

• Theory of Computation

↓

what which How

Problem Algorithm efficiently.
can be solved

Symbol → a, b, 0, 1 - - -

Alphabet → "Σ": {a, b}

String → a, aa, aba, abb - - -

Language → 'L' = set of all strings of length 2
 $\{aa, ab, ba, bb\}$

finite Automata → $w \in L$ or $w \notin L$



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

L = set of all strings of length 2
 $\{aa, ab, ba, bb\}$

$w_1 = ab$

$w_2 = bbb$

L = set of all strings start with 'a'
 $\{a, aa, ab, aab, \dots\}$

$w_1 = abb$

$w_2 = baa$

- Length of string: no. of symbol in string
"ISI"
eg: $S = abcd$, $|S| = 4$ if $|S|=0$ it's ' ϵ '

- Kleene star

$$\hookrightarrow \Sigma^* \text{ (universal set)}$$

$\Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \dots \Sigma_k$ where Σ_k is
set of all possible strings of length k

eg: If $\Sigma = \{a, b\}$

$$\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, \dots\}$$

- Kleene closure/plus

$$\hookrightarrow \Sigma^+ \\ \downarrow \\ " \Sigma^* - \epsilon "$$

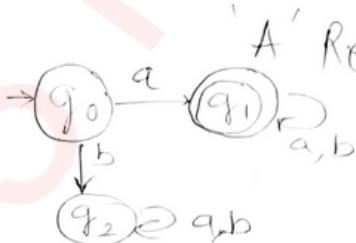
eg: $\Sigma = \{a, b\}$

$$\Sigma^+ = \{a, b, aa, ab, \dots\}$$



Accepts/

'A' Represents 'L'



Generates

$$L = \{a, aa, ab, \dots\} \quad \begin{cases} S \rightarrow ax \\ x \rightarrow ax / bx \end{cases}$$

finite Automata

without
output

- DFA
- NFA

with
Output

- Moore m/c
- Mealy m/c

* Deterministic finite Automata (DFA)

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

↓ ↓ ↓ ↓ ↓ ↓
 Set of states I/P Transition function Start/Initial State final state

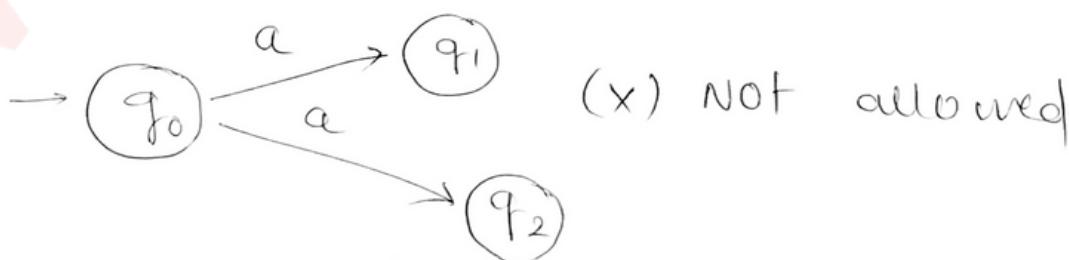
$$\delta : Q \times \Sigma \rightarrow Q \quad \Sigma = \{a, b\}$$



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

δ	Σ
Q	Transition Table

what is "Deterministic" ?

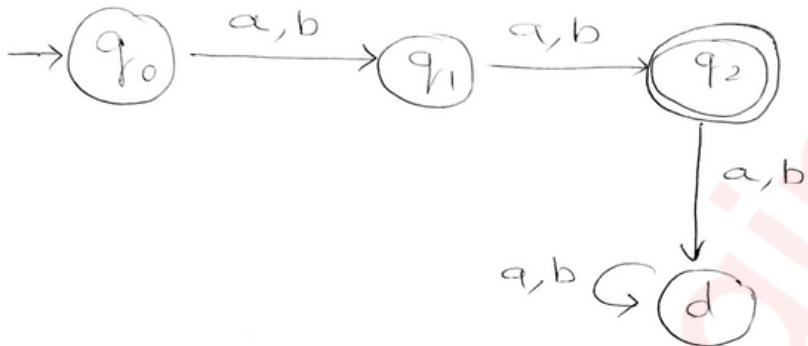


Eg: Length based:

$$|w|=2 \text{ or } |s|=2$$

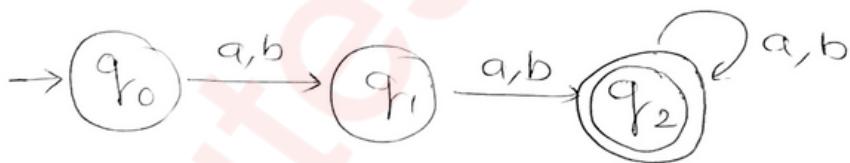
$$\Sigma = \{a, b\}$$

$$L = \{aa, ab, ba, bb\}$$



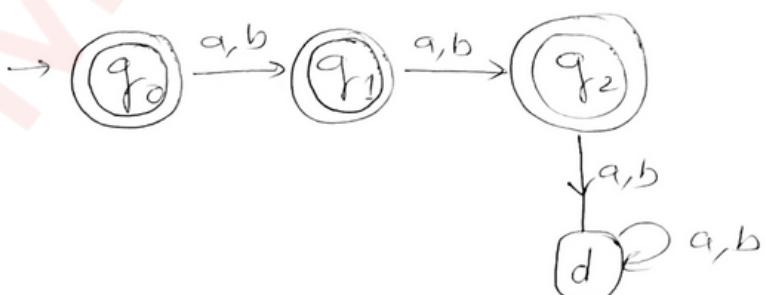
$$|s| \geq 2$$

$$L = \{aa, ab, ba, bb, aaa, aba \dots\}$$



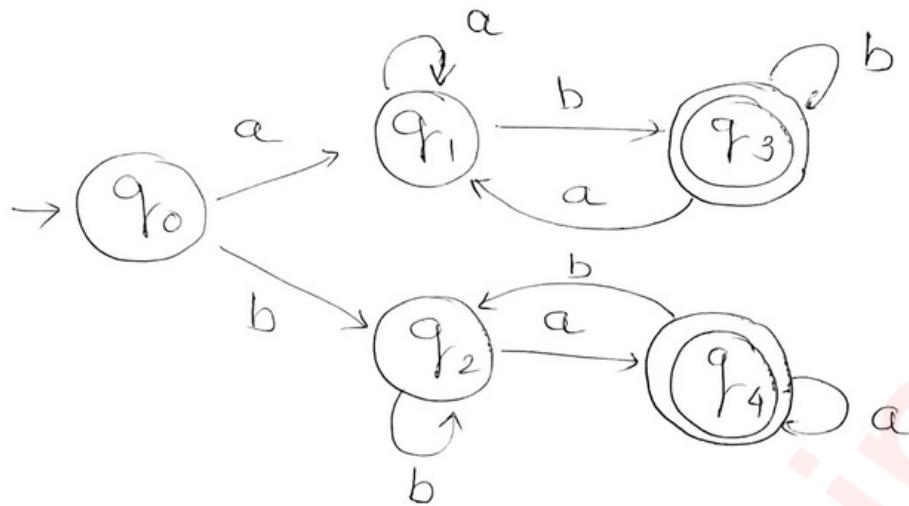
$$|s| \leq 2$$

$$L = \{\epsilon, a, b, aa, ab, ba, bb\}$$



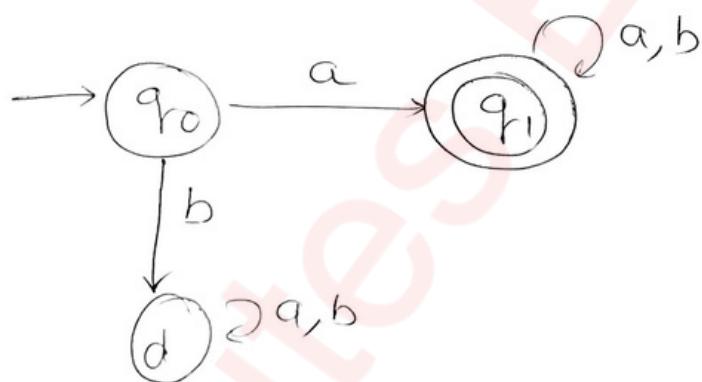
Q: Starts & Ends with different symbol.

$$L = \{ab, aab, abb, baa, bba \dots\}$$



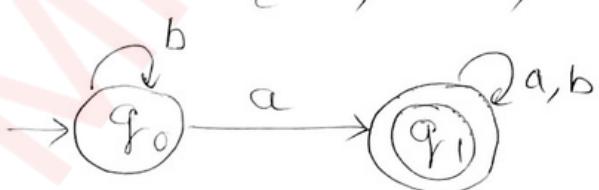
Q: Starts with 'a'

$$L = \{a, aa, ab, abb \dots\}$$



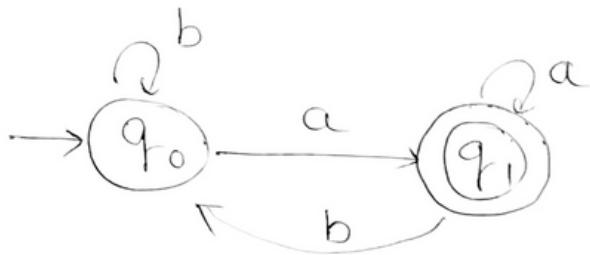
Q: Contain 'a'

$$L = \{a, ab, ba, aab, aba \dots\}$$



Q: Ends with 'a'

$$L = \{ a, ba, aa, bba, \dots \}$$

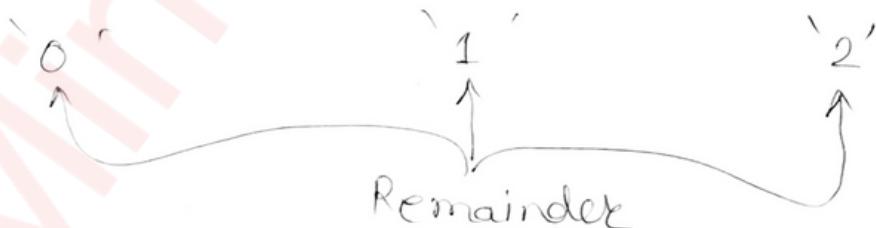
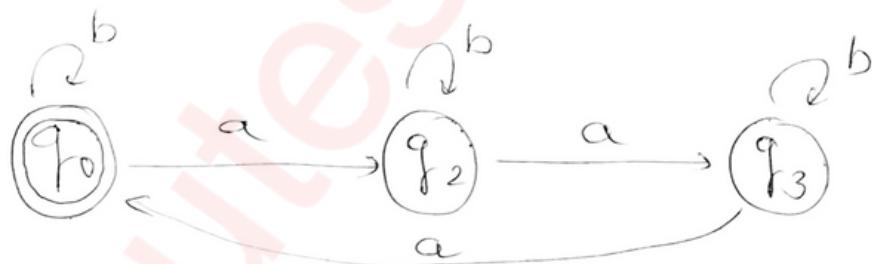


Q: Divisible (mod) problems.

$$n(a) \bmod 3 = 0$$

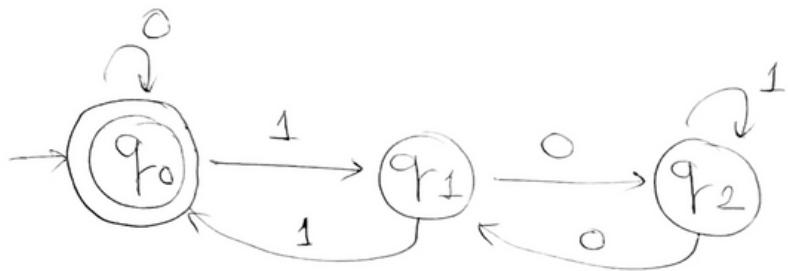
$$\Sigma = \{a, b\}$$

$$L = \{ \epsilon, aaq, aaaaqaa, bbbbaaq, \dots \}$$



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

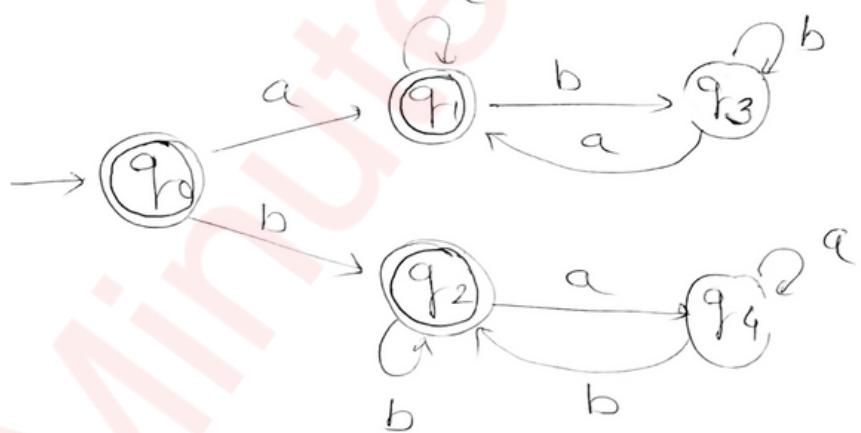
Binary no. divisible by 3.



	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

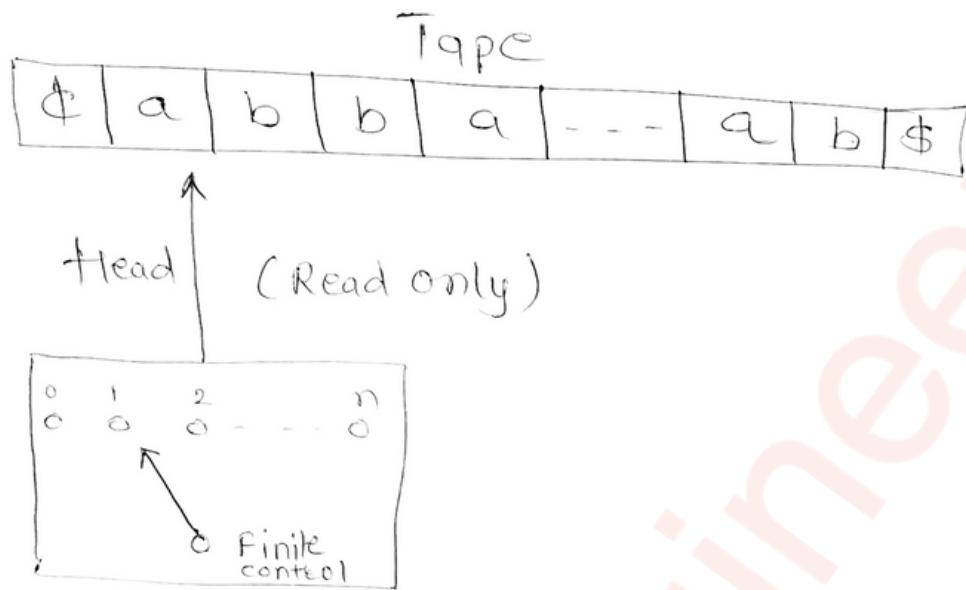
Q : starts & ends with same symbol

$$L = \{aa, aaaa, abab, bab, \dots\}$$



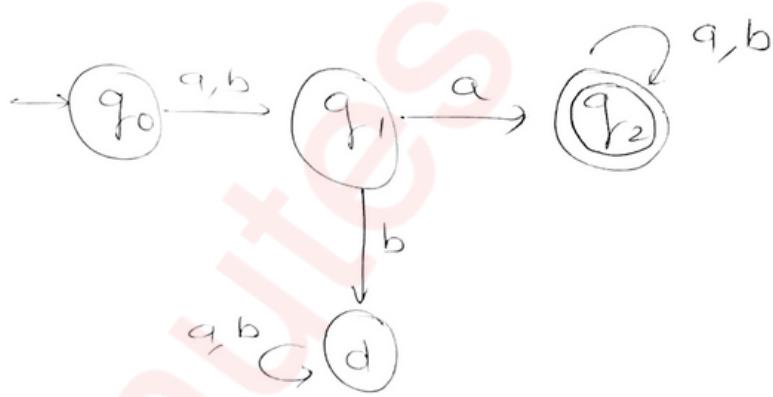
Completing final \Rightarrow Non final.

