CS & IT Engineering

Finite Automata:
Doubt Clearing Session

Lecture No 13

Deva sir

Previous Class Summary:

closure property

$\rightarrow$ 3 min $\Leftarrow$ 4 Hrs

Topics to be covered Today:

- Doubts clearing
- Algorithms :
  - DFA $\Rightarrow$ NFA
  - NFA $\Rightarrow$ DFA
  - RegExp $\Leftrightarrow$ FA
  - Reg Gram $\Leftrightarrow$ FA

I. Regular Expression $\Rightarrow$ FA

II. FA $\Rightarrow$ Reg Exp

III. FA $\Rightarrow$ RLG

IV. RLG $\Rightarrow$ FA

V. FA $\Rightarrow$ LLG

VI. LLG $\Rightarrow$ FA

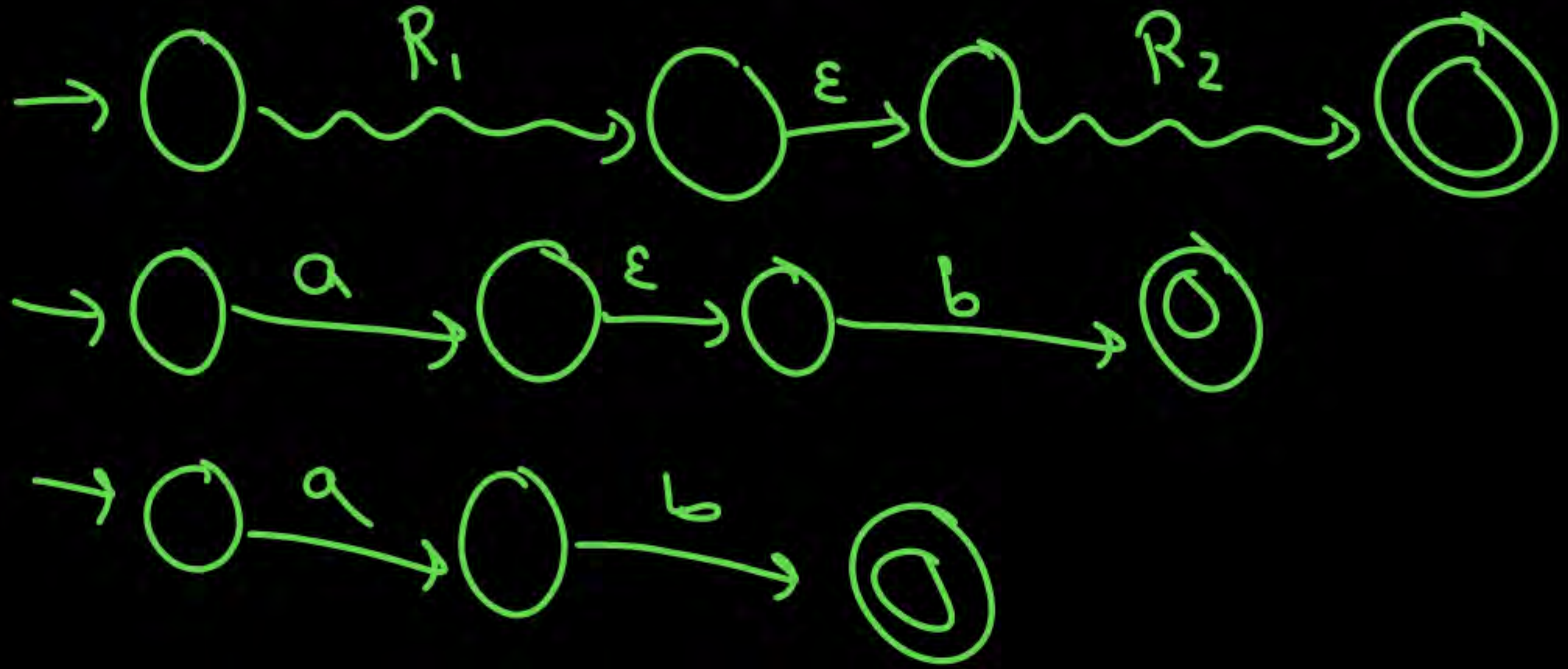
