CS & IT ENGINEERING

Wait for 5 min

Theory of Computation

Finite Automata:

DFA-2

Lecture No. 6

By- DEVA Sir

01 Regular Expressions

02 Finite Automata

03 Deterministic FA

Topics :

- Finite Automata
- DFA construction

aX

bX

Xa

Xb

XaX

$Xaa X$

$Xaa X$

$|w| = 2$

$|w| \leq 2$

$|w| \geq 2$

$|w| \,\%\, 3 = 0$

$n_a(w) = 2$

$n_a(w) \leq 2$

$n_a(w) \geq 2$

$|w| = \text{even}$

$|w| = \text{odd}$

$n_a(w) = \text{even}$

$n_a(w) = \text{odd}$

$a^* b^*$

$a^+ b^+$

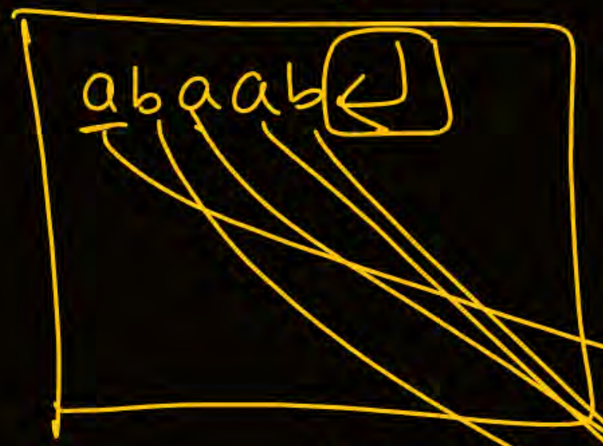
ϕ

ε

$(a+b)^*$

How many regular languages ?

↳ Infinite



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

```
g()
{
  ch = getch();
  if(ch=='a'||ch=='b')
    g();
  if(ch=='\n')
    Reject
}
```

```
f()
{
  ch = getch();
  if(ch=='a'||ch=='b')
    f();
  if(ch=='\n')
    Accept
}
```

```
char ch;
Void main()
{
  ch = getch();
  if(ch=='a')
    f(); => Accepting
  if(ch=='b')
    g(); => for rejecting
  if(ch=='\n')
    Reject
}
```

Finite Automata (Finite State Machine) (Finite Machine)

- It is a machine which represents a regular language. (regular set)
- It accepts (recognizes) a regular set

Finite Automata (FA)

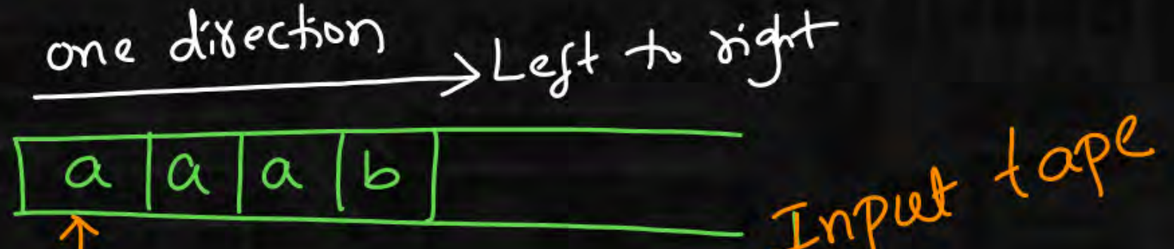
After reading whole string

If $w \in L$ (Valid string), FA accepts
FA halts at final state

If $w \notin L$ (Invalid string), FA not accepts
FA halts at non final state

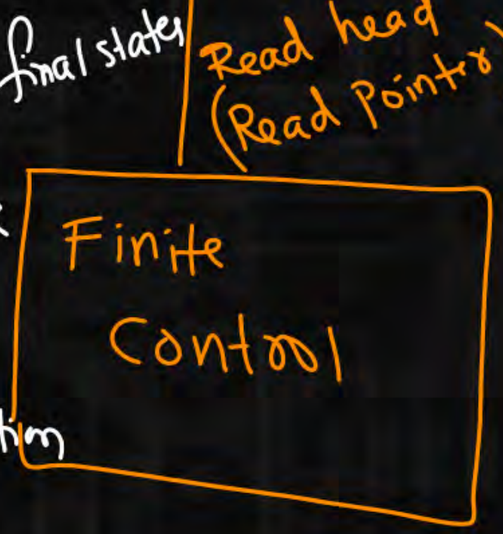
Finite Automata

Configuration of FA:



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

- Q → Set of states
- Σ → Input Alphabet
- δ → Transition Function
- q_0 → start state
- F → Set of final states