° FA Block Diagram

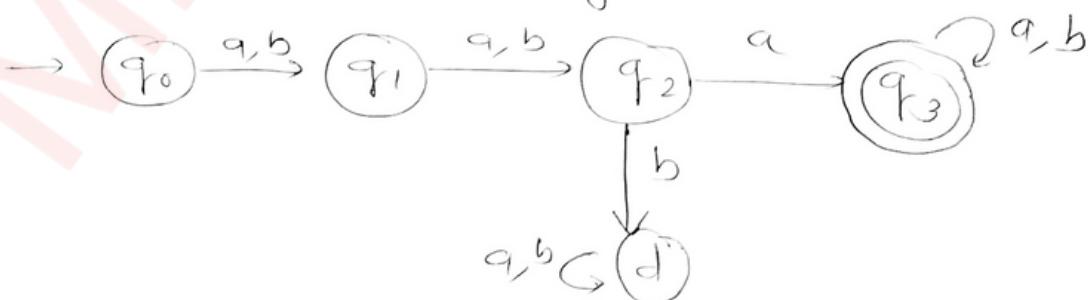


Q: $\Sigma = \{a, b\}$

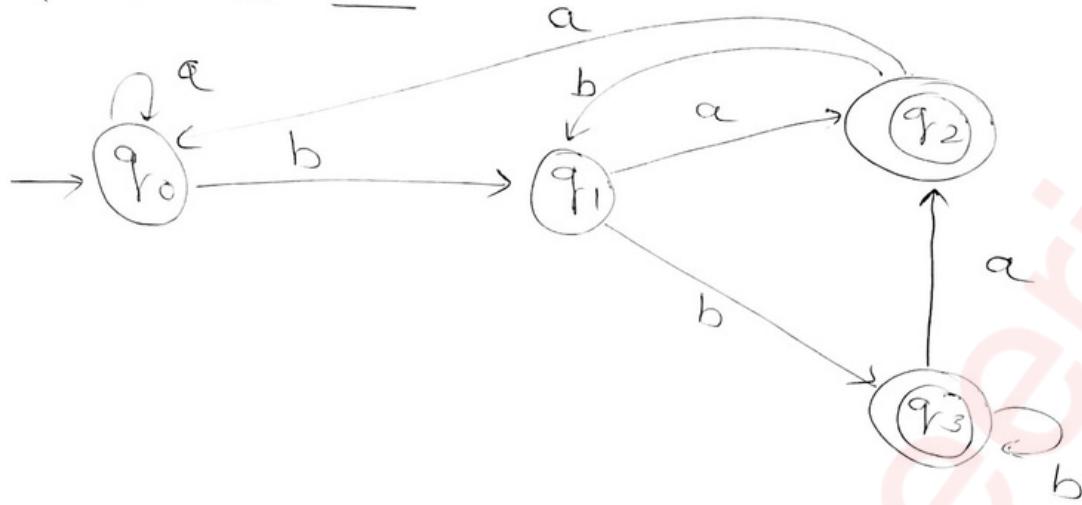
2nd is 'a' (from Left)



3rd is 'a' (from Left)



Q: 2nd last is 'b'



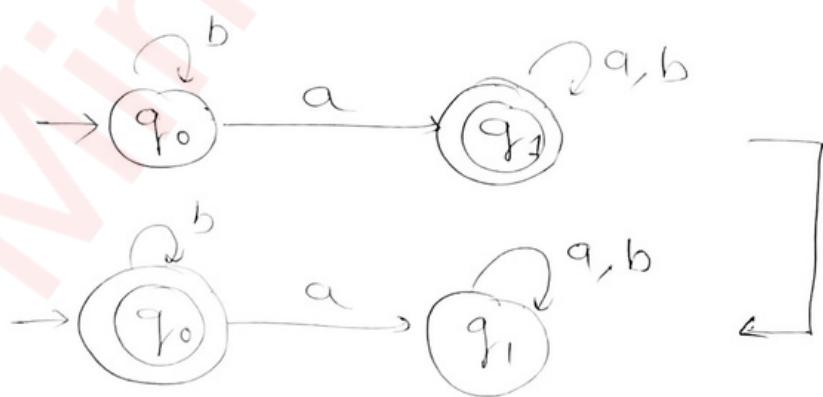
Q {

S	Σ	
	b	a
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

$2^2 = 4$ states

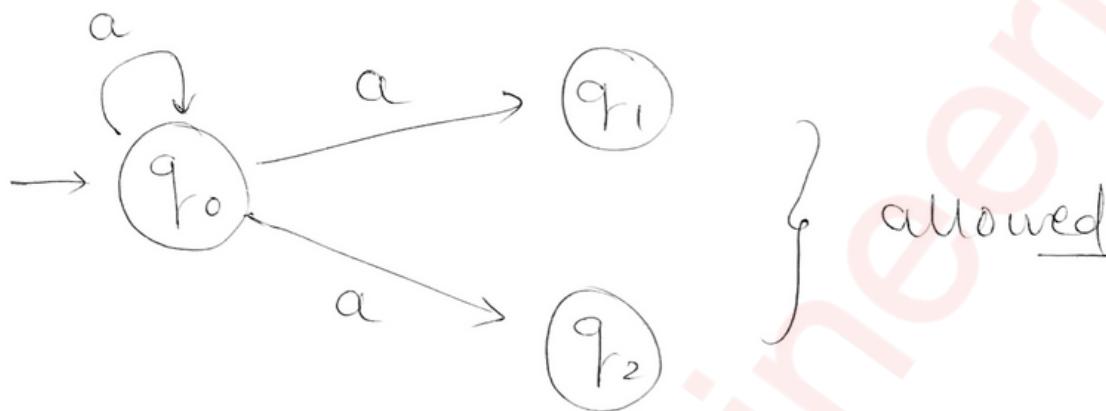
• Complement of DFA.

→ does not contain 'a'



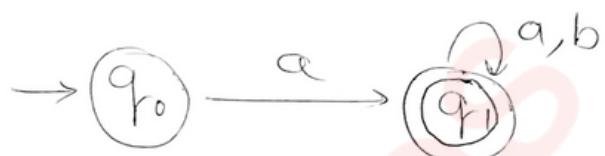
• NFA : Non Deterministic FA

$$\{ Q, \Sigma, \delta, q_0, F \}$$

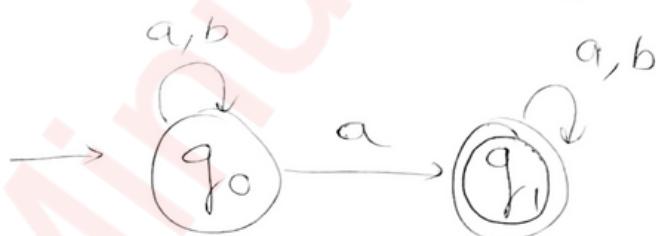


$$\delta: Q \times \Sigma \rightarrow 2^Q$$

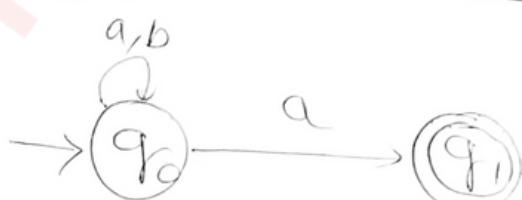
ϱ : Starts with 'a'



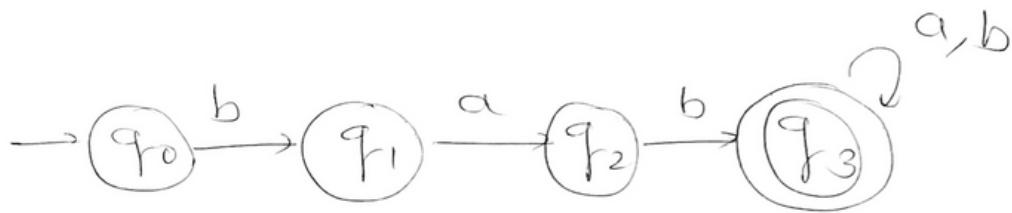
ϱ : Contains 'a'



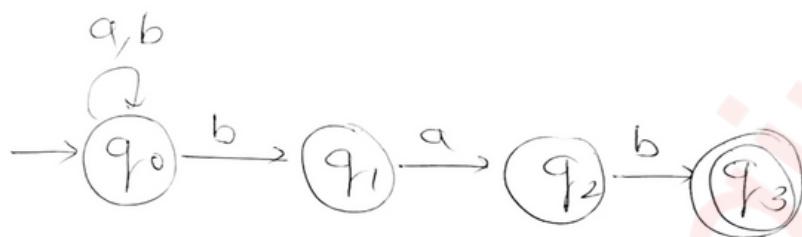
ϱ : Ends with 'a'



Q: Contains 'bab'



Q: Ends with 'bab'

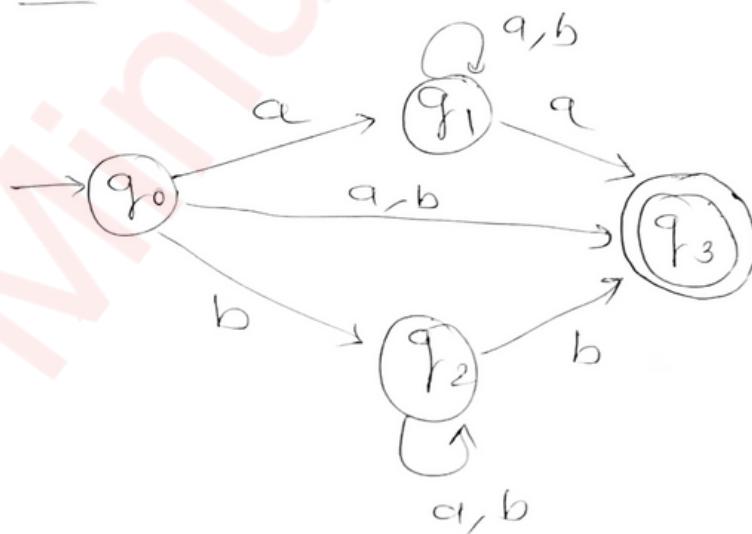


Invalid string cannot be accepted.

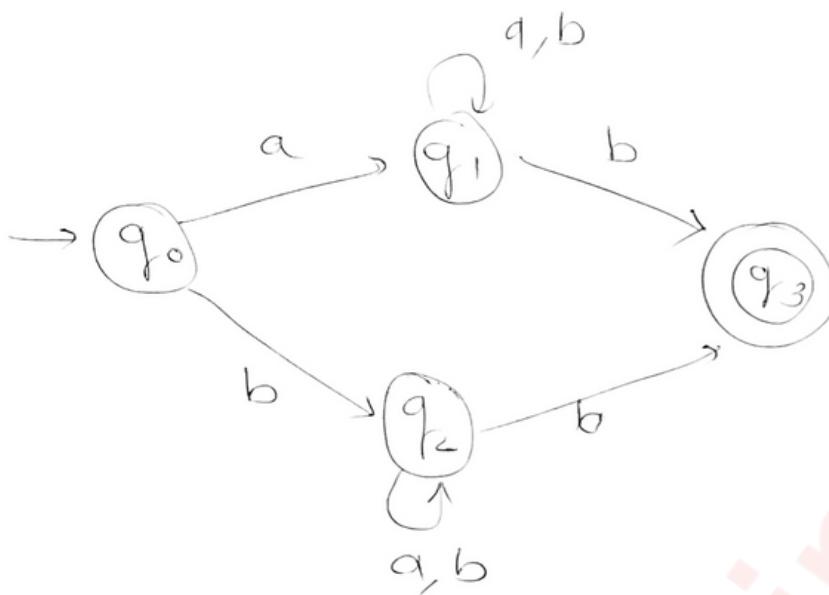
Valid string $\xrightarrow{\text{can be}} \text{Accepted}$
 $\xrightarrow{\text{Not be}} \text{Accepted}$.

eg: aabab. it can go to q_3
or q_0

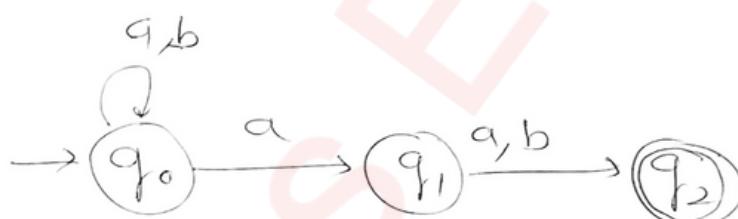
Q: Start end with same symbol.



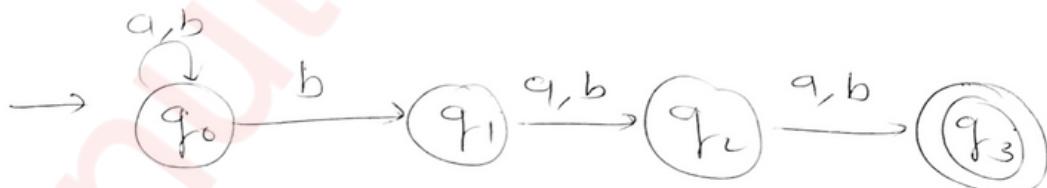
Q: Start & End with different symbol.



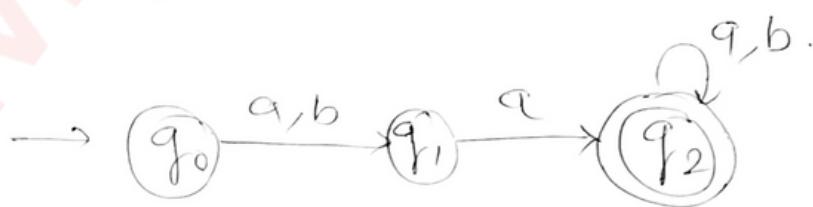
Q: 2nd Last is 'a'



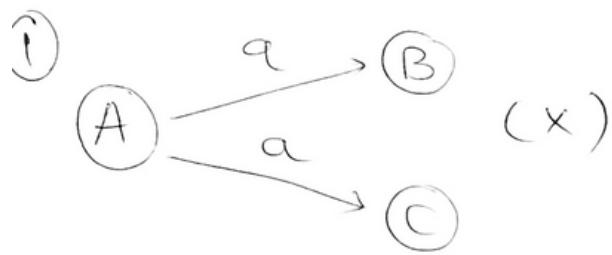
Q: 3rd Last is 'b'



Q: 2nd from Left 'a'



DFA



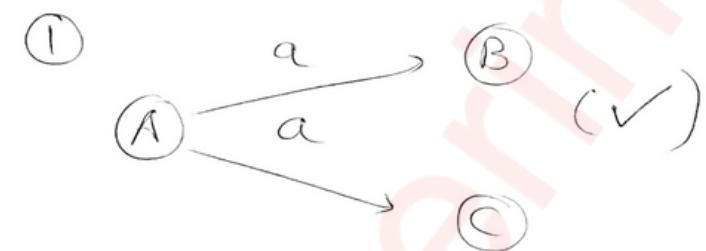
② Difficult to construct

③ All DFA are NFA

④ $Q \times \Sigma \rightarrow Q$
(unique)

⑤ Need more space

NFA

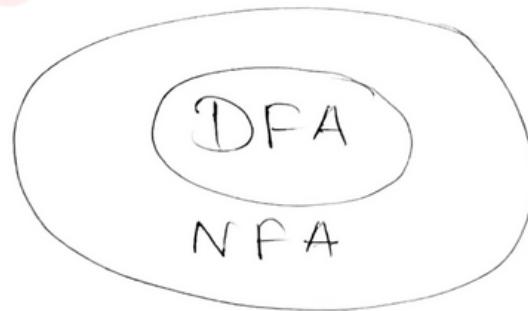


② Easy to construct

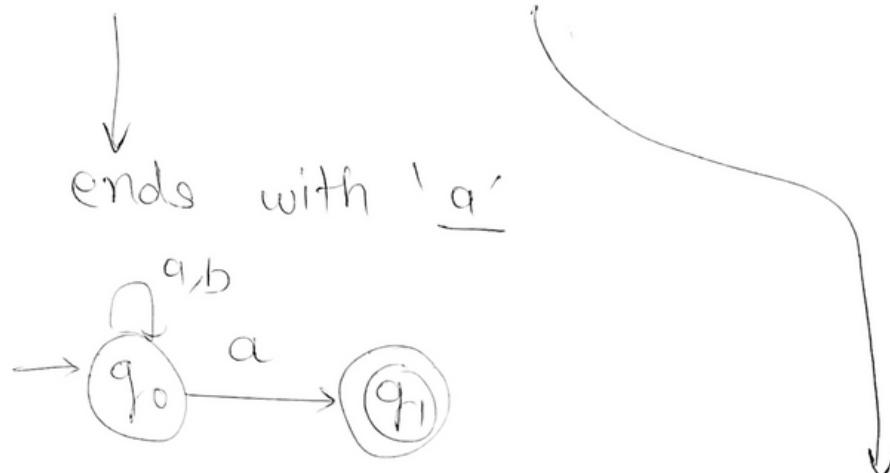
③ Not All NFA are DFA

④ $Q \times \Sigma \rightarrow 2^Q$

⑤ Need Less space

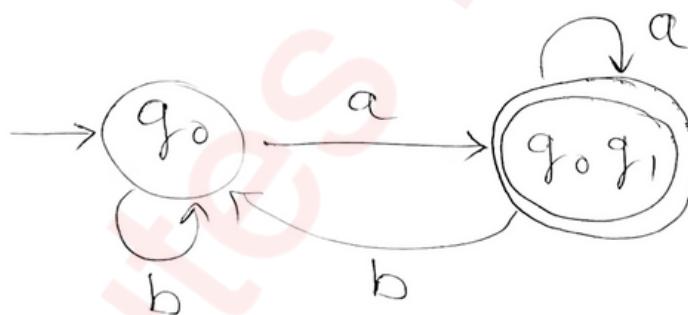


• NFA to DFA conversion



	a	b		a	b
q_0	{ q_0, q_1 }	{ q_0 }	*	[q_0, q_1]	[q_0]
q_1	{ ϕ }	{ ϕ }	*	[q_0, q_1]	[q_0]

$x [q_1]$
↳ not reachable.



NFA has 'n' states then .

DFA has $1 \leq m \leq 2^n$
m states

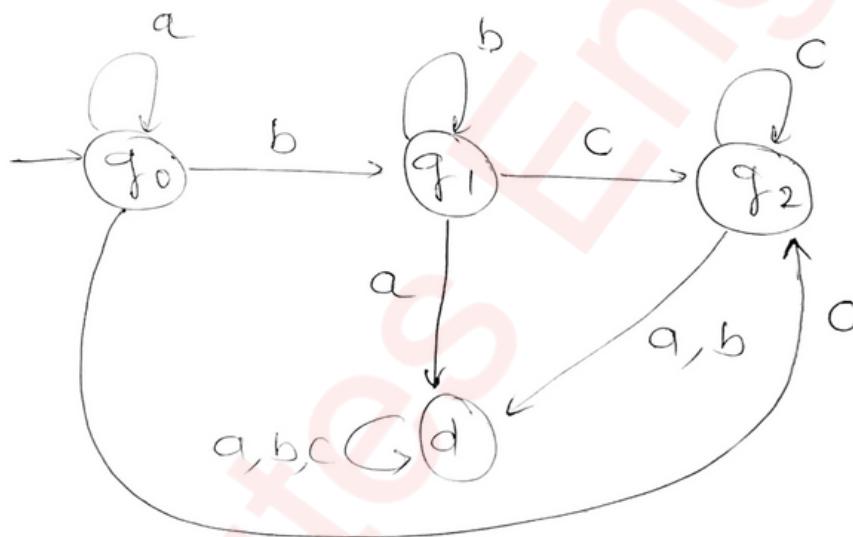
• ϵ -NFA (NFA with ϵ move)

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

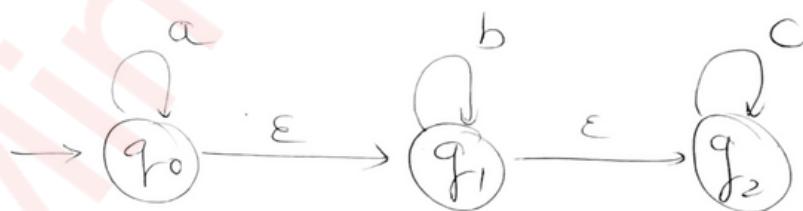
$$\delta: Q \times \{\Sigma \cup \epsilon\} \rightarrow 2^Q$$



Eg: DFA for $[a^x b^y c^z \mid x, y, z \geq 0]$



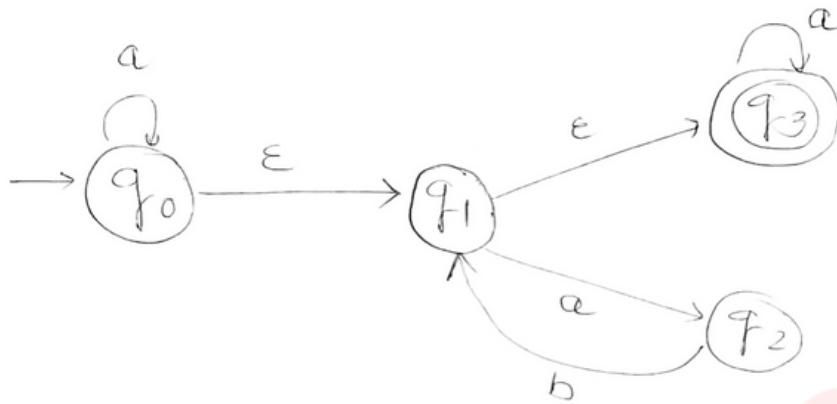
But ϵ -NFA



$$\begin{aligned} \text{closure}(q_1) &\left\{ \begin{aligned} q_0 &= q_0, q_1, q_2 \\ q_1 &= q_1, q_2 \\ q_2 &= q_2 \end{aligned} \right. \end{aligned}$$

E-NFA to NFA conversion

Eg:-



$$Q = \{q_0, q_1, q_2, q_3\} \quad \Sigma = \{0, 1\}$$

$$F = \{q_3\}$$

① Find Null closure of all states.

$$q_0 \rightarrow \{q_0, q_1, q_3\}$$

$$q_1 \rightarrow \{q_1, q_3\}$$

$$q_2 \rightarrow \{q_2\}$$

$$q_3 \rightarrow \{q_3\}$$

② Check transition now for given Input i.e $\delta[\epsilon\text{-closure}(q_i), \alpha]$ & take again a ϵ -closure of that.

