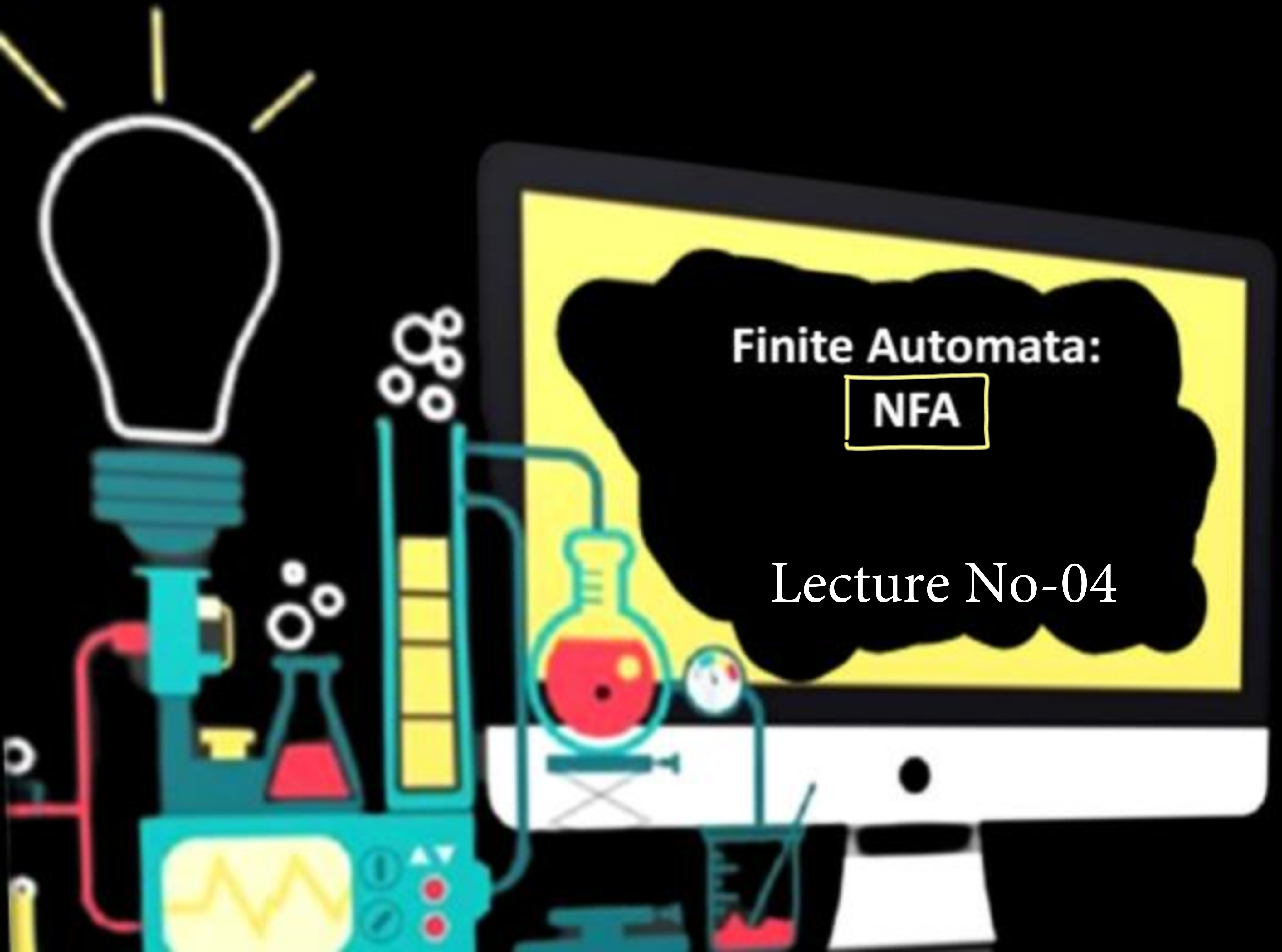
CS & IT Engineering



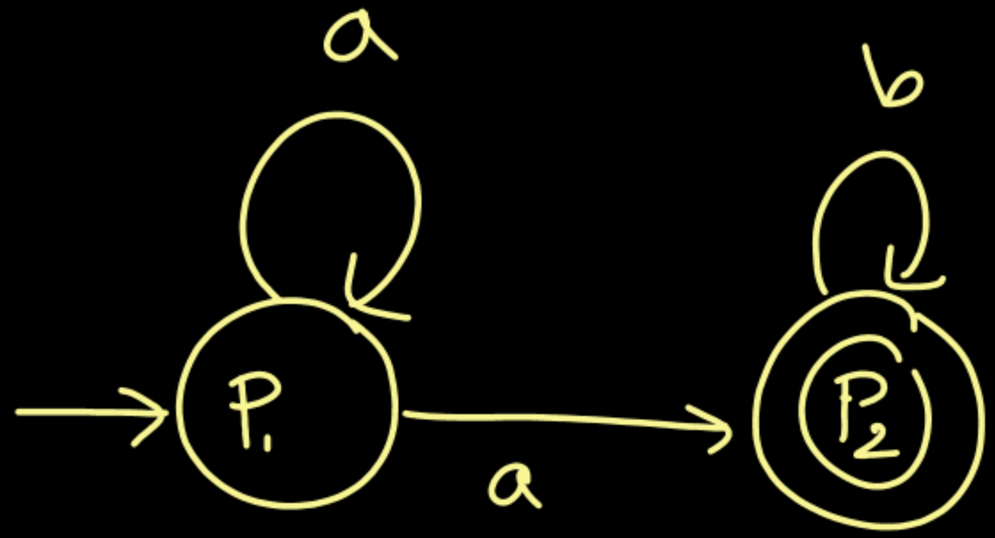
Deva sir

Topics:

1. NFA without ε moves ✓
2. Construct NFA ✓
3. Conversion from NFA to DFA ✓
4. DFA Minimization Algorithm (to minimize DFA)
5. NFA Vs DFA

$$\delta_{NFA \text{ without } \varepsilon \text{ moves}} : \quad Q \times \Sigma \to 2^Q$$

$$Q \times \Sigma \to \text{Powerset}(Q)$$



$Q = \{P_1, P_2\}$

$2^Q = \{\{\}, \{P_1\}, \{P_2\}, \{P_1, P_2\}\}$

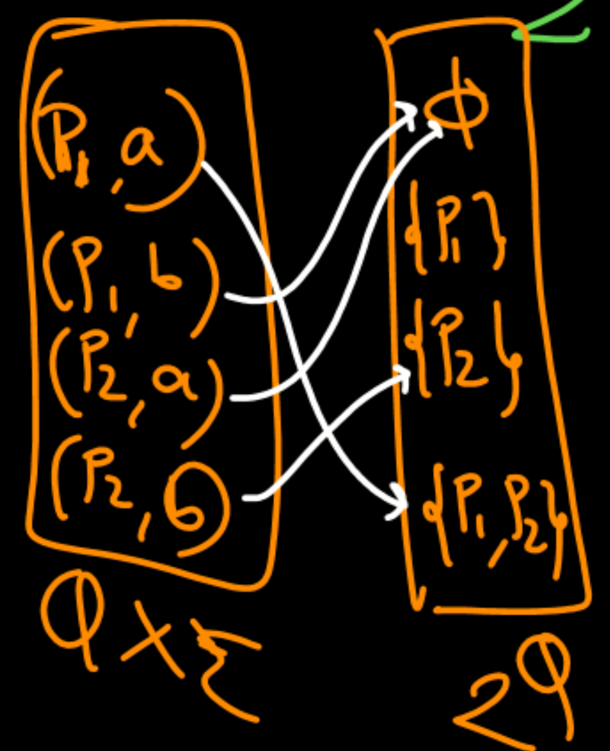
$= \{\phi, \{P_1\}, \{P_2\}, Q\}$

$\delta(P_1, a) = \{P_1, P_2\}$

$\delta(P_1, b) = \{\ \}$

$\delta(P_2, a) = \{\ \}$

$\delta(P_2, b) = \{P_2\}$



$w \longrightarrow \boxed{\text{DFA}}$

Accepts L

If $w \in L$, exactly 1 path exist, halts at final

If $w \notin L$, exactly 1 path exist, halts at nonfinal

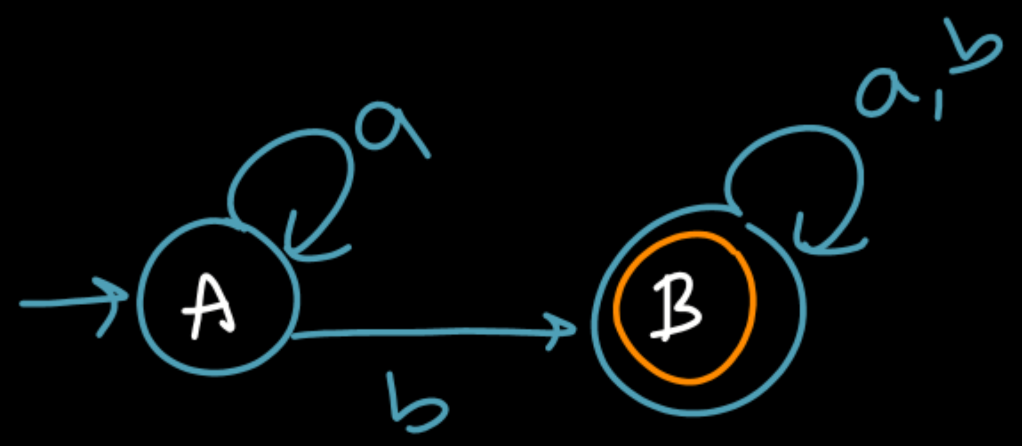
$w \longrightarrow \boxed{\text{NFA}}$

Accepts L

If $w \in L$, atleast 1 path exist, some path halts at final

If $w \notin L$, either "no path exist" or "every path halts at non final"