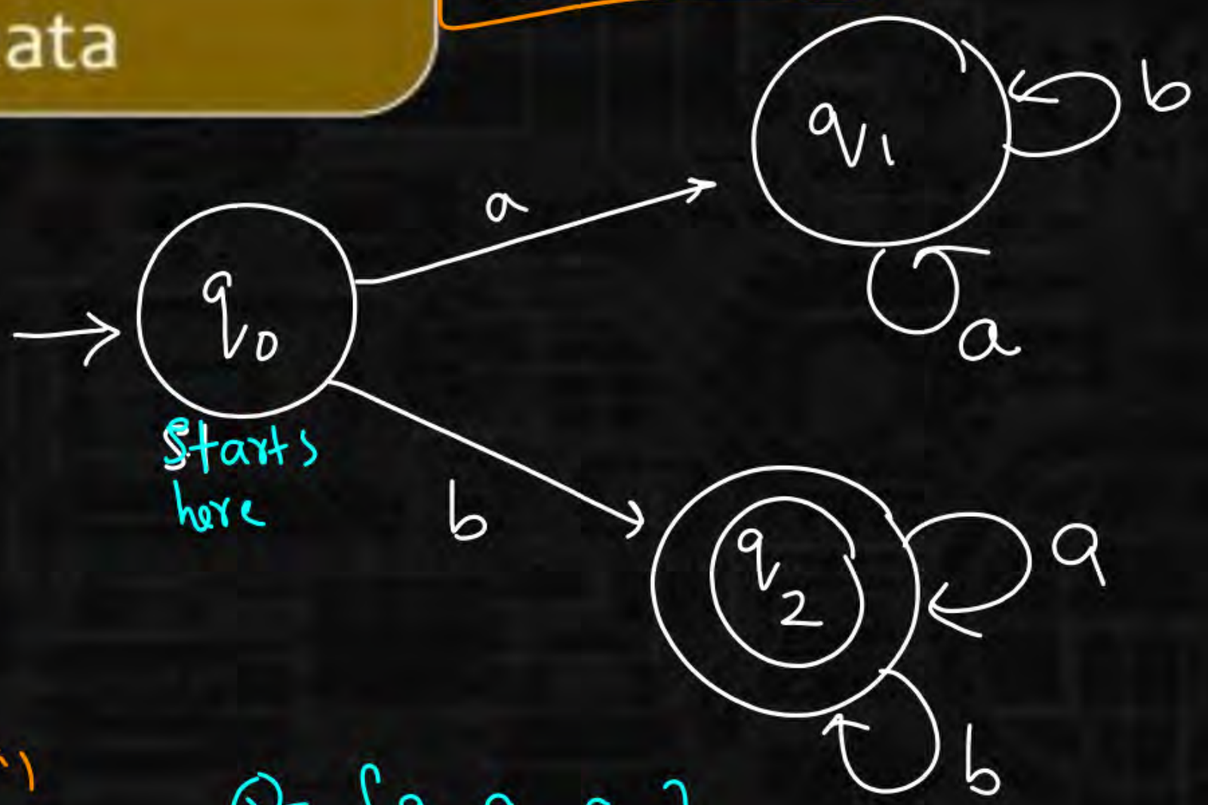


$q_0 \in Q$

$F \subseteq Q$

Finite Automata

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



$Q = \{q_0, q_1, q_2\}$

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

$q_0 = q_0$

$F = \{q_2\}$

Ex

a x

b ✓ valid string

✗ aa : $q_0 \xrightarrow{a} q_1 \xrightarrow{a} q_1$ (nonfinal)

✗ ab : $q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_1$ (nonfinal)

✓ ba : $q_0 \xrightarrow{b} q_2 \xrightarrow{a} q_2$ (final)