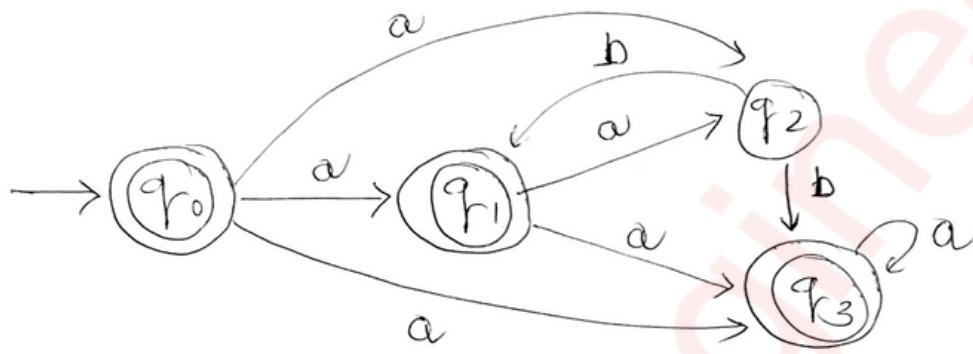
$$\text{eg: } \delta(q_0, a) \Rightarrow \delta(\{q_0, q_1, q_3\}, a)$$

$$\downarrow$$

$$\epsilon\text{-closure}\{q_0, q_2, q_3\}$$

$$\{q_0, q_1, q_2, q_3\}$$

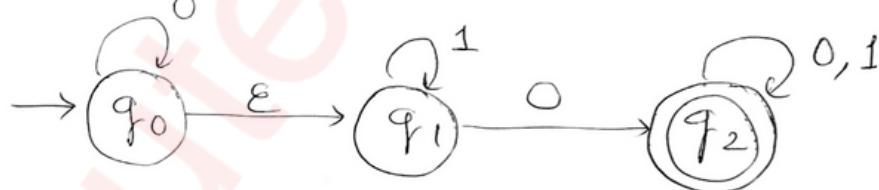
	<u>a</u>	<u>b</u>
q_0	$\{q_0, q_1, q_2, q_3\}$	
q_1	$\{q_2, q_3\}$	
q_2		$\{q_1, q_3\}$
q_3	$\{q_3\}$	



q_0, q_1 & q_3 null closure contains E
But for q_2 not so not a final state

Conversion of ε-NFA to DFA

Eg:



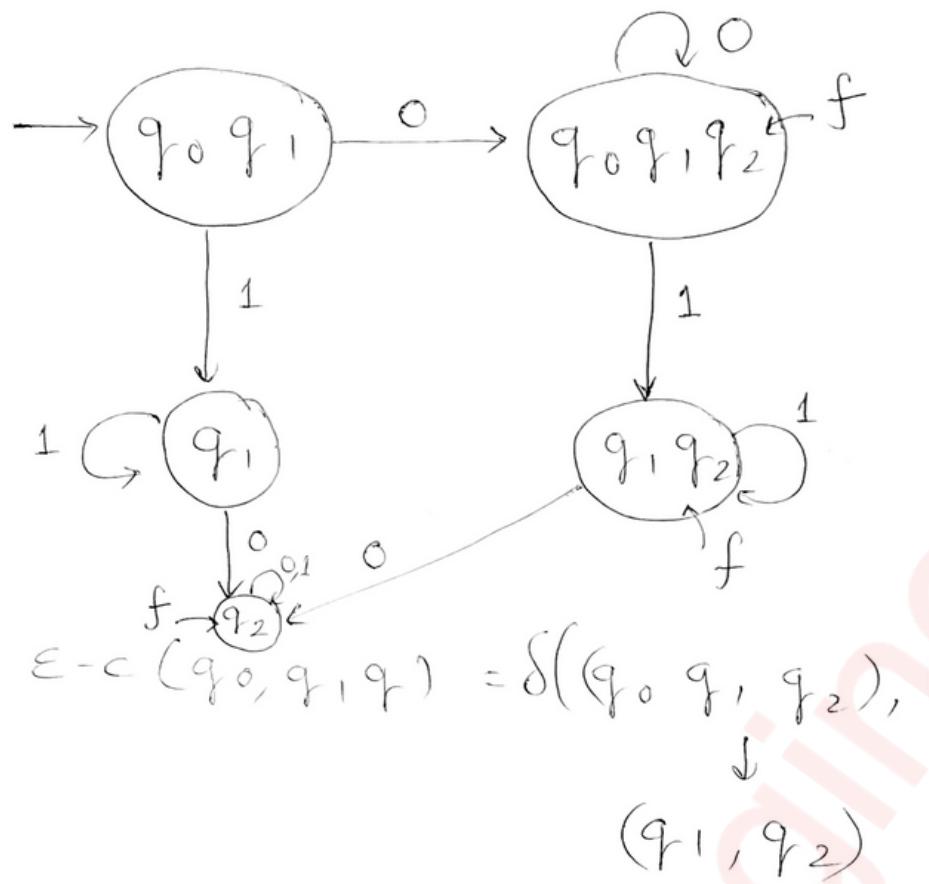
$\rightarrow \epsilon\text{-closure of } q_0 = q_0 q_1$

$$\delta((q_0, q_1), 0)$$

$$\downarrow \\ \epsilon\text{-c}(q_0 q_2) \Rightarrow q_0, q_1, q_2$$

$$\& \delta((q_0, q_1), 1)$$

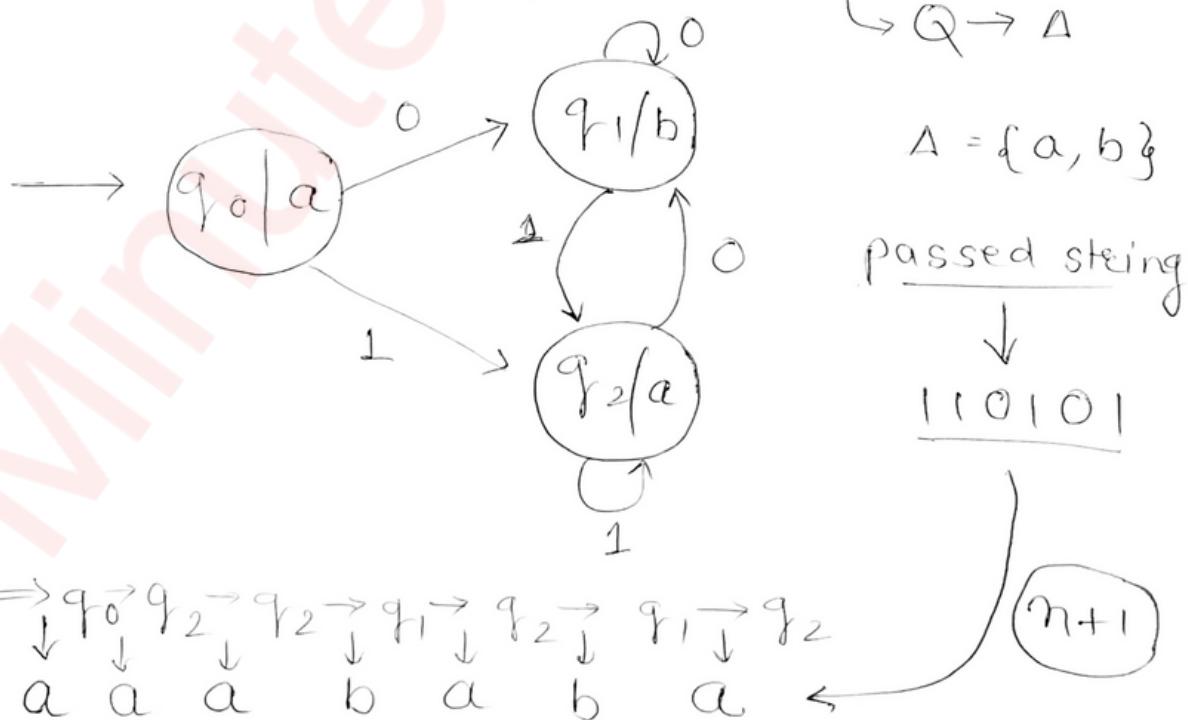
$$\hookrightarrow \epsilon\text{-c}(q_1) \rightarrow q_1$$



• Moore m/c

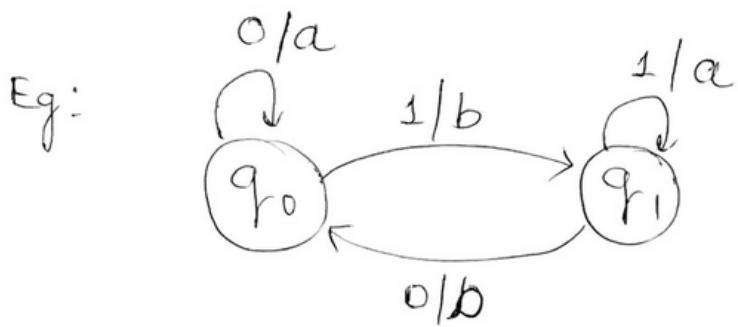
$$M = \{ Q, \Sigma, A, \delta, \lambda, q_0 \}$$

$\xrightarrow{Q \times \Sigma \rightarrow Q}$
 \downarrow
 o/p symbol \downarrow
 $\xrightarrow{\text{o/p } f^n}$
 \downarrow
 $\xrightarrow{Q \rightarrow A}$



• Mealy M/C

$$\hookrightarrow \{ Q, \Sigma, \Delta, \delta, \lambda, q_0 \}$$



Passing string: 001101

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

Transition table

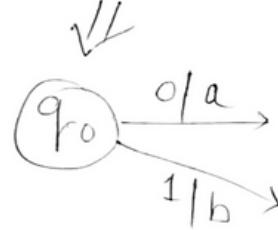
CS	NS	
	$I/P = 0$	$I/P = 1$
q_0	$q_0 a$	$q_1 b$
q_1	$q_0 b$	$q_1 a$

Moore m/c { mealy m/c

① O/p → PS



① O/p → PS & PI/p



② I/p string length = n

$$\text{so } o/p = n+1$$

② I/p string length = n
so o/p = n

③ $\lambda : Q \times \Sigma \rightarrow \Delta$

③ $\lambda : Q \rightarrow \Delta$

④ O/p is placed on State

④ O/p is placed on transition

⑤ I/p changes, O/p doesn't change

⑤ I/p change, O/p also change.

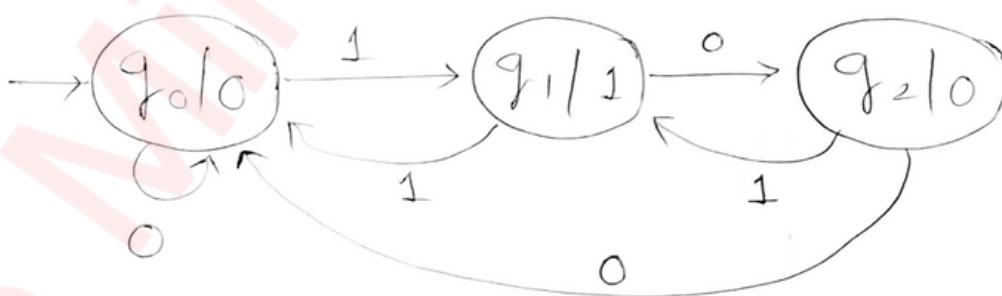
⑥ Easy to design

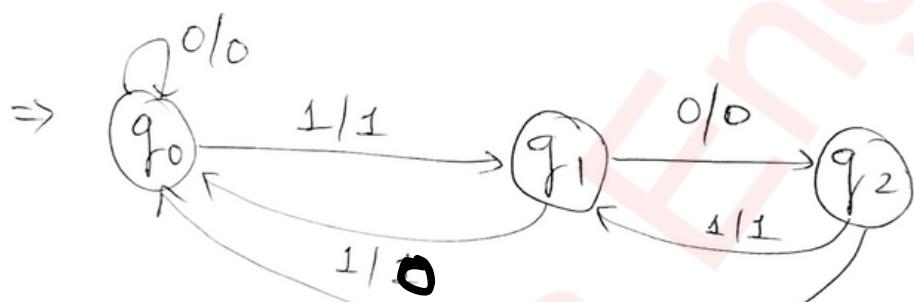
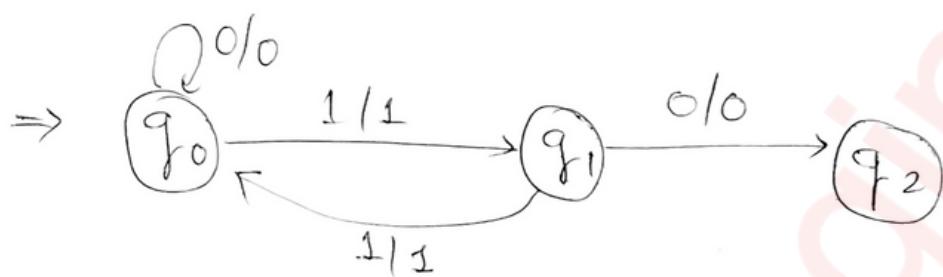
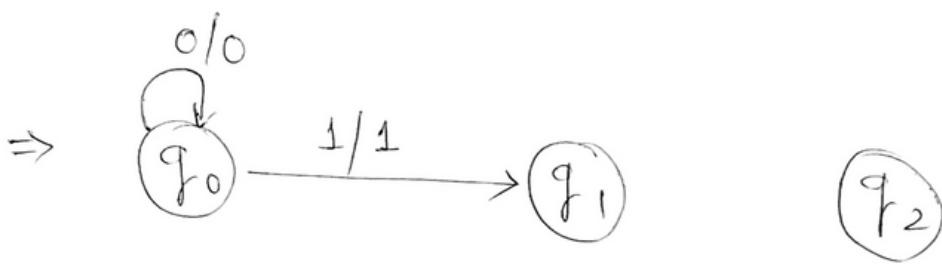
⑥ Difficult to design

Moore to mealy Conversion



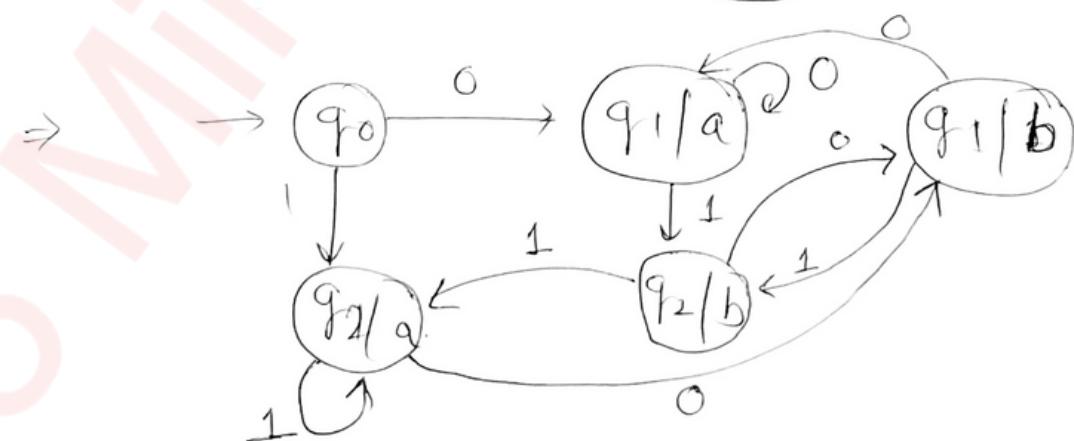
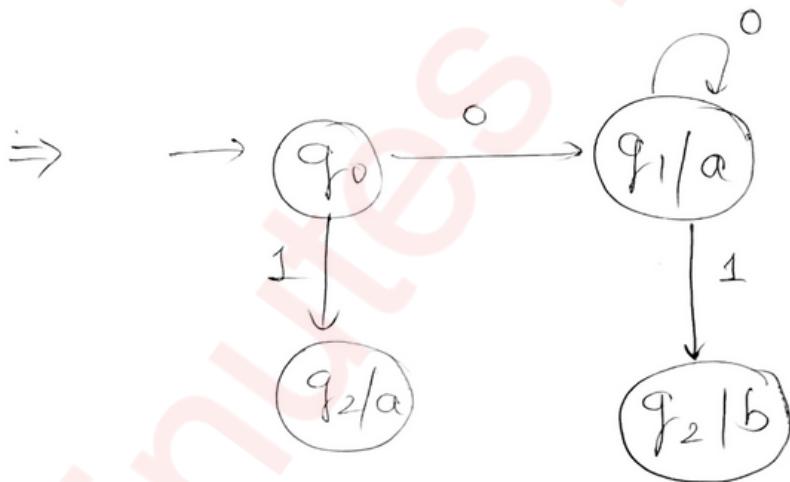
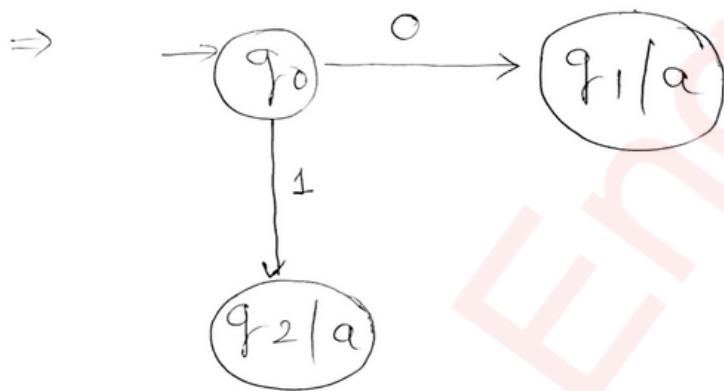
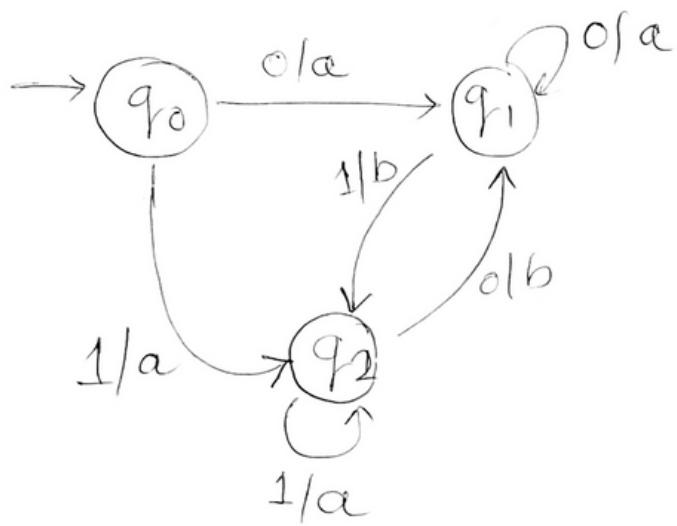
Given