RegExp ⇒ FA

① R

② $R = a$

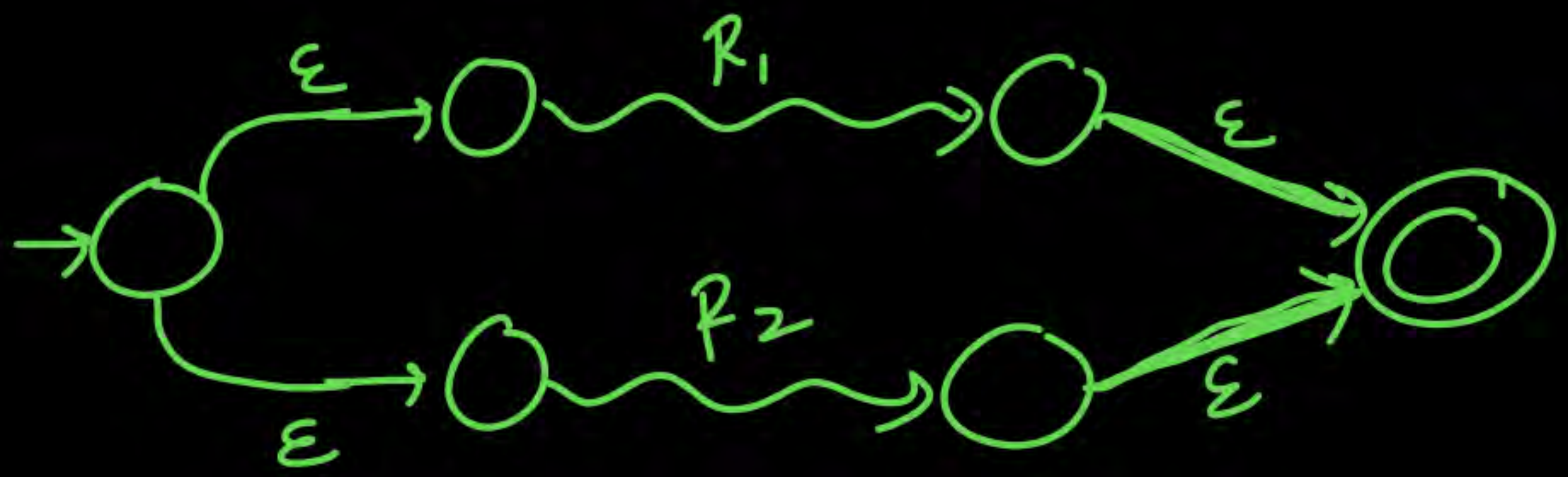
③ $R = a \cdot b$

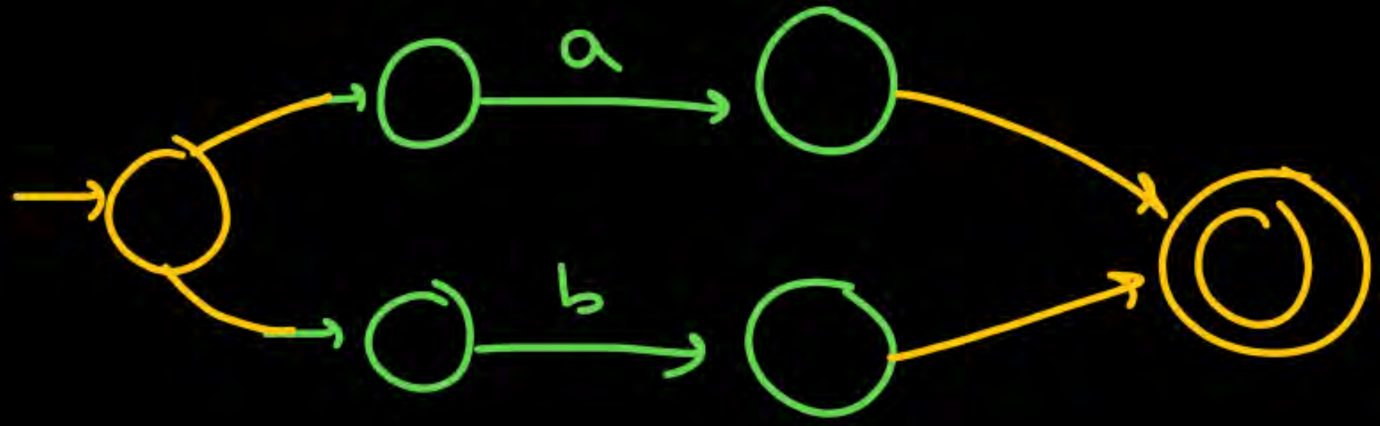
$= R_1 \cdot R_2$



4 $R = a + b$

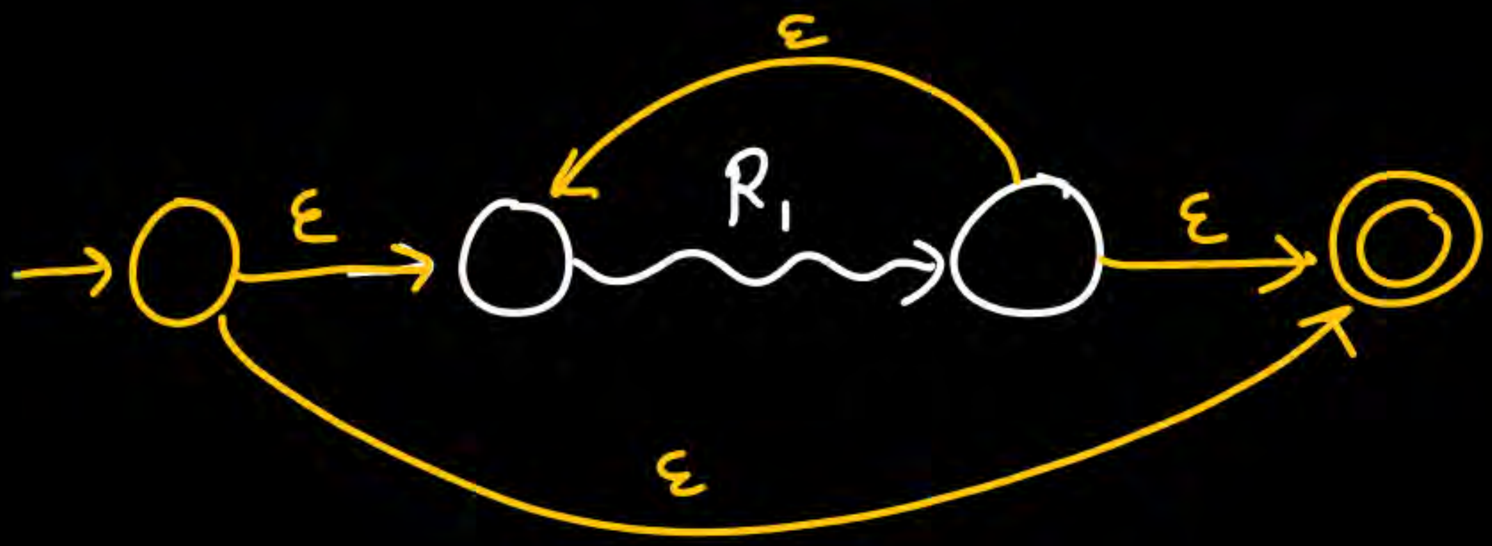
$= R_1 + R_2$

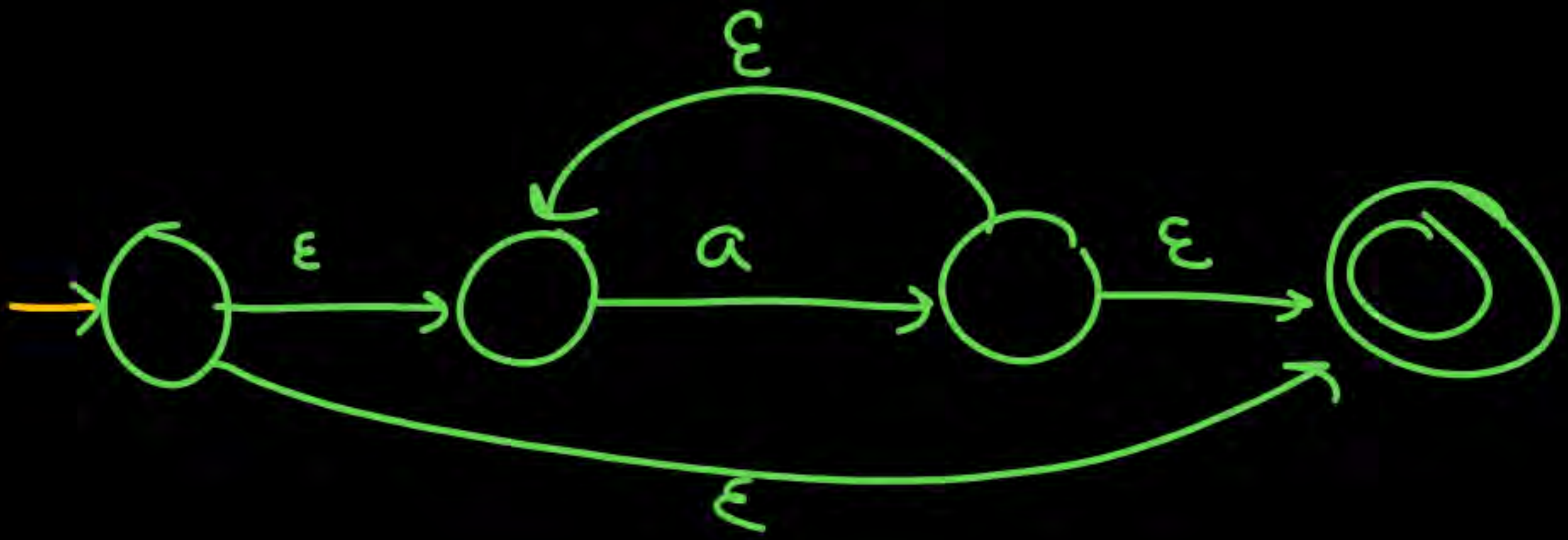




(5) $R = a^*$

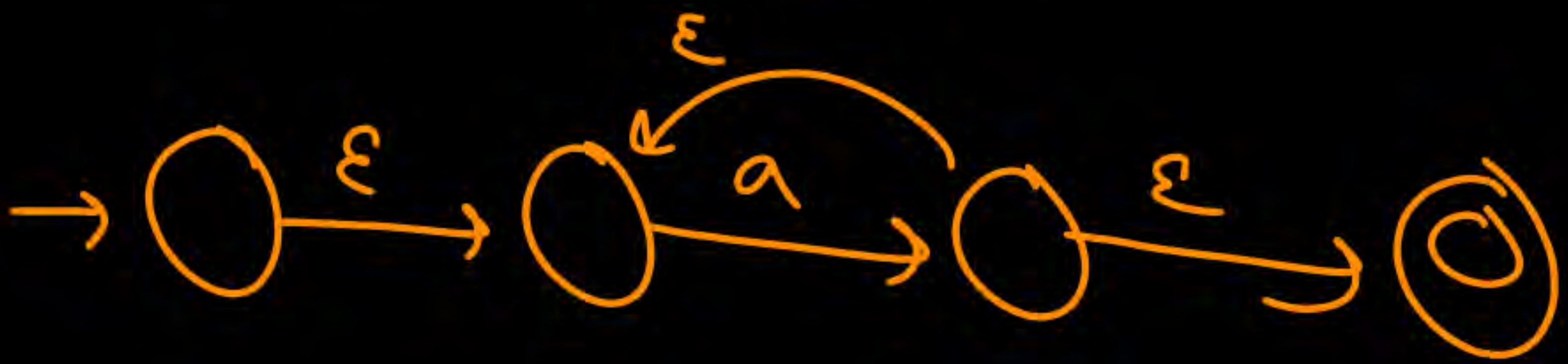
$= R_1^*$





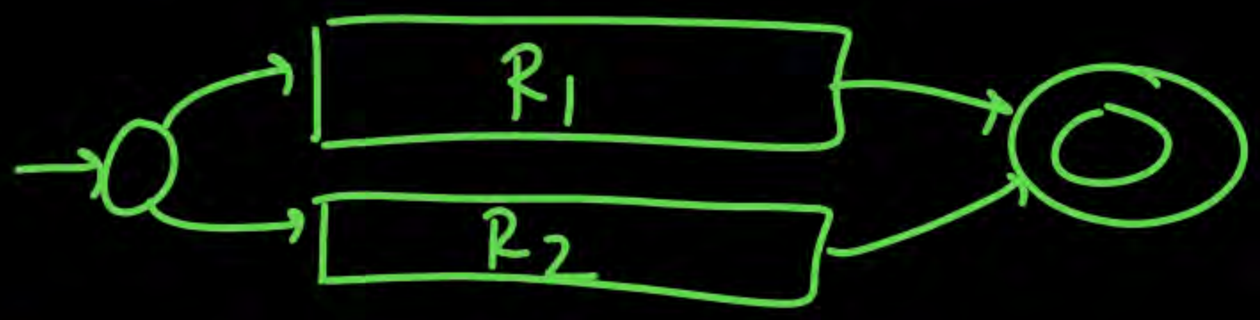
(6) $R = a^+$

$= R_1^+$



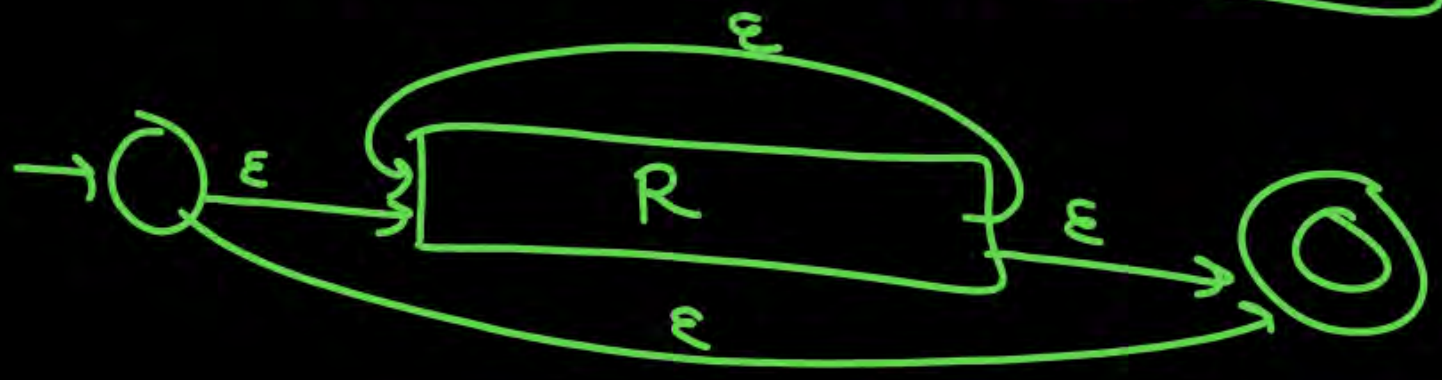
R

$R_1 + R_2$



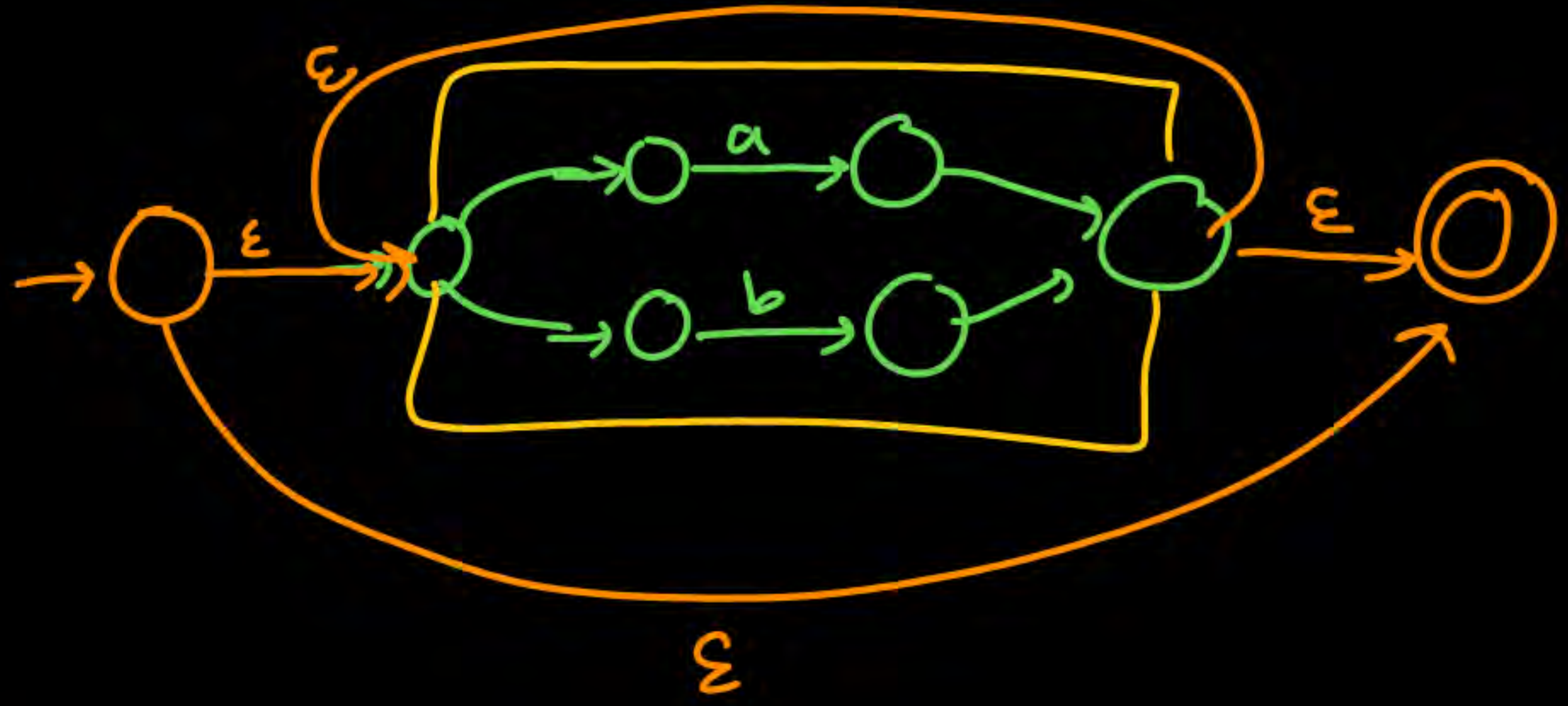
$R_1 . R_2$

R^*



R^+

① $R = (\boxed{a}+\boxed{b})^*$



Shortcut:

