Computer Science

Theory of Computation

Regular Languages &
Non Regular Languages

Lecture No.- 5

Mallesham Devasane Sir

Recap of Previous Lecture

Topic Regular Grammar

Topics to be Covered

- Topic: NFA to DFA Conversions
- Topic: Regular Expression to FA Conversion
- Topic: FA to Regular Expression Conversion
- Topic: FA to Regular Grammar Conversion
- Topic: Regular Grammar to FA Conversion
- Topic: FA, RG and Reg exp comparison

Conversion

Regular Language

$\equiv$

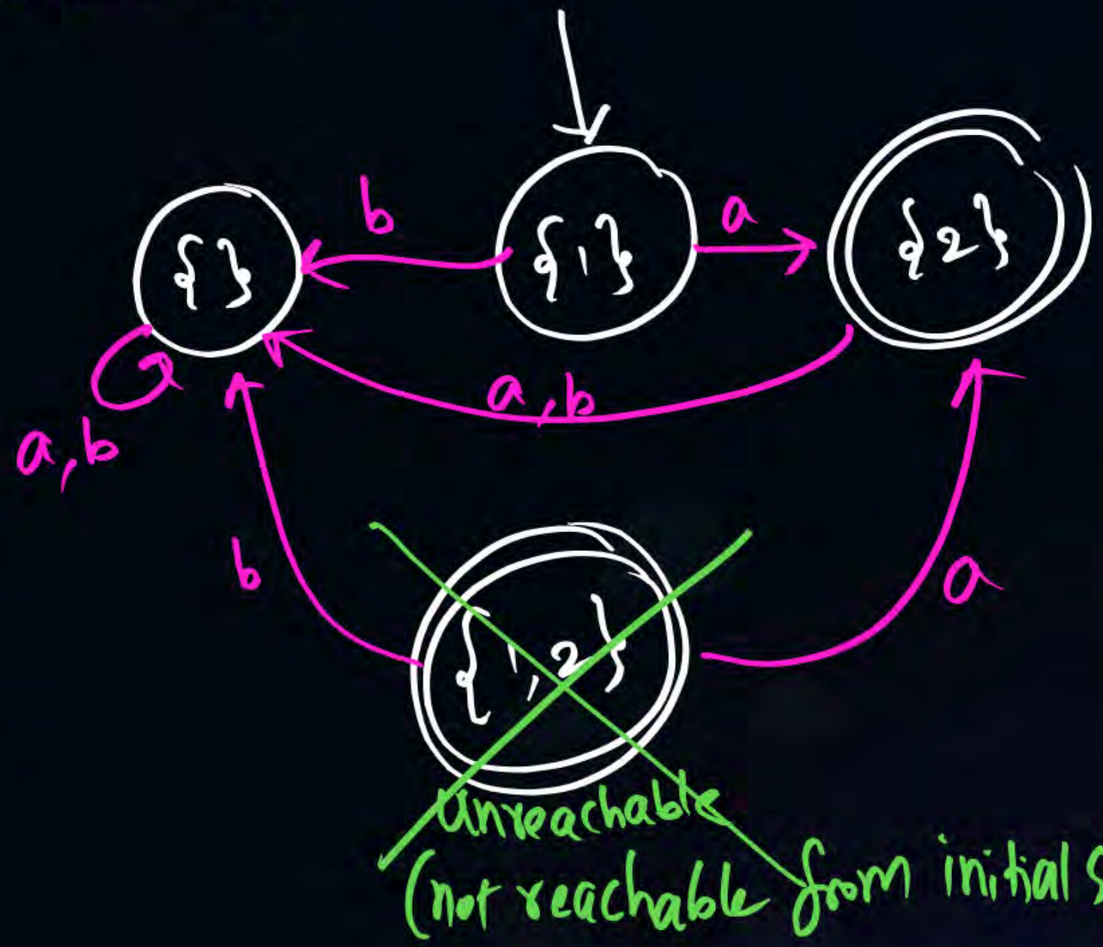
$\equiv$

Regular Exp $\equiv$ DFA $\equiv$ NFA $\equiv$ LLG $\equiv$ RLG $\equiv$ RG $\equiv$ FA

Conversion

I) NFA to DFA : [Subset Construction]

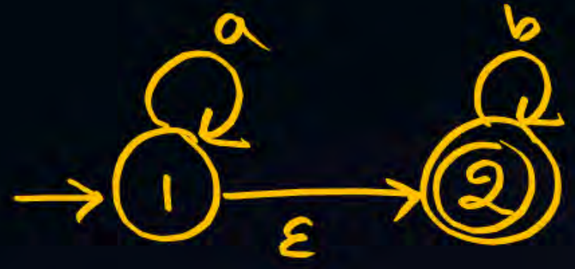
$\Sigma = \{a, b\}$



Conversion

II) NFA to DFA [Subset Construction]

(with ε moves)



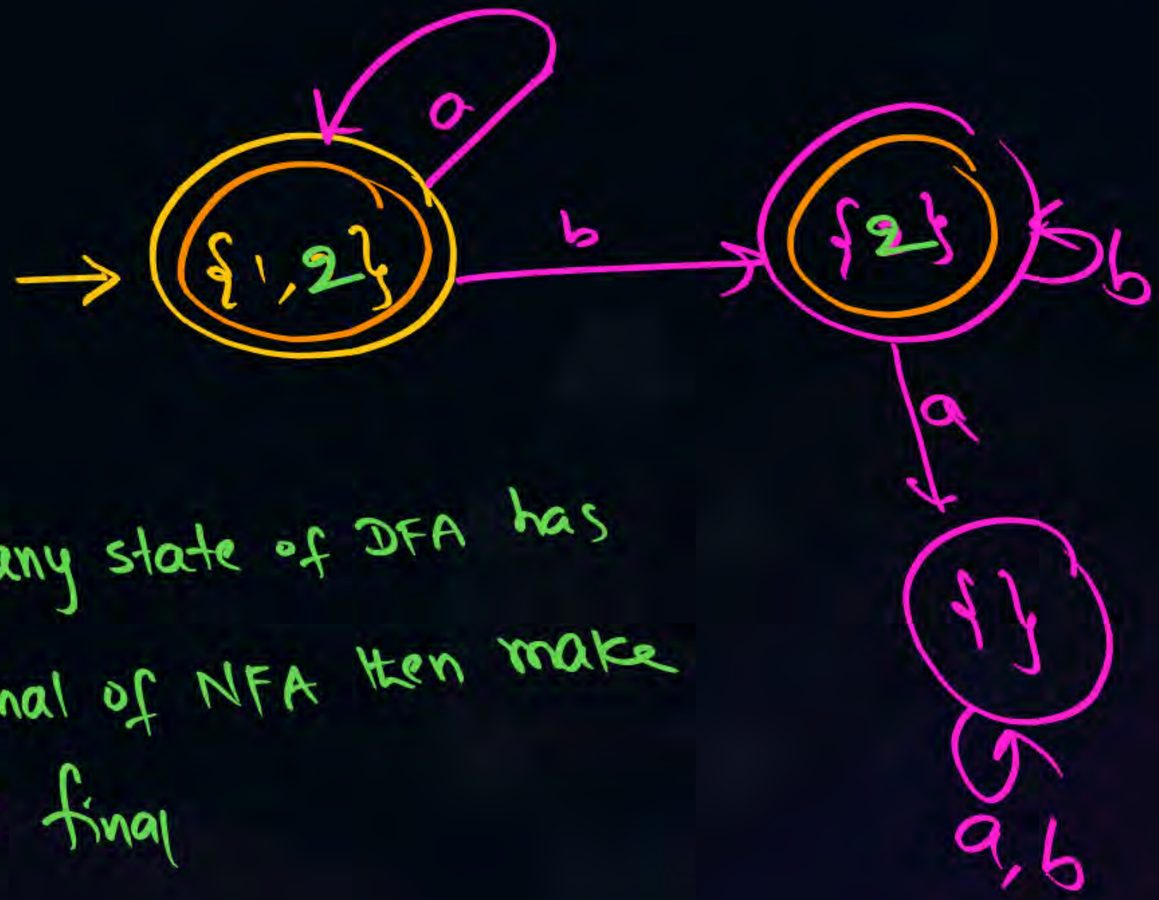
$\delta(1,a) = \{1\}$ $\quad \varepsilon\text{-clo}(1) = \{1,2\}$