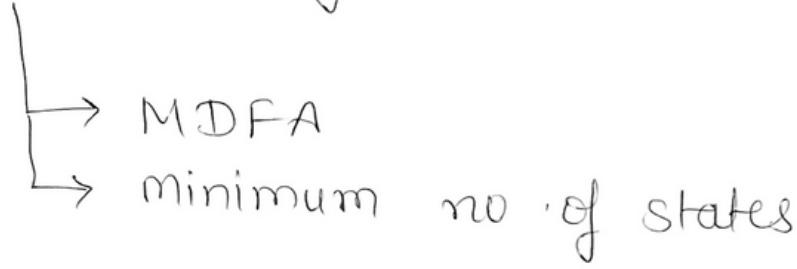


<u>CS</u>	<u>NS</u>	<u>IP = 0</u>	<u>IP = 1</u>		<u>CS</u>	<u>NS</u>	<u>O/P</u>
q_0	q_0	0	q_1	1	q_0	$\frac{0}{q_0}$	0
q_1	q_2	0	q_0	0	q_1	$\frac{1}{q_1}$	1
q_2	q_0	0	q_1	1	q_2	$\frac{0}{q_2}$	0

Mealy to Moore Conversion

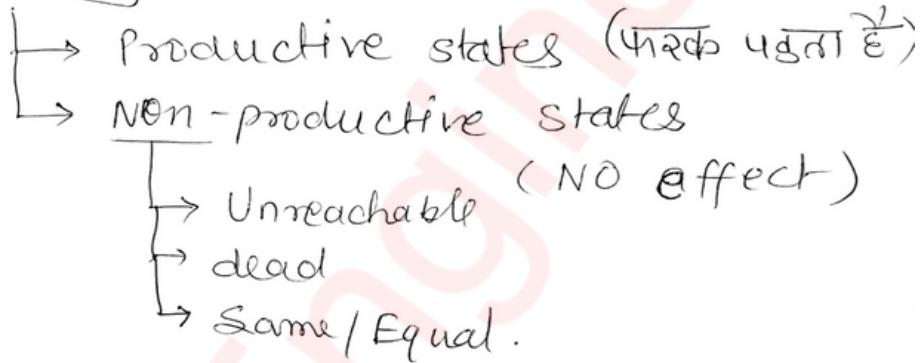


• Minimization of finite Automata

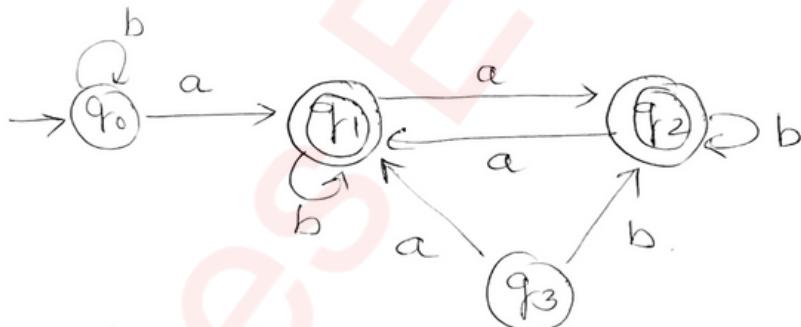


ϵ -NFA — NFA — DFA — MDFA

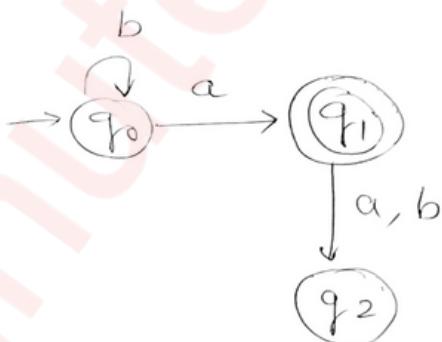
• Productivity



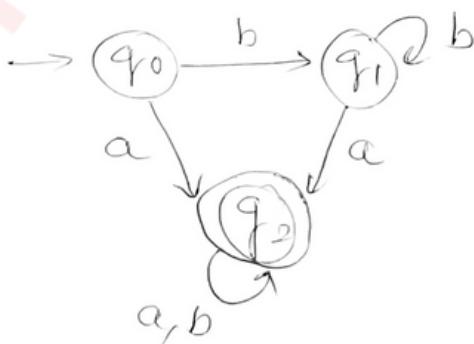
Eg: ①



②



③



• Regular Expressions

↳ Used to denote RL

describle
represent

Regular Grammer

↳ Generates RL

• Regular Language is accepted by FA

⇒ { Σ , ϵ , ϕ }

e.g:-

RE

a

RL

{a}

b

{b}

a.b

{ab} no choice

a+b

{a, b} choice

a^*

{ ϵ , a, aa, aaa, aaaa, ---}

a^+

{ a, aa, aaa, --- }

a + b + c

{ a, b, c }

RL

→ RE1

RE2

→ RE2

→ RL1

(✓)

→ RL2
(X)

Q: RE for $\Sigma = \{a, b\}$ where every string starts with 'a'

$$\Rightarrow a \cdot (a+b)^*$$

Q: Contains 'a'

$$\Rightarrow (a+b)^* \cdot a \cdot (a+b)^*$$

Q: ends with 'a'

$$\Rightarrow (a+b)^* \cdot a$$

Q: start & end with a.

$$\Rightarrow a \cdot (a+b)^* \cdot a$$

Q: start & end with same symbol.

$$a + a \cdot (a+b)^* \cdot a + b + b \cdot (a+b)^* \cdot b$$

Q: start & end with different symbol.

$$a \cdot (a+b)^* \cdot b + b \cdot (a+b)^* \cdot a$$

Q: Starts with [aba or bab] _{sub string}

$$\Rightarrow (aba + bab) \cdot (a+b)^*$$

Q: Ends with substring aba or bab

$$\Rightarrow (a+b)^* \cdot (aba + bab)$$

Q: Length based: $\leq(a, b)$

$$\circ |s| = 2$$

$$\circ |s| \leq 2$$

$$\Rightarrow (a+b)^2$$

$$\circ |s| \geq 2$$

$$\left. \begin{array}{l} \varepsilon + \\ (a+b) \\ (a+b)^2 \end{array} \right\} \text{choice}$$

$$(a+b)^2 \cdot (a+b)^*$$

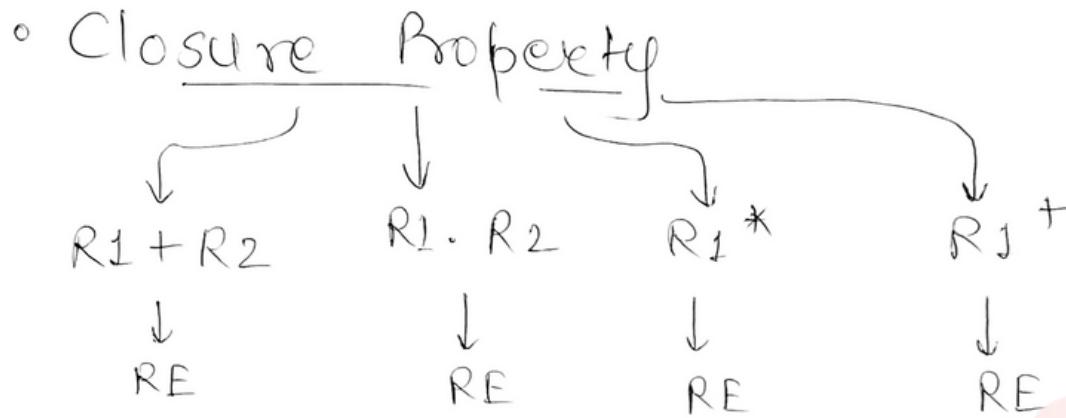
Q: $\Sigma = \{a, b\}$

2nd last is 'a'

$$\Rightarrow (a+b)^* a (a+b)$$

Q: 2nd from Left is 'a'

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



◦ Associative Property

$$(R1 + R2) + R3 = R1 + (R2 + R3)$$

$$(R1 \cdot R2) \cdot R3 = R1 \cdot (R2 \cdot R3)$$

◦ Distributive Property

$$R1 \cdot (R2 + R3) = R1 \cdot R2 + R1 \cdot R3$$

$$(R1 + R2) \cdot R3 = R1 \cdot R3 + R2 \cdot R3$$

◦ Commutative Property

$$R1 + R2 = R2 + R1$$

$$R1 \cdot R2 \neq R2 \cdot R1$$

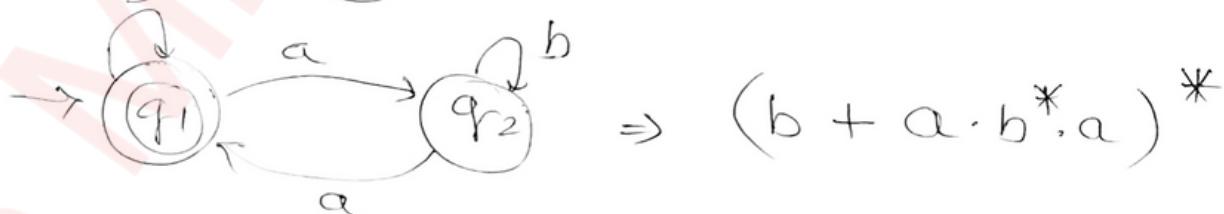
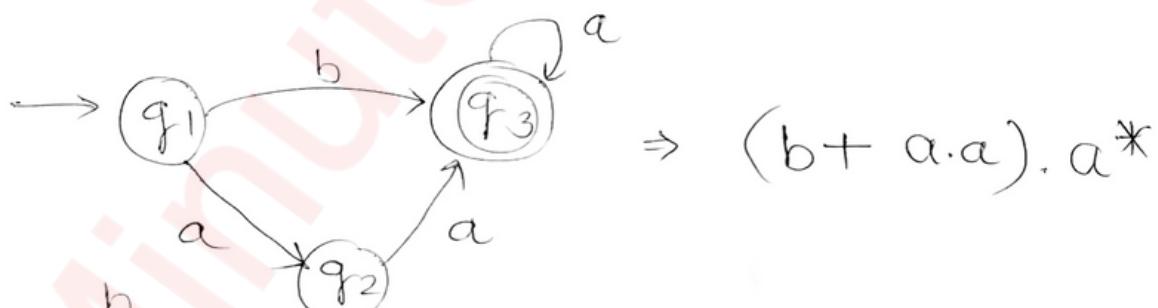
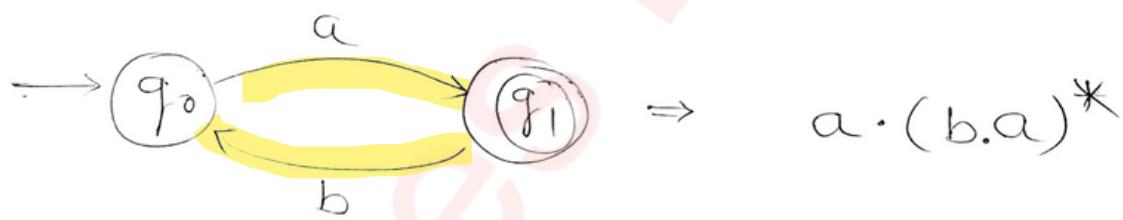
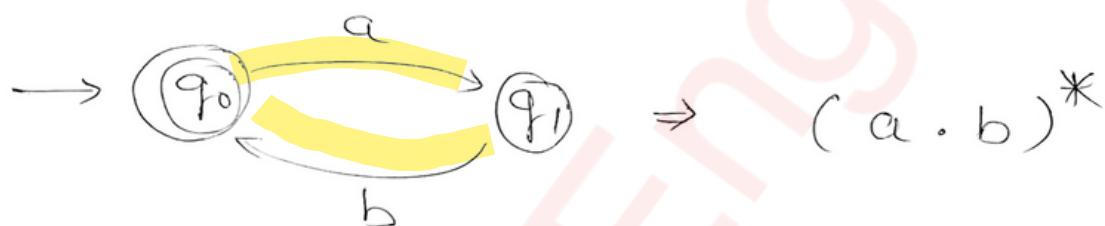
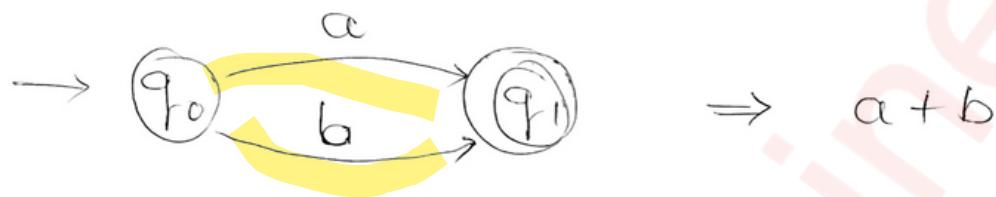
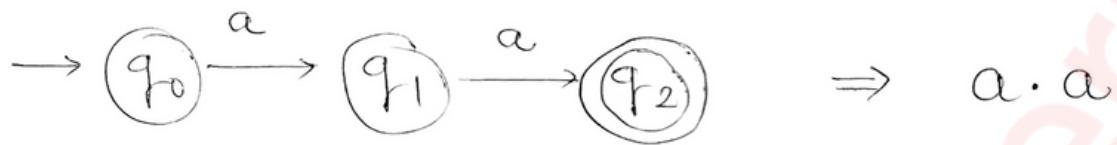
◦ Identity Property

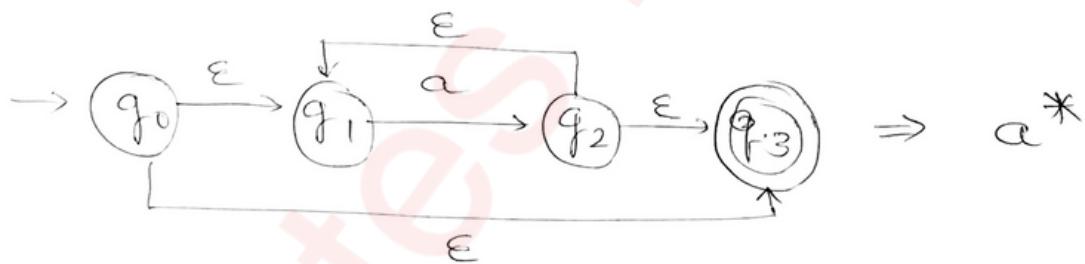
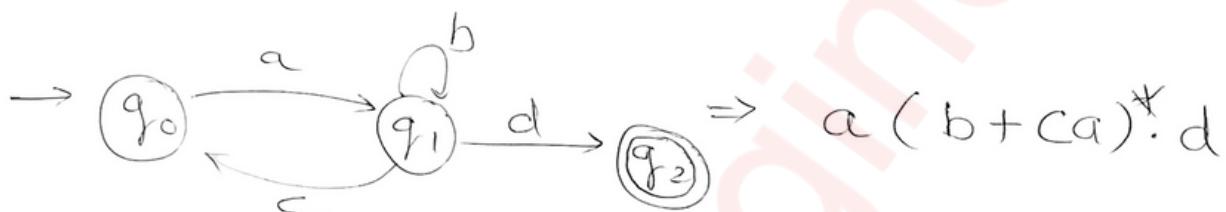
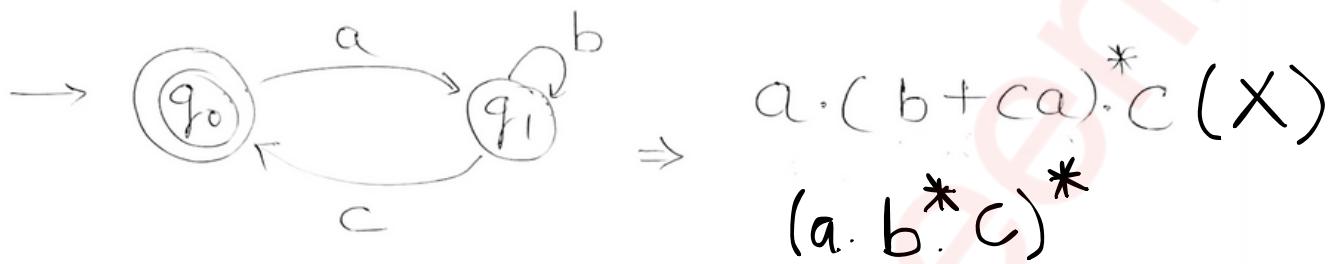
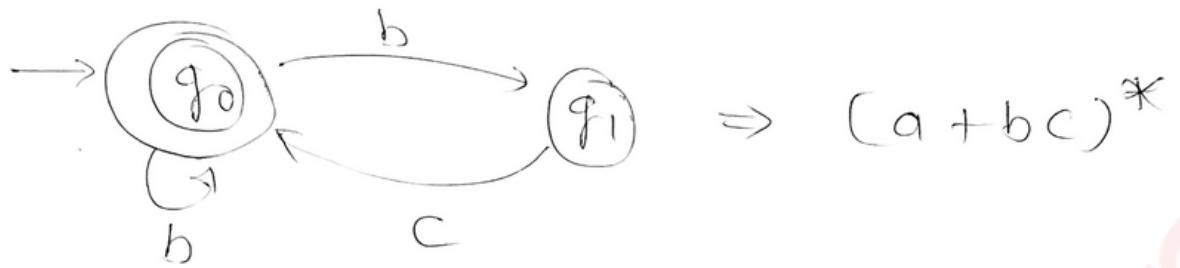
$$R \cdot E = R \quad R + \phi = R$$