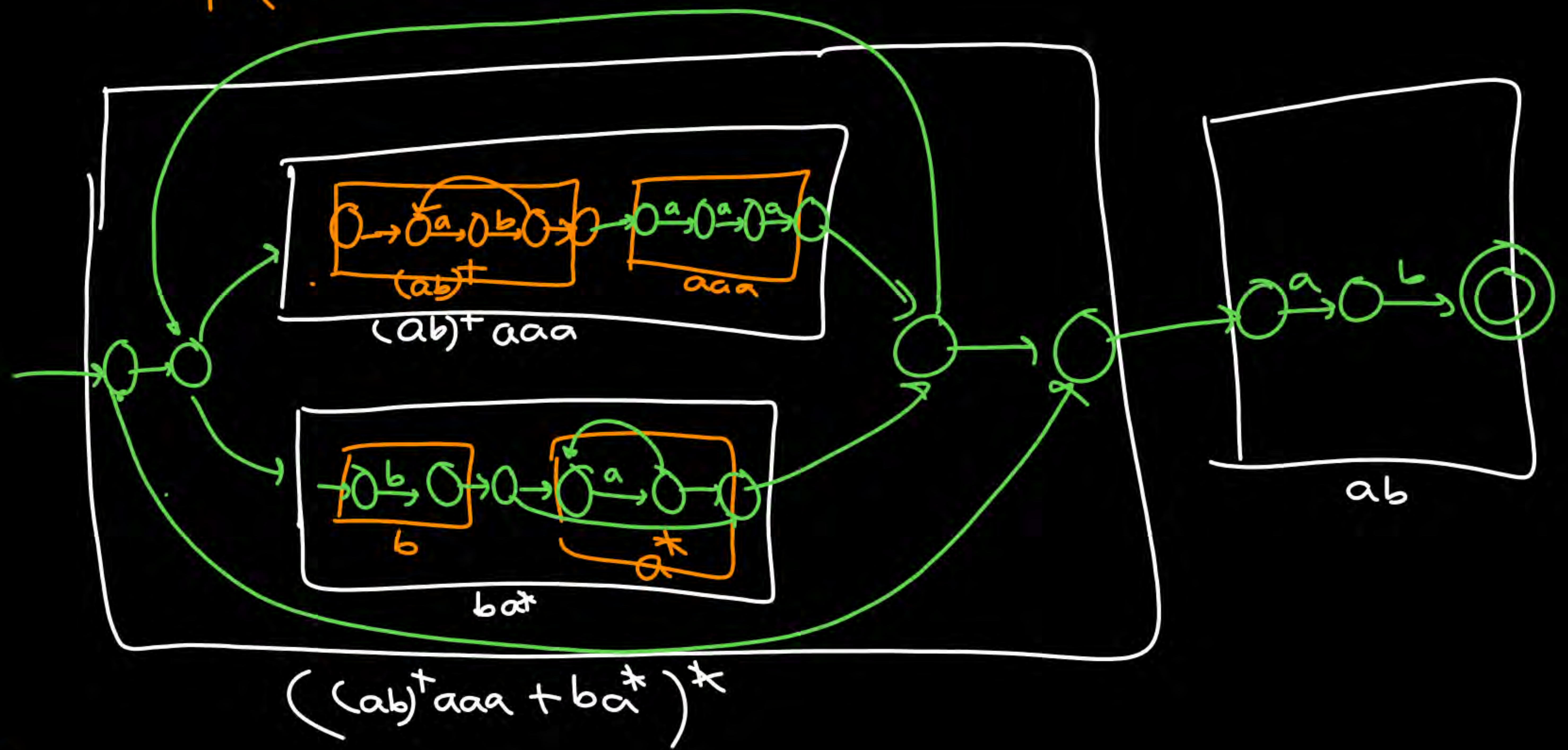
a ✓

b ✓

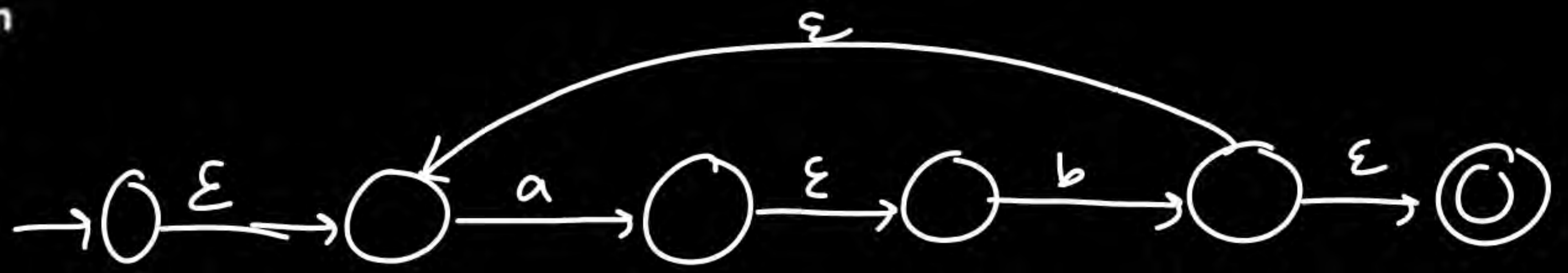
a+b ✓

$(a+b)^*$ ✓

② $((ab)^+ aaa + ba^*)^* ab$

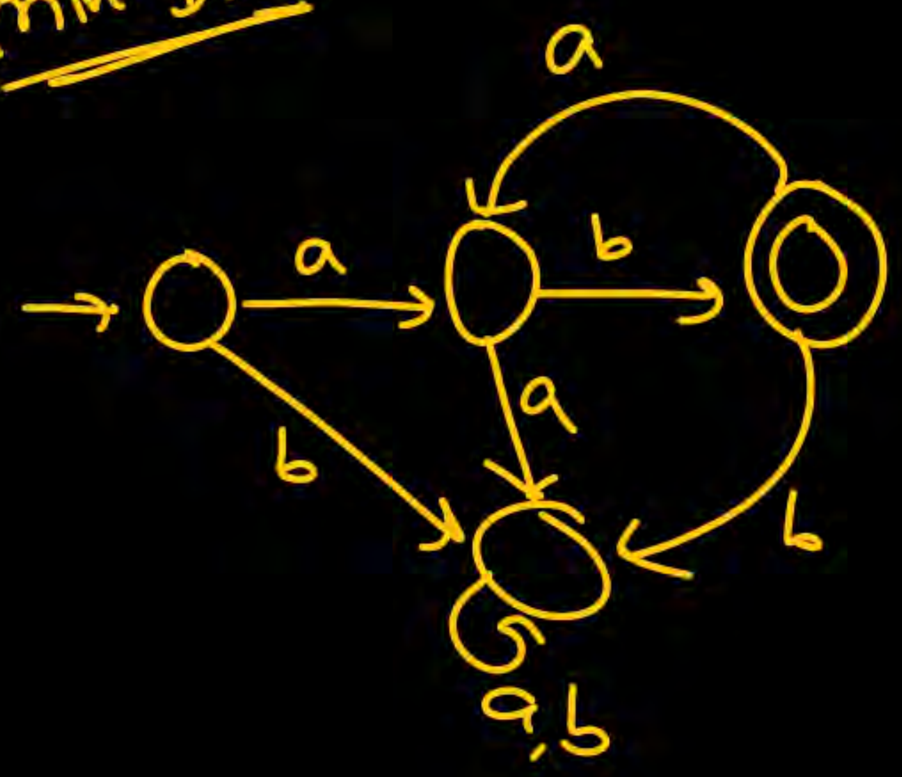


(3)



$$L = (ab)^+$$

min DFA



4. $(ab)^*$

5. $(ab)^*a$

6. $(ab)^+a$

7. $(ba)^*b$

8. $(ba)^*a$

9. $(aaa)^*b$

H.W.
Find no. of states in min DFA.
(No algo required)

FA $\Rightarrow$ Reg Exp :

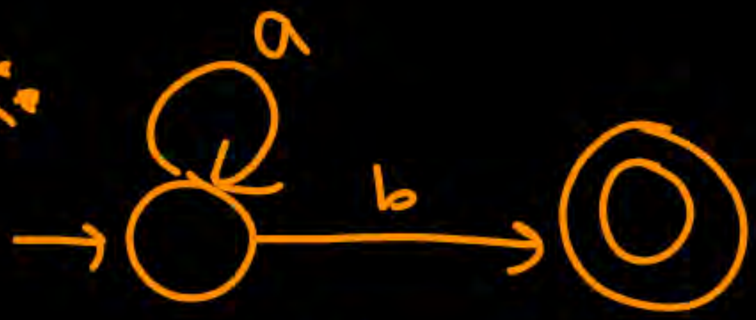
1. Kleene Method → $R_{ij}^{K} = R_{ij}^{K-1} + R_{iK}^{K-1} (R_{KK}^{K-1})^{*} R_{Kj}^{K-1}$
2. Arden's method → If $R = Q + RP$ then $R = QP^{*}$
3. State Elimination

State Elimination Method :