DFA

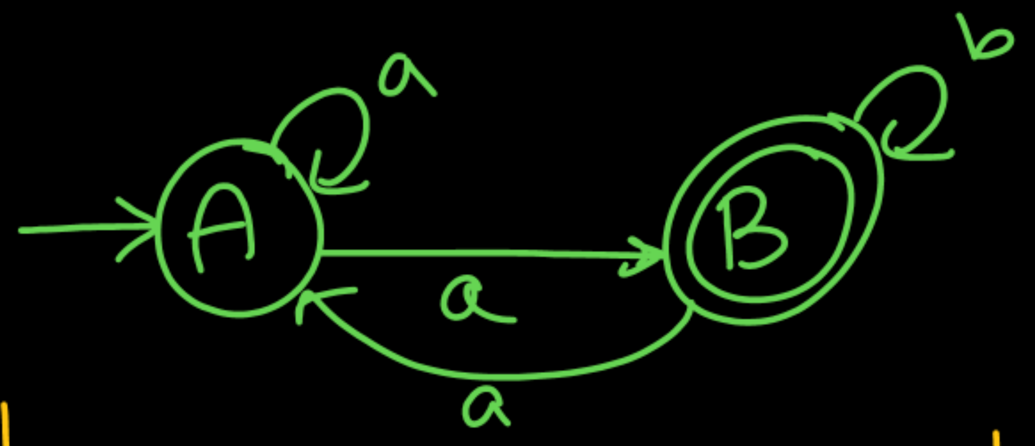


No. of paths for :

	path	no of paths
× 1) ε	A	1
× 2) a	$A \xrightarrow{a} A$	1
✓ 3) b	$A \xrightarrow{b} B$	1
× 4) aa	$A \xrightarrow{a} A \xrightarrow{a} A$	1
✓ 5) ab	$A \xrightarrow{a} A \xrightarrow{b} B$	1

⋮

NFA
(non-deterministic FA)

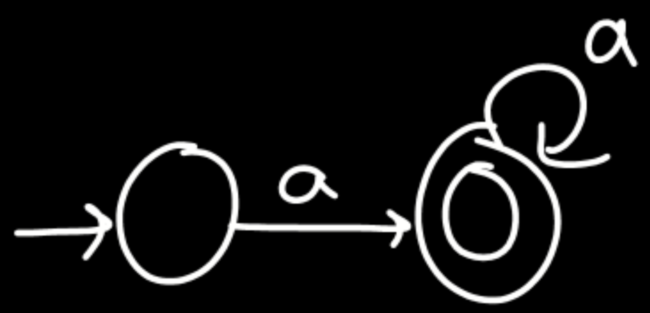


String	Path	No. of paths	valid/invalid
ε	A	1	Invalid
a	① $A \xrightarrow{a} A$ ② $A \xrightarrow{a} B$ (final)	2	valid
b	A → no path	0	Invalid
aa	① $A \xrightarrow{a} A \xrightarrow{a} A$ ② $A \xrightarrow{a} A \xrightarrow{a} \boxed{B}$ ③ $A \xrightarrow{a} B \xrightarrow{a} A$	3	Valid

DFA	NFA
① For every string, No. of paths = 1	① For every string, No. of paths ≥ 0
② For valid string, No. of paths = 1	② For valid string, No. of paths ≥ 1
③ For Invalid string, No. of Paths = 1	③ For Invalid string, No. of Paths ≥ 0
④ For every combination of state and input symbol, No. of transitions = 1	④ For every combination of state and i/p symbol, No. of transitions ≥ 0

DFA

(5) No. of minimum DFAs $= 1$



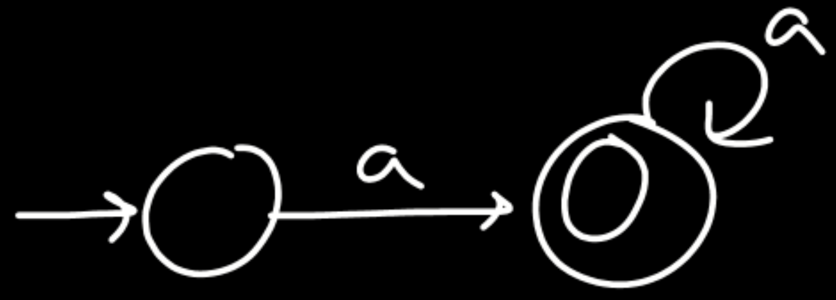
$\Sigma = \{a\}$

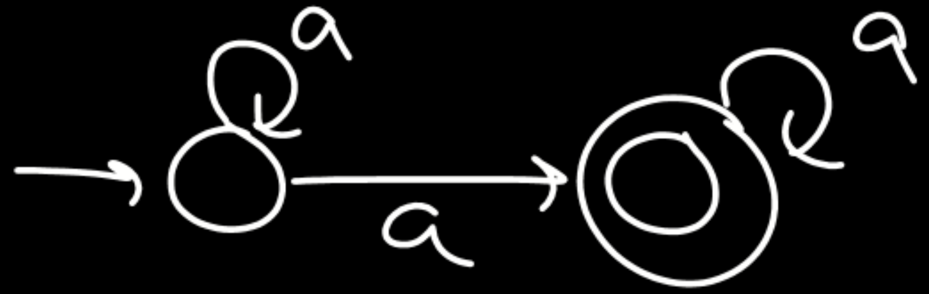
Ex $L = a^+$

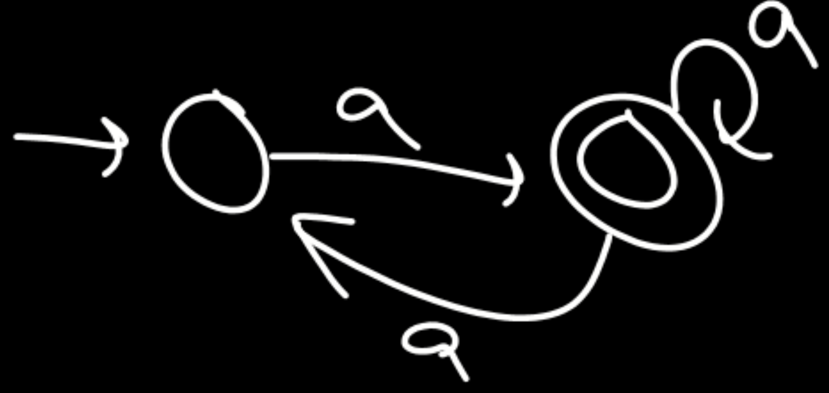
a ✓
a^2 ✓
a^3 ✓

NFA

(5) No. of min NFAs ≥ 1

i)


ii)


iii)


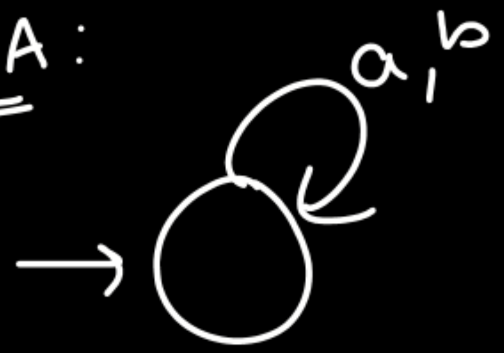
⋮

all are min NFAs

NFA construction:

① $L = \phi$ over $\Sigma = \{a, b\}$

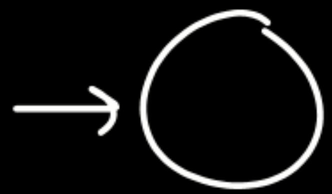
DFA:

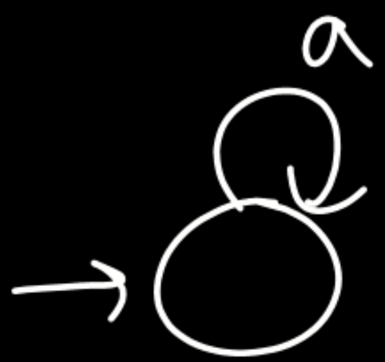


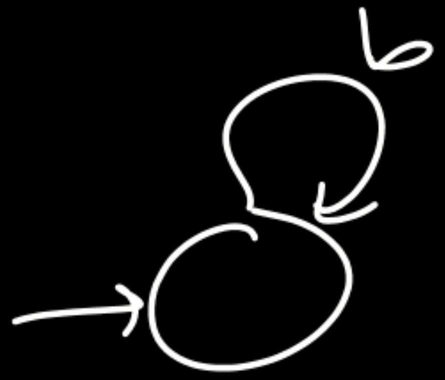
1 min DFA

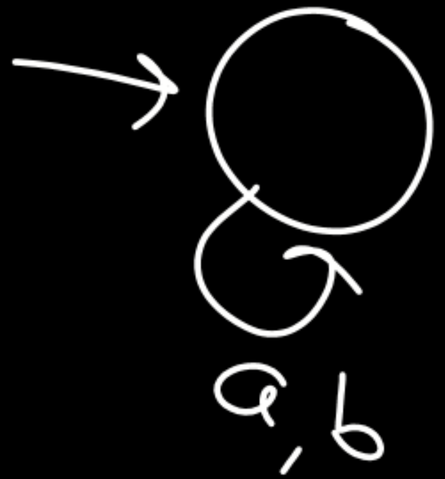
1 state

NFA:









4 min NFAs

1 state

Note:

1) Every DFA is NFA.