◦ Idempotent Property

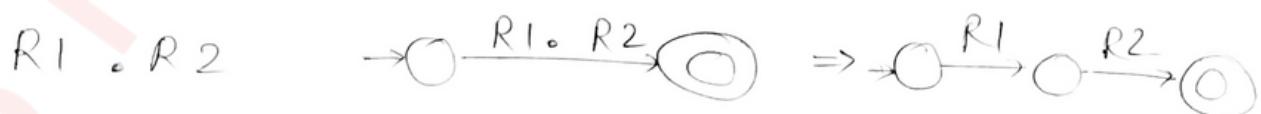
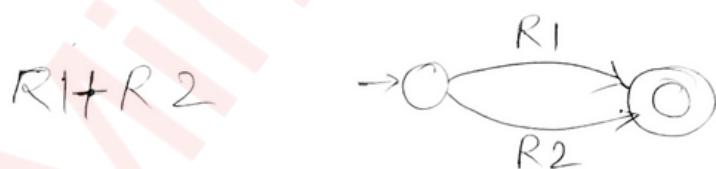
$$R1 + R1 = R1 \quad R1 \cdot R1 \neq R1$$

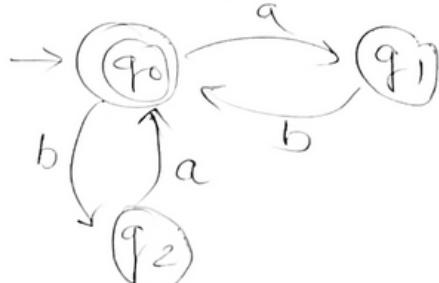
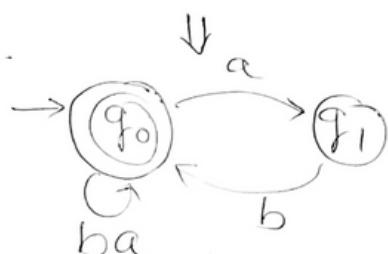
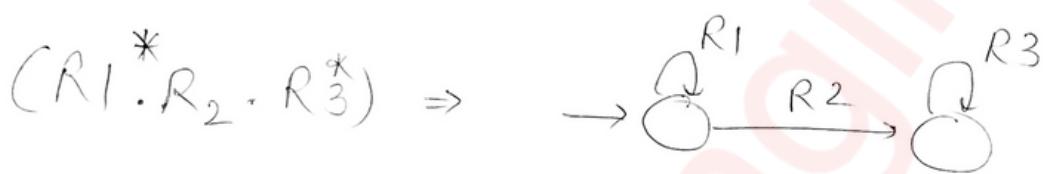
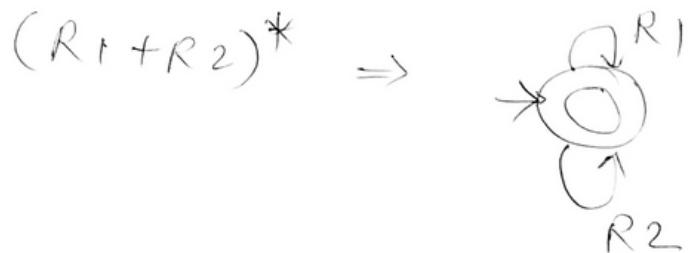
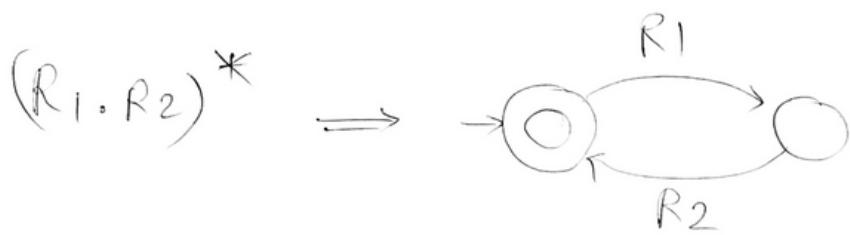
• Convert FA $\xrightarrow{\text{to}}$ RE
 Given



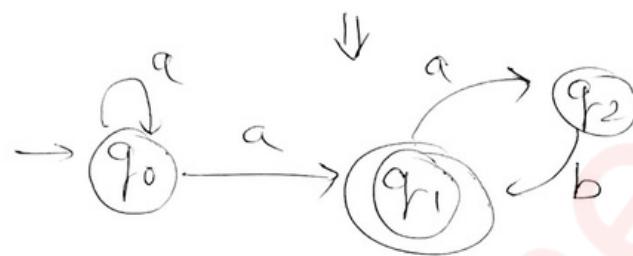
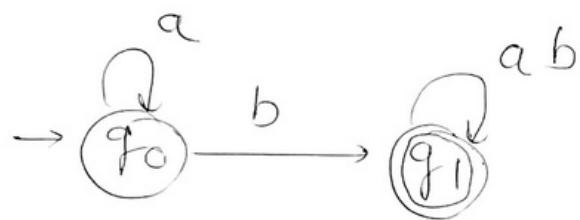


• Convert RE $\xrightarrow{\text{to}}$ FA

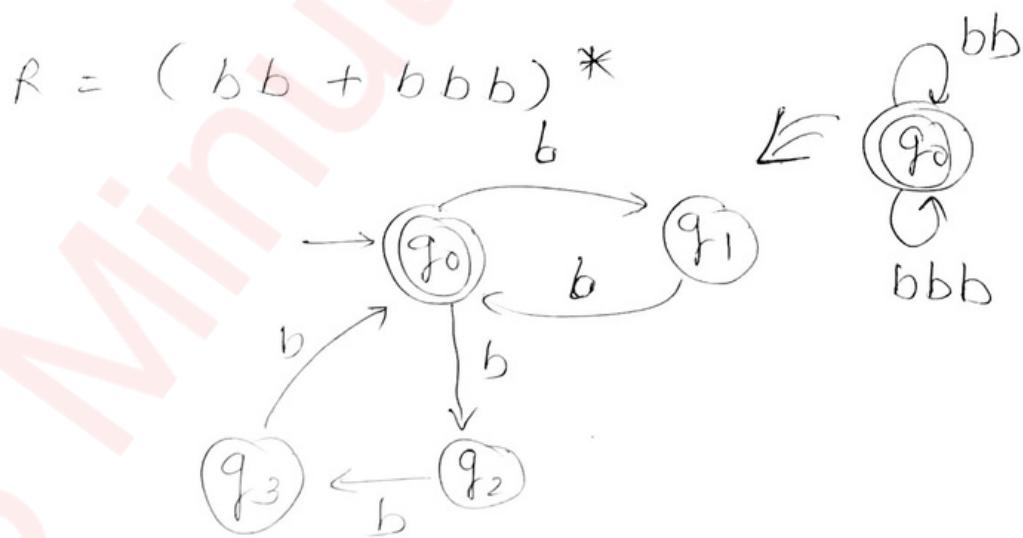
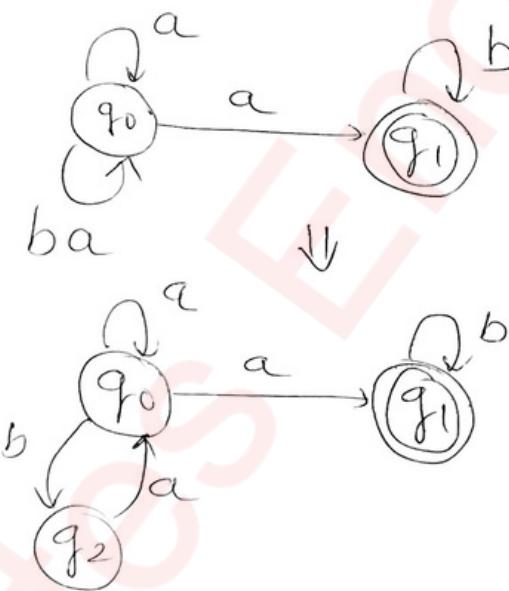




$$R = a^* b (ab)^*$$

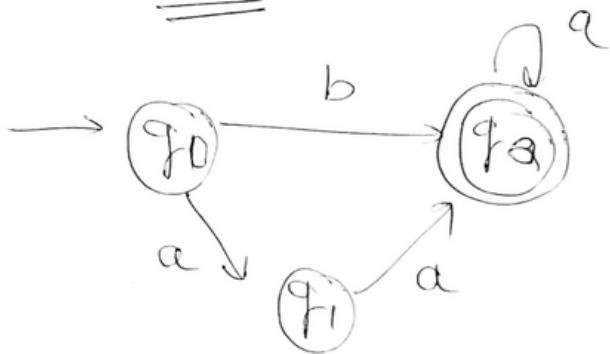


$$R = (a + ba)^* ab^*$$



- Arden's theorem: $\underline{RE} \leftarrow \underline{FA}$
↳ on DFA

e.g:



form eqⁿs:

$$q_0 = \epsilon \rightarrow ①$$

$$q_1 = q_0 \cdot a \rightarrow ②$$

$$q_2 = q_0 \cdot b + q_1 \cdot a + q_2 \cdot a \rightarrow ③$$

Bring final state in form $R = Q + RP$

$$q_1 = \epsilon \cdot a = a \rightarrow ④$$

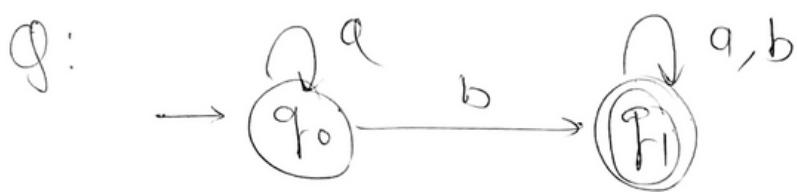
$$(R = Q P^*)$$

↓
soln ↗

Use ①, ④ in ③

$$\begin{aligned}
 q_2 &= \epsilon \cdot b + a \cdot a + q_2 \cdot a \\
 &= (b + a \cdot a) + q_2 \cdot a \Rightarrow (Q + RP)
 \end{aligned}$$

$$R = (b + a \cdot a) \cdot a^*$$



$$q_0 = \varepsilon + q_0 \cdot a \rightarrow ①$$

$$q_1 = q_0 \cdot b + q_1(a+b) \rightarrow ②$$

final state in $R = Q + RP$ $\xrightarrow{\text{soM}}$
 $(\text{in } ①) \uparrow$ $R = Q P^*$

$$q_0 = \varepsilon \cdot a^* = a^* \rightarrow ③$$

use ③ in ②.

$$q_1 = \underbrace{a^* \cdot b}_Q + \underbrace{q_1(a+b)}_{R P}$$

$$R = a^* b \cdot (a+b)^*$$



$$q_0 = \varepsilon + q_1 \cdot b \rightarrow ①$$

$$q_1 = q_0 \cdot a \rightarrow ②$$

use ① in ②.

$$q_1 = (\varepsilon + q_1 \cdot b) \cdot a$$

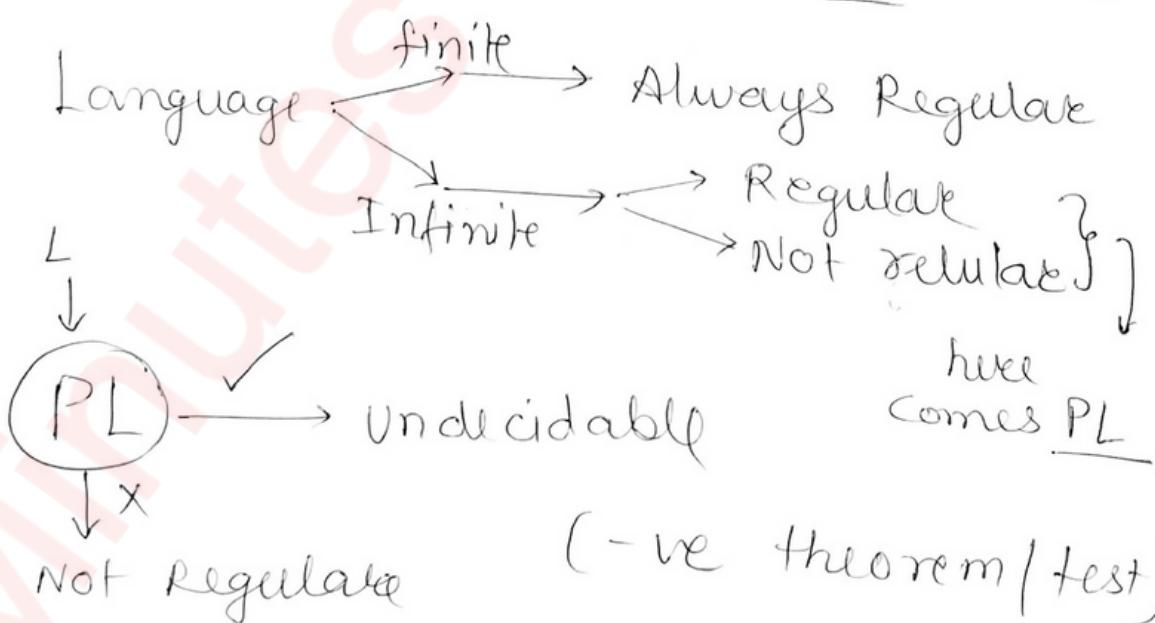
$$= a \cdot \varepsilon + q_1 \cdot b \cdot a$$

$$= \underbrace{a}_{Q} + \underbrace{q_1}_{R} \underbrace{(b \cdot a)}_{P}$$

$$\therefore R = a \cdot (b \cdot a)^*$$

• Pumping Lemma:

↳ Check if 'L' is RL or not



Eg: $L = a^n b^n c^n \quad n \geq 0$

$\omega: \frac{aa}{x} \frac{bb}{y} \frac{cc}{z}$

\downarrow Pump $y \rightarrow y^i (bb)^{j-1}$
 \downarrow $(bb)^2$
 $\frac{aa}{x} \frac{bb \cancel{bb}}{y} \frac{cc}{z} \notin L$

So, it's not Regular.

Eg: $L = a^n b^{2n} \quad n \geq 0$

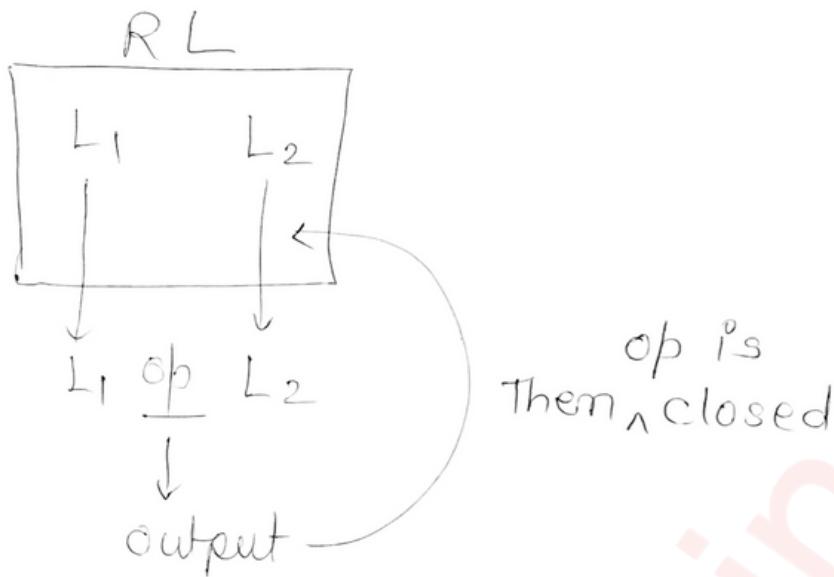
$\omega: \frac{a}{x} \frac{ab}{y} \frac{bb}{z}$

\downarrow Pump y

$\frac{a}{x} \frac{ab ab}{y} \frac{bb b}{z} \notin L$

So, it's not Regular.

Closure Properties of RL



- ① CLOSURE (L^*)
- ② UNION ($L_1 \cup L_2$) (+)
- ③ INTERSECTION ($L_1 \cap L_2$)
- ④ CONCATENATION ($L_1 \cdot L_2$)
- ⑤ COMPLEMENTATION (\bar{L})
- ⑥ DIFFERENCE ($L_1 - L_2$)
- ⑦ REVERSAL (L^R)
- ⑧ HOMOMORPHISM
- ⑨ REVERSE HOMOMORPHISM
- ⑩ SUBSTITUTION.
- ⑪ INFINITE UNION
- ⑫ INIT OPERATION.