①

$R = a$

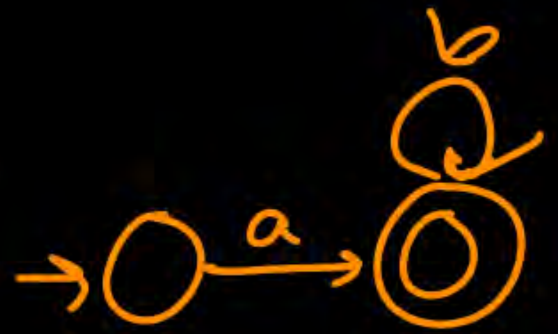
② FA

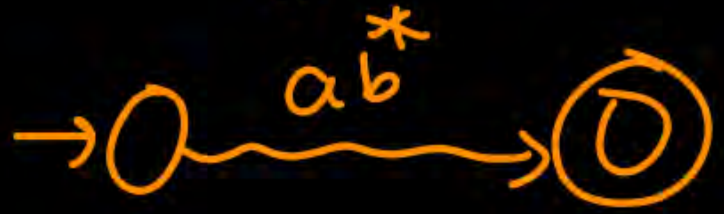


Transition graph

$R = a^* b$

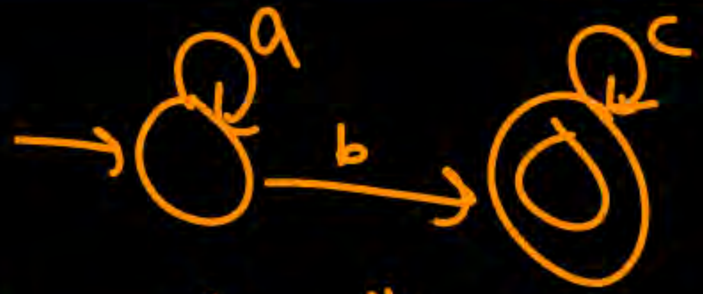
③





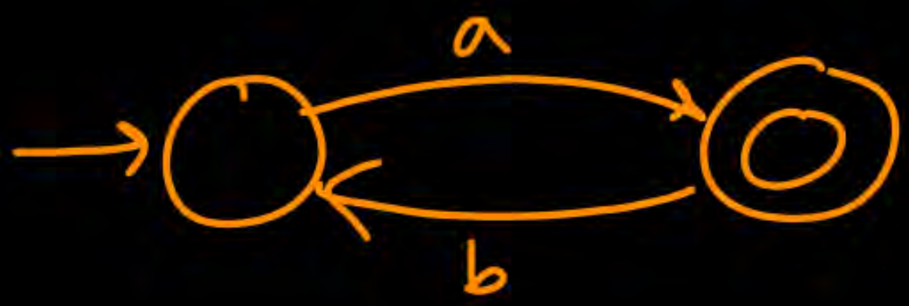
$R = ab^*$

④

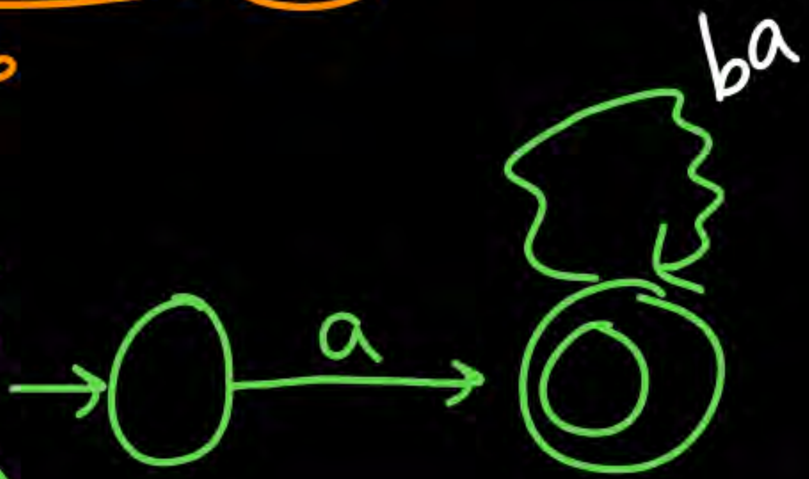


$R = a^* b c^*$

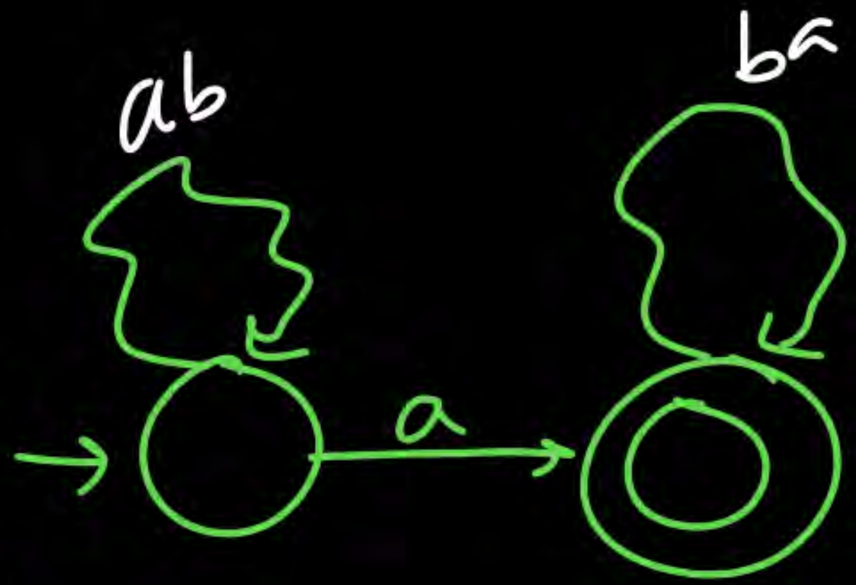
5



$R = (ab)^* a$



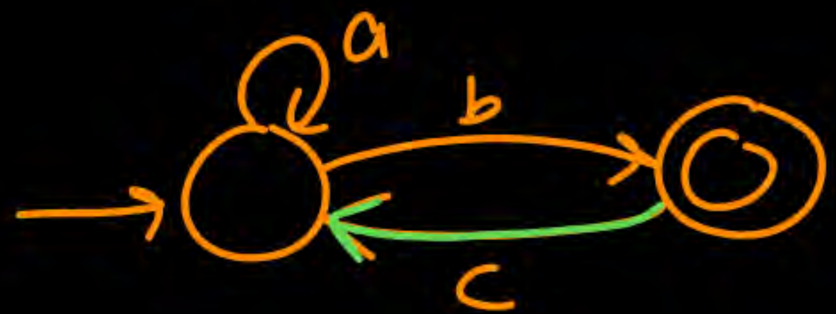
$a(ba)^*$

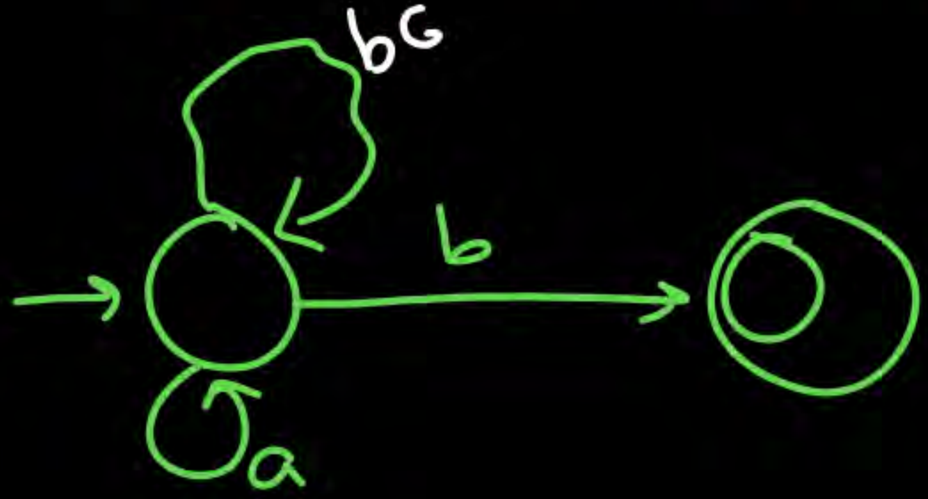


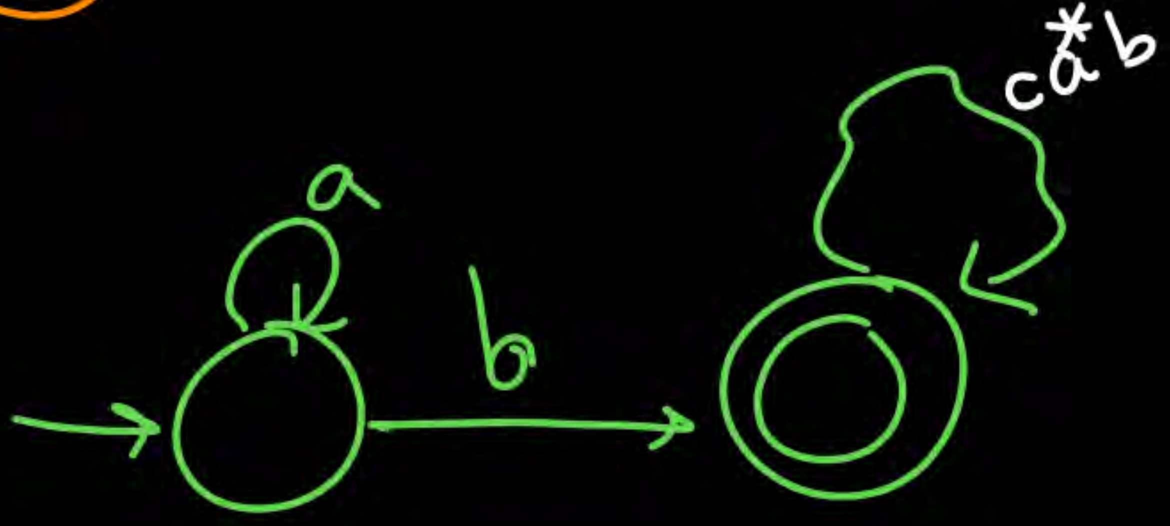