✓ bb

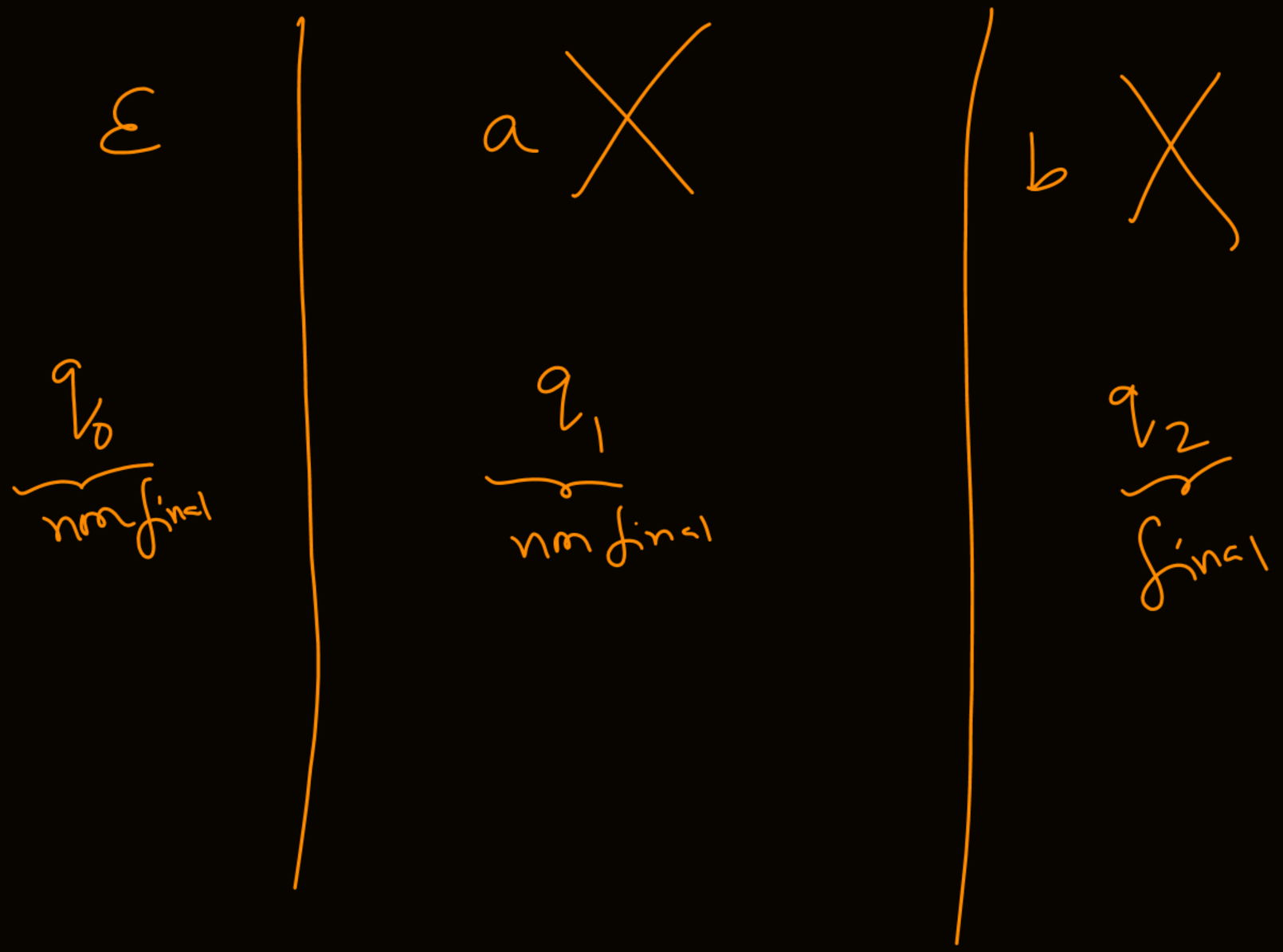
ε : Invalid halts at q_0

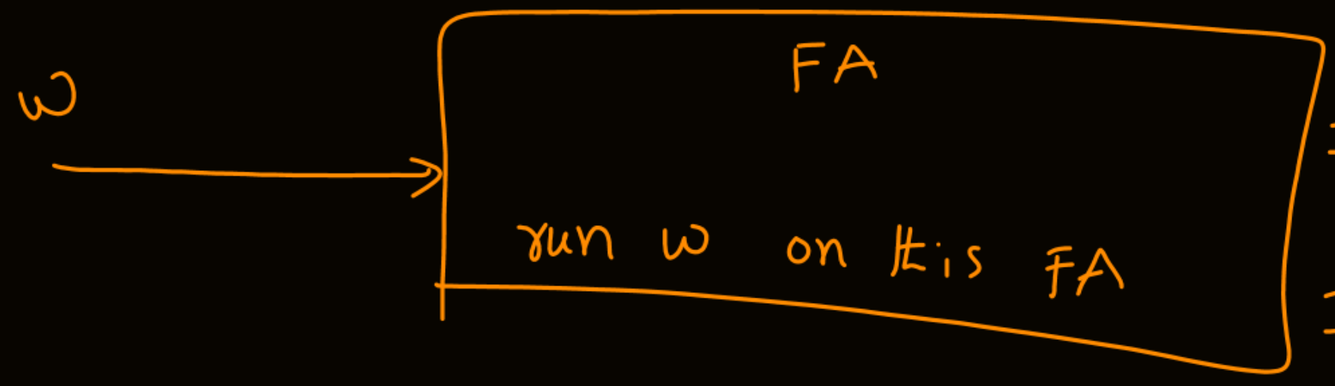
q_0

$q_0 \xRightarrow{\varepsilon} q_0$

$\omega = a$ Invalid halts at nonfin

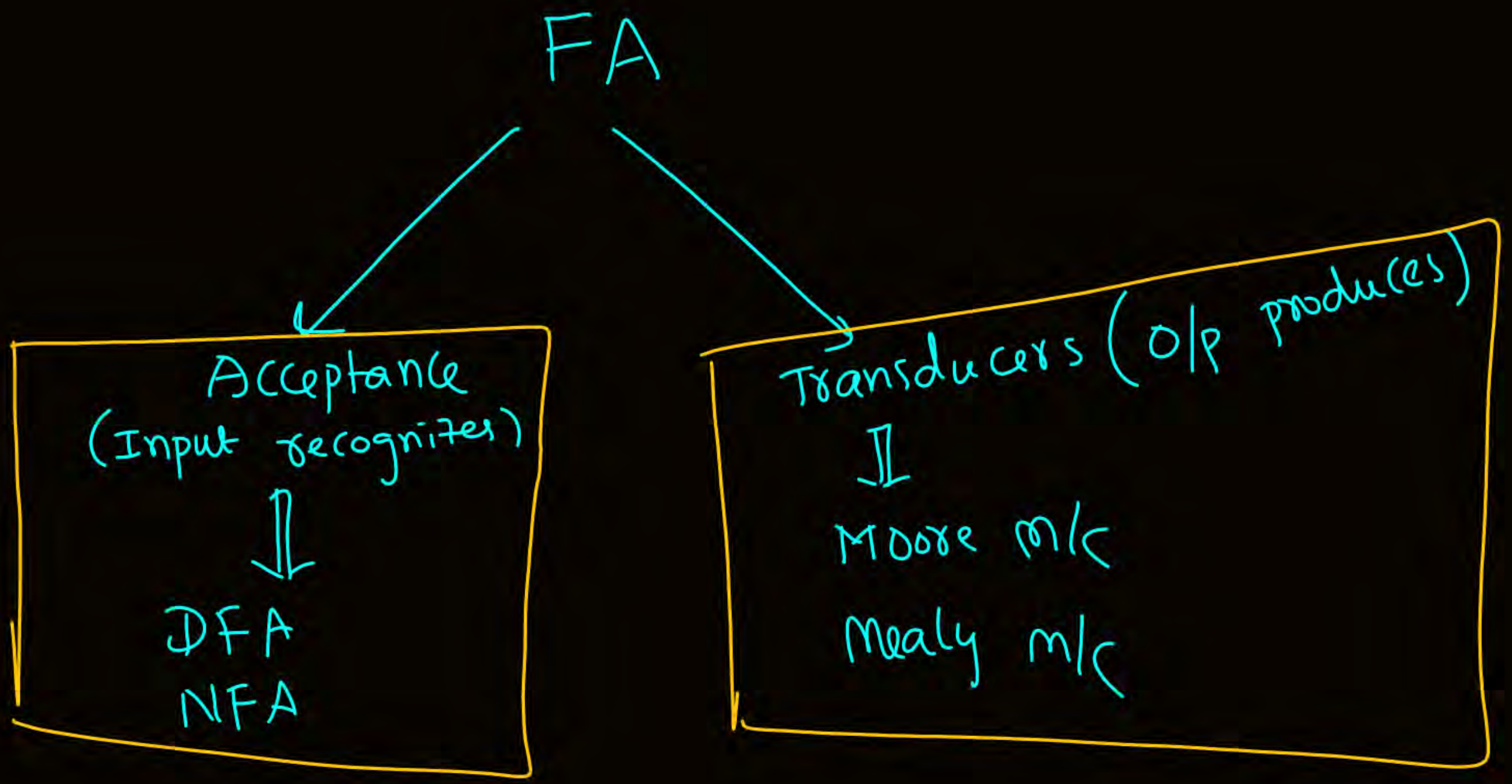
$q_0 \xrightarrow{a} q_1$ (Halt)


ε
a
b
q_0
non final
q_1
non final
q_2
final



If it halts at final state
$w \in L$ (valid)

If it halts at non final
$w \notin L$ (invalid)


FA
Acceptance
(Input recognizers)
DFA
NFA
Transducers (O/P produces)
Moore m/c
Mealy m/c

Finite Automata

Transition Function δ

- DFA (Deterministic FA)

$$\delta: Q \times \Sigma \rightarrow Q$$

- NFA (Non-deterministic FA)
  - without ε moves (NFA)

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

  - with ε moves (ε-NFA)