2) Min DFA need not be min NFA

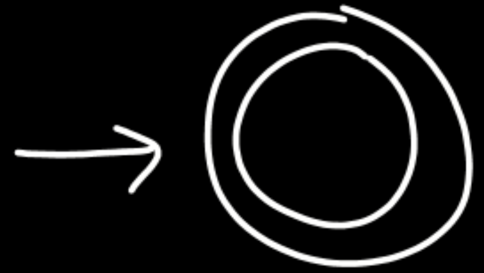
② $L = \{\varepsilon\}$ over $\Sigma = \{a, b\}$

DFA

It is NFA
but not min NFA

min DFA

= 2 states

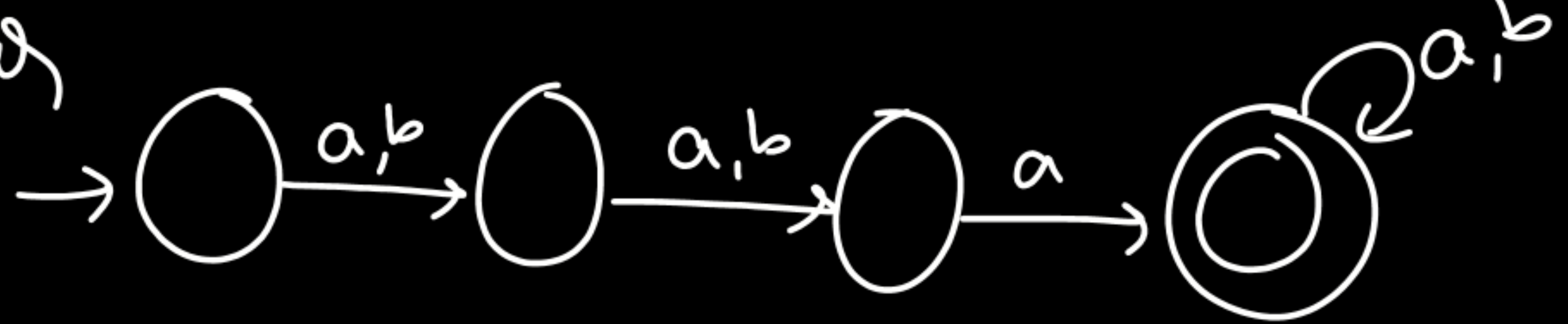


Min NFA

= 1 state

(3) $L = (a+b)^2 \underline{a} (a+b)^*$ [3rd symbol from begin is 'a']

= 4 states

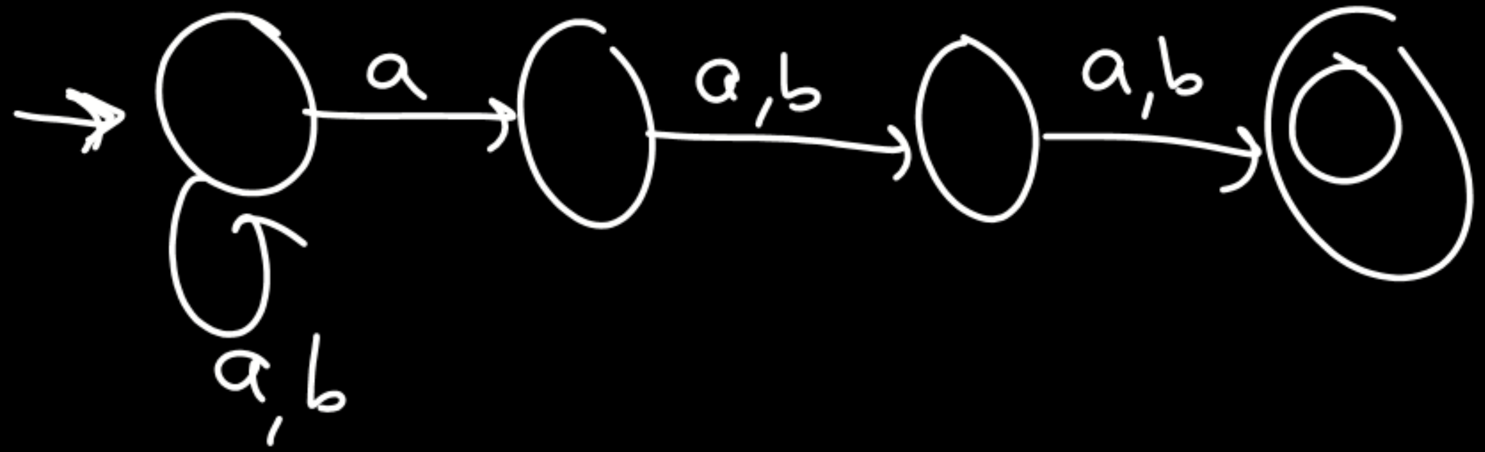


In DFA:

$3+2 = 5$ states

(4) $L = (a+b)^* a (a+b)^2$ [3rd symbol from end is 'a']

= 4 states

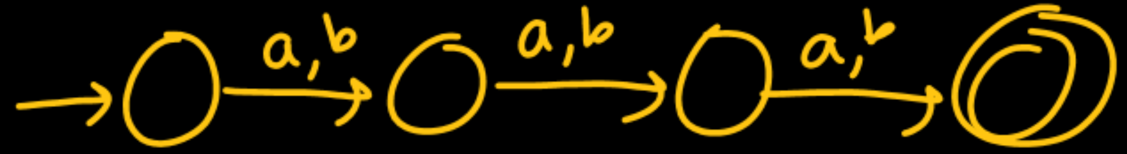


In DFA:

$2^3 = 8$ states

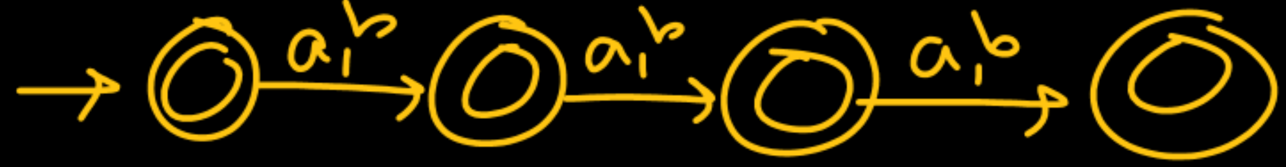
(5) $\{w \mid w \in \{a,b\}^*, |w| = 3\}$

4 states



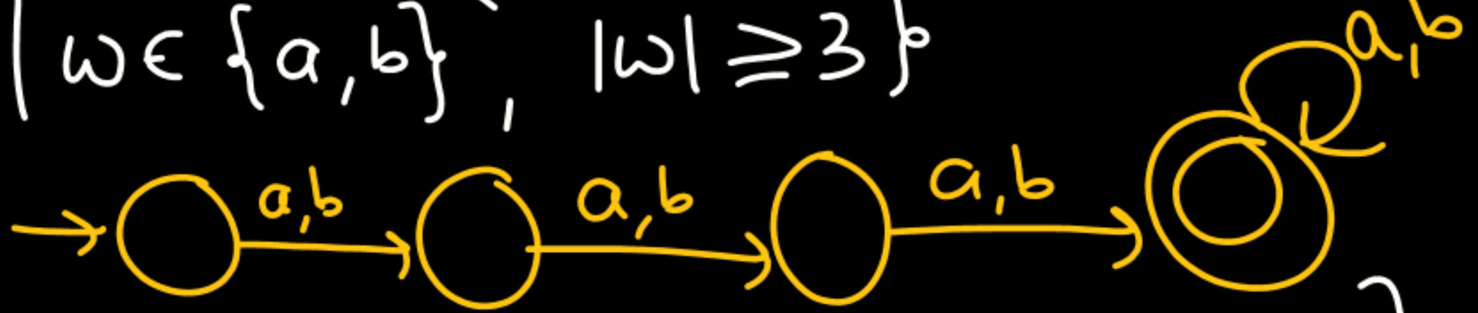
(6) $\{w \mid w \in \{a,b\}^*, |w| \leq 3\}$

4 States



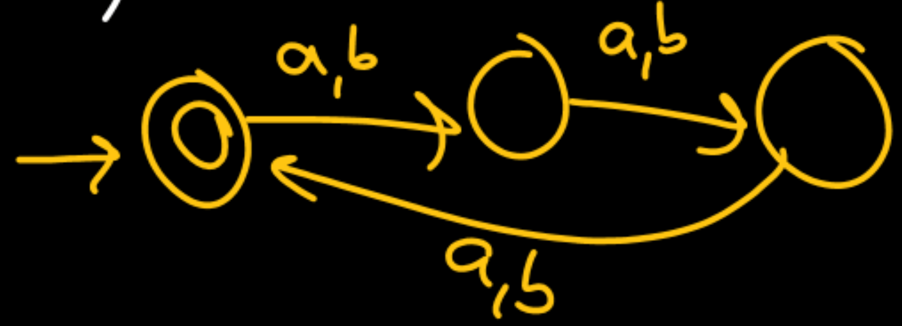
(7) $\{w \mid w \in \{a,b\}^*, |w| \geq 3\}$