	<u>Regulate L</u>	<u>DCFL</u>	<u>CFL</u>	<u>CSL</u>	<u>REC</u>	<u>RE</u>
U		X				
n		X	X			
L ^c			X			X
-		X	X			X
.						
*		X				
+		X				
L ^R						
H		X		X	X	
IH						

DCFL → Deterministic Context free Language

CFL → Context free L

CSL → Context sensitive L

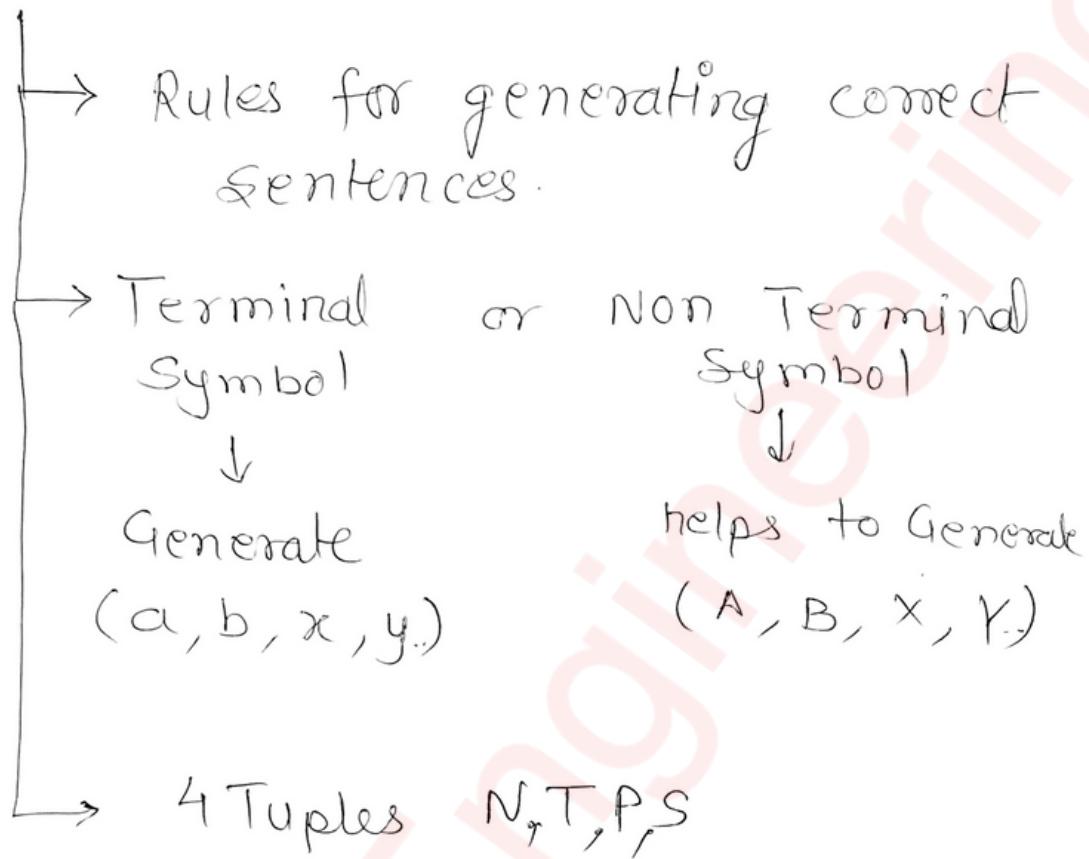
REC → Recursive

RE → Recursive Enumerable.

• Decidability & Undecidability Table

	RL	DCFL	CFL	CSL	REC	RE
Membership problem	✓	✓	✓	✗	✗	✗
Emptiness problem	✓	✓	✓	✗	✗	✗
Equality Problem	✓	✓	✗	✗	✗	✗
Ambiguity	✓	✓	✗	✗	✗	✗
Infiniteness Problem	✓	✓	✓	✗	✗	✗
Complete	✓	✓	✗	✗	✗	✗
Subset Problem	✓	✗	✗	✗	✗	✗

Grammars



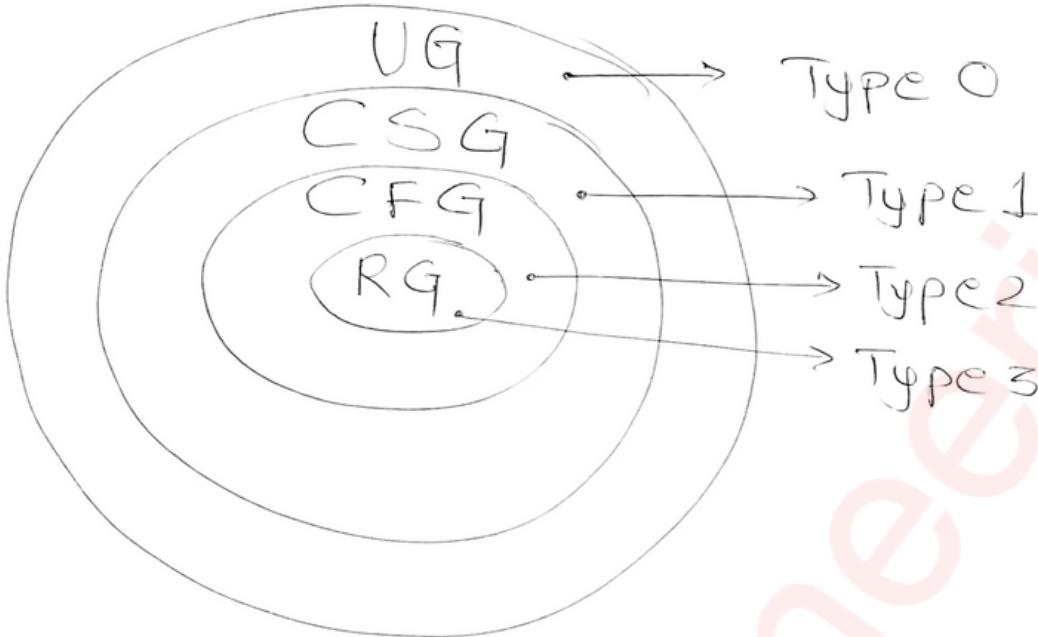
N : Non-Terminal symbols (set of)

T : Finite set of Terminal symbols.

P : Production Rules

S : Start symbol.

Type of ' P '	Types	No. of strings
	$D \uparrow T$	
• Type-0	• Ambiguous	• Recursive
• Type-1	• Unambiguous	• Non-Recursive
• Type-2		
• Type-3		



RG Accepted by FA

CFG Accepted by Push down Automata

CSG Accepted by Linear Bound Automata

UR Recognized by Turning Machine.

Grammatic

Type 0

Type 1

Type 2

Type 3

Language

Recursively
Enumerable

CSL

CFL

RL

Automata

TM

LBA

PDA

FA

'P'

$Sxy \rightarrow yx$
 $X \rightarrow S$

$S \rightarrow XY$
 $XY \rightarrow xyz$

$S \rightarrow XY$
 $X \rightarrow xc$
 $Y \rightarrow y$
 $S \rightarrow x$

° Type - 0 Grammar (Unrestricted)

→ No restriction on Production Rule.

$$\alpha \longrightarrow \beta$$

$$\alpha \in \{TUV\}^* \quad \beta \in \{TUV\}^*$$

at least '1' variable

° Type - 1 Grammar (CSG)

↳ Non-Conteacting grammar

$$\alpha \longrightarrow \beta$$

$$\alpha \in \{TUV\}^* \quad \beta \in \{TUV\}^*$$

$$\beta \in \{TUV\}^*$$

$$|\alpha| \leq |\beta|$$

° Generates Context sensitive Language.

CFG : (V, T, P_S)

Type-2

$\alpha \rightarrow \beta$

↑ ↑

only one variable (Variable as well as Terminal)

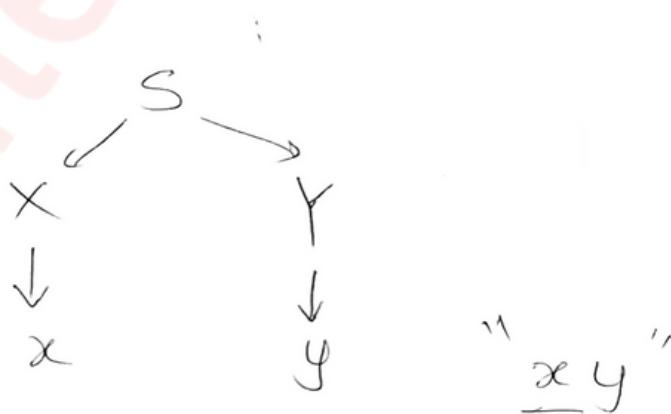
$$\alpha \in V; |\alpha|=1 \quad \beta \in \{T \cup V\}^*$$

eg:- $S \rightarrow xY$
 $X \rightarrow x$
 $Y \rightarrow y$

Start symbol: {s}

Variable : { s, x, y }

Terminal: {x, y}



° Type-3 Grammer (Regular)

→ Most Restriction.

→ Left Regular

$$X \rightarrow x / Y_x$$

$$|X| = |Y| = 1 ; X, Y \in V$$

→ Right Regular

$$X \rightarrow x / x Y$$

$$|X| = |Y| = 1 ; X, Y \in V$$

• Convert CFL \longrightarrow CFG

$$Q : L = a^n b^n, n \geq 0$$

$$S \rightarrow a S b / \epsilon$$

$$\begin{array}{c} S \\ | \\ \epsilon \end{array}$$

$$\begin{array}{c} S \\ | \\ a S \\ | \\ \epsilon \end{array}$$

$$\begin{array}{c} S \\ | \\ a S \\ | \\ a S \\ | \\ \epsilon \end{array}$$

$$\begin{array}{c} S \\ | \\ a S \\ | \\ a S \\ | \\ a S \\ | \\ \epsilon \end{array}$$

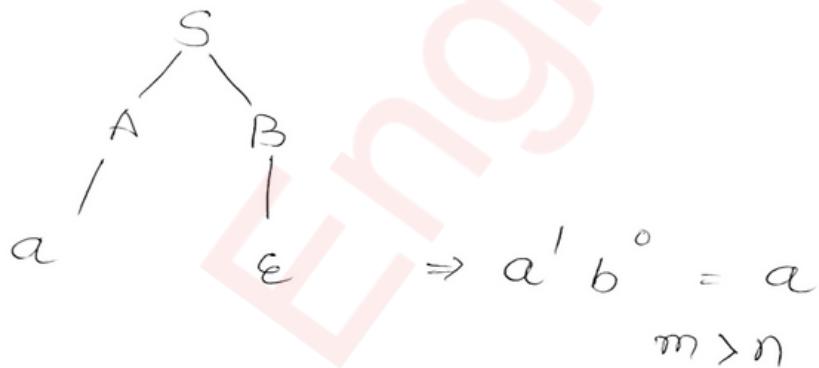
$\mathcal{L} : L = a^m b^n, m > n$

↑
DCFL

$\Rightarrow S \rightarrow A B$

$B \rightarrow a B b / \epsilon$

$A \rightarrow a A / a$



$\mathcal{L} : L = a^m b^n, m \geq n$

↑
DCFL

$\Rightarrow S \rightarrow A B$

$B \rightarrow a B b / \epsilon$

$A \rightarrow a A / \epsilon$

$\mathcal{J}: L = a^m b^n, m \leq n$

$\Rightarrow S \rightarrow A B$

$A \rightarrow a A b / \epsilon$

$B \rightarrow b B / \epsilon$

Q: $a^m b^m c^n$, $m, n \geq 0$

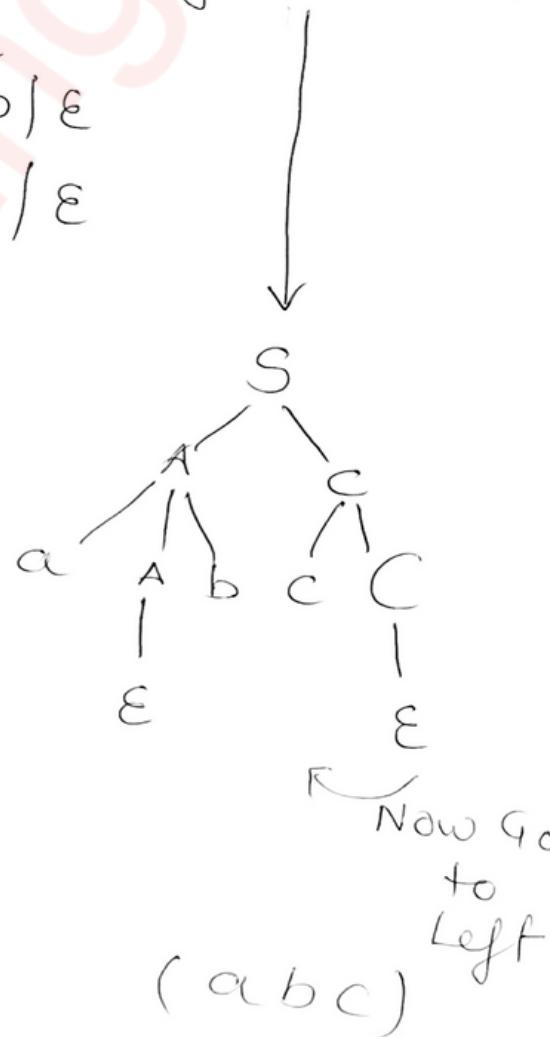
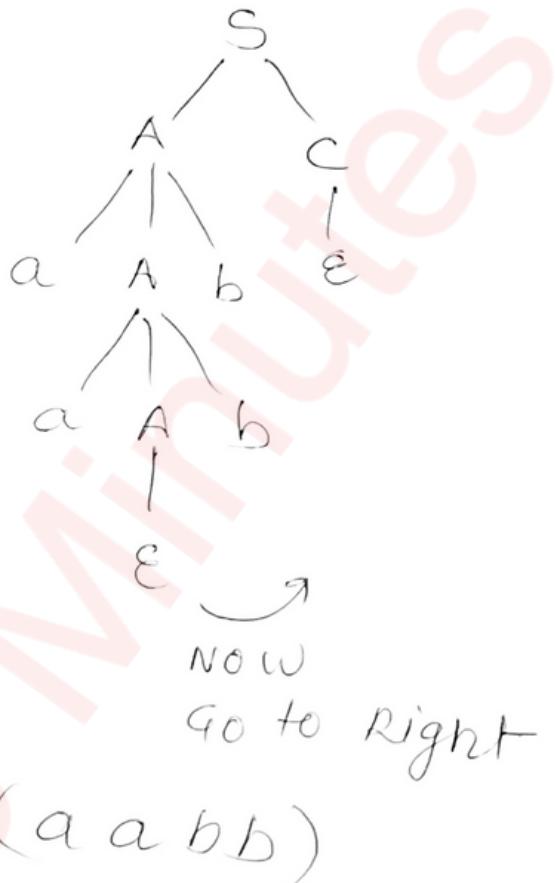
$$S \rightarrow AC$$

$$A \rightarrow aAb / \epsilon$$

$$C \rightarrow CC / \epsilon$$

• Left Most Derivation & Right Most.D

$$\begin{aligned} S &\rightarrow AC \\ A &\rightarrow aAb / \epsilon \\ C &\rightarrow CC / \epsilon \end{aligned}$$



◦ Ambiguous Grammar

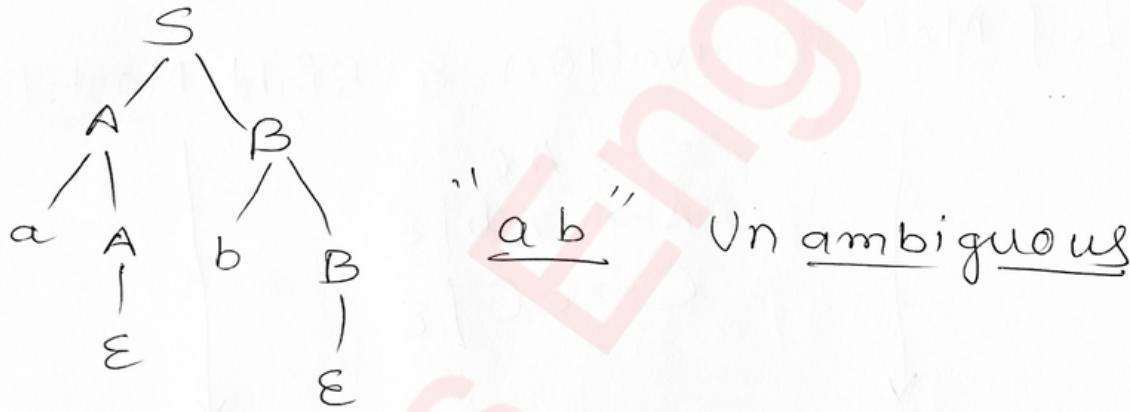
↳ for a string if there are more than one DT

(LMDT or RMDT)

$$\text{eg: } S \rightarrow A B$$

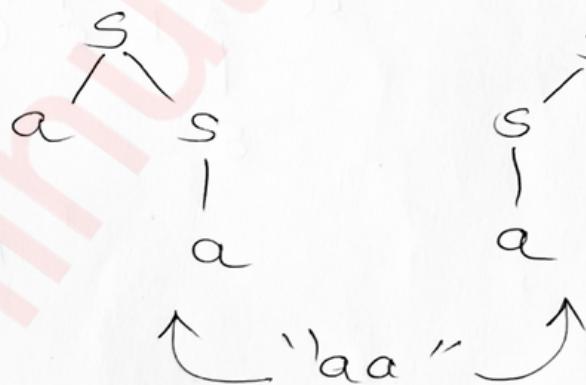
$$A \rightarrow aA / \epsilon$$

$$B \rightarrow bB / \epsilon$$



"ab" Un ambiguous

$$\text{eg: } S \rightarrow aS / Sa / a \Rightarrow a^+$$



But

$$\begin{aligned} &\curvearrowright a \\ S \rightarrow aS/a \end{aligned}$$

Unambiguous

• Minimization of CFG

$\begin{cases} \rightarrow \epsilon/\text{null} \\ \rightarrow \text{unit} \\ \rightarrow \text{useless} \end{cases}$ } productions
 elimination

$$\text{eg: } S \rightarrow XxY$$

$$X \rightarrow x / \epsilon$$

$$Y \rightarrow y / \epsilon \leftarrow Y \rightarrow \epsilon \text{ remove}$$

\Downarrow

$X \rightarrow \epsilon \text{ remove}$

$$S \rightarrow XxY / Xx / xY / x$$

$$X \rightarrow x$$

$$Y \rightarrow y$$

$$\text{eg: } S \rightarrow A$$

$$A \rightarrow B$$

$$B \rightarrow C$$

$$C \rightarrow x$$

Removing unit production

$$\Rightarrow S \rightarrow x$$

eg: $S \rightarrow Aa$

$A \rightarrow a/B$

$B \rightarrow d$

\Downarrow

$S \rightarrow Aa$

$A \rightarrow a/d$

$B \rightarrow d \rightarrow$ now here $B \rightarrow d$

is never used/reach-
able

\Downarrow

Its useless so remove
it.

$S \rightarrow Aa$

$A \rightarrow a/d$

eg: $S \rightarrow xX/yY$

$X \rightarrow y$

So here $S \rightarrow yY$ \rightsquigarrow Its dead
here.