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

In maths:

function $f : A \longrightarrow B$

i) $f \subseteq A \times B$ [Relation]

ii) $\forall a \in A,\ f(a) = \underbrace{b}_{\text{some } b \in B}$ [Existence]

iii) Uniqueness

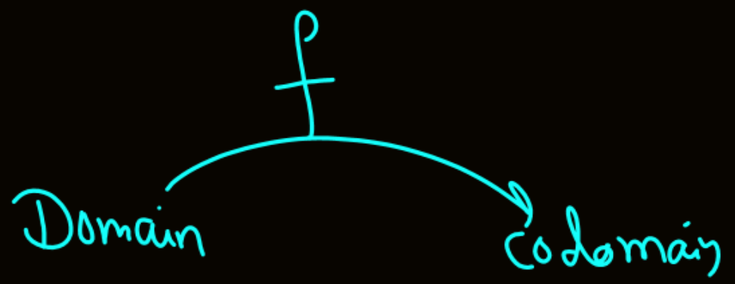
$$\left.\begin{matrix} f(a) = b_1 \\ f(a) = b_2 \end{matrix}\right\} \Rightarrow b_1 = b_2$$

Program

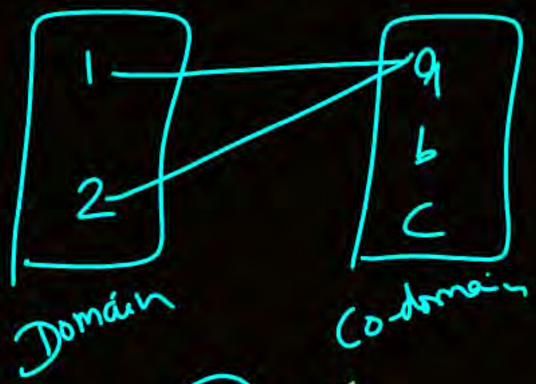
↳ set of functions

function

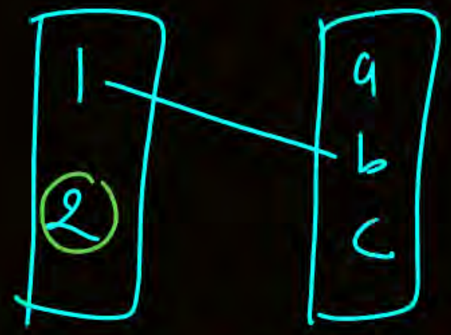
↳ ?



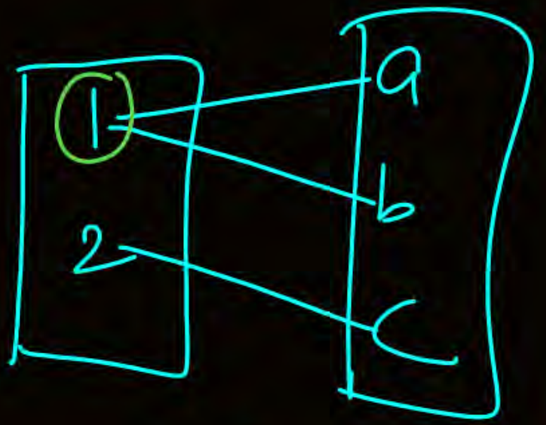
Identify functions:



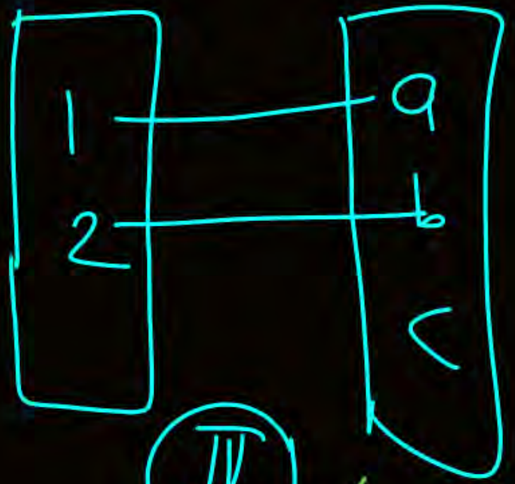
(I) ✓



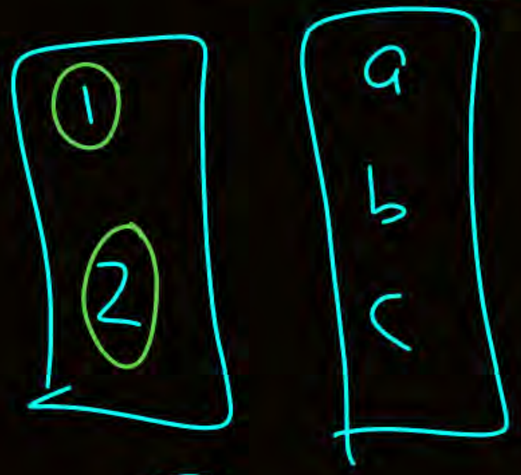
(II) X



(III) X



(IV) ✓



(V) X

Finite Automata (DFA) Representation

$\delta: Q \times \Sigma \rightarrow Q$ DFA

① State Diagram (δ)

No. of edges = 3

No. of transitions = 4

→○ : Initial state

○ : non final

◎ : final state

$\xrightarrow{0}$: Transition

② Transition Table (δ)

δ	0	1
→ A	B	A
* B	B	B

or

Ⓑ

	Σ
Q	Q

Q : codomain

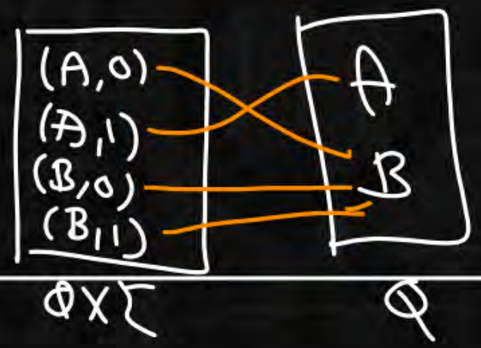
③ Set(δ)

$FA = (\{A,B\}, \{0,1\}, \delta, \underbrace{A}_{\text{Initial}}, \underbrace{\{B\}}_{\text{finals}})$