4 states



(8) $\{w \mid w \in \{a,b\}^*, |w| \text{ is divisible by } 3\}$

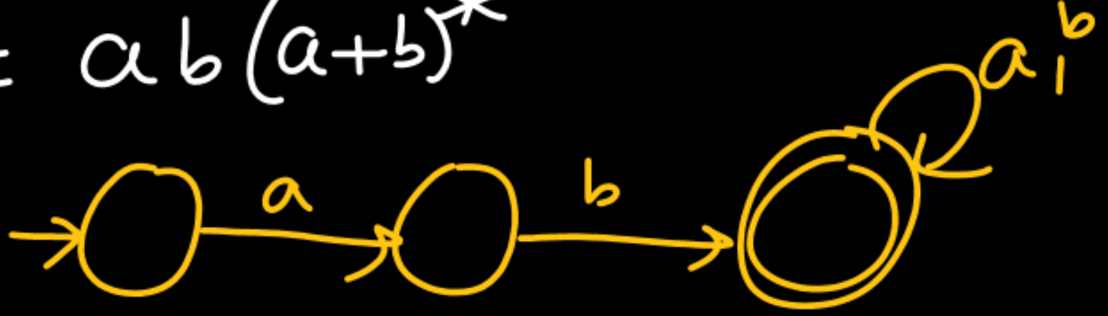
3 States



NFA

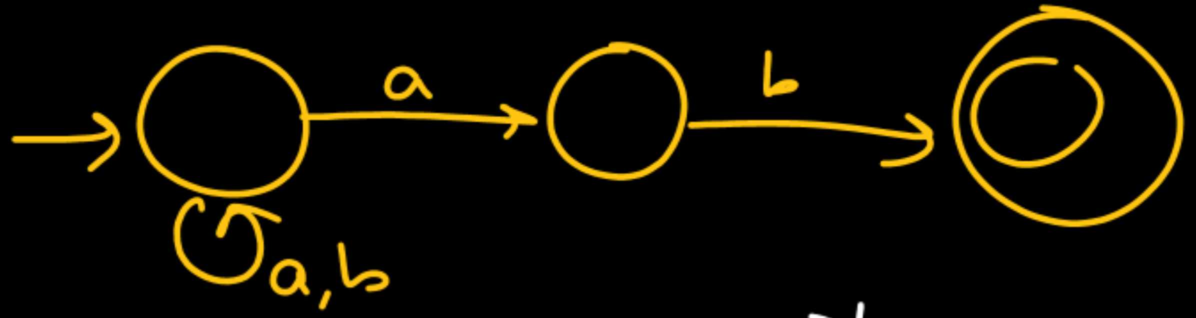
9 $L = ab(a+b)^*$

=3 states



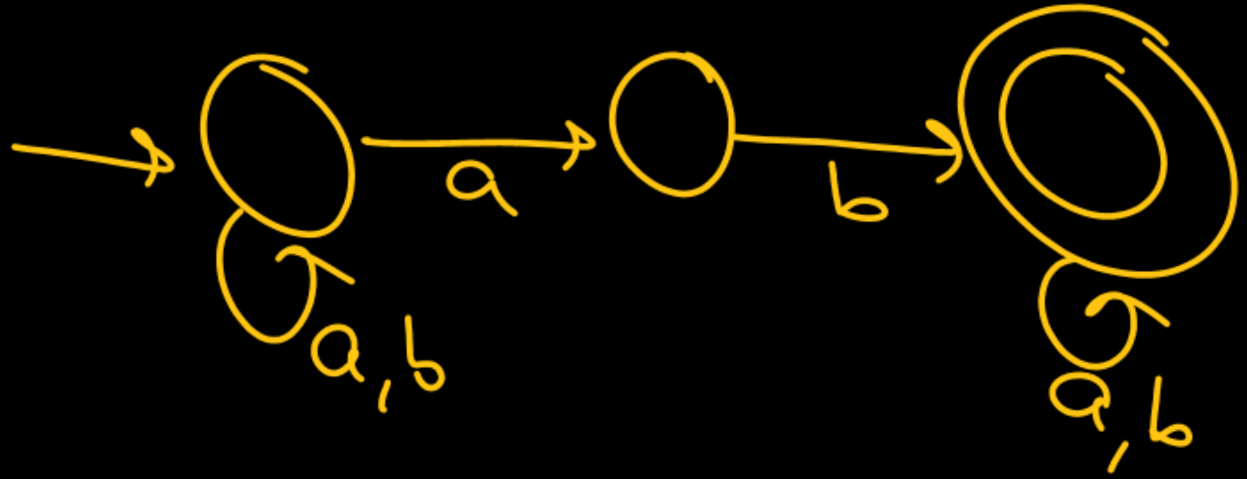
10 $L = (a+b)^*ab$

=3 states



11 $L = (a+b)^*ab(a+b)^*$

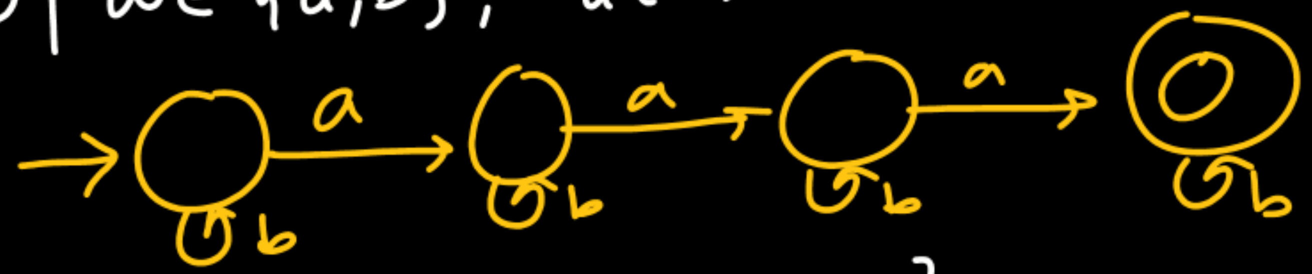
=3 states



NFA

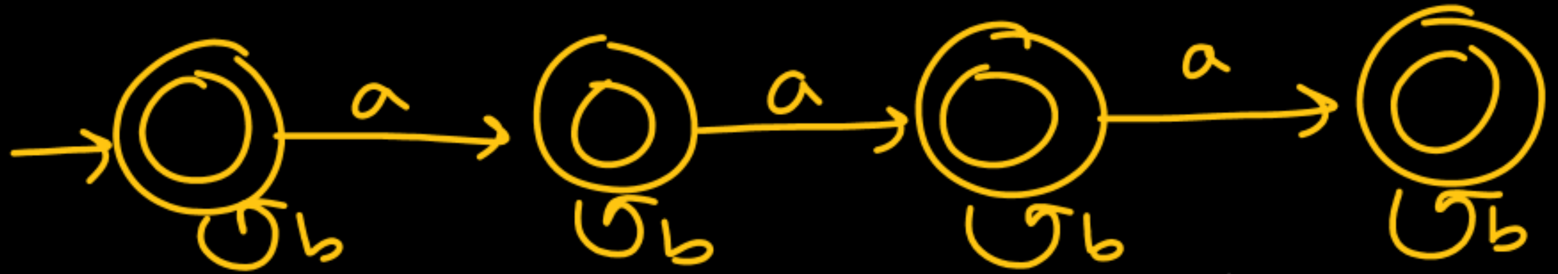
(12) $L = \{w \mid w \in \{a,b\}^*, n_a(w) = 3\}$

= 4 states



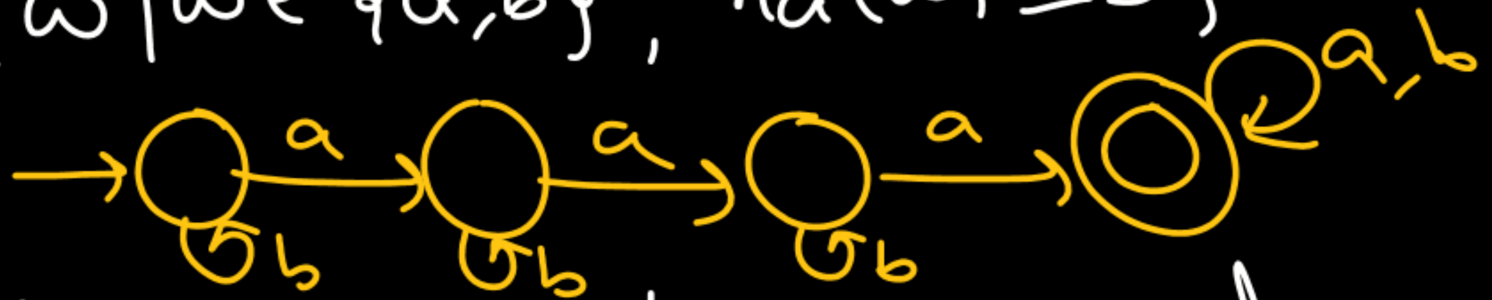
(13) $L = \{w \mid w \in \{a,b\}^*, n_a(w) \le 3\}$

