Theory of Computation

- What — Problem can be solved
- which — Algorithm
- How — efficiently.

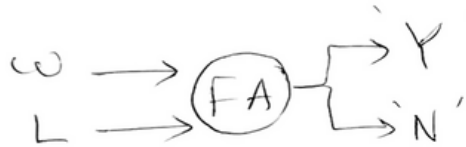
Symbol → a, b, 0, 1 - - -

Alphabet → "Σ" : {a, b}

Steing → a, aa, aba, abb - - -

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

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



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

L = set of all steings of length 2
{aa, ab, ba, bb}

ω_1 = ab ω_2 = bbb

L = set of all steings start with 'a'
{a, aa, ab, aaa, - - - }

ω_1 = abb ω_2 = baa

- Length of string : no of symbol in string "|S|"
  eg: S = abcd , |S| = 4 if |S|=0 its 'ε'
- Kleene star
  ↳ Σ^* (universal set)

  $\Sigma_0 \cup \Sigma_1 \cup \Sigma_2 \cup \Sigma_3 \ldots \Sigma_\infty$ where Σ_k is set of all possibled strings of length k

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

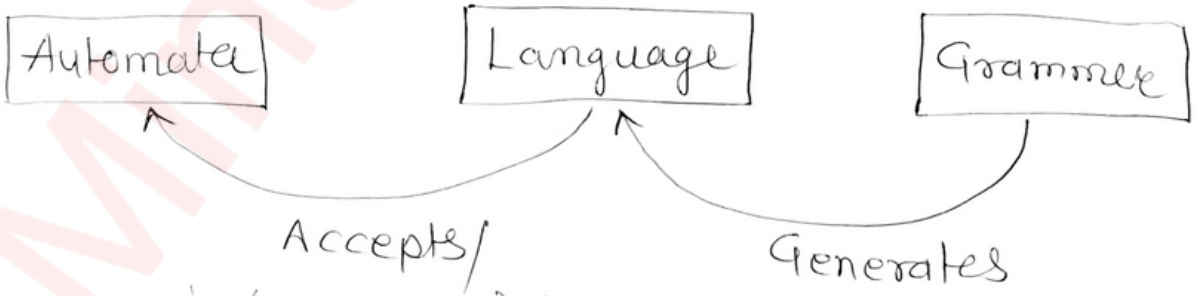
  $\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, \ldots\}$

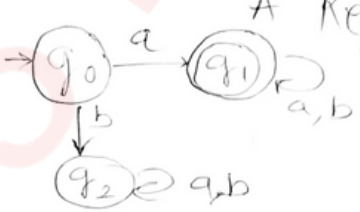
- Kleene closure/plus
  ↳ Σ^+

  " $\Sigma^* - \varepsilon$ "

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

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





$L = \{a, aa, ab \ldots\}$

$\begin{cases} S \to aX \\ X \to aX / bX \end{cases}$

finite Automata:

- Without Output
  - DFA
  - NFA
- With Output
  - Moore m/c
  - Mealy m/c

* Deterministic finite Automata (DFA)

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

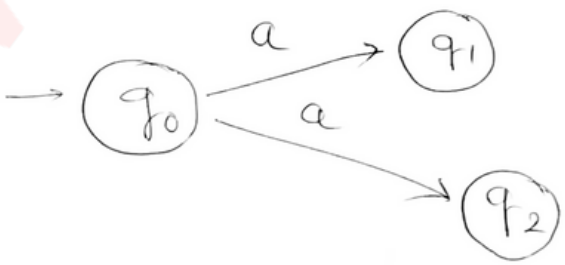
- Q : Set of states
- Σ : I/p
- δ : Transition function
- q_0 : Start/Initial State
- F : final State

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

$Q = \{q_0, q_1\}$

δ	Σ
Q	Transition Table

What is "Deterministic" ?