$\delta = \{((A,0),B), ((A,1),A), ((B,0),B), ((B,1),B)\}$

OR

① $\delta(A,0) = B$
② $\delta(A,1) = A$
③ $\delta(B,0) = B$
④ $\delta(B,1) = B$



$f(1) = a$

$f(2) = b$

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

$\in$ Domain $\in$ co-domain

$\delta(A, 0) = B$

$\delta = \{ ((A, 0), B) \ldots \}$

$(A, 0) \in Q \times \Sigma$ (Domain)

$B \in Q$ (co-domain)

$\delta : Q \times \Sigma \rightarrow Q$

$Q \times \Sigma$: Domain

Q: co-domain

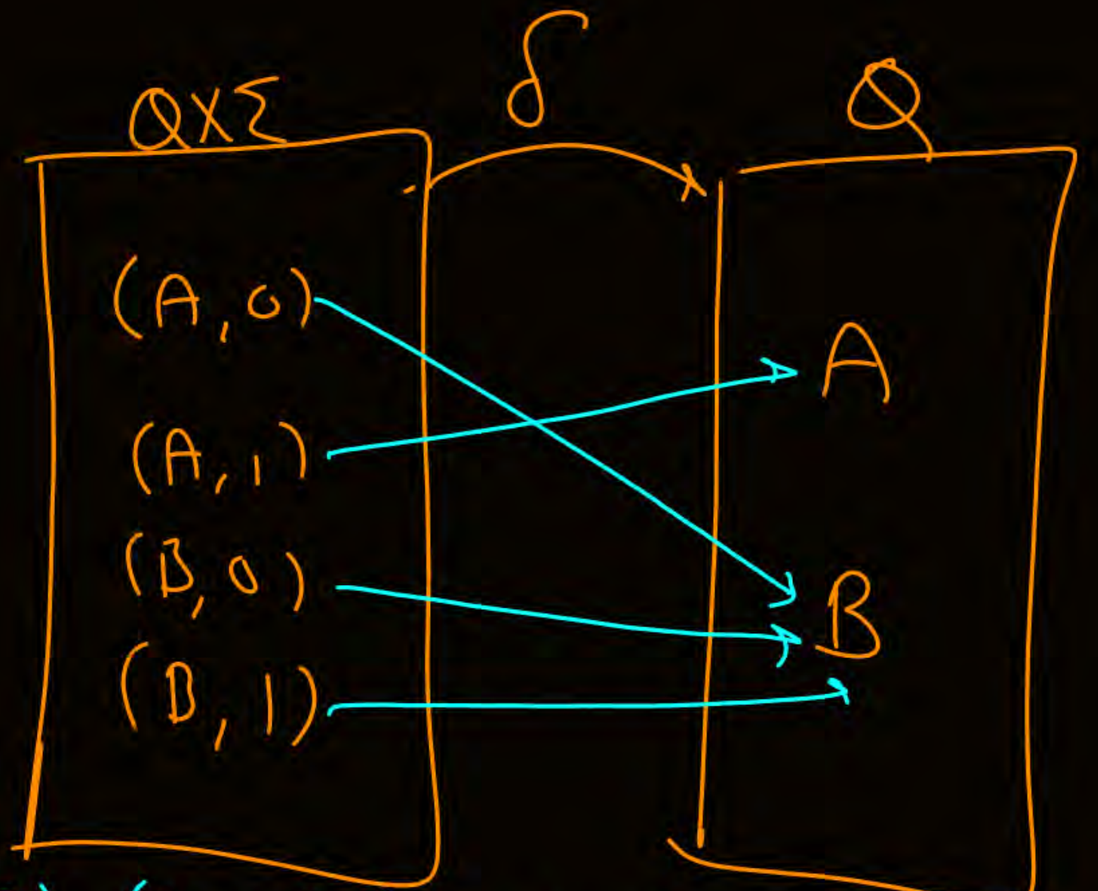
$$\delta : Q \times \Sigma \longrightarrow Q$$

Domain — Co-domain

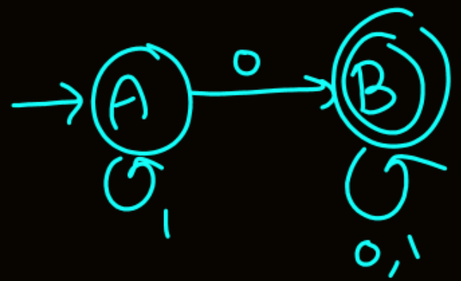
$Q = \{A, B\}$

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

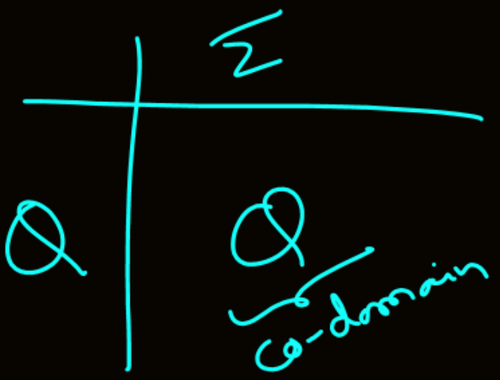
$Q \times \Sigma = \{(A,0), (A,1), (B,0), (B,1)\}$