$\delta(1,b) = \phi$ $\quad \varepsilon\text{-clo}(2) = \{2\}$

$\delta(2,a) = \phi$

$\delta(2,b) = \{2\}$

Initial state of DFA = $\varepsilon\text{-clo}(1) = \{1,2\}$



If any state of DFA has final of NFA then make it final

δ NFA without ε moves $\Longrightarrow$ δ' DFA :

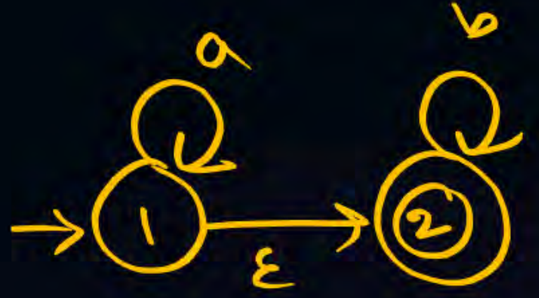
$$\delta'(\{p,q\}, i) = \delta(p,i) \cup \delta(q,i)$$

δ NFA with ε moves $\Longrightarrow$ δ' DFA

$$\delta'(p, i) = \varepsilon\text{-clo}\left(\delta(p,i)\right)$$

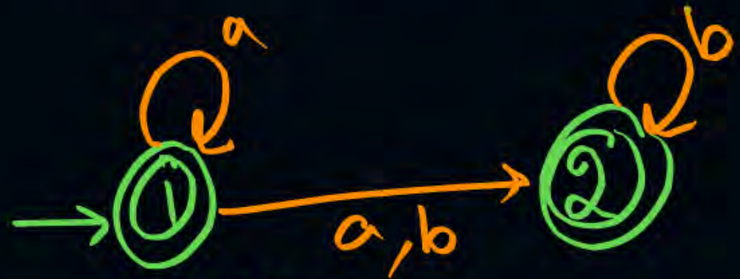
$$\delta'(\{p,q\}, i) = \varepsilon\text{-clo}\left(\delta(p,i) \cup \delta(q,i)\right)$$

III) NFA with ε moves ⇒ NFA without ε moves



$\delta(1,a)=\{1\}$
$\delta(1,b)=\phi$
$\delta(2,a)=\phi$
$\delta(2,b)=\{2\}$

$\varepsilon\text{-clo}(1)=\{1,2\}$
$\varepsilon\text{-clo}(2)=\{2\}$



$\delta'(1,a)=\{1,2\}$
$\delta'(1,b)=\{2\}$
$\delta'(2,a)=\phi$
$\delta'(2,b)=\{2\}$

$2 \in \varepsilon\text{-clo}(1)$ — final
$2 \in \varepsilon\text{-clo}(2)$

$$\delta'(1,a) = \varepsilon\text{-clo}\left[\delta(\varepsilon\text{-clo}(1), a)\right]$$

$$= \varepsilon\text{-clo}\left[\delta(\{1,2\}, a)\right]$$

$$= \varepsilon\text{-clo}\left[\delta(1,a) \cup \delta(2,a)\right]$$

$$= \varepsilon\text{-clo}(1) = \{1,2\}$$

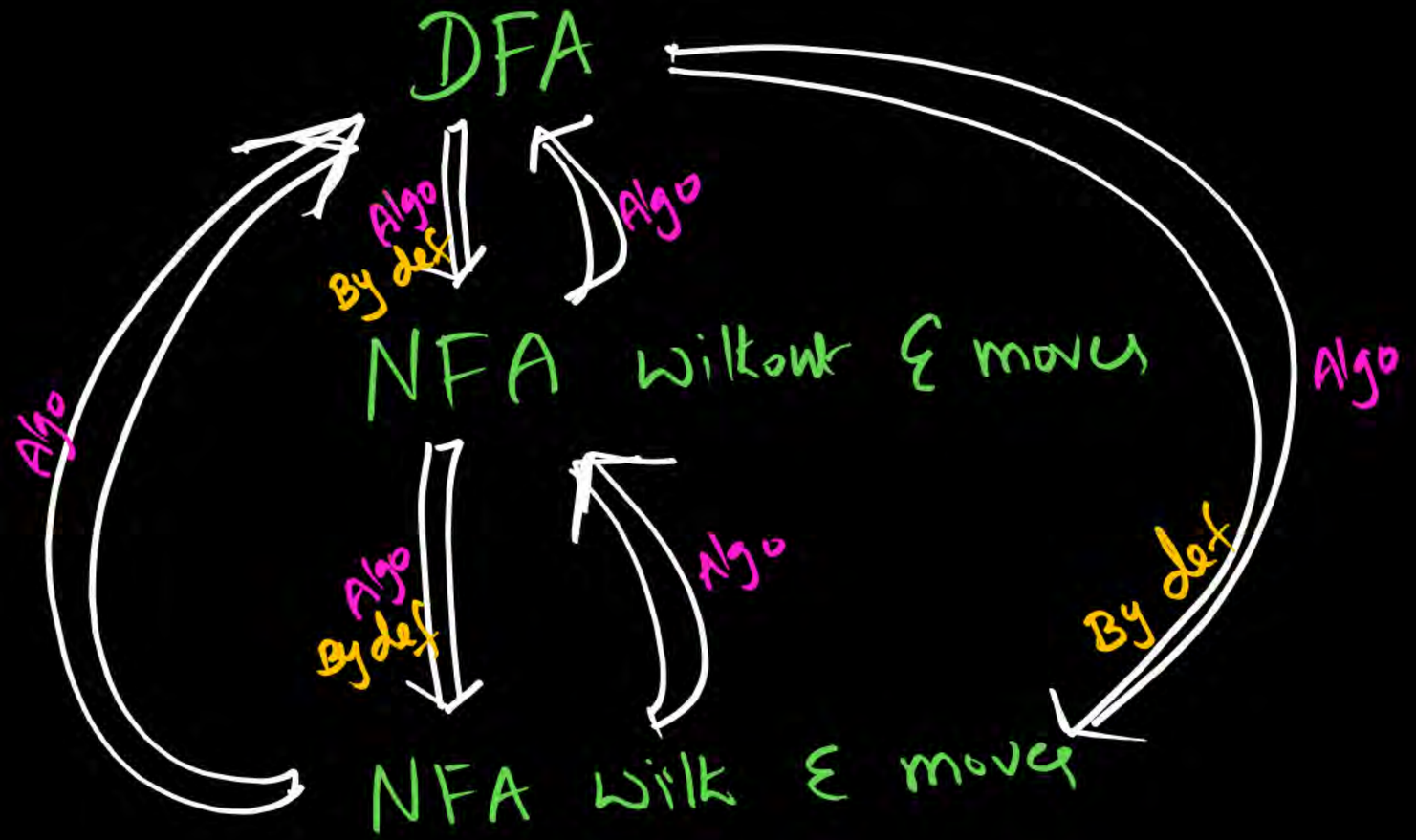
NFA
with ε moves
(δ)
Given

$\Longrightarrow$

NFA
without ε moves
(δ')

$$\delta'(p, i) = \varepsilon\text{-clo}\left[\delta\left(\varepsilon\text{-clo}(p), i\right)\right]$$