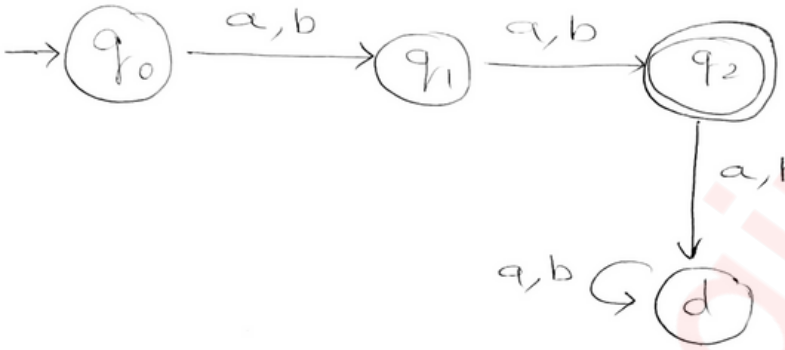
(x) Not allowed

Eg: Length based:

$|w| = 2$ or $|s| = 2$

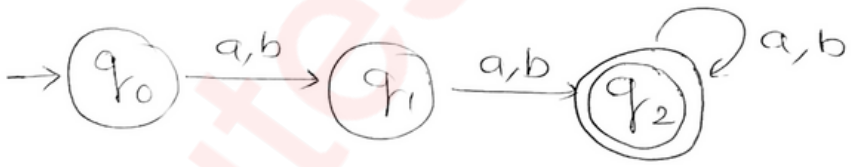
$\Sigma = \{a, b\}$

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



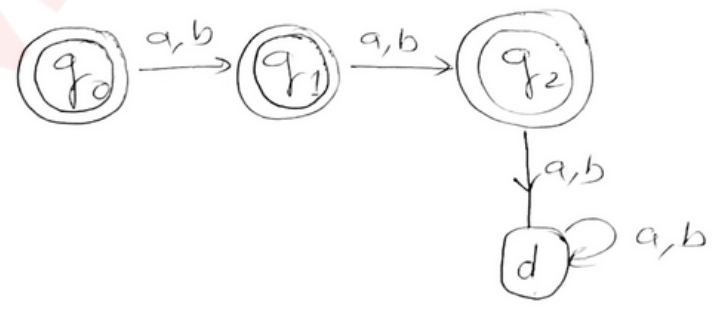
- $|s| \geq 2$

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



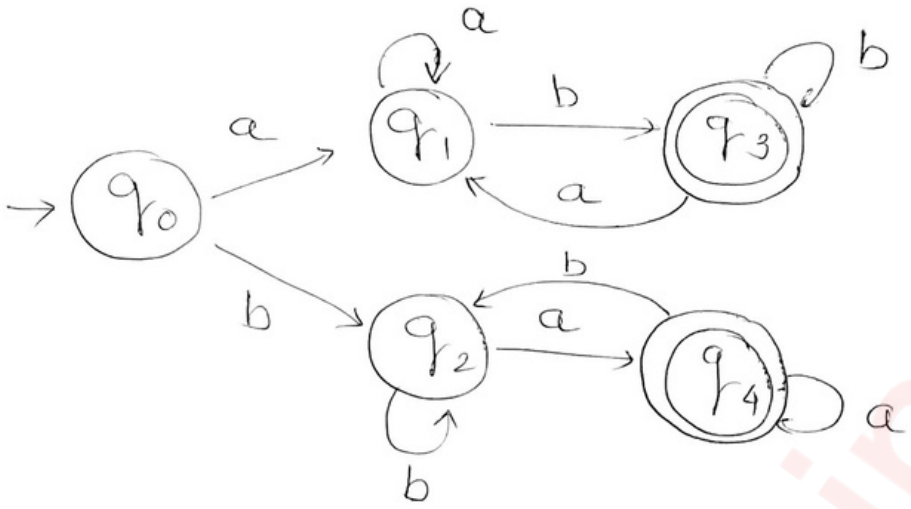
- $|s| \leq 2$

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



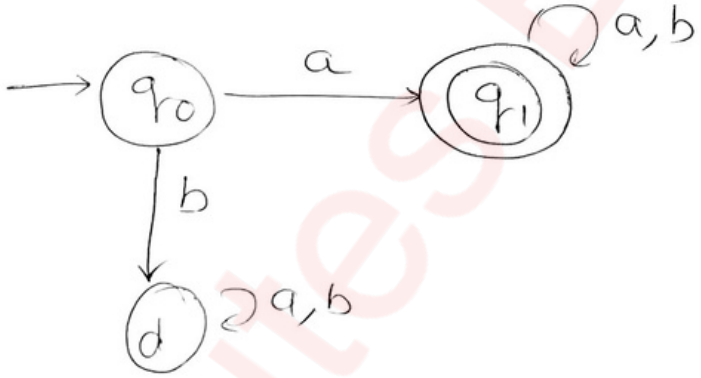
Q: Starts & Ends with different symbol.