\Downarrow

so remove it

$S \rightarrow xX$

$X \rightarrow y$

- Chomsky Normal form (CNF)

↳ It's a way of rewriting a CFG to simplify parsing.

Production form



$$X \rightarrow YZ | x$$

$\underbrace{}_{V} \quad \underbrace{}_{V} \quad \underbrace{}_{T}$

e.g.: $S \rightarrow xSy | xy$

$$X \rightarrow x \quad Y \rightarrow x$$

so.

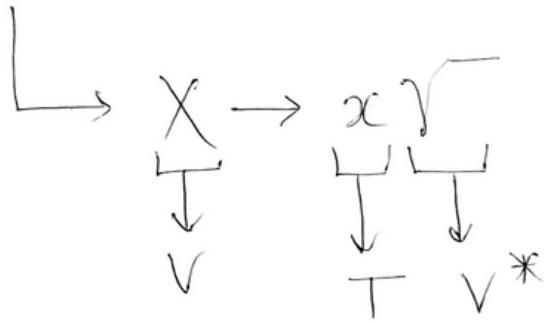
$$S \rightarrow \frac{XS Y}{\downarrow} | XY$$

$$S_1 \rightarrow XS$$

So finally:

$$\begin{aligned} S &\rightarrow S_1 Y | XY \\ S_1 &\rightarrow XS \\ X &\rightarrow x \\ Y &\rightarrow y \end{aligned}$$

Greibach Normal form (GNF)



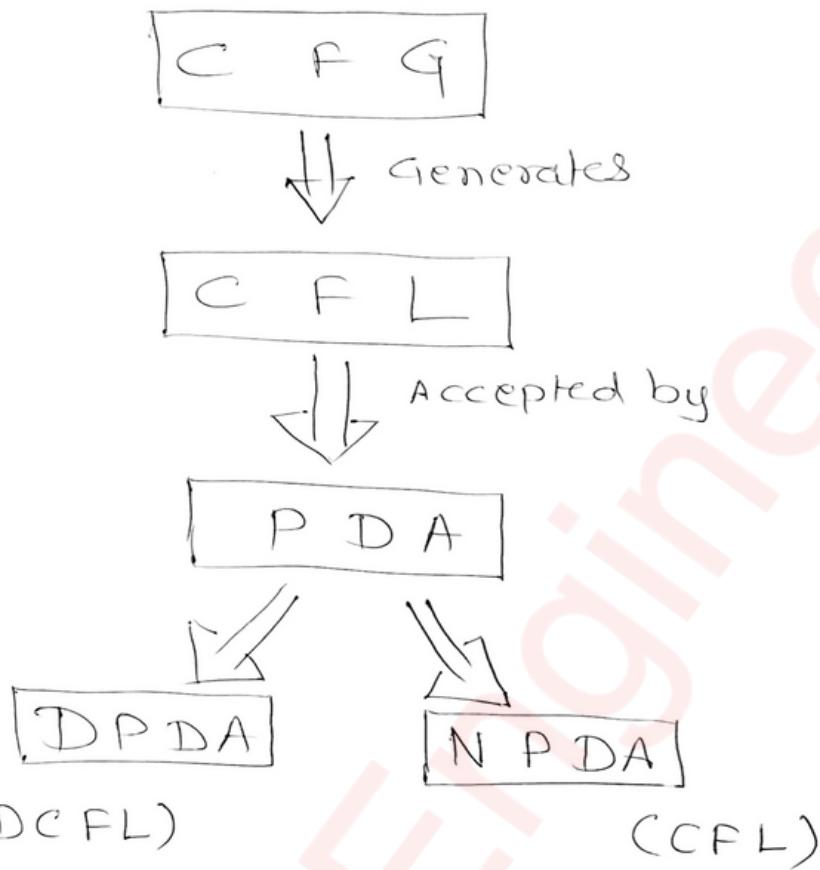
eg: $S \rightarrow xSy | xy$

$Y \rightarrow y$ } just add this PR

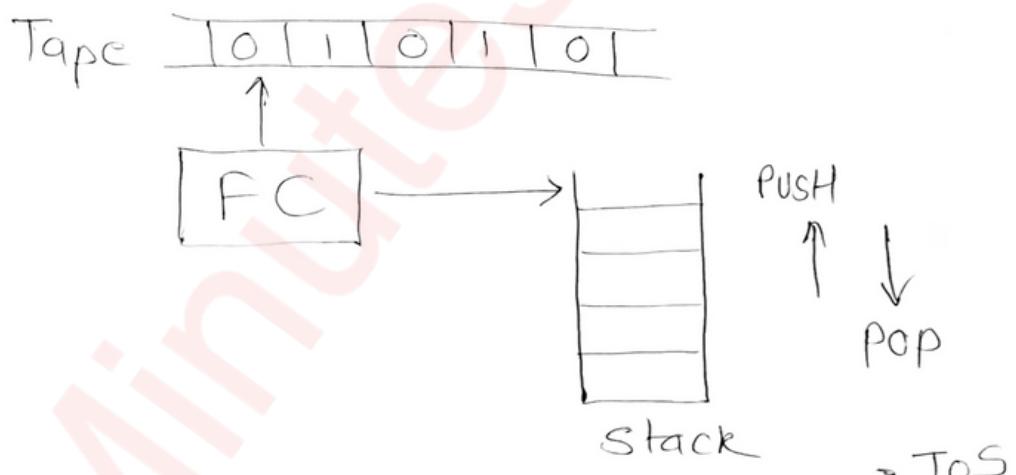
So, $S \rightarrow xSY | xY$ } In GNF
 $Y \rightarrow y$ }

eg:- $S \rightarrow aAB | aB$
 $A \rightarrow aaA | a$
 $B \rightarrow bbB | b$ } Already In
GNF.

◦ Push down Automata



$$P \ D \ A = F \ A + S \ t \ a \ c k$$

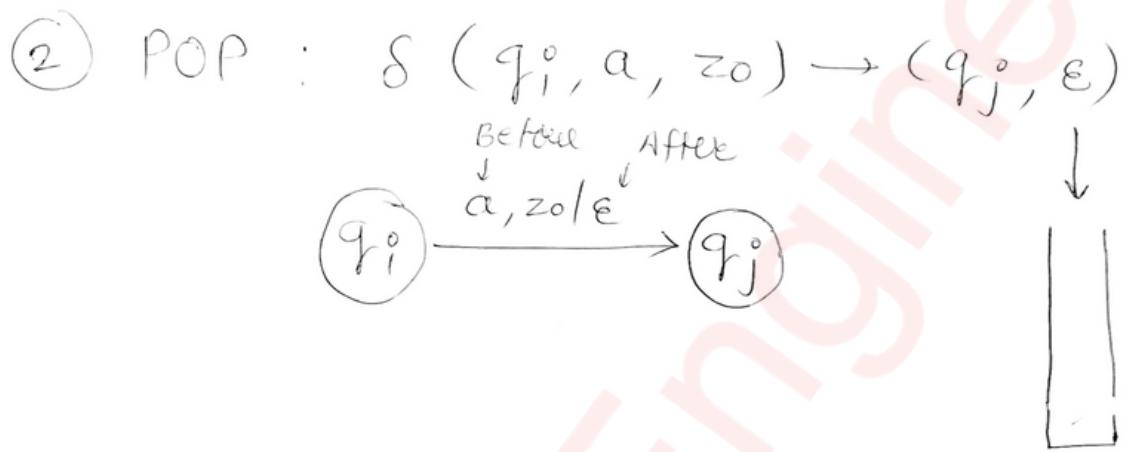
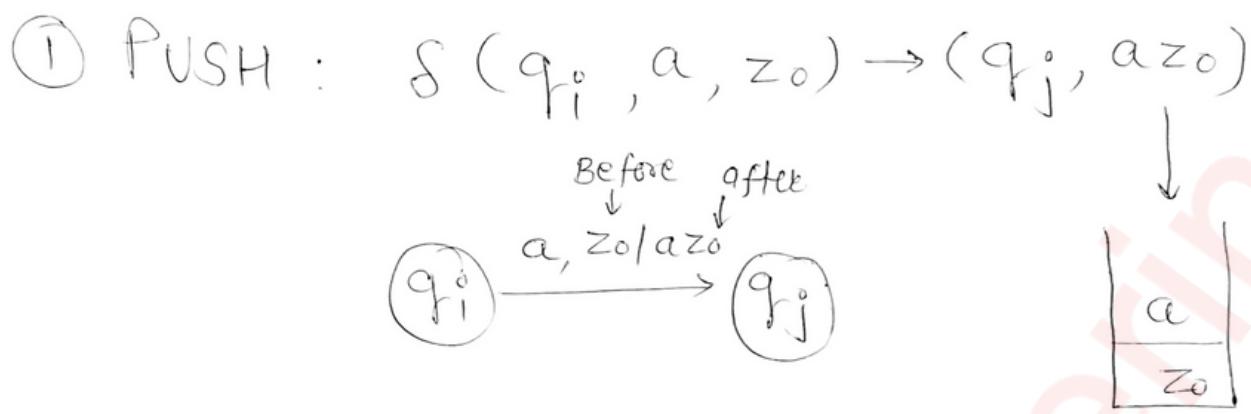


$$P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$$

$$\text{Stack} \quad Q \times (\Sigma \cup \epsilon) \times \Gamma^* \rightarrow Q \times \Gamma^*$$

\downarrow \downarrow

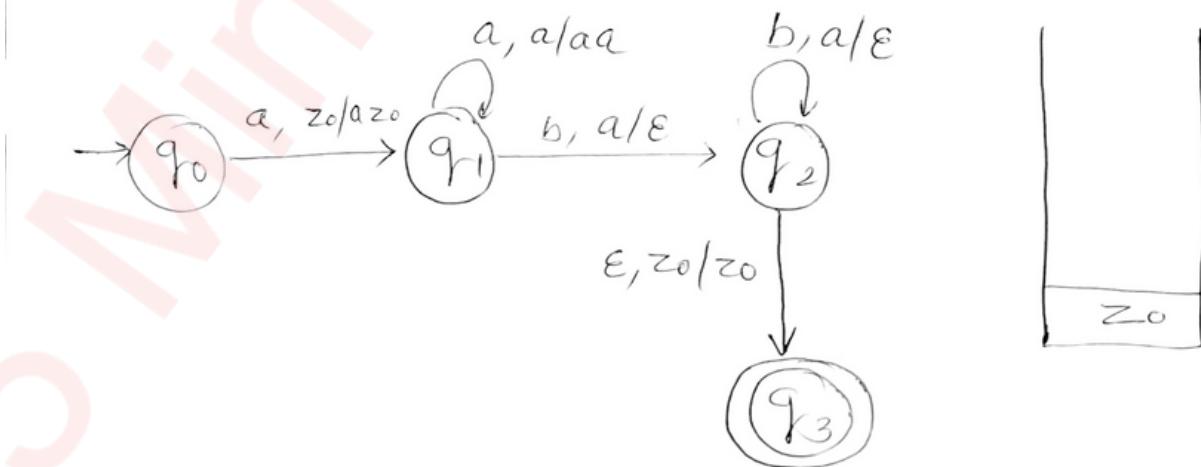
Alphabet



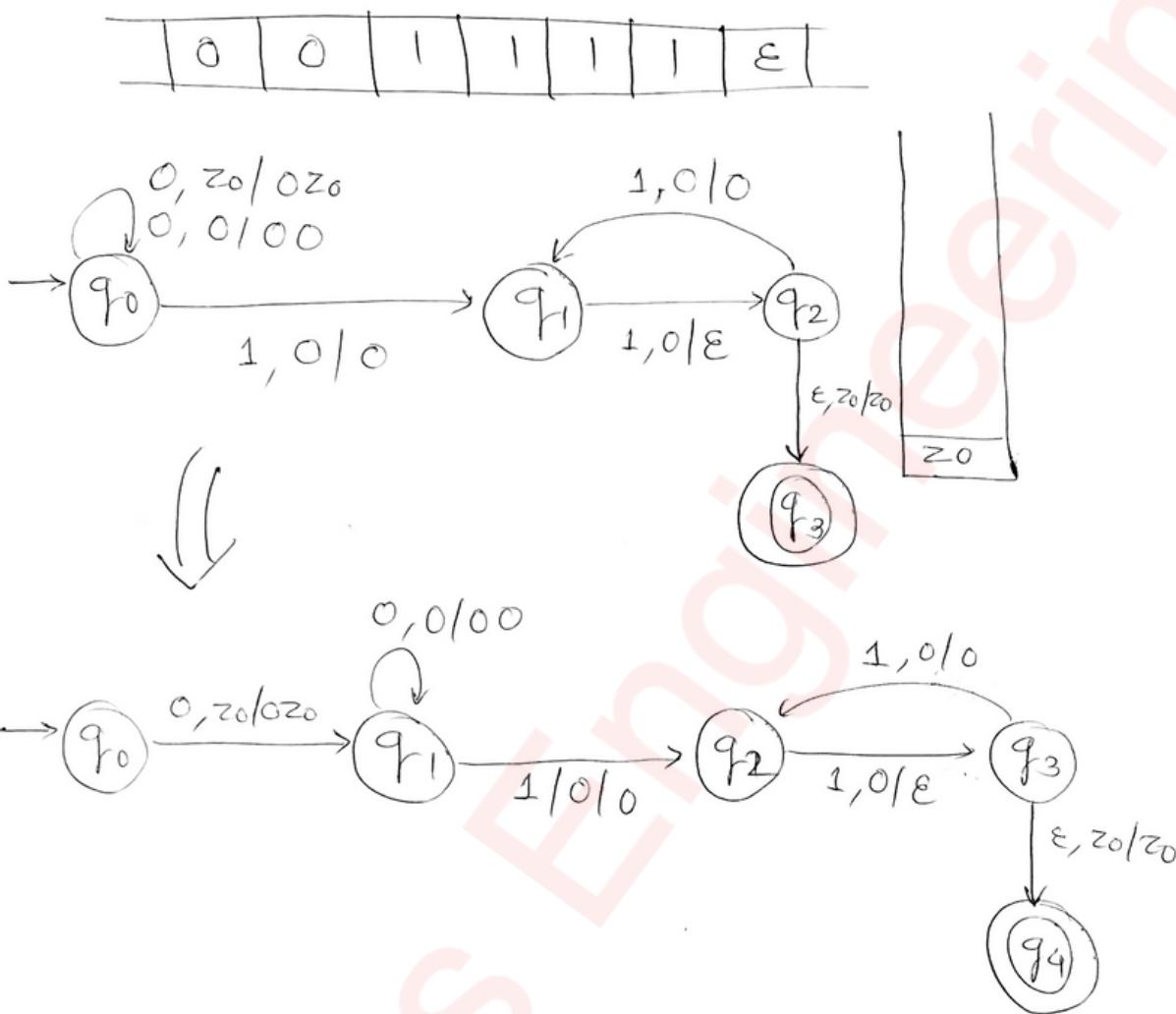
③ SKIP : $\delta(q_i^o, a, z_0) \rightarrow (q_j^o, z_0)$

Eg: $a^n b^n ; n \geq 1$

a | a | a | b | b | b | ε |

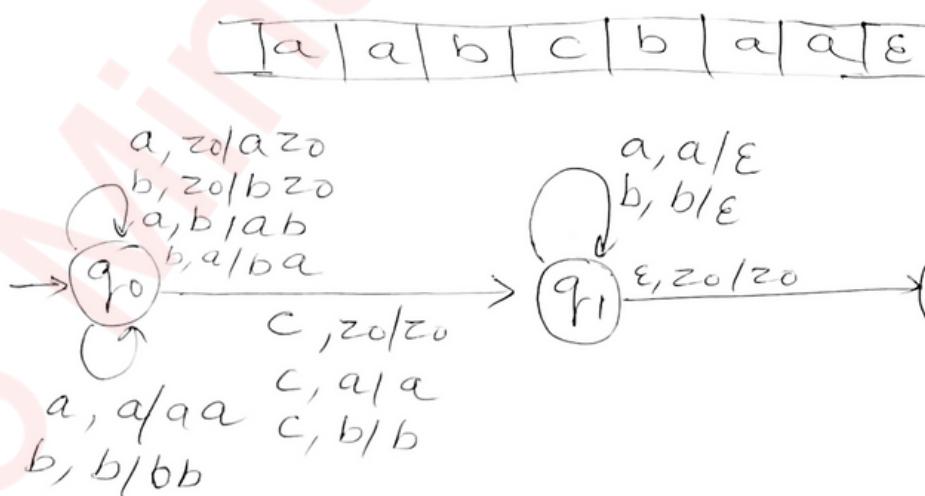


$$Q: L = 0^n 1^{2n} ; n > 0$$



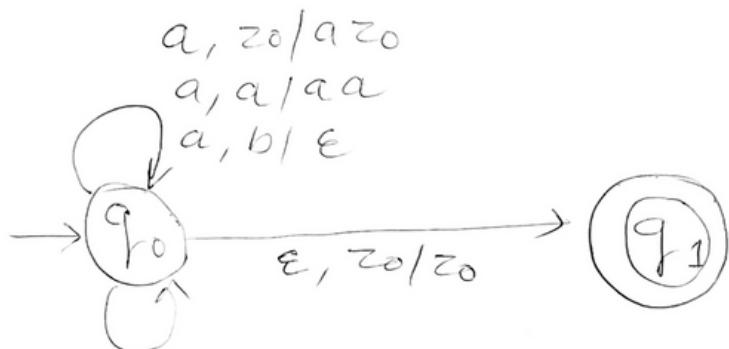
$$Q: L = \{ w \in w^R \mid w \in (a, b)^* \}$$