1st
2nd
3rd

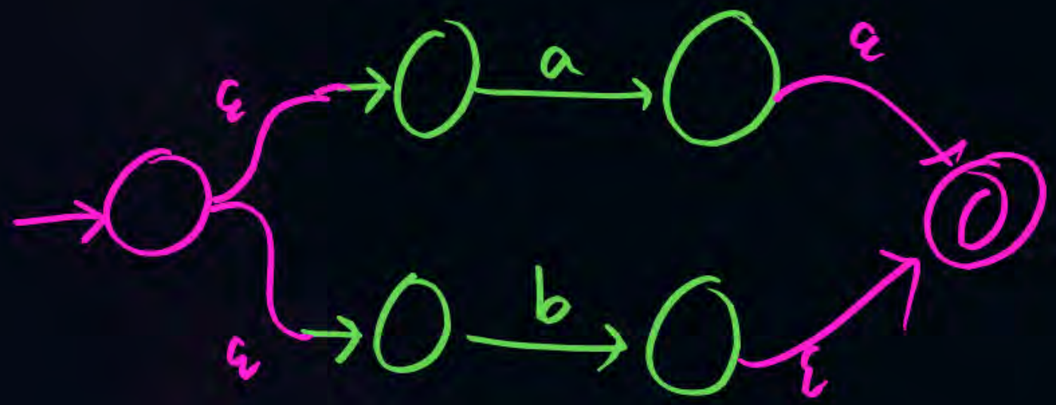

DFA
Algo
By def
Algo
NFA without ε moves
Algo
Algo
By def
Algo
By def
NFA with ε moves

NFA $\cong$ DFA

IV) Reg Exp ⟹ FA [Induction Method]