$L = \{ab, aaab, abb, baa, bba \ldots\}$



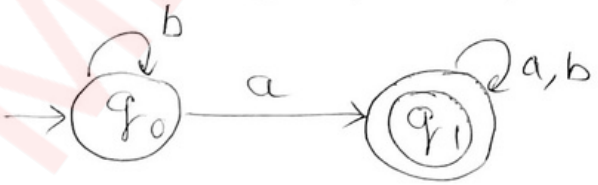
Q: Starts with 'a'

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



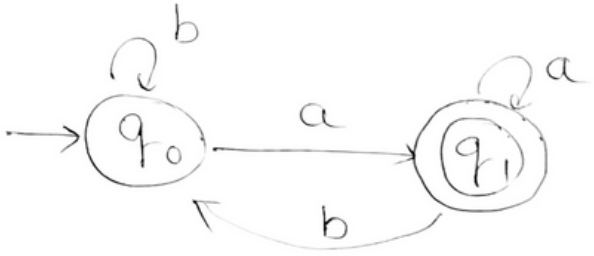
Q: Contain 'a'

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



Q: Ends with 'a'

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

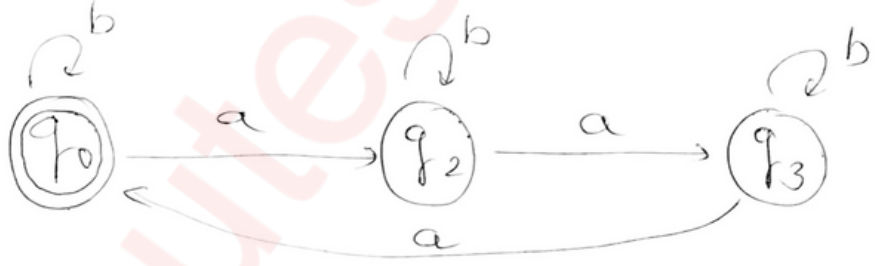


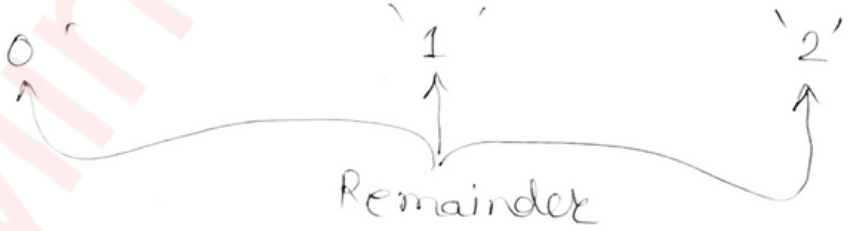
Q: Divisible (mod) problems.

$n(a) \bmod 3 = 0$

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

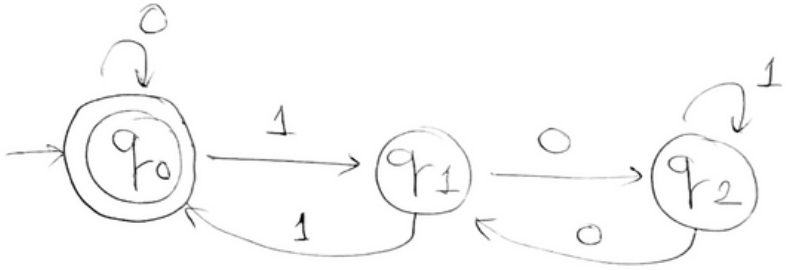
$L = \{ \varepsilon, aaa, aaaaaa, bbbaaa \cdots \}$