= 4 states



(14) $L = \{w \mid w \in \{a,b\}^*, n_a(w) \ge 3\}$

= 4 states



(15) $L = \{w \mid w \in \{a,b\}^*, n_a(w)$ is divisible by 3$\}$

= 3 states

NFA

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

	L	Min DFA	Min NFA
1	Σ^*	1 State	1 State
2	ϕ	1	1
3	$\{\varepsilon\}$	2 states	1 states
4	$a\Sigma^*$	3 States	2 states
5	$\Sigma^* a$	2 states	2 states
6	$\Sigma^* a \Sigma^*$	2 states	2 states
7	$w \in \Sigma^*, \lvert w \rvert = K$	K+2 States	K+1 States
8	$w \in \Sigma^*, \lvert w \rvert \le K$	K+2	K+1
9	$w \in \Sigma^*, \lvert w \rvert \ge K$	K+1	K+1

K is constant

Note:

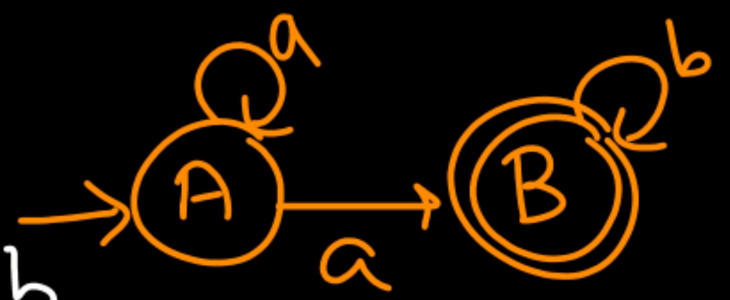
For any regular language L :

1. $\boxed{\text{No. of states (Min NFA)} \le \text{No. of states (Min DFA)}}$
2. No. of min NFAs ≥ 1
   No. of min DFAs $= 1$

$$NFA \xrightarrow{\text{Subset construction}} DFA$$

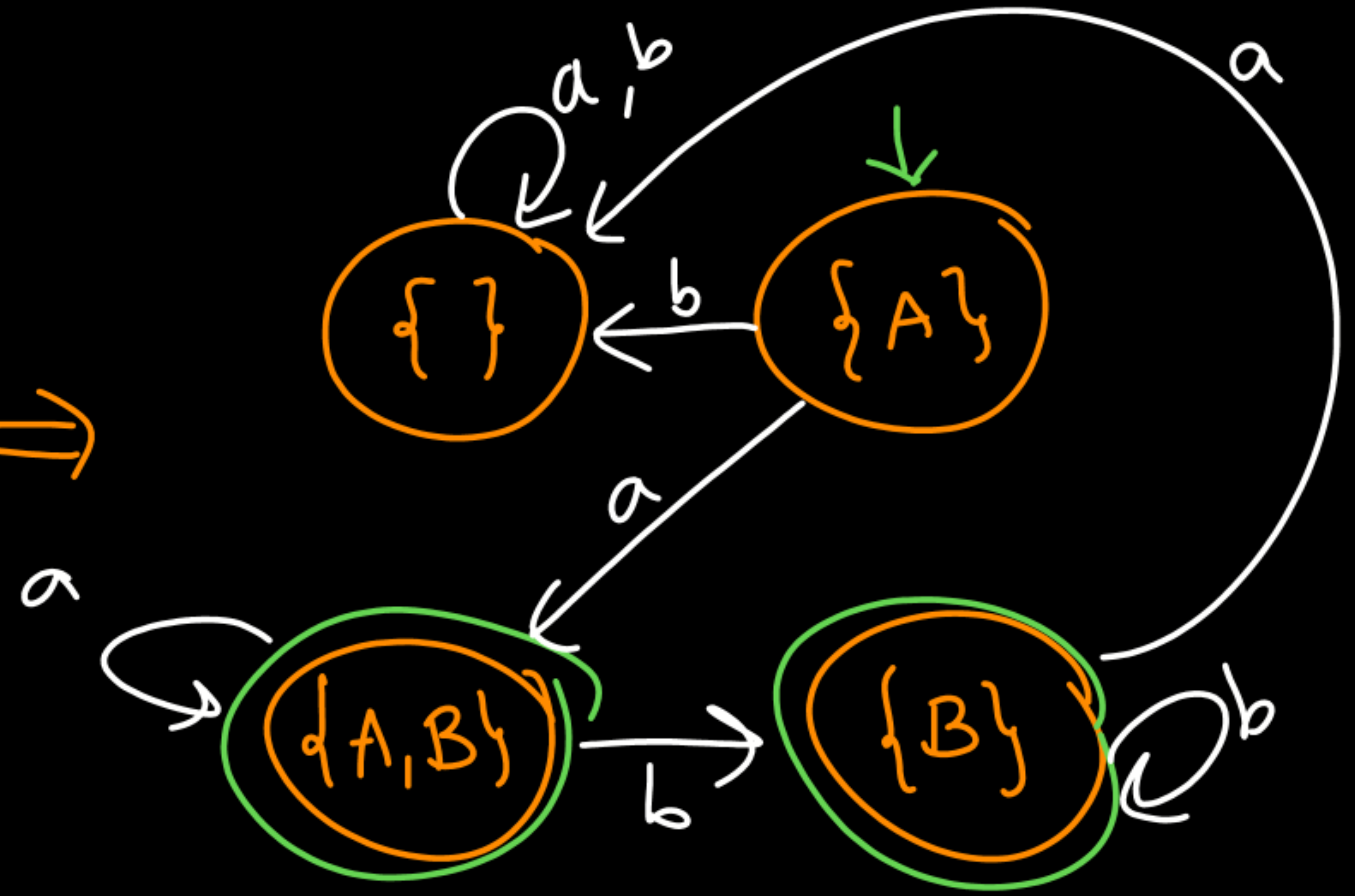
$$(Q, \Sigma, \delta_{NFA}, q_0, F) \qquad (2^Q, \Sigma, \delta_{DFA}, \{q_0\}, F')$$