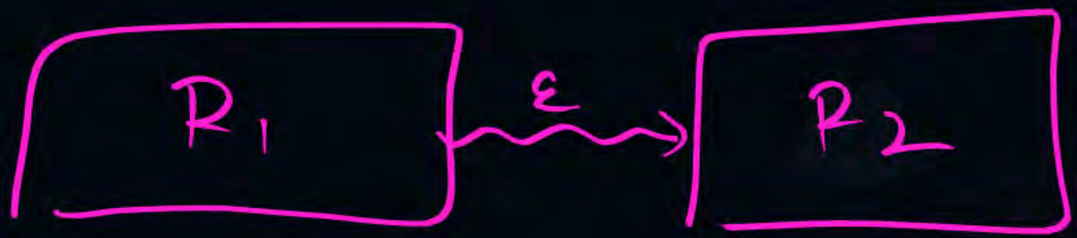
1) a

2) a+b



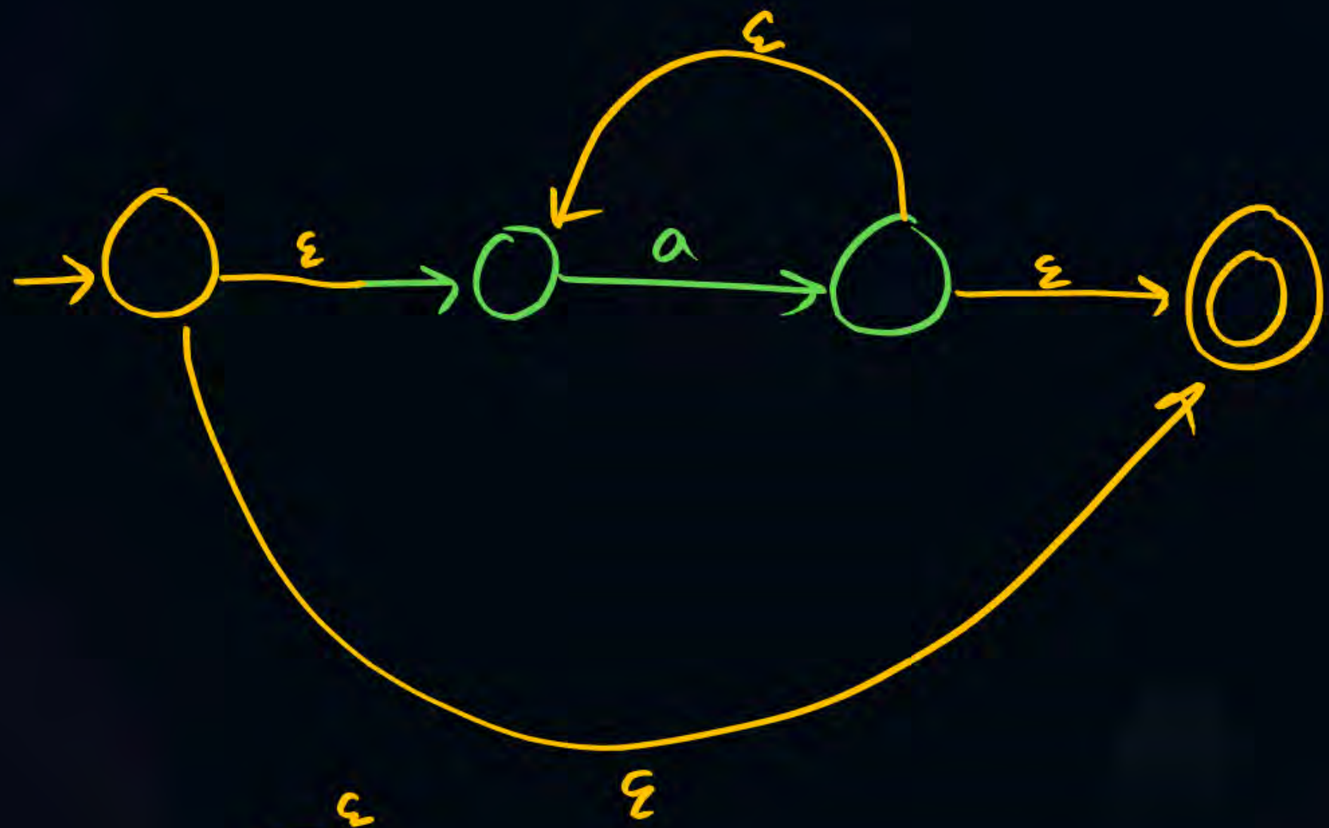
3) $a.b$

$R_1.R_2$

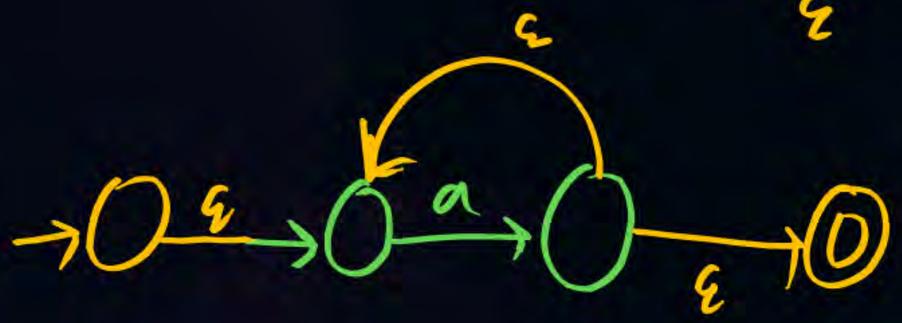


Conversion

④ a^*

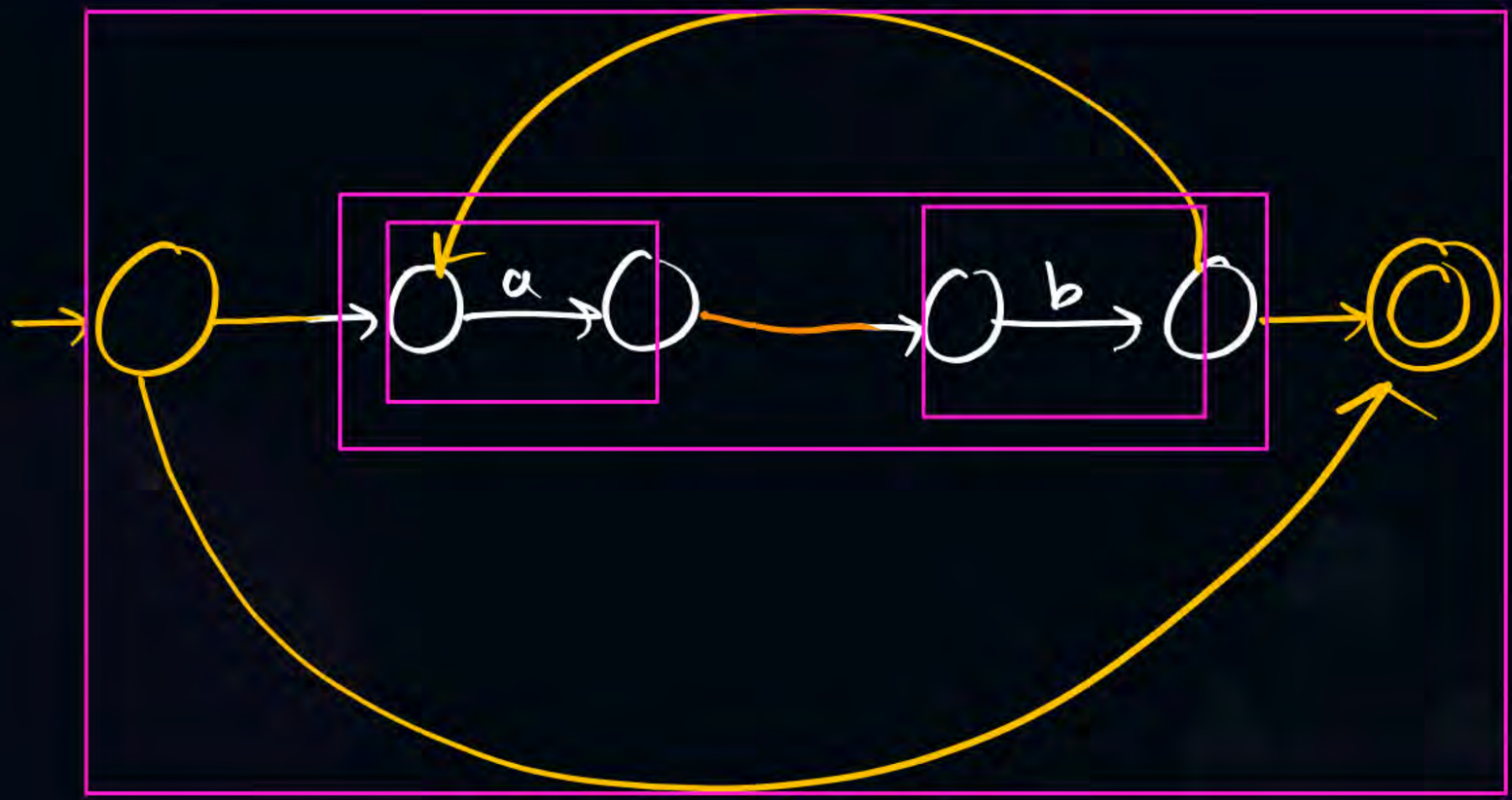


⑤ a^+



Conversion

$R = (ab)^*$



a ✓

b ✓

ab ✓

$(ab)^*$ ✓

Conversion

$$R = (ab + aaa)^* a + ba$$

a ✓

b ✓

ab ✓

aaa ✓

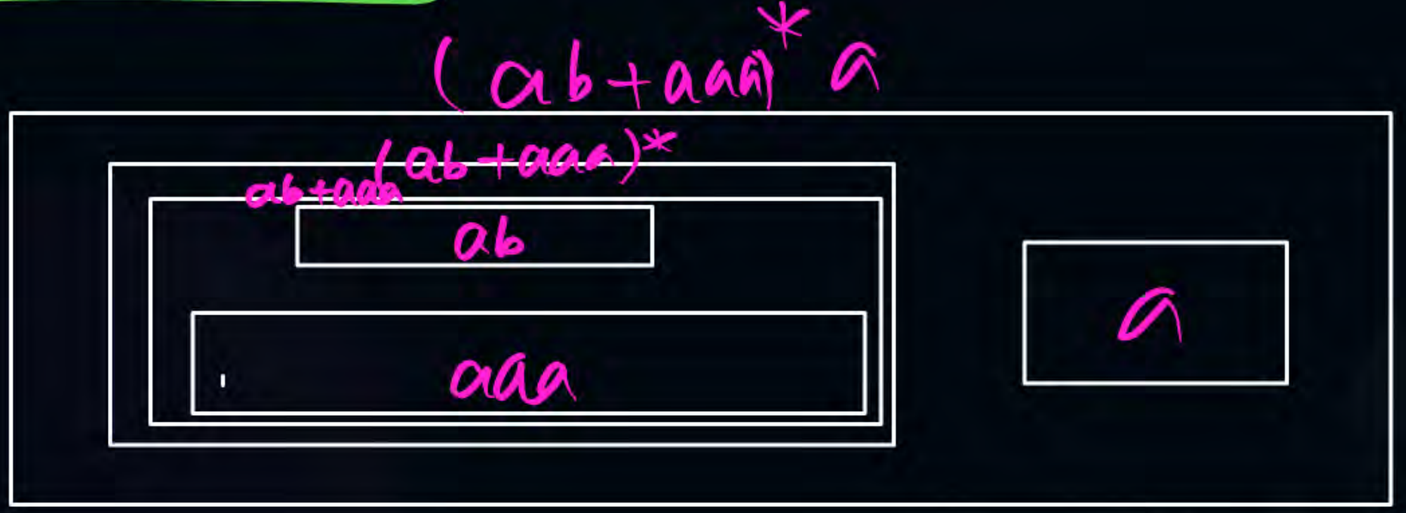
ba ✓

ab+aaa ✓

$(ab+aaa)^*$ ✓

$(ab+aaa)^* . a$ ✓

$(ab+aaa)^* a + ba$ ✓



Conversion