\downarrow
palindrome: a a b c b a a



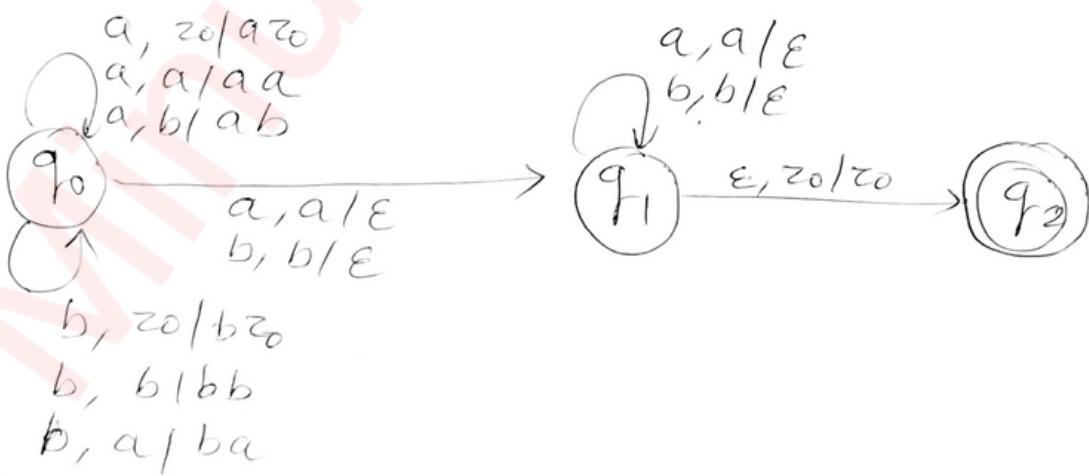
\varnothing : $L = \{ \text{no. of } a = \text{no. of } b \}$
in any order

a | a | b | a | b | b | ε |



\varnothing : $L = \{ w w^R \mid w \in (a, b)^* \}$

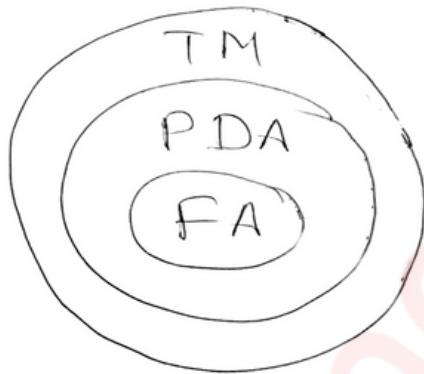
T a | b | a | a | b | a | ε |



• Turing Machine

$\text{FA} \subset \text{PDA} \subset \text{TM}$

$\underline{a^n}$ $\underline{a^n b^n}$ $\underline{a^n b^n c^n}$

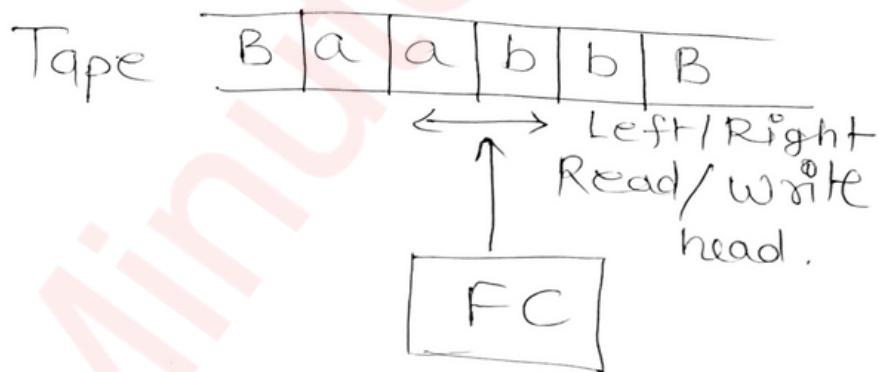


$(Q, \Sigma, \Gamma, \delta, q_0, B, F)$



$Q \times \Gamma \rightarrow Q \times \Gamma \times (L, R)$

\downarrow \downarrow
Left Right



Linear Bounded Automata (LBA)

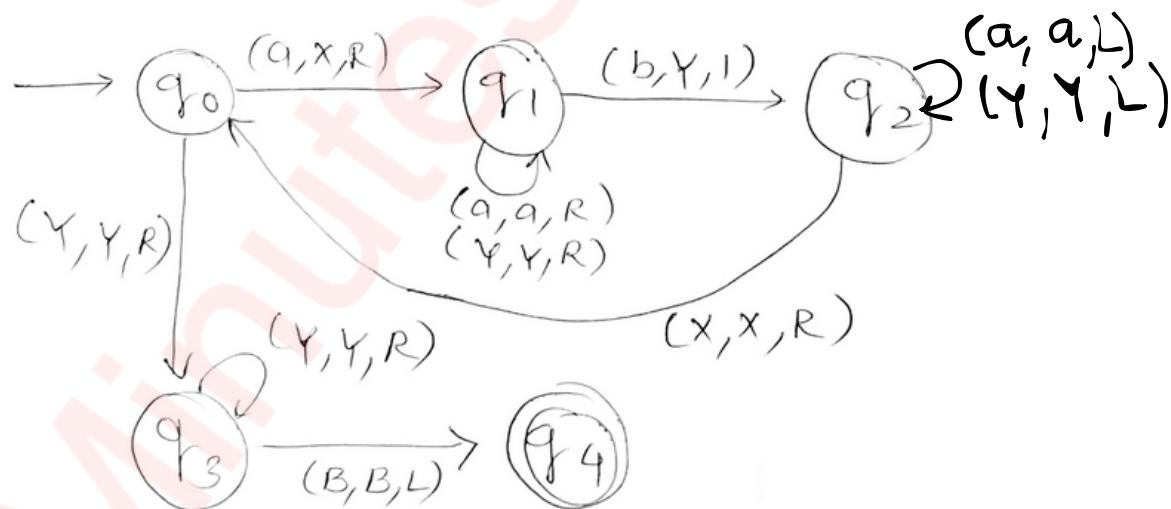
\rightarrow TM + Limit on Tape
 based on I/P size
 \rightarrow Context sensitive Language

FA < PDA < LBA < TM

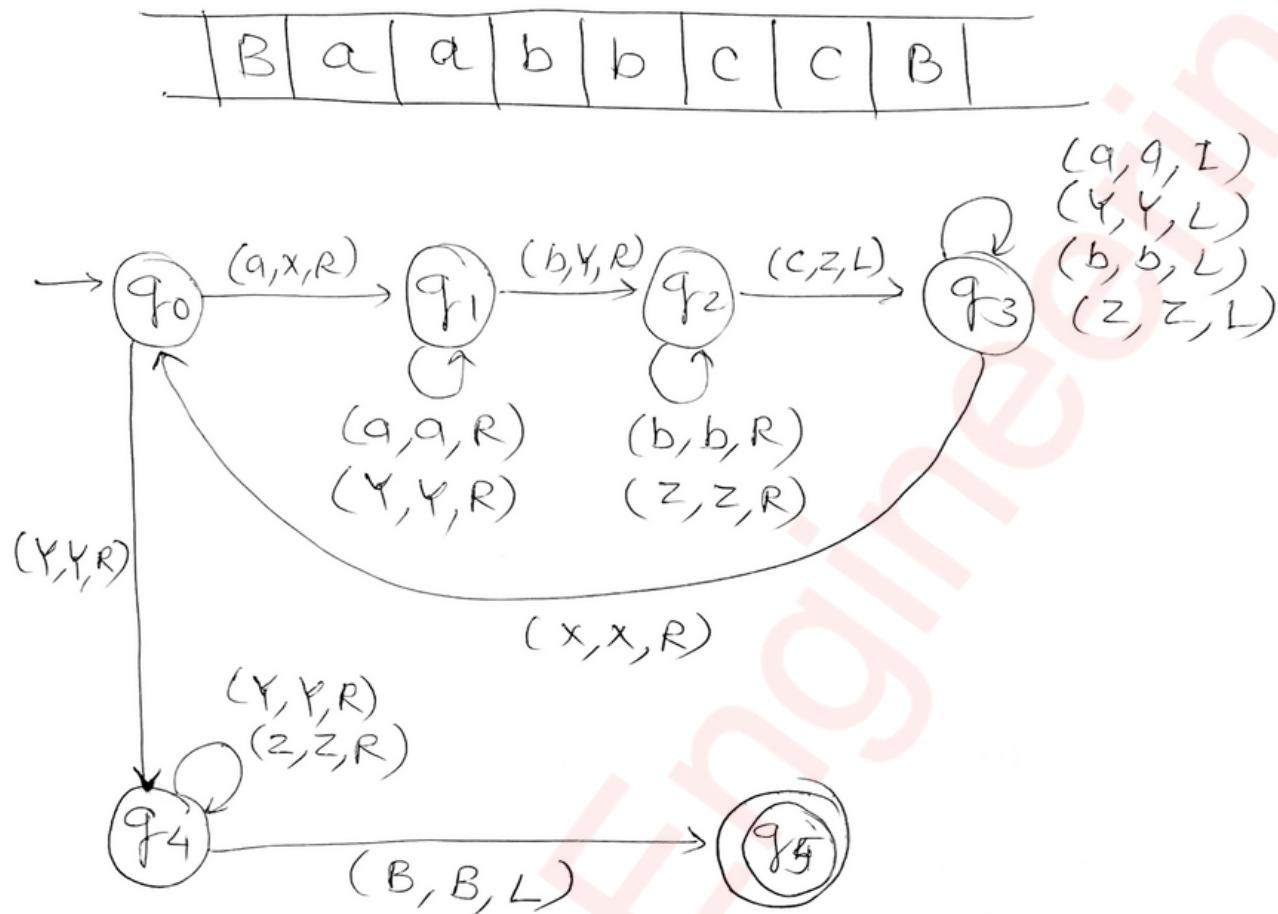
$a^n b^n c^n, n \geq 1$
 a^n for $n \rightarrow$ prime
 $n \rightarrow$ nonprime

Q: $L = a^n b^n | n \geq 1$

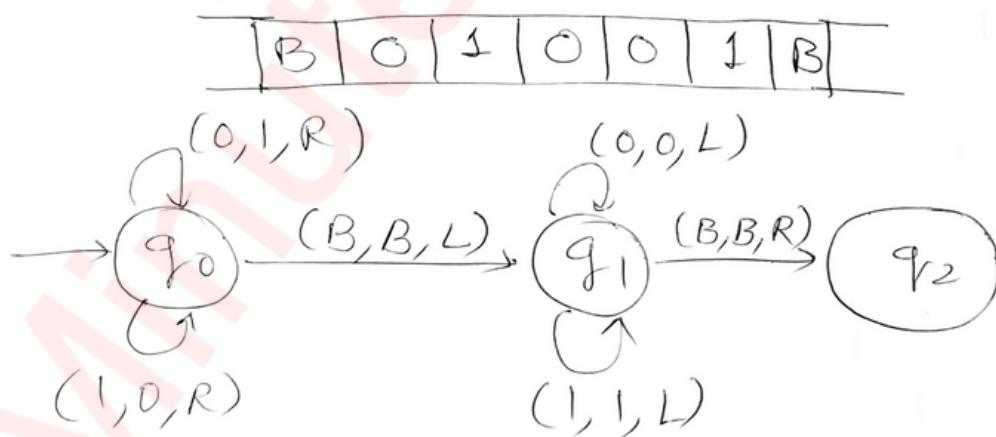
B | B | a | a | b | b | B | B



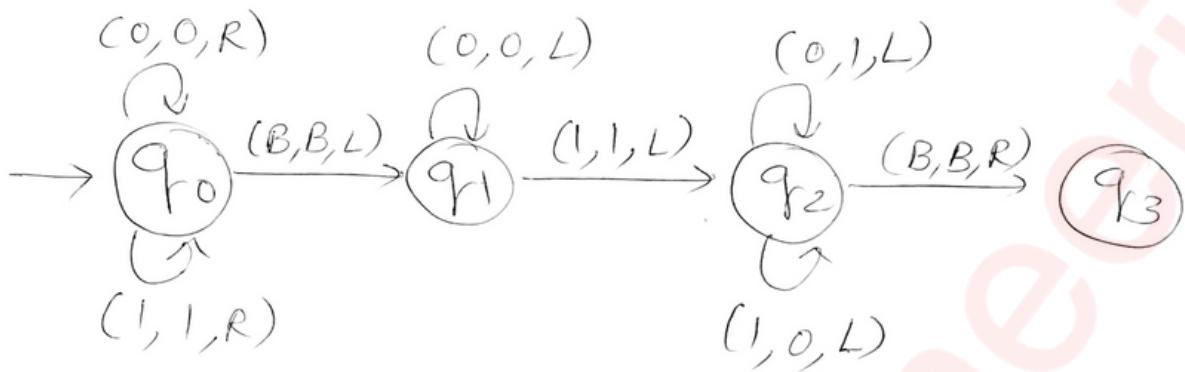
Q: $L = \{a^n b^n c^n \mid n \geq 1\}$



Q: 1's Complement



Q: 2's Complement



B|0|1|0|1|0|0|B|

- Modification / variation / Versions of Turing machine.

Note: The more languages accepted, the more powerful Automata.

→ Same power:

- ① stay option
- ② semi infinite tape
- ③ offline TM
- ④ Multidimensional TM
- ⑤ multtape TM
- ⑥ Multihead TM
- ⑦ NDTM
- ⑧ Universal TM
- ⑨ Jumping TM
- ⑩ Non-Erasing TM
- ⑪ Always write TM

FA < DPDA < NPDA < LBA < TM

• RE Language } REC Language

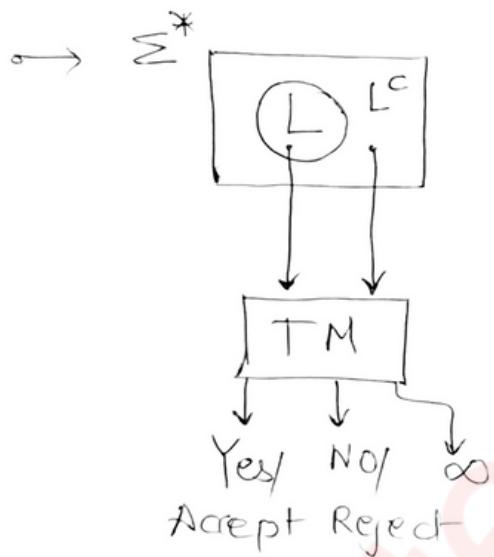
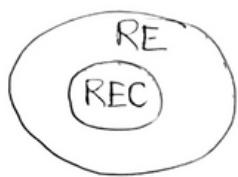
→ TM

→ Can go in ∞ Loop
(Never halt)
state

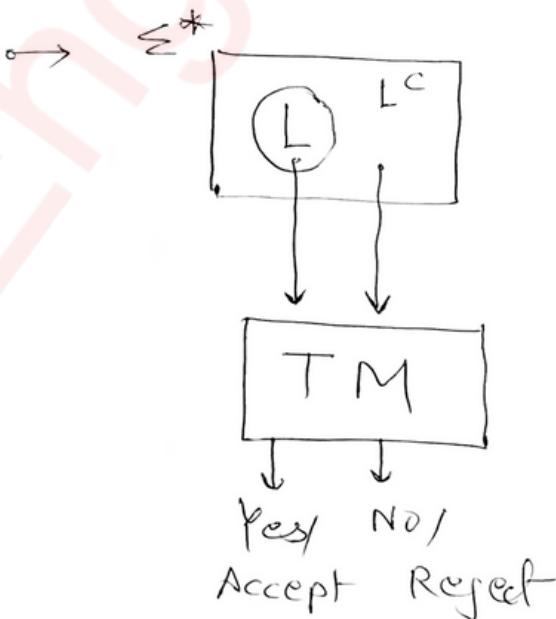
→ Halting TM

→ No such state

→



→ Subset of RE



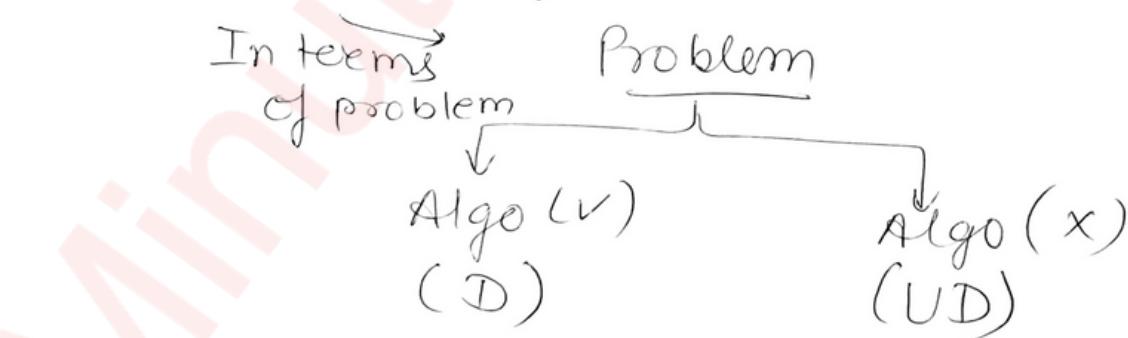
→ $\{ -, L^C \}$ Not closed

→ Homomorphism &
Substitution not
closed.

◦ Turing Thesis

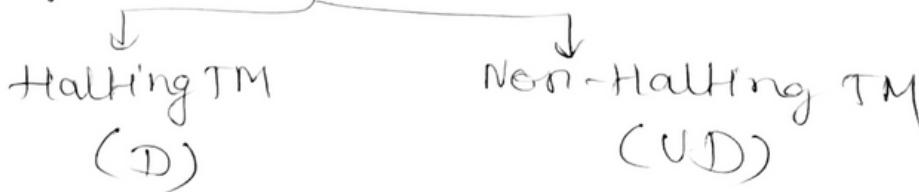
- digital computer → TM (✓)
- No one literally no one has given a problem which is solvable by some algo, for which a TM program can't be written
- TM is most powerful, as none of the Alt. models have power more than TM

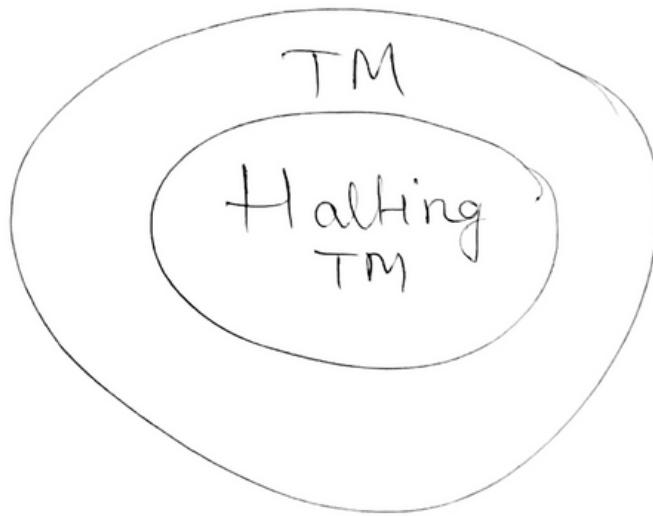
◦ Undecidability



In terms of Lang.

Language





Set of
All Languages

Set of
all TM

Set of
All Halt TM