Given



δ_{NFA}	a	b
$\rightarrow A$	$\{A, B\}$	ϕ
$*B$	ϕ	$\{B\}$

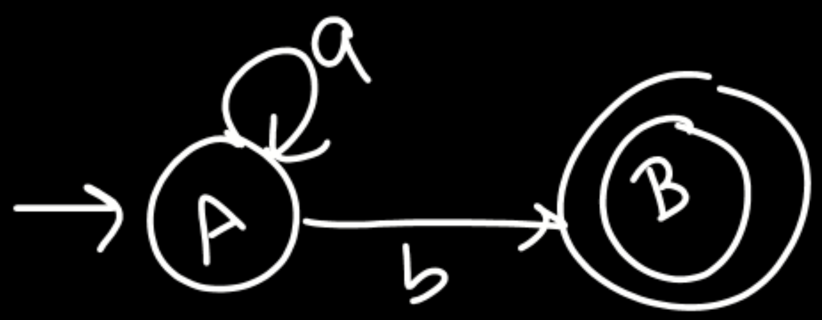
$\Longrightarrow$



n States in NFA $\Big\} \Longrightarrow$

1) We can always design DFA with 2^n states

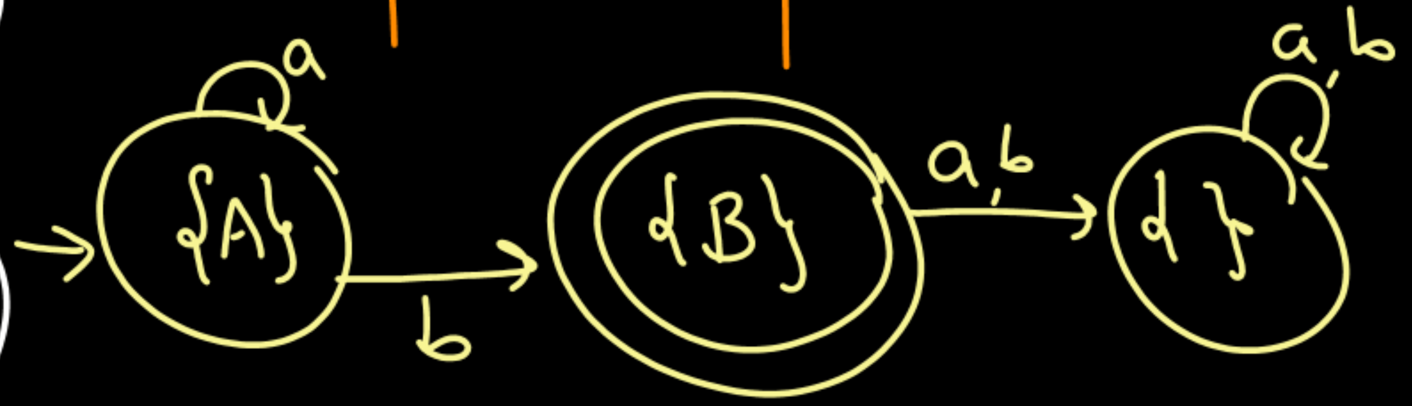
*2) Min DFA contain atmost 2^n states



NFA

NFA	a	b
→A	{A}	{B}
*B	{ }	{ }

DFA	a	b
→{A}	{A}	{B}
*{B}	{ }	{ }
{ }	{ }	{ }



Initial state of DFA' = {Initial of NFA}

NFA
without ε moves

δ_{NFA}

DFA

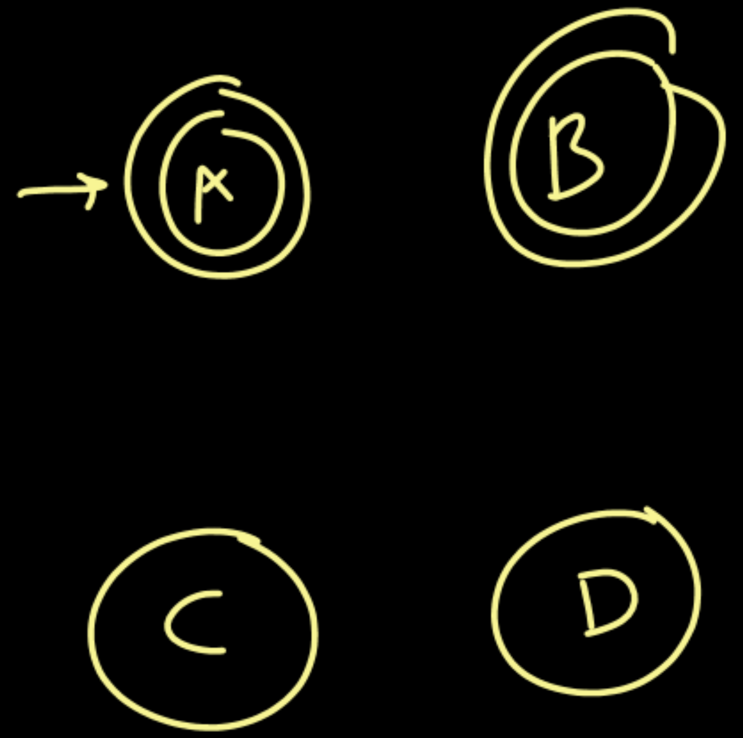
$\delta_{DFA}(\{q_1, q_2\}, a)$

$= \delta_{NFA}(q_1, a) \cup \delta_{NFA}(q_2, a)$

F = set of finals in NFA

F' = Set of finals in DFA

If any subset of Q (State in DFA) contain final state of NFA, then make it as final in DFA.



In NFA

$\Rightarrow$

{ } {A} {B} {C} {D}

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

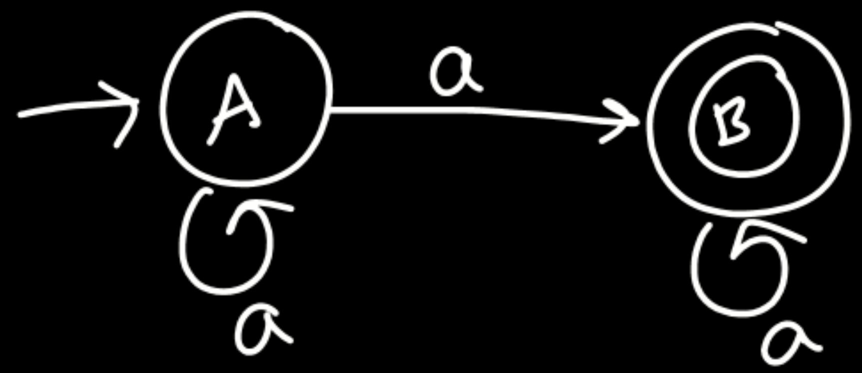
{B,D} {C,D} {A,B,C}

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

n states $\longrightarrow$ $\underbrace{\leq 2^n \text{ states}}_{\text{max } 2^n \text{ states}}$ in min DFA

(atmost 2^n states)

NFA



$\Sigma = \{a\}$

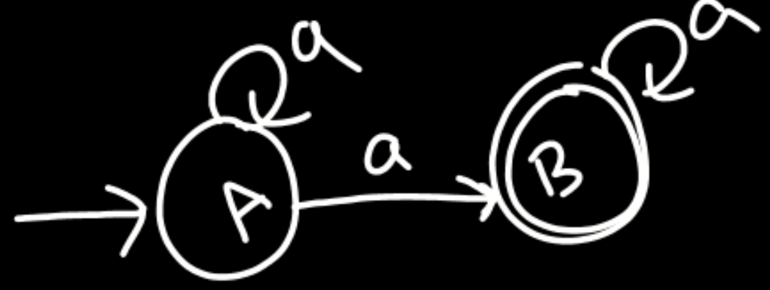
NFA	a
$\rightarrow A$	$\{A, B\}$
$*B$	$\{B\}$

$\Rightarrow$

DFA	a
$\rightarrow \{A\}$	$\{A, B\}$
$*\{A, B\}$	$\{A, B\}$

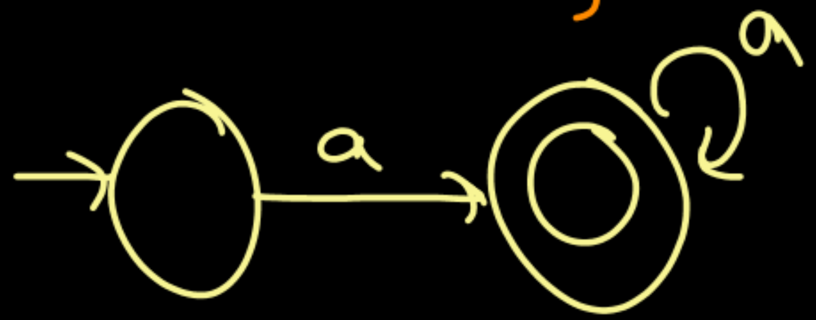
1. For the following NFA, find $\boxed{\text{no.of states}}$ in equivalent min DFA

Ex
a ✓
aa ✓
aaa ✓
⋮



Given

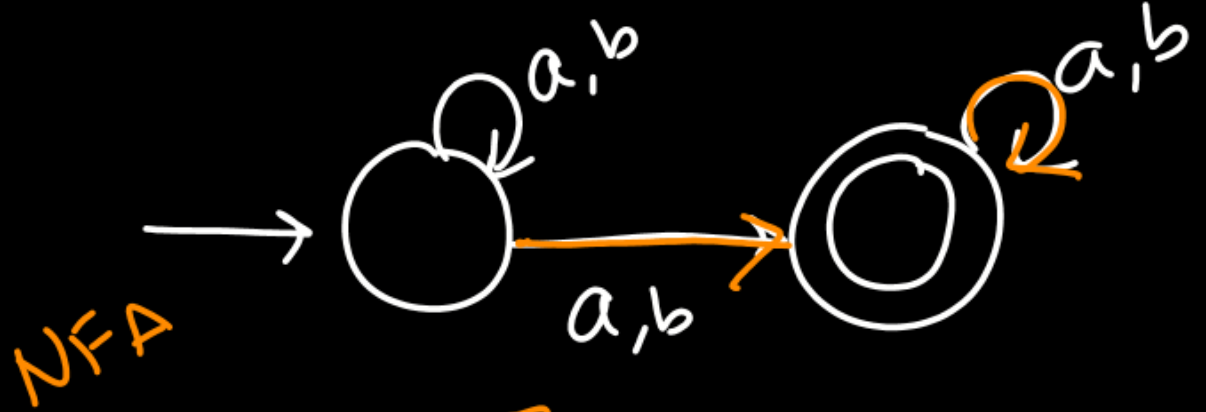
Best
Method 1: i) Understand $L(NFA) = a^+$
ii) Find min DFA directly



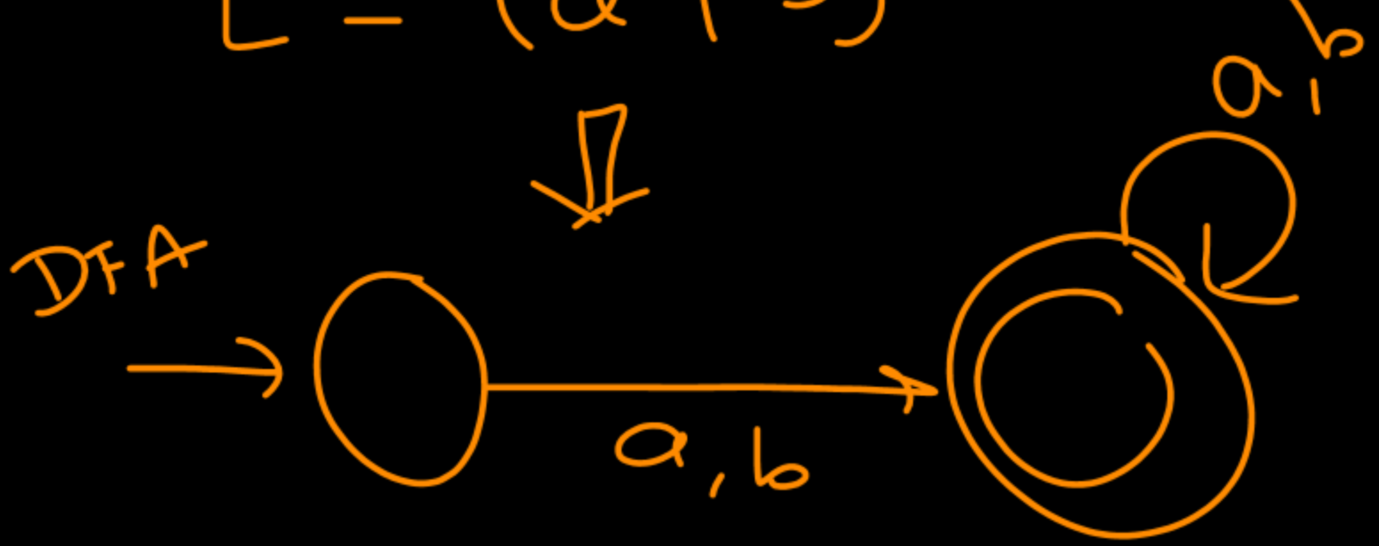
= 2 states in min DFA

Lengthy
Method 2:
i) Apply Subset construction
ii) Apply minimization Algo to get min DFA

(2)



$$L = (a+b)^{+}$$



Ex
- a ✓
- b ✓
- aa ✓
- ab ✓
- ba ✓
- bb ✓

some path halts at final

= 2 states in min DFA

Minimization Algorithm (Partition Method):

n states DFA for L ⟹ Minimization Algorithm ⟹ Minimized DFA for L ($\leq n$ states)

(minimal DFA)

(minimum DFA)

(DFA with less no. of states)

(Best DFA)

- No. of states may reduce
- If any 2 states are equivalent in DFA then Algo combines those states into single state to get "min DFA".