$$R = (ab + aaa)^* a + ba$$

a ✓

b ✓

ab ✓

aaa ✓

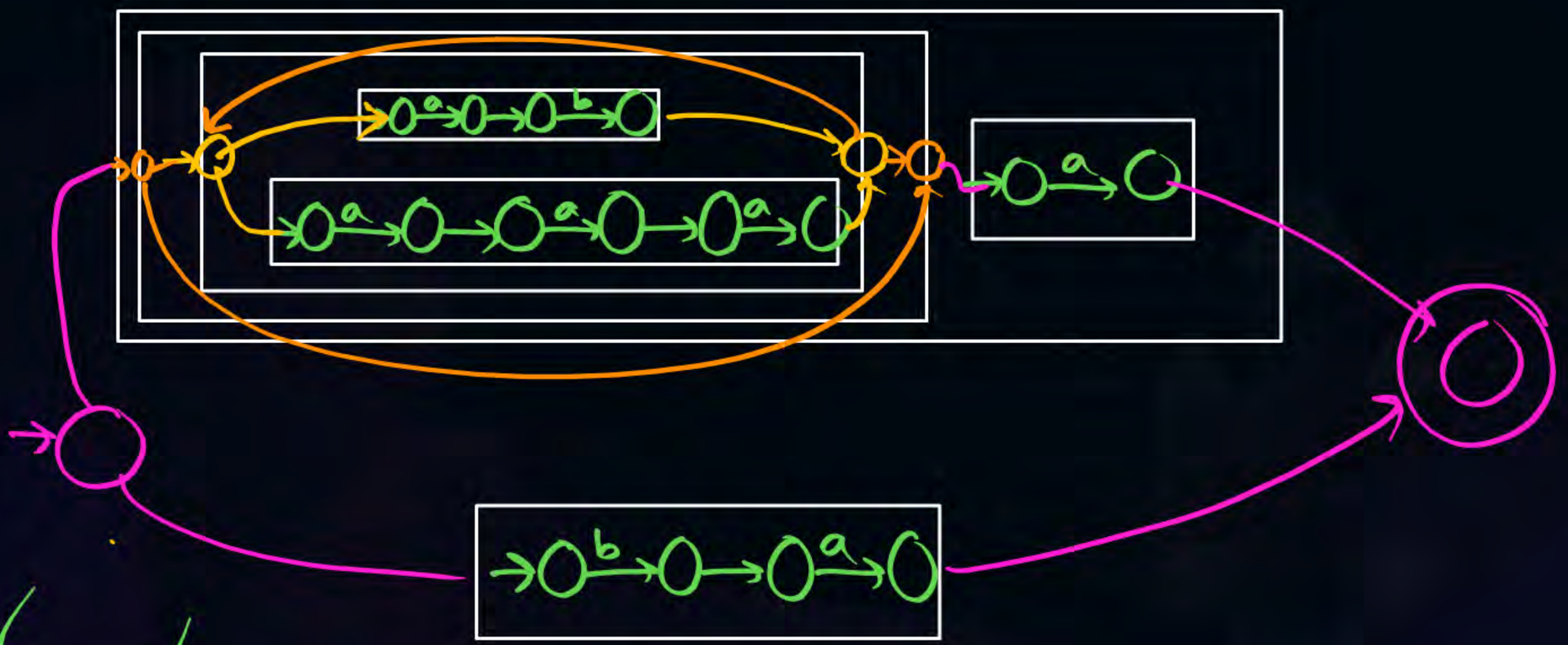
ba ✓

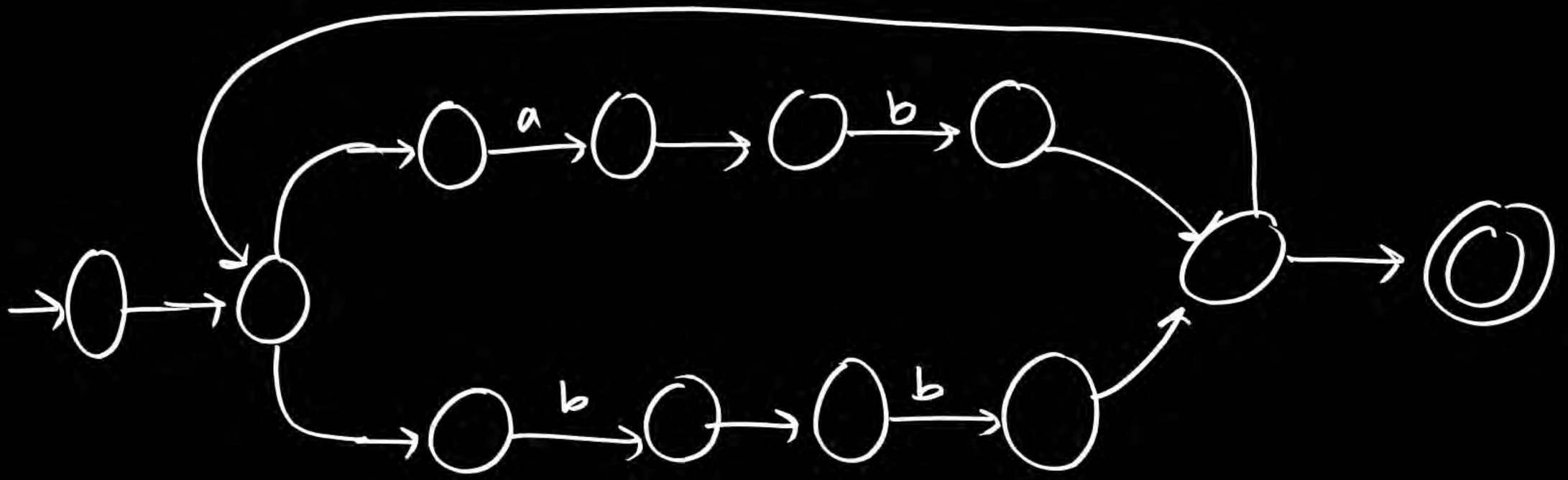
ab+aaa ✓

$(ab+aaa)^*$ ✓

$(ab+aaa)^* . a$ ✓

$(ab+aaa)^* a + ba$ ✓





$$L = (ab + bb)^{+}$$

V) FA $\Longrightarrow$ Reg Exp

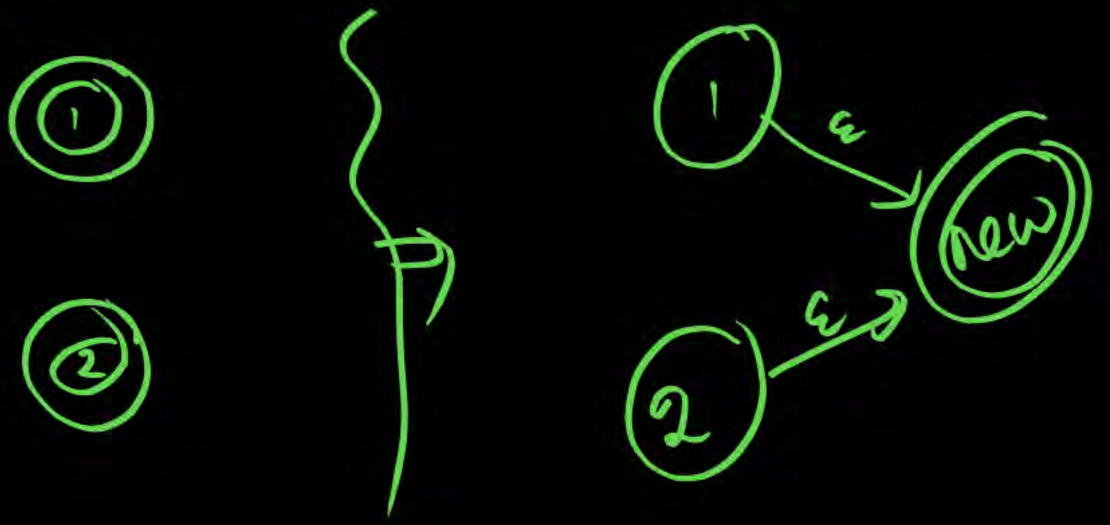
Algo 1 : State elimination method

Algo 2 : Arden's method

Algo 3 : Kleene method

State elimination Method:

i) If more than one final state then convert into one final.