$(ab)^* a (ba)^*$

Doubt Clearing Session

6



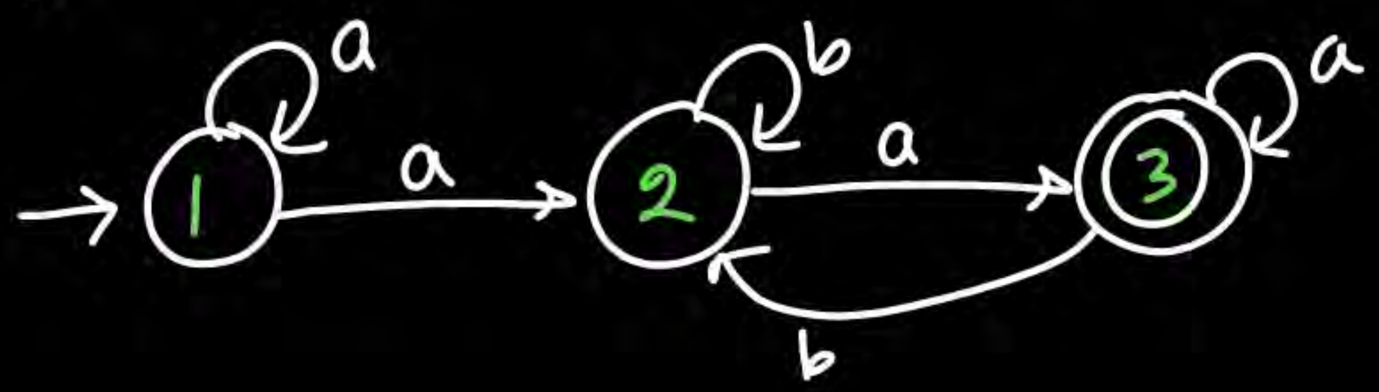




$$= (a+bc)^* b$$

$$= a^* b (ca^* b)^*$$

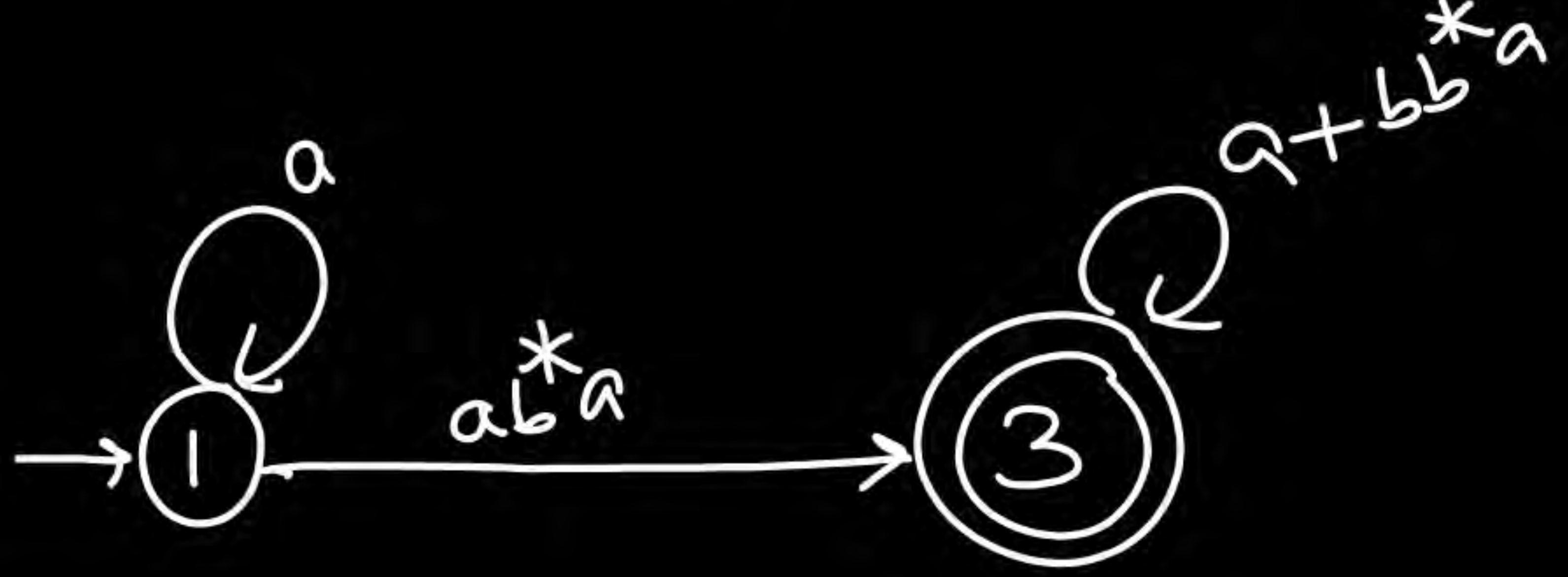
1)



Delete 2

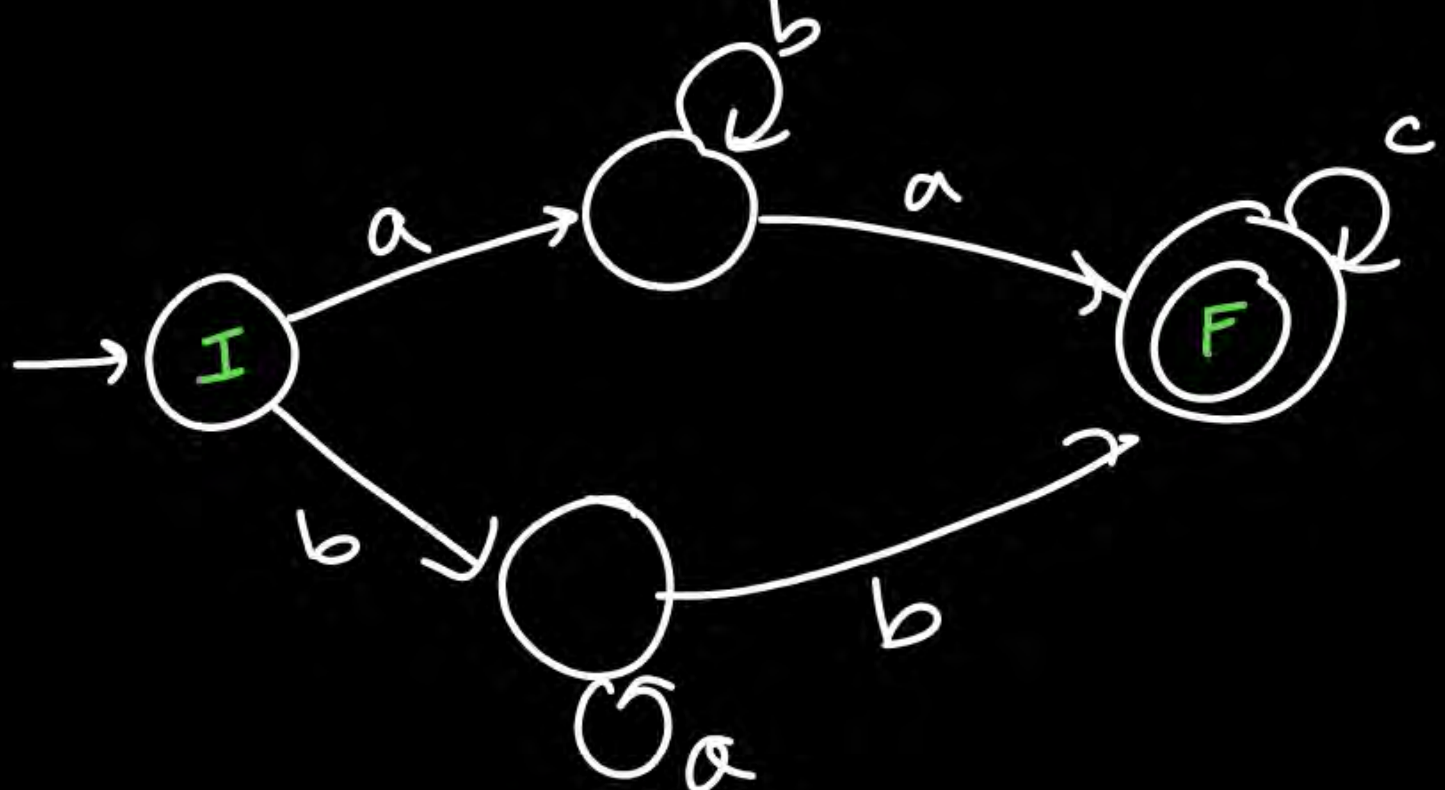
$$1 \xrightarrow{ab^*a} 3$$

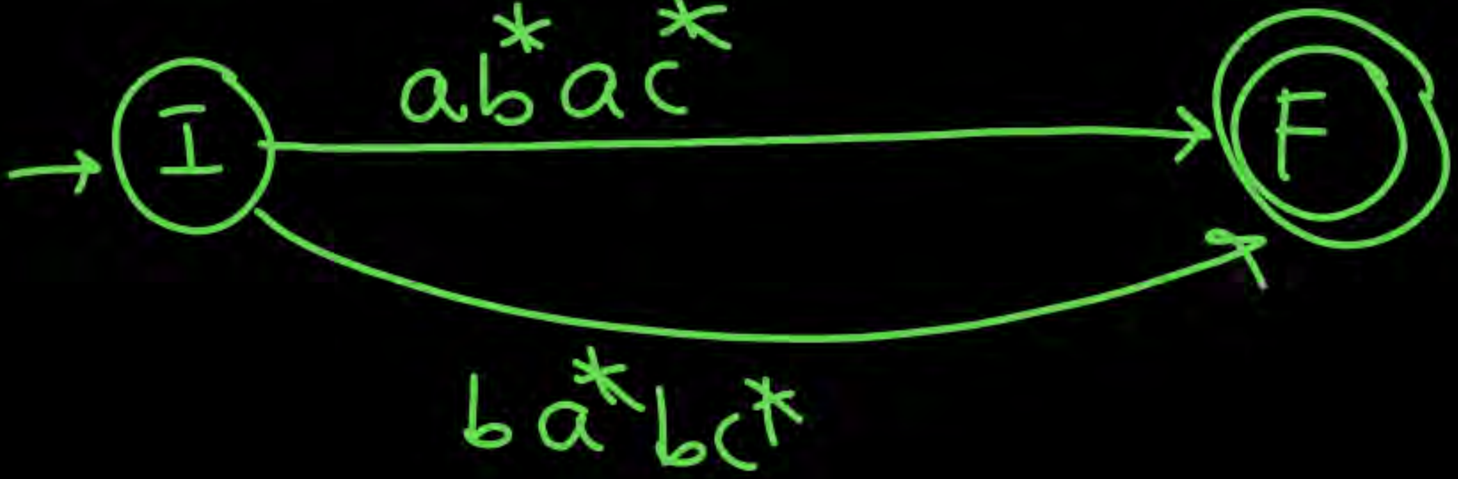
$$3 \xrightarrow{bb^*a} 3$$



$$= a^* . (ab^*a) . (a+bb^*a)^*$$

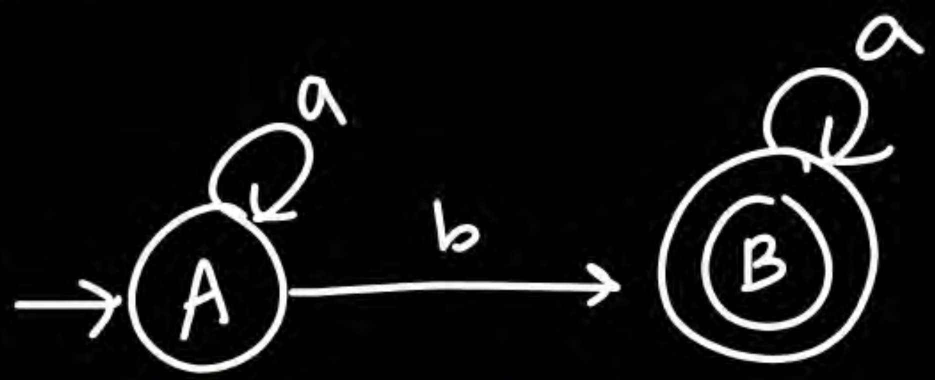
2)





$$= ab^*ac^* + ba^*bc^*$$

FA $\Rightarrow$ RLG :