$\delta = \{((A,0), B), ((A,1), A), ((B,0), B), ((B,1), B)\}$



δ	0	1
$\rightarrow A$	B	A
$*B$	B	B



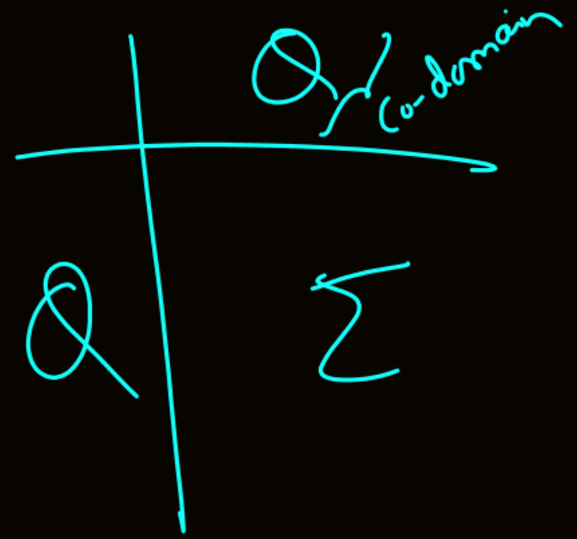
$\delta(A,0) = B$

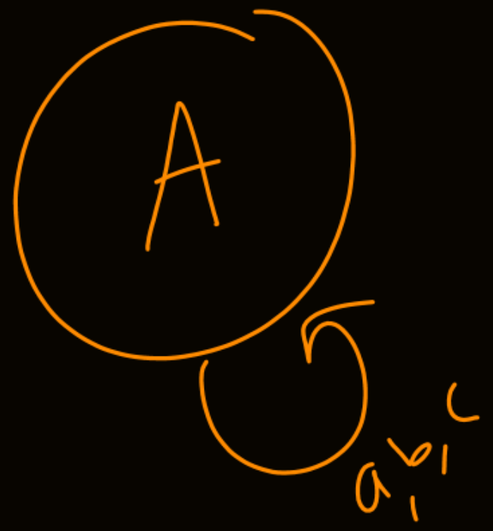
$\delta(A,1) = A$

$\delta(B,0) = B$

$\delta(B,1) = B$

δ	B	A
$\rightarrow A$	0	1
$*B$	0,1	





$\delta(A, a) = A$

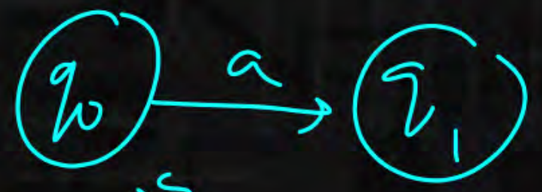
$\delta(A, b) = A$

$\delta(A, c) = A$

One edge

3 transitions

Finite Automata



$\delta(q_0^{PS}, a^{i/p}) = q_1^{NS}$

	i/p
PS	NS

$f : A \to B$

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

$((x, y), z) \in \delta$

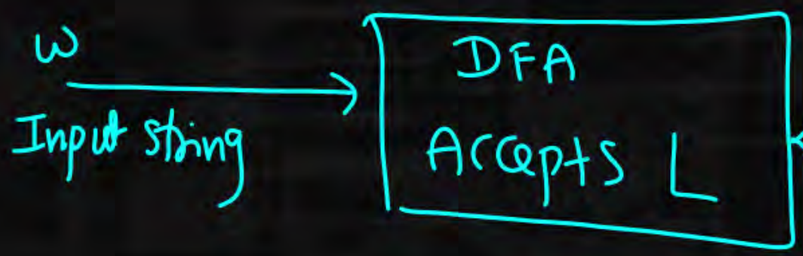
$\in Q \times \Sigma$

DFA Definitions

I) $\delta : Q \times \Sigma \longrightarrow Q$

II) From every state, for every i/p symbol, exactly one transition to next state is present.

III)



If $w \in L$, DFA halts at final state
exactly one path that halts at final

If $w \notin L$, exactly one path halts at non final state

(IV) No. of paths for every string in the Σ^* = 1 path

(V) No. of transitions in DFA = $|\delta|$
$= |Q \times \Sigma|$
$= |Q| \times |\Sigma|$

(VI) No. of Initial states = 1

(VII) No. of Final states = Depends on problem (≥ 0)

(VIII) No. of non-final states = Depends on problem (≥ 0)

(IX) No. of states
= depends on problem
= finite no. of states

$$f : A \rightarrow B$$

For Every element of A, there is corresponding single element in B

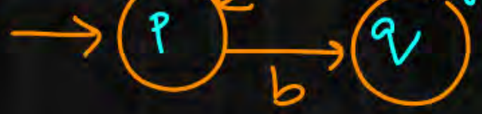
$$\delta : Q \times \Sigma \rightarrow Q$$

From every state ($\in Q$), for every i/p ($\in \Sigma$), there is exactly one transition to next state ($\in Q$)

DFA Construction

Identify DFAs. $\Sigma = \{a, b\}$

① Not DFA



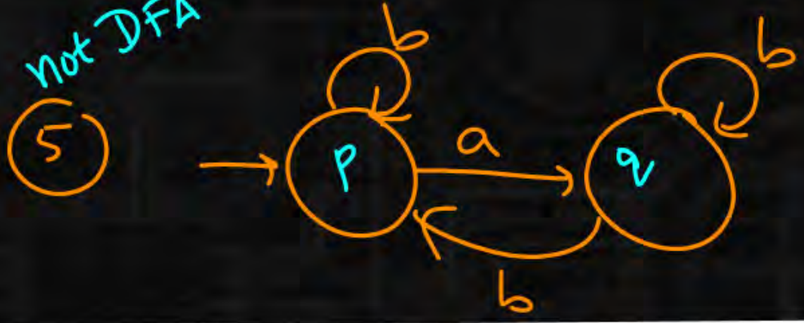
$\delta(q, a) = X$
$\delta(q, b) = X$

④ not DFA

$\delta(p, b) = X$
$\delta(q, a) = X$

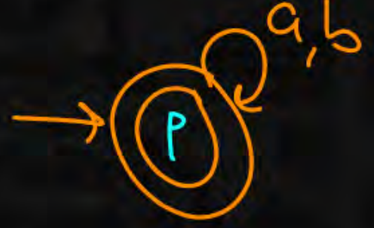
② DFA

⑤ not DFA

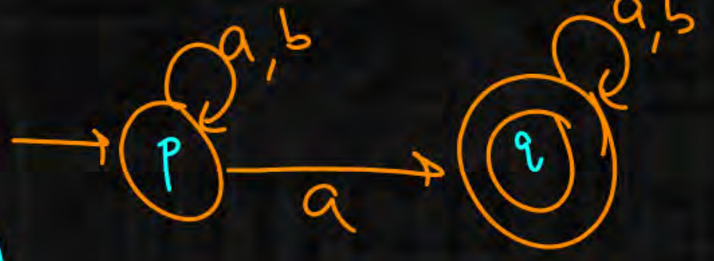


$\delta(q, a) = X$
$\delta(q, b) = \{p, q\}$

③ DFA



⑥ not DFA



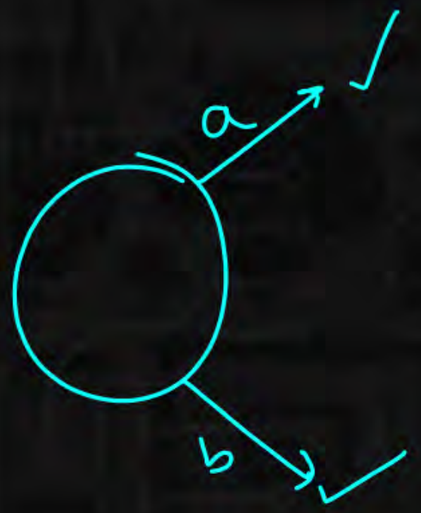
$\delta(p, a) = \{p, q\}$
2 transitions for same i/p