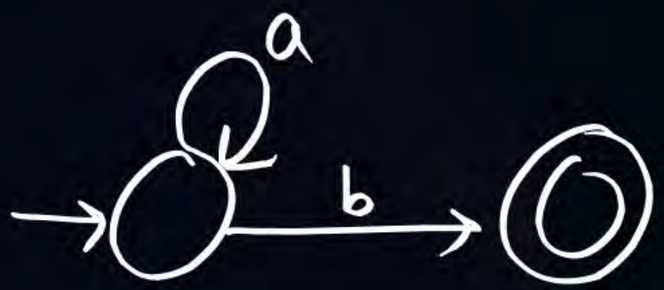
ii) Delete all states except initial and final using elimination method.

Conversion FA ⇒ Reg Exp

1)

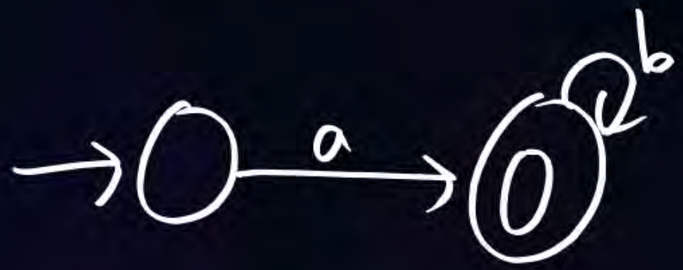
$R = a$

2)



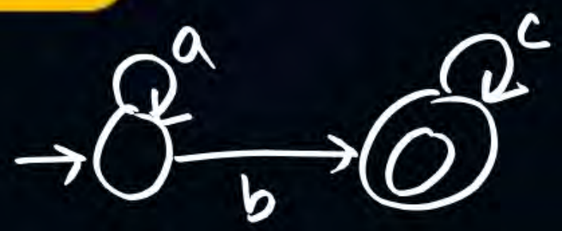
$R = a^* b$

3)



$R = ab^*$

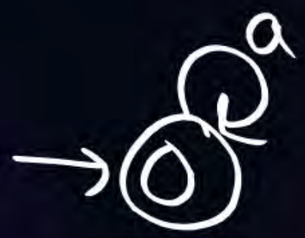
Conversion

4) 

$R = a^* b c^*$

5) 

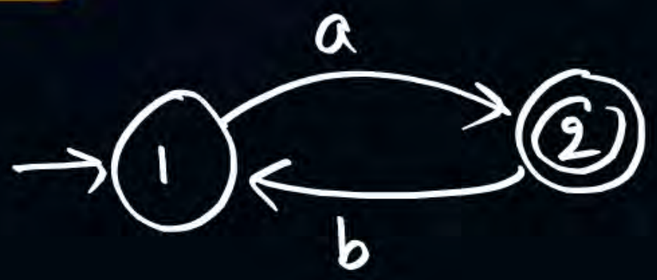
$R = \varepsilon$

6) 

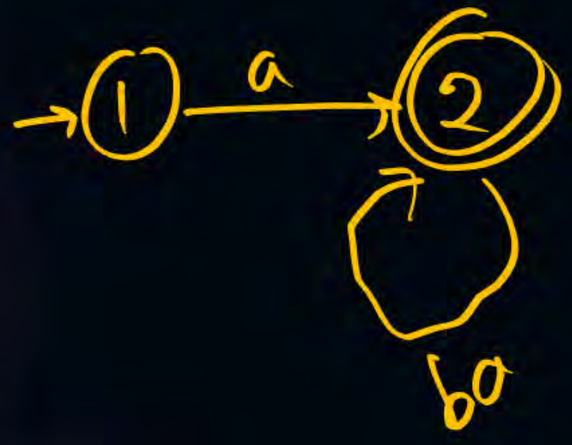
$R = a^*$

Conversion

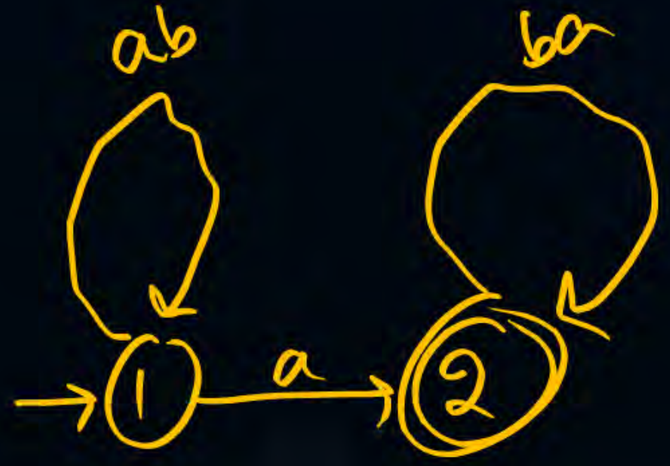
*** 7)



$R = (ab)^* a$

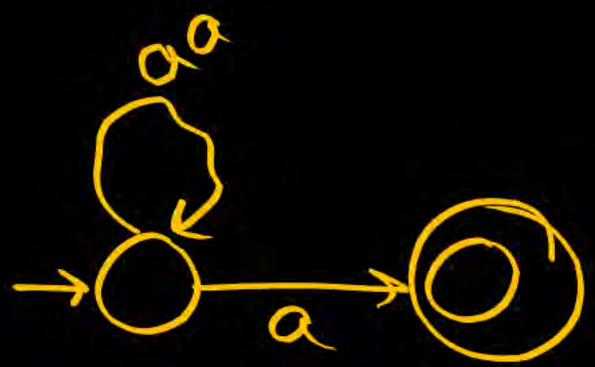


$R = a(ba)^*$

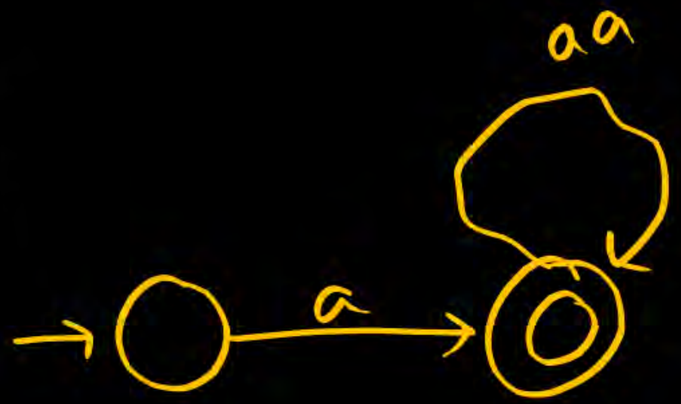


$R = (ab)^* a (ba)^*$

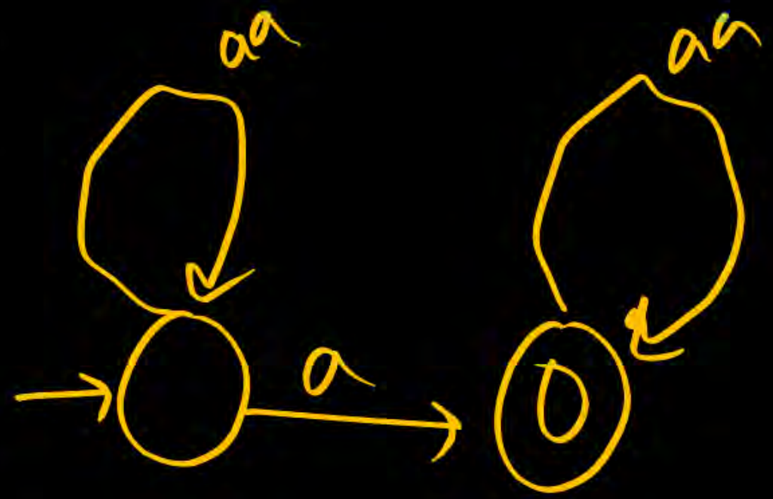
8)