3 states

A
B
C } equivalent

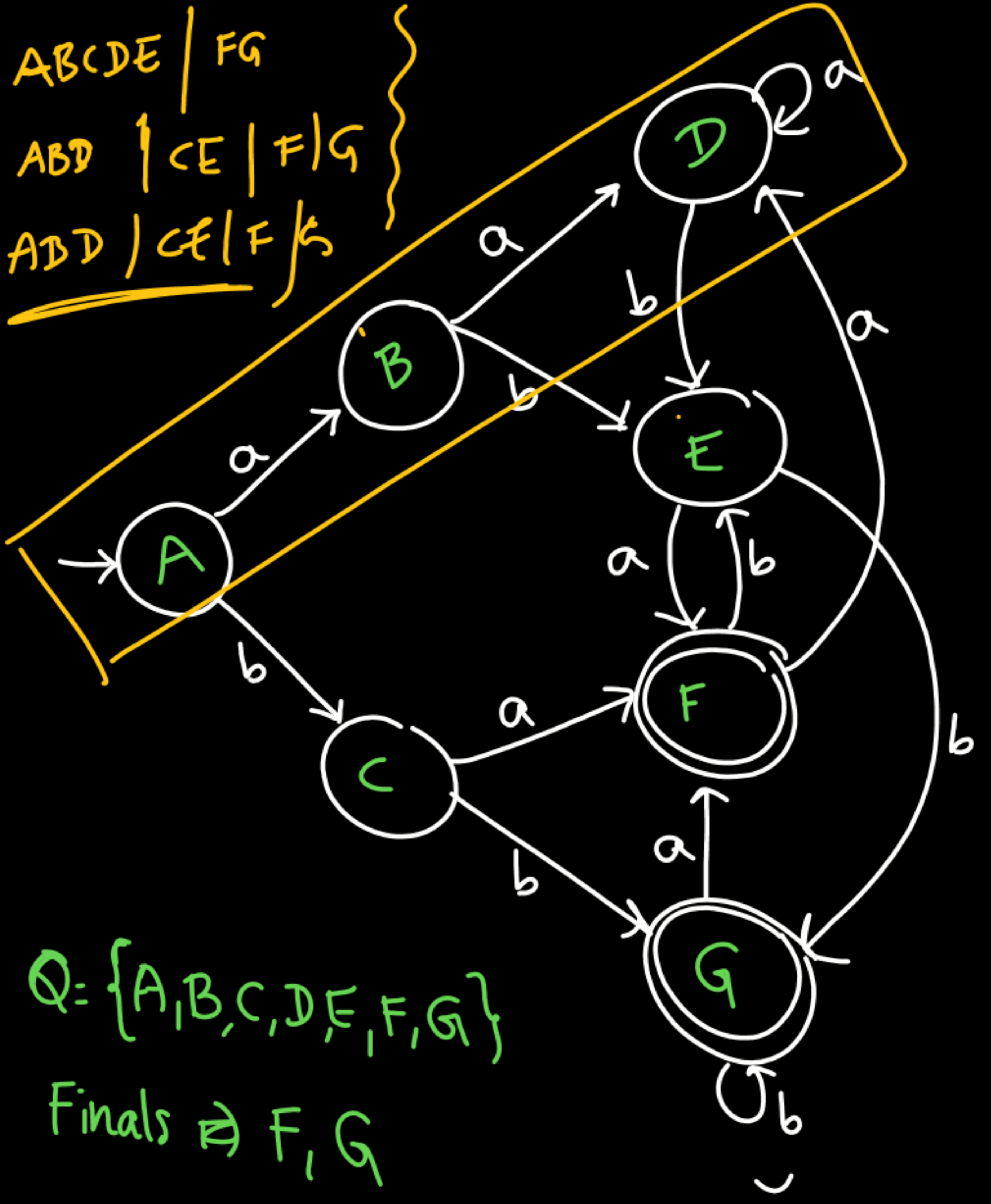
1 state

min DFA

ABCDE | FG

ABD | CE | F | G

ABD | CE | F | G



Q: {A, B, C, D, E, F, G}

Finals ⇒ F, G

Minimization Algorithm ⟹

Step 1: {A, B, C, D, E} {F, G}

all non finals   all finals

Step 2: {A, B, D} {C, E} {F} {G}

Step 3: {A, B, D} {C, E} {F} {G}

A, B: $\xrightarrow{b}$ C, E (in same set); $\xrightarrow{a}$ B, D (in same set)

A, C: $\xleftarrow{b}$; $\xrightarrow{a}$ B, F (not in same set) ✗

A, D: $\xrightarrow{b}$ C, E (in same set); $\xrightarrow{a}$ B, D (in same set)