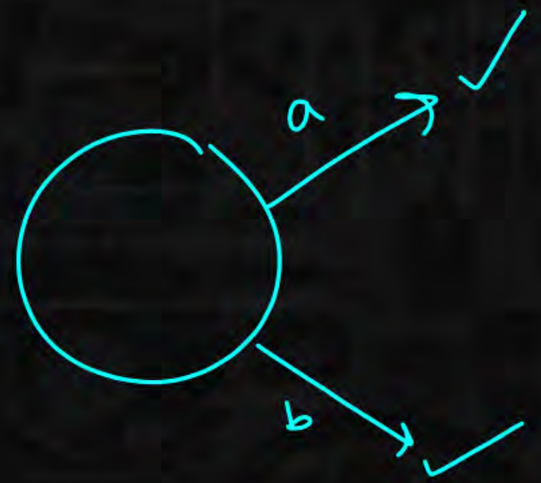
DFA Construction

$|\delta| = |Q \times \Sigma|$

$= |Q| \times |\Sigma|$

No. of transitions



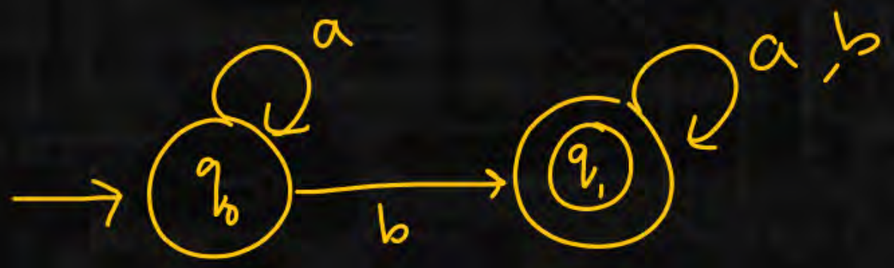


2 states
2 i/p symbols in Σ
$\Rightarrow$ $2 \times 2 \Rightarrow 4$ transitions

1 state
2 i/p symbols
$\Rightarrow$ $1 \times 2 \Rightarrow 2$ transitions

DFA Construction

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



$(a+b)^* = \Sigma^*$

↳ every string

IS it DFA ? ⇨ It is DFA

How many paths for every string in Σ^* ?

String	Path	No. of Paths	Valid / Invalid
ε	q_0	1	✗
a	$q_0 \xrightarrow{a} q_0$	1	✗
b	$q_0 \xrightarrow{b} q_1$	1	✓
aa	$q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0$	1	✗
ab	$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1$	1	✓
ba	$q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_1$	1	✓
bb	$q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_2$	1	✓

