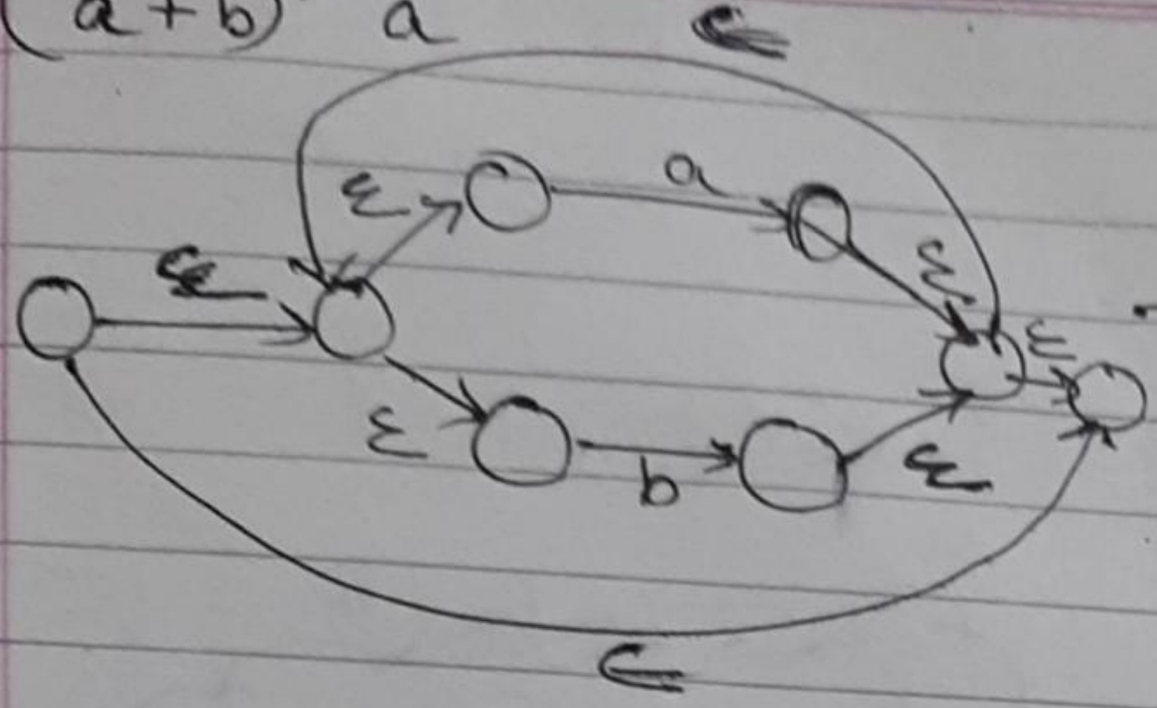
3. closure a*



DATE: / /

PAGE:

$(a+b)^* a$



Thomson structure