A, E: $\xleftarrow{b}$; $\xrightarrow{a}$ B, F (in diff sets) ✗

C, E: $\xrightarrow{b}$ G, G; $\xrightarrow{a}$ F, F

F, G: $\xrightarrow{a}$ D, F (diff) ✗

$\{A, B, C, D, E\}$ $\{F, G\}$

no need to compare every two states

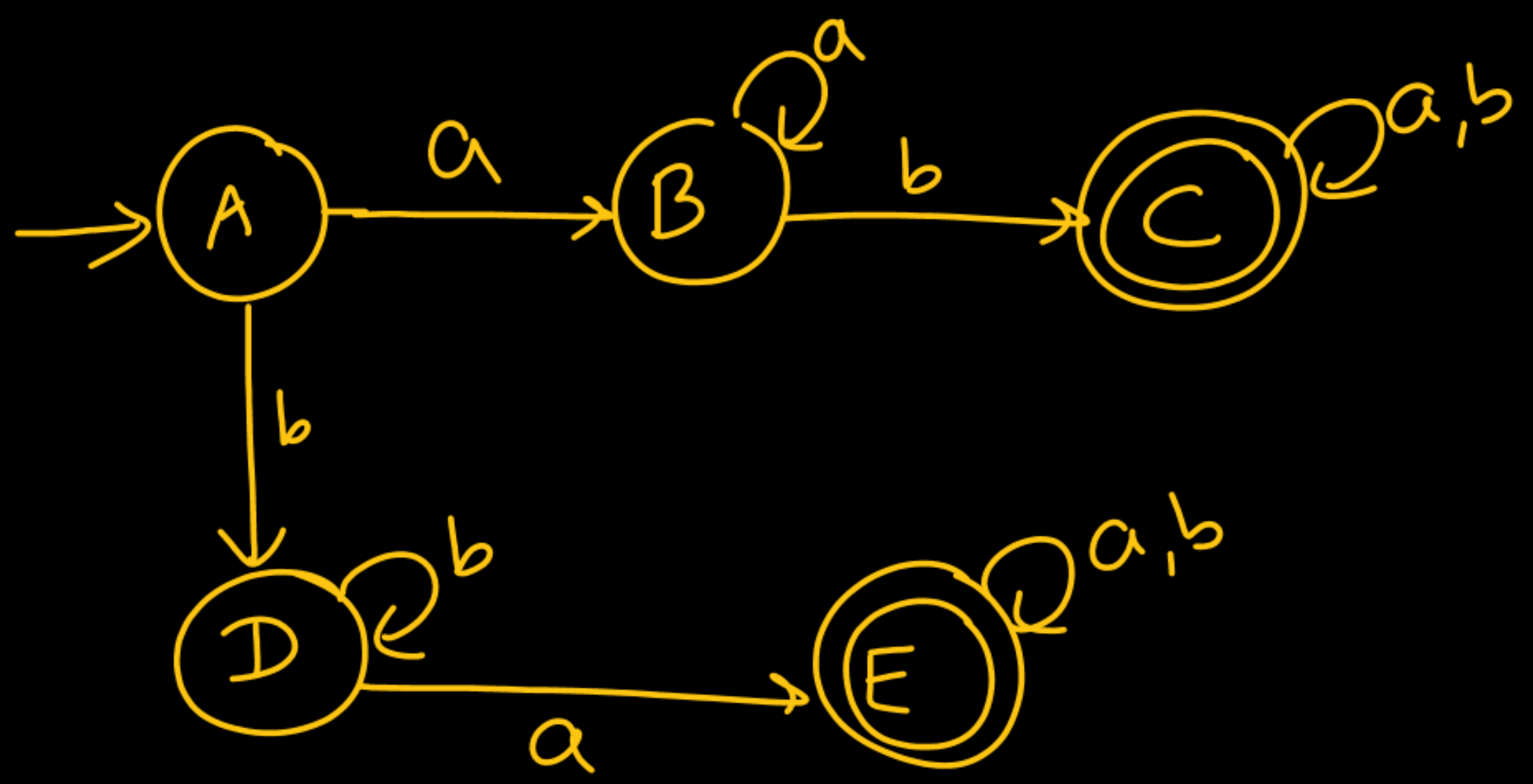
$\{\boxed{A}, B, C, D, E\}$

$\{A, B, E\}$

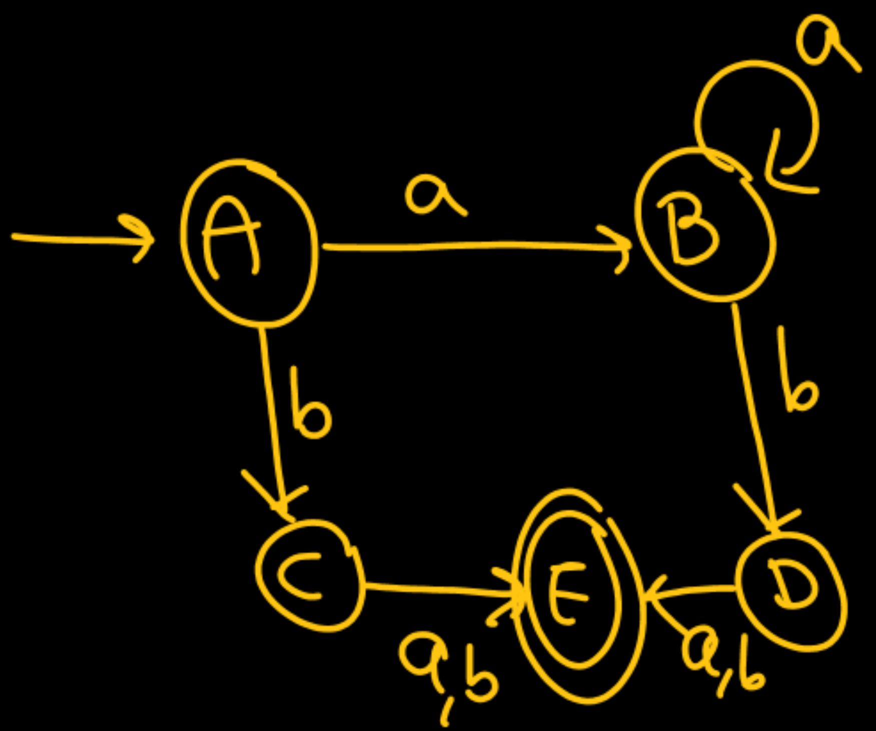
C & D will compare

H.W.

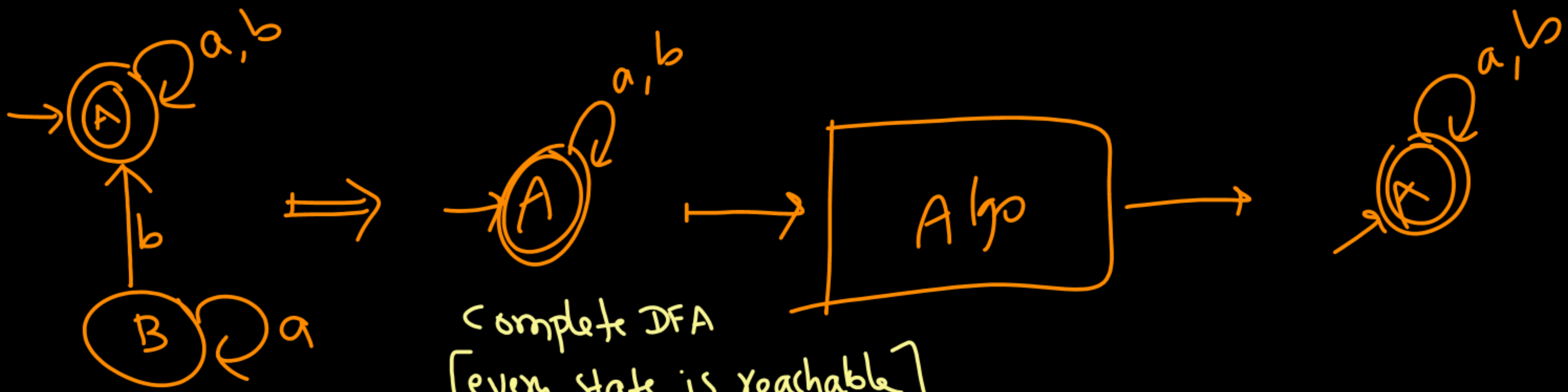
2



3



DFA $\xrightarrow{\text{Delete all unreachable states from initial state}}$ Take complete DFA $\longrightarrow$ $\boxed{\text{minimization Algo}}$ $\longrightarrow$ min DFA



DFA

Complete DFA
[every state is reachable from initial state]

B is not reachable from initial

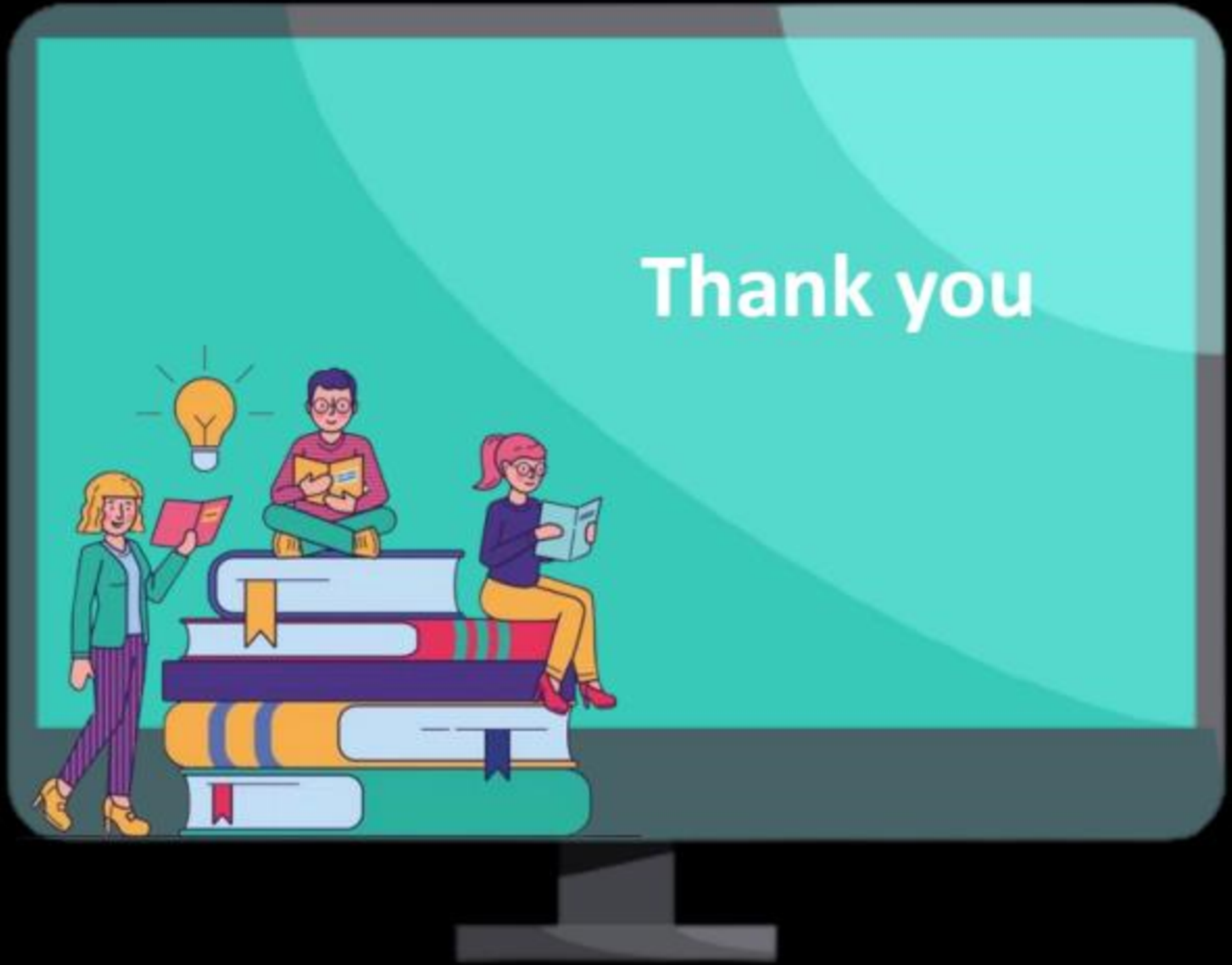
Summary

NFA ✓

NFA ⇒ DFA ✓

$n \quad \leq 2^n$ in min DFA

Minimization of DFA ✓


Thank you