$(aa)^*a$

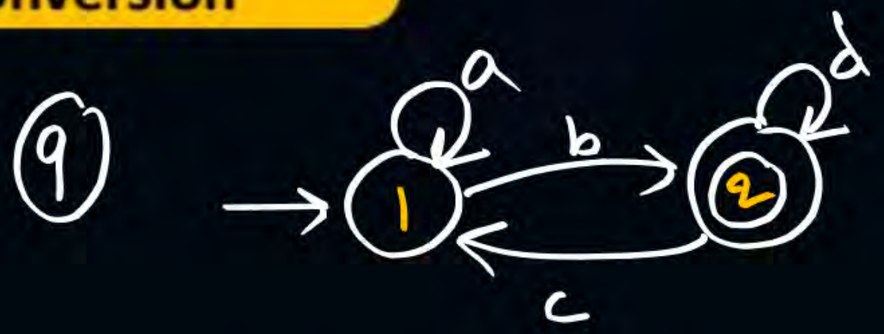


$a(aa)^*$

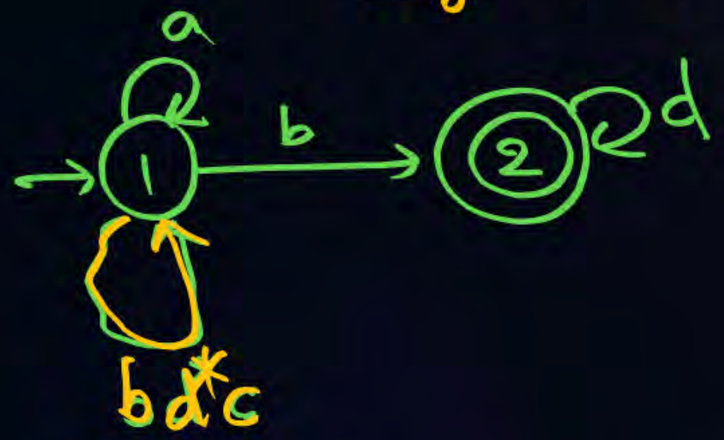


$(aa)^*a(aa)^*$

Conversion

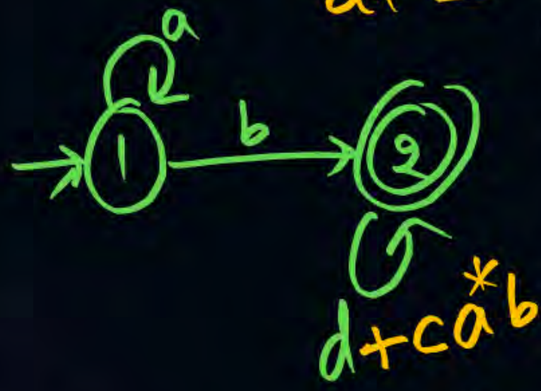


Delete cycle by covering self loop at 1



$$(a+bd^*c)^* b d^*$$

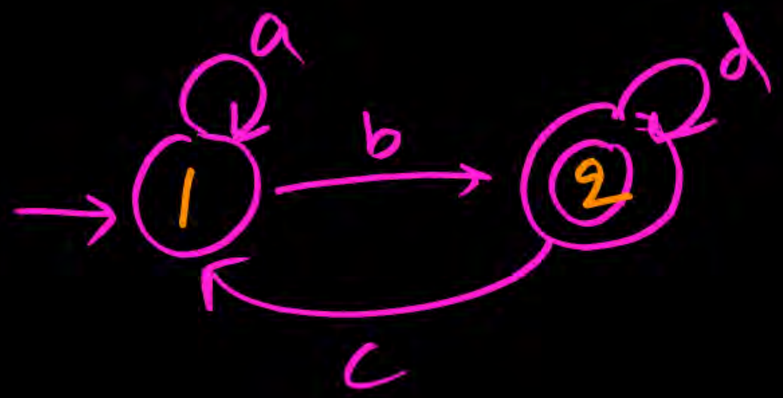
Delete cycle by covering self loop at 2

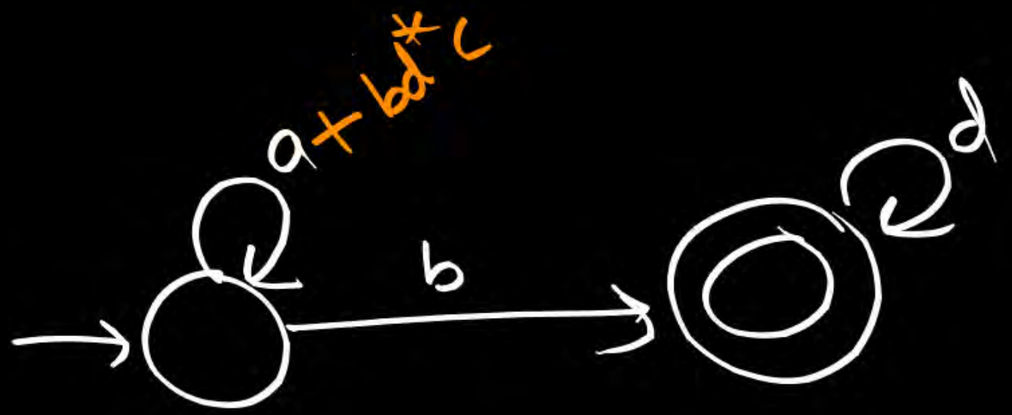


$$= a^* b (d+ca^*b)^*$$

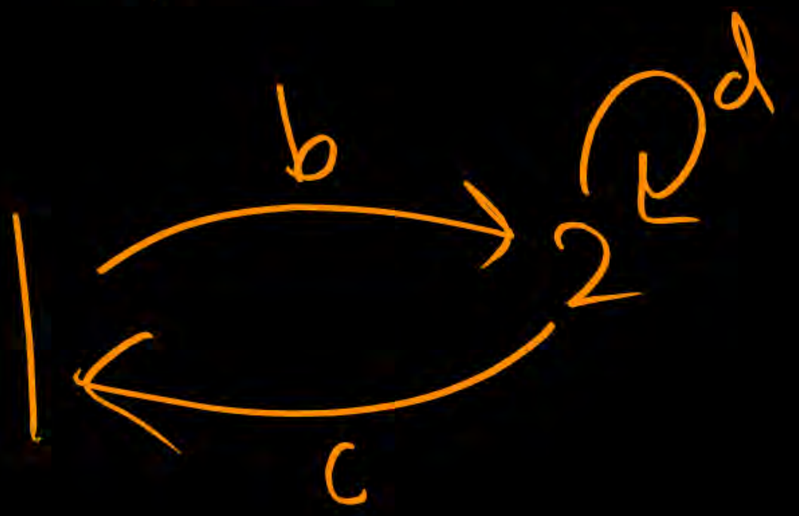
Cover self loop at both 1 & 2

$$(a+bd^*c)^* b (d+ca^*b)^*$$

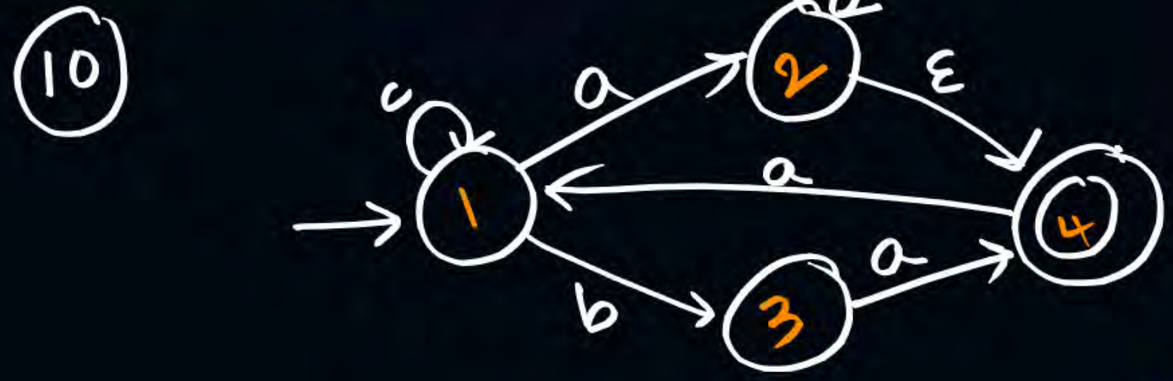




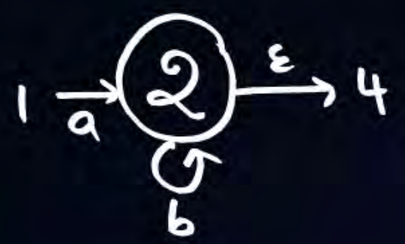
$$(a+bd^*c)^* b\, d^*$$

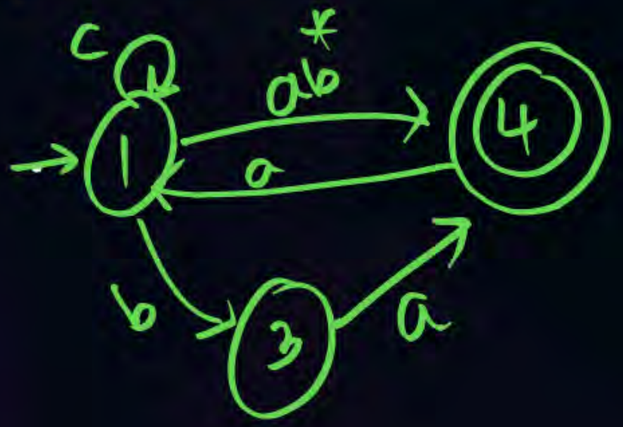


Conversion