Valid & Invalid

⋮

DFA Construction

$\delta(q_0, \text{symbol})$

δ → Transition

(q0, symbol) → Transition

$q_0 \xrightarrow{a} q_1$

$\delta(q_0, a) = q_1$

$\hat{\delta}(q_0, w)$

$\hat{\delta}$ → path extended transition

w → string

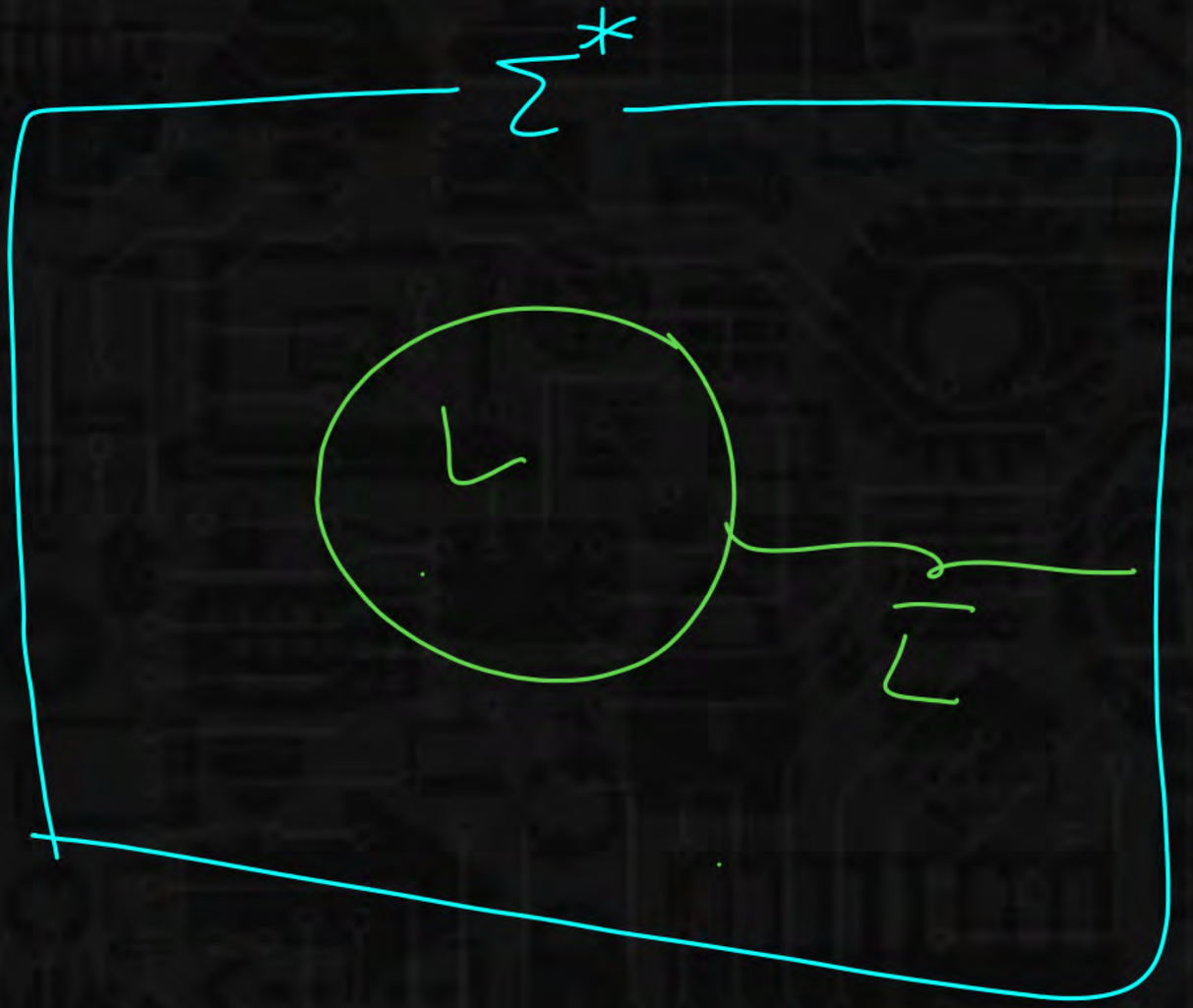
(q0, w) → path

$q_0 \xrightarrow{abb} q_2$

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_2$

path

DFA Construction



$$L \cup \bar{L} = \Sigma^*$$

$w \in L$ iff $w \notin \bar{L}$

$w \notin L$ iff $w \in \bar{L}$

DFA Construction

$L = ab(a+b)^*$ — this is your problem

$L \subseteq \Sigma^*$

$L \subseteq (a+b)^*$

Valid string $\in L$

ab, $ab\square$, $ab\square\square$, . . .

Invalid string $\in \bar{L}$

ε
a
b
aa
ba
bb
⋮

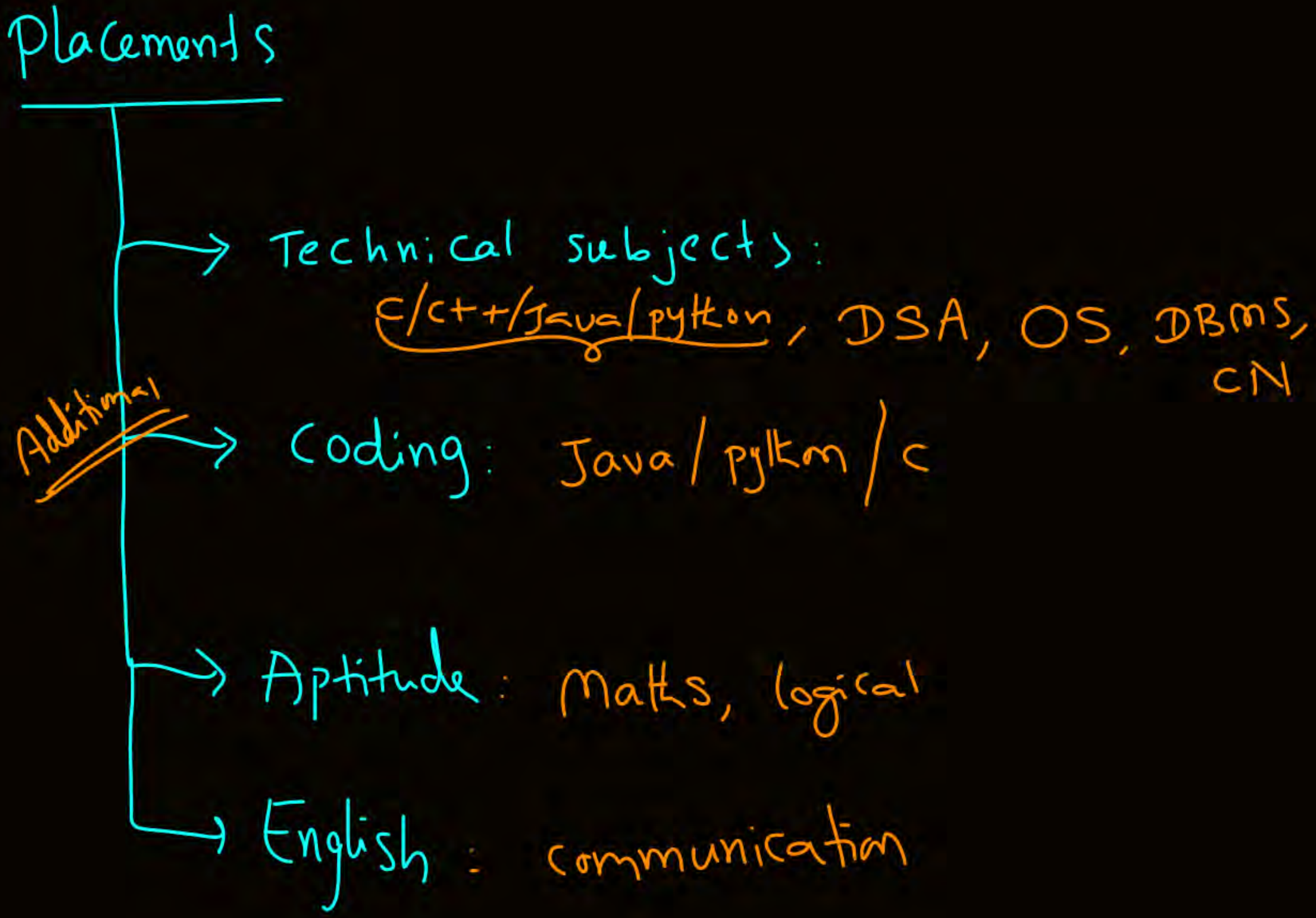
$\square\square$ . . . .

$\neq a$ $\neq b$

Invalid

Summary

- What is FA?
- What is DFA?


Placements
Technical subjects:
C/C++/Java/python, DSA, OS, DBMS, CN
Additional
Coding: Java / python / c
Aptitude: Maths, logical
English: communication


THANK YOU GW
SOLDIERS !