Given FA

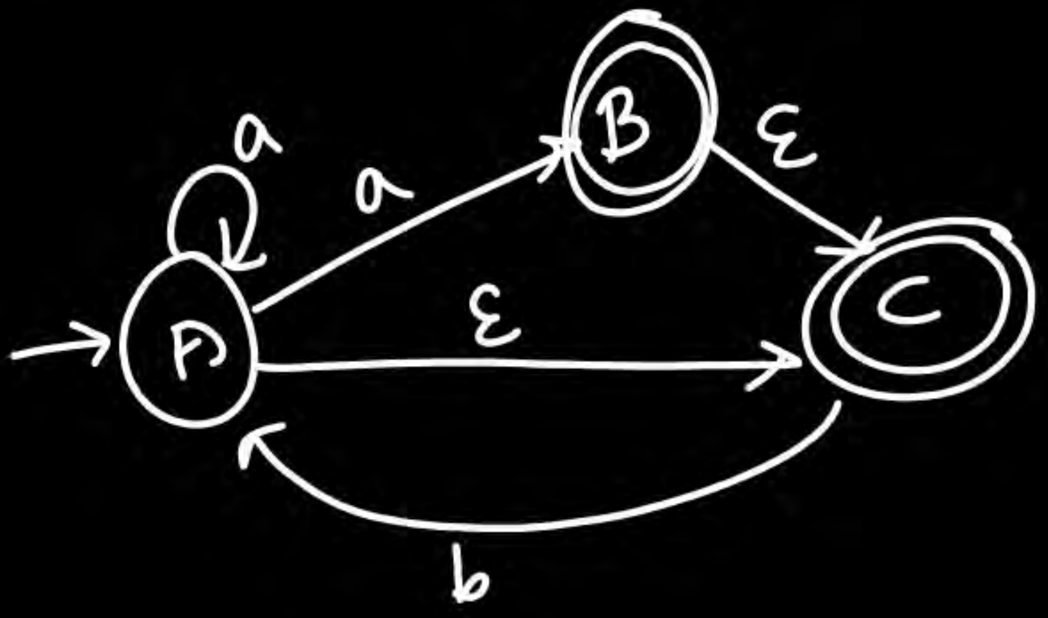
$X \rightarrow iY$

Look at outgoing edges :

$A \rightarrow aA \mid bB$

$B \rightarrow aB \mid \underbrace{\varepsilon}_{\text{for final state}}$

RLG



FA

$\cong$

$$A \rightarrow aA \mid aB \mid C$$
$$B \rightarrow C \mid \varepsilon$$
$$C \rightarrow bA \mid \varepsilon$$

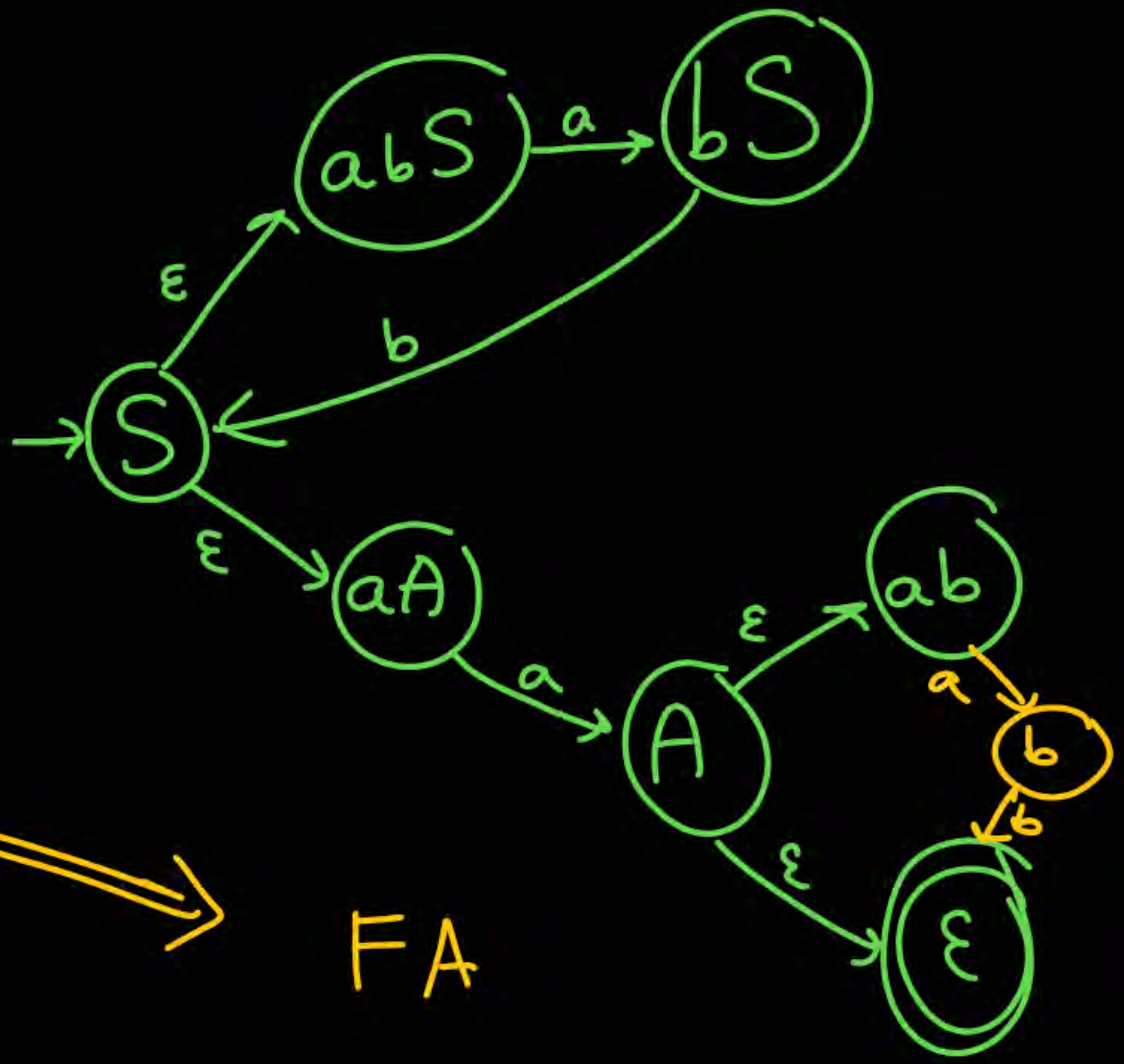
RLG

RLG $\Rightarrow$ FA :

$S \rightarrow abS \mid aA$

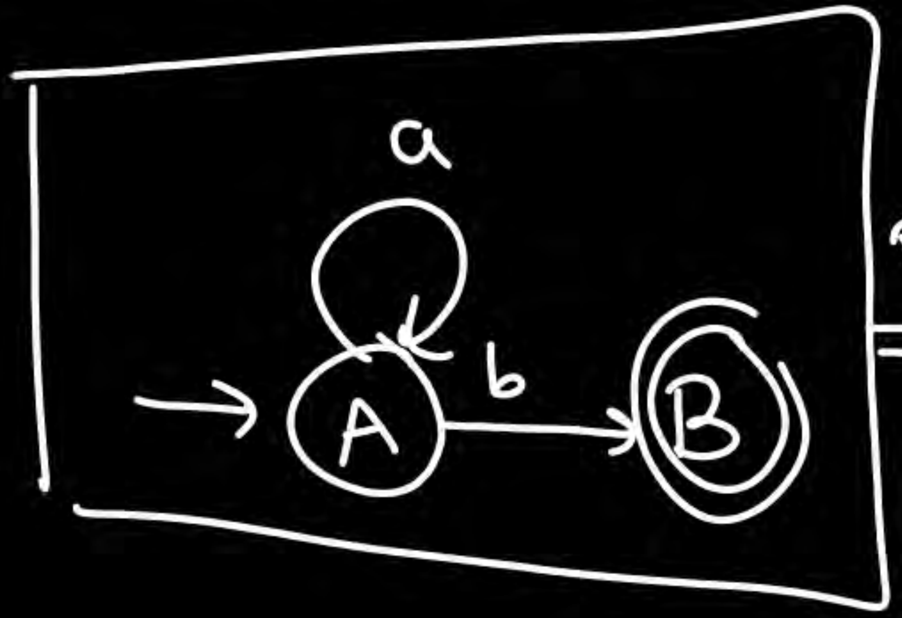
$A \rightarrow ab \mid \varepsilon$

Given RLG $\Rightarrow$ FA

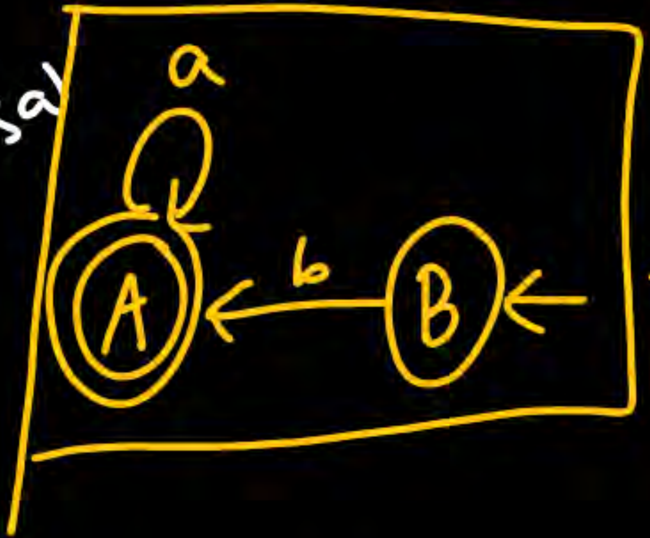


FA $\Rightarrow$ LLG :

$$\underset{L}{\boxed{\text{FA}}} \xRightarrow{\text{Reversal}} \underset{L^{Rev}}{\text{FA}} \Longrightarrow \underset{L^{Rev}}{\text{RLG}} \xRightarrow{\text{Reversal}} \underset{L}{\boxed{\text{LLG}}}$$