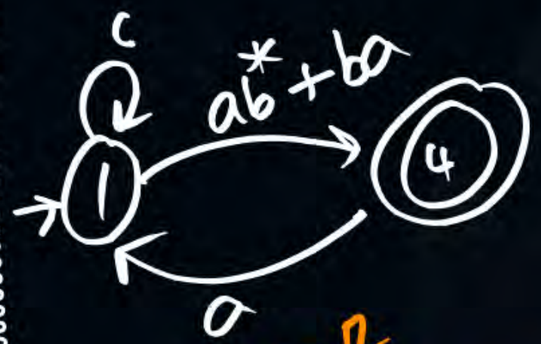


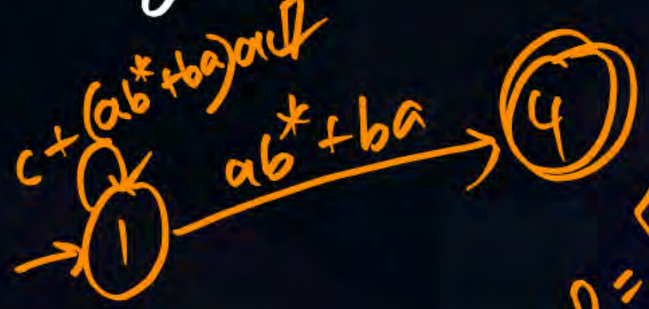
Step1: Delete state 2





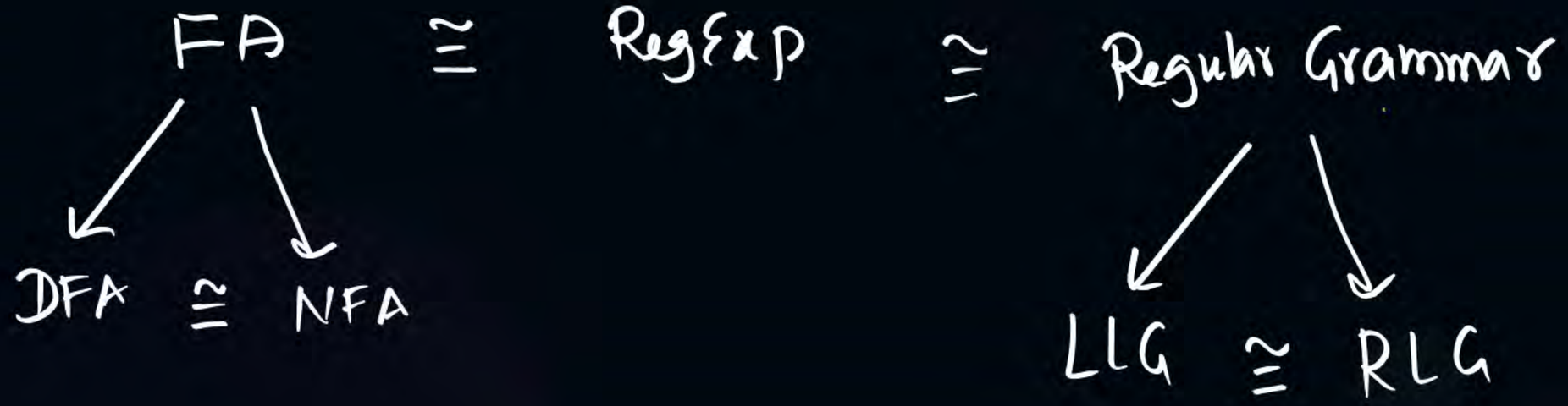
Step2: Delete state 3





$$R = [c+(ab^*+ba)a^*](ab^*+ba)$$

Conversion



Conversion

Algo : DFA ⇒ NFA ✓

Algo : NFA ⇒ DFA ✓

Algo : FA ⇒ RegExp ✓

Algo : RegExp ⇒ FA ✓

Algo : LLG ⇒ RLG ✓

Algo : RLG ⇒ LLG ✓

Algo : LLG ⇒ FA ✓

Algo : FA ⇒ LLG ✓

Algo : RLG ⇒ FA ✓

Algo : FA ⇒ RLG

2 mins Summary

Topic	NFA to DFA Conversions
Topic	Regular Expression to FA Conversion
Topic	FA to Regular Expression Conversion
Topic	FA to Regular Grammar Conversion
Topic	Regular Grammar to FA Conversion
Topic	FA, RG and Reg exp comparison

THANK - YOU