Reversal $\Rightarrow$



$\Rightarrow$

RLG

$B \rightarrow bA$

$A \rightarrow aA \mid \varepsilon$

Reversal $\Rightarrow$

LLG

$B \rightarrow Ab$

$A \rightarrow Aa \mid \varepsilon$

$L = a^* b$

$L^{Rev} = ba^*$

$L^{Rev} = ba^*$

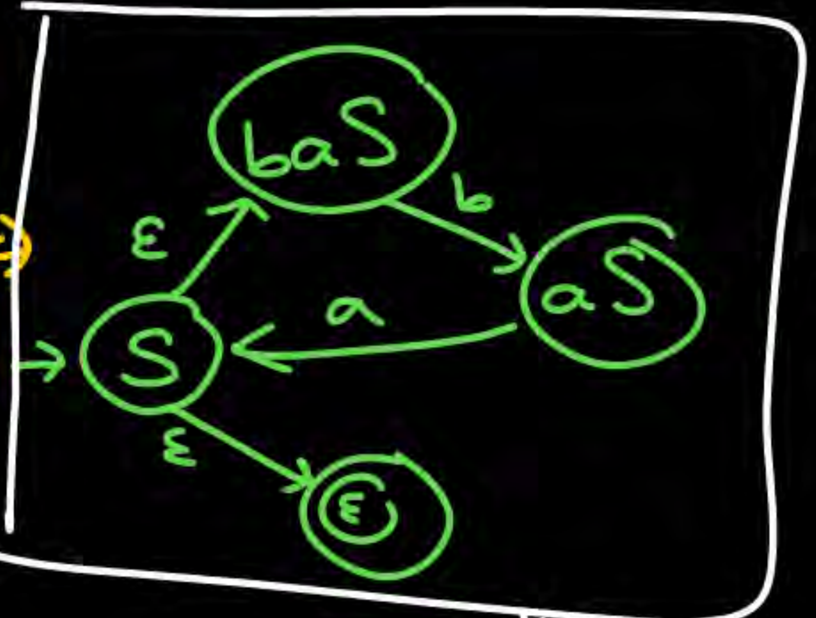
$L = a^* b$

$LLG \Rightarrow FA$:

$$\underset{L}{\boxed{LLG}} \xRightarrow{\text{Reversal}} \underset{L^{Rev}}{RLG} \Rightarrow \underset{L^{Rev}}{FA} \xRightarrow{\text{Reversal}} \underset{L}{\boxed{FA}}$$

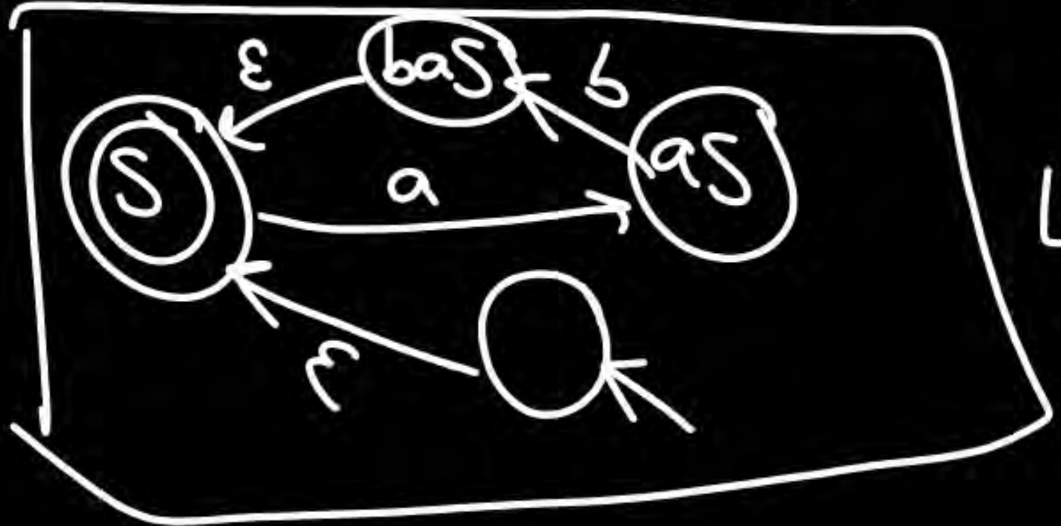
$$\boxed{S \rightarrow Sab \mid \varepsilon} \Rightarrow \boxed{S \rightarrow baS \mid \varepsilon} \Rightarrow$$

LLG



$L^{Rev} = (ba)^*$

⇓ Reversal



$L = (ab)^*$

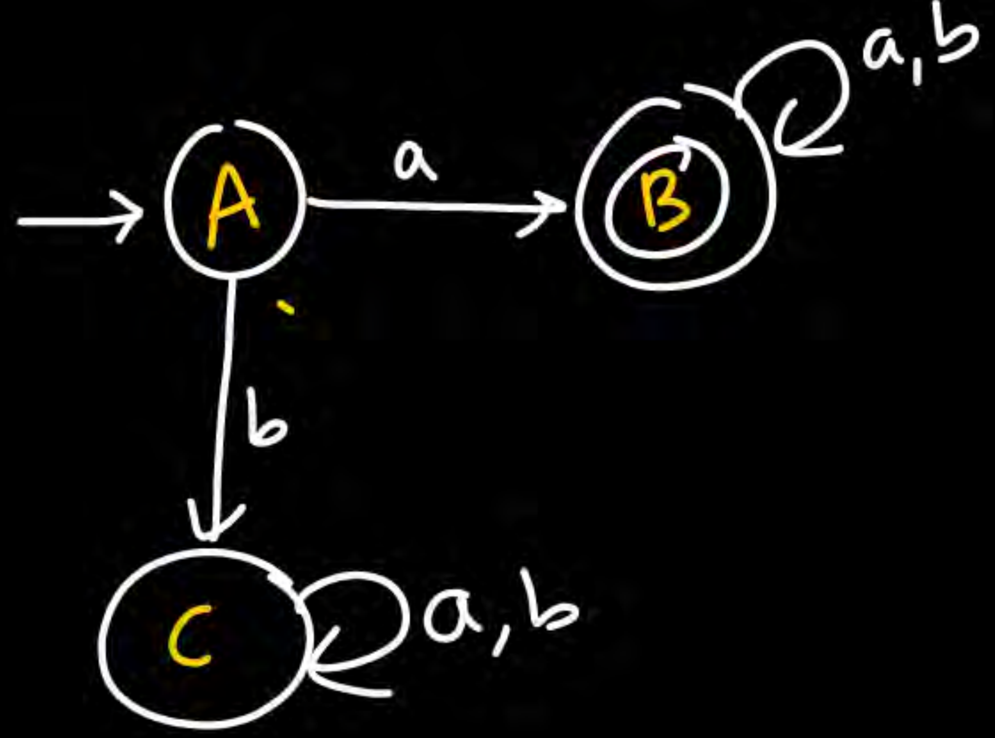
$L = (ab)^*$

$L^{Rev} = (ba)^*$

$$\boxed{\begin{array}{c} DFA \\ \cong \\ NFA \\ \cong \\ LLG \\ \cong \\ RLG \\ \cong \\ RegExp \end{array}}$$

$$\boxed{FA \cong RG \cong Reg\,Exp} \cong \text{Regular Language}$$

$L = a(a+b)^*$:



$[A] = \varepsilon$

$[B] = a(a+b)^*$

$[C] = b(a+b)^*$

$A \cap B = \phi$
$A \cap C = \phi$
$B \cap C = \phi$

$\boxed{A \cup B \cup C = \Sigma^*}$

How many equivalence classes are there for $L = a(a+b)^*$?

$= 3$

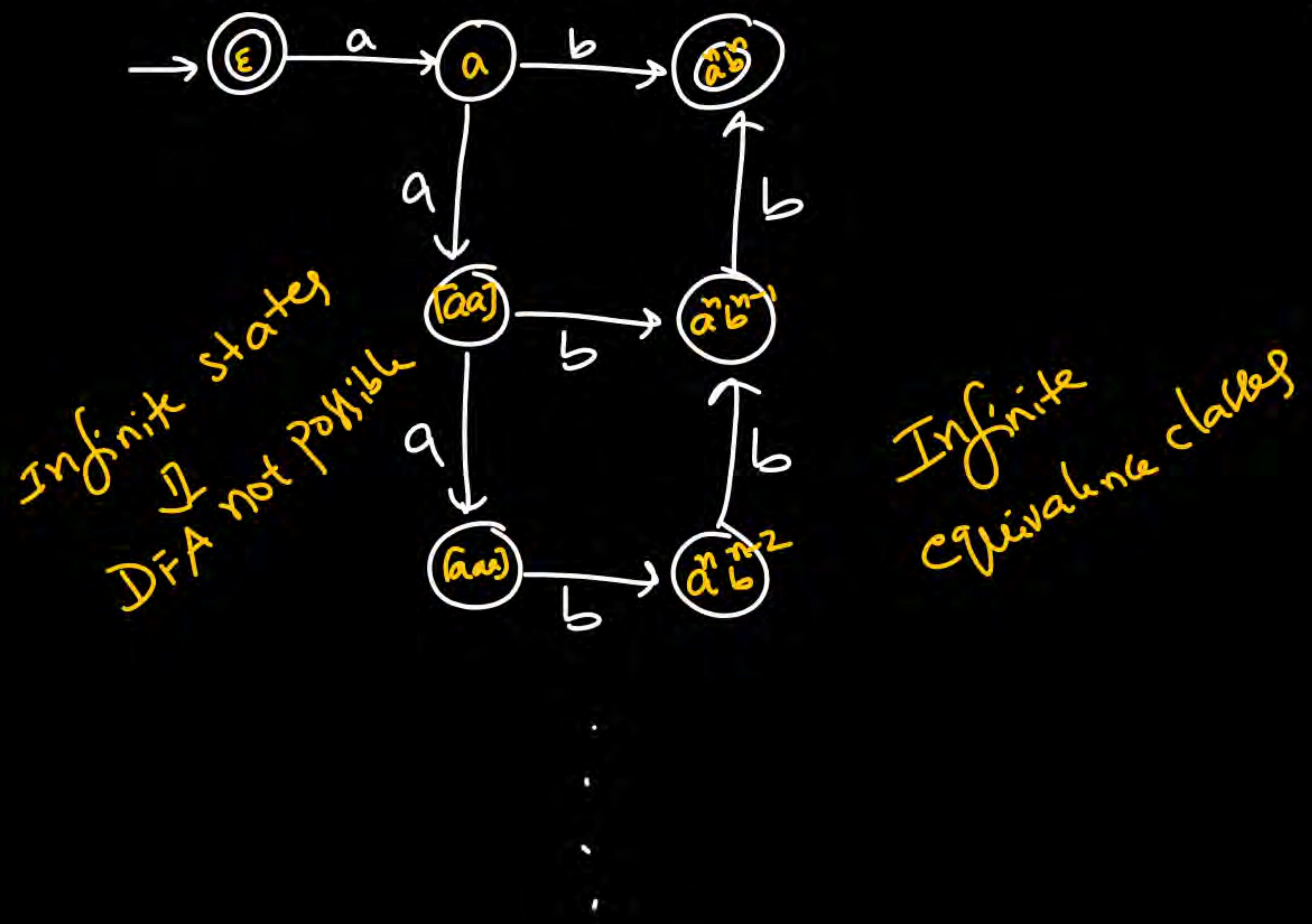
$$\boxed{\begin{gathered}\text{No. of states in min DFA that accepts } L \\ = \\ \text{No. of equivalence classes for } L\end{gathered}}$$

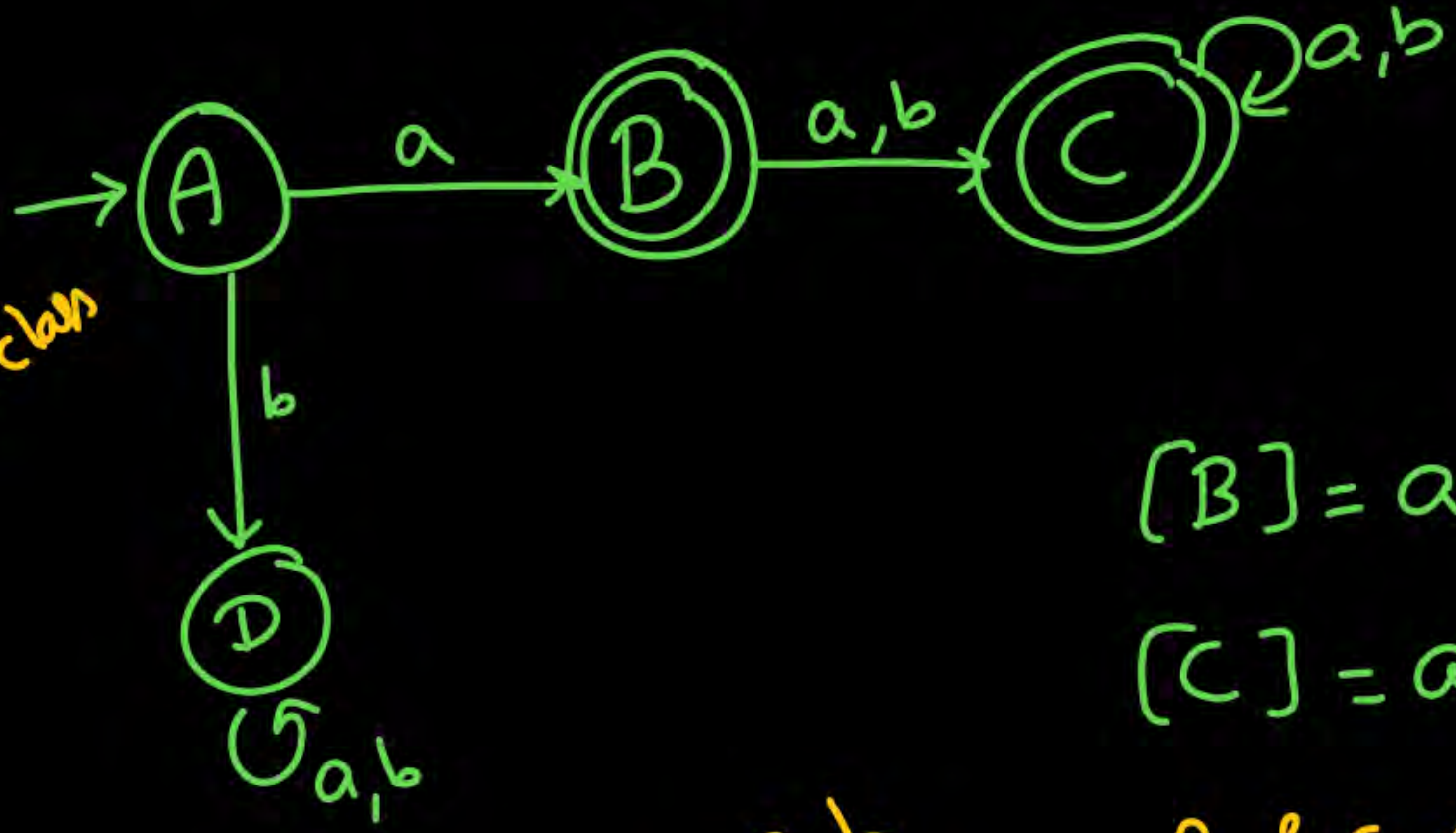
Every regular language has unique min DFA.

I) No. of equivalence classes for every regular language is finite

II) No. of equivalence classes for every non regular language is Infinite

$L = a^n b^n$


a
b
a
b
b
a
b
b
ε
a
aa
aaa
Infinite states
DFA not possible
Infinite equivalence classes



equivalent states so, we will make into 1 class

$\boxed{B \cong C}$

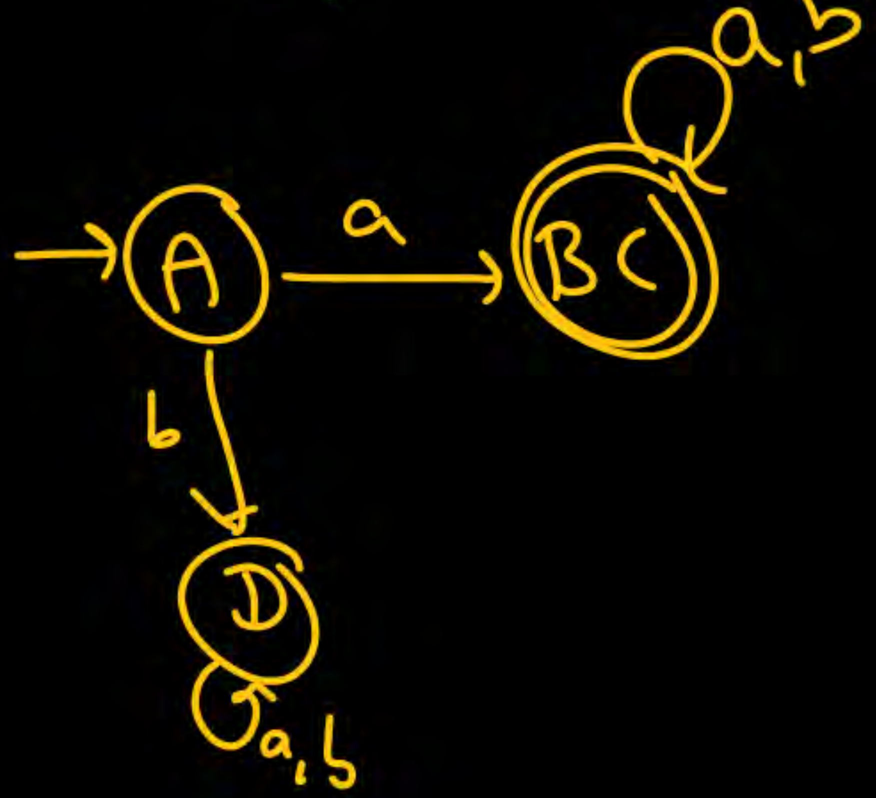
$A \ncong B$

$A \ncong C$

$A \ncong D$

$B \ncong D$

$C \ncong D$



$[B] = a$

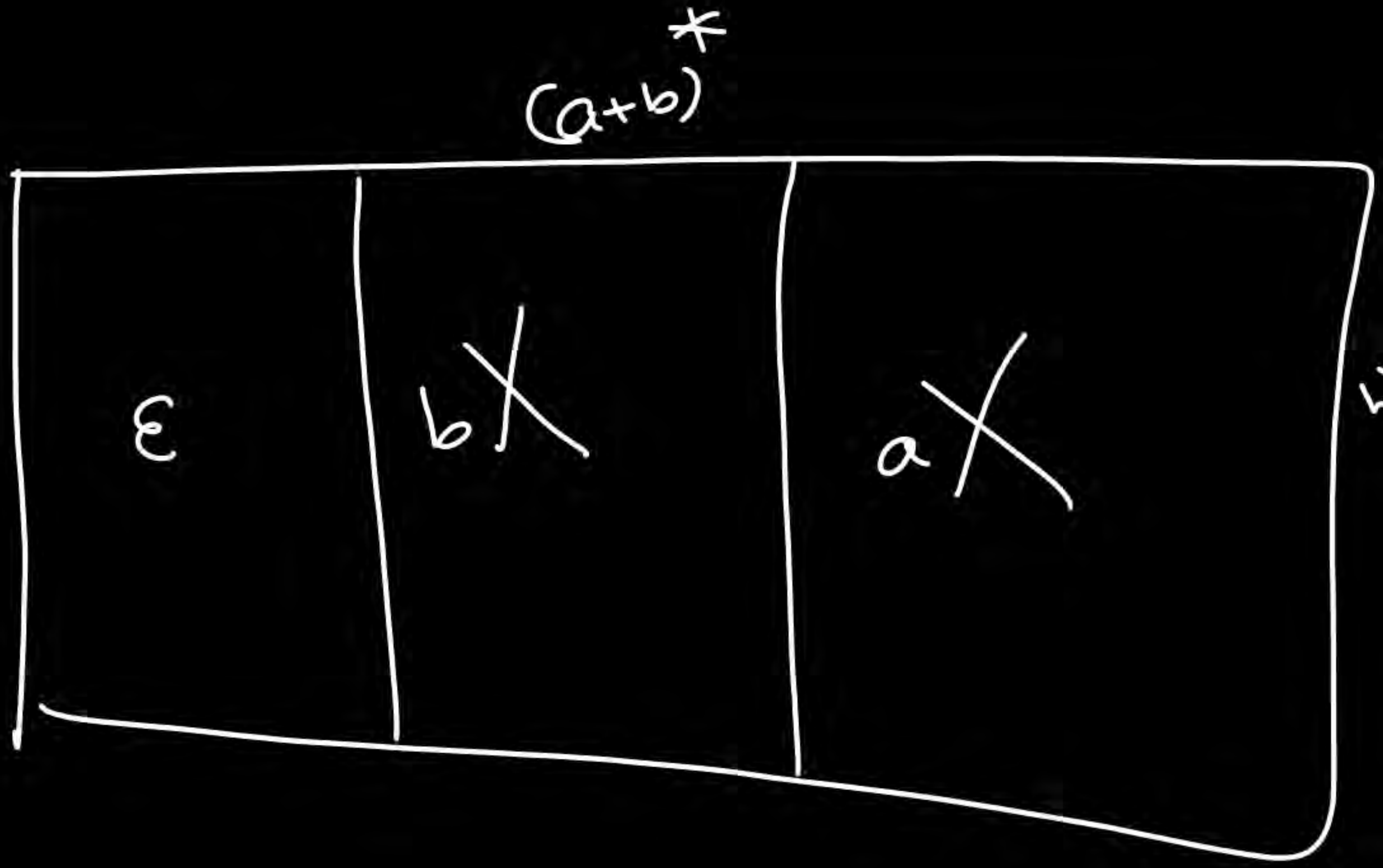
$[C] = a(a+b)(a+b)^*$

B & C are not two different equivalence class

B&C will be combined

$\boxed{B \cong C}$ using minimization Algo

$L = a(a+b)^*$



We need min DFA to understand no. of equivalence classes for reg lang.

Summary

Algorithms

$\hookrightarrow$ 99% Never apply in exam

[A] $\quad w_1 s \in L$

[B] $\quad w_2 s \notin L$

Thank you