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

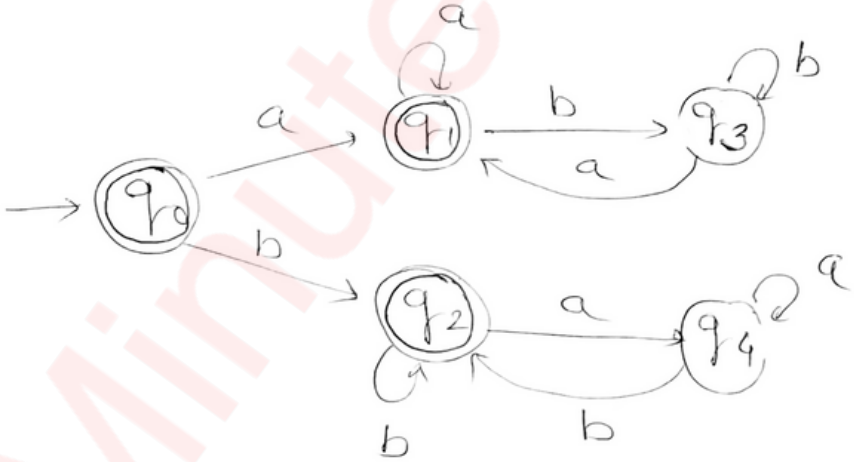
Binary no. divisible by 3.



	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2

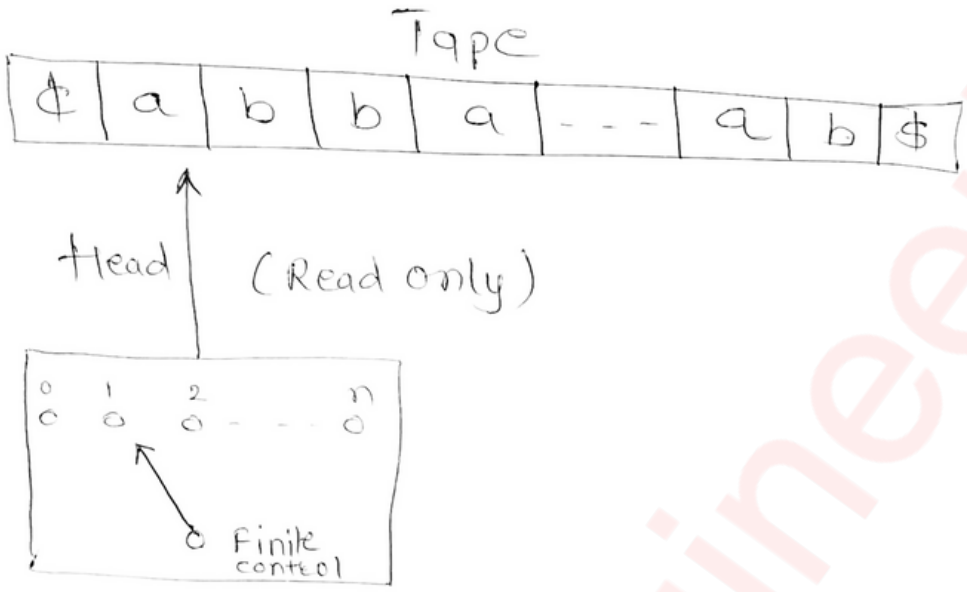
Q: starts & ends with same symbol

$L = \{aa, aaa, aba, bab \ldots\}$



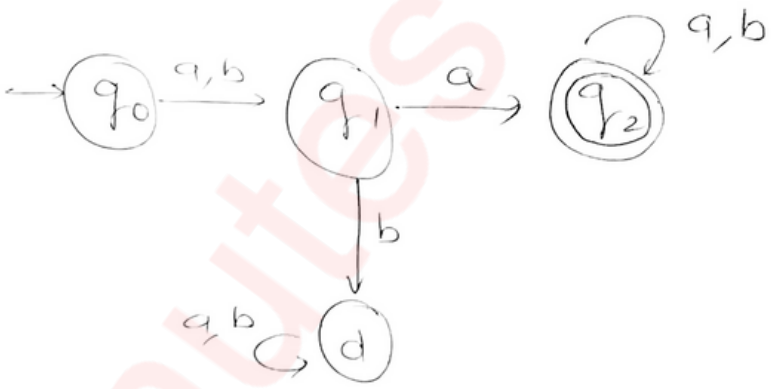
Complementing final $\rightleftharpoons$ Non final.

FA Block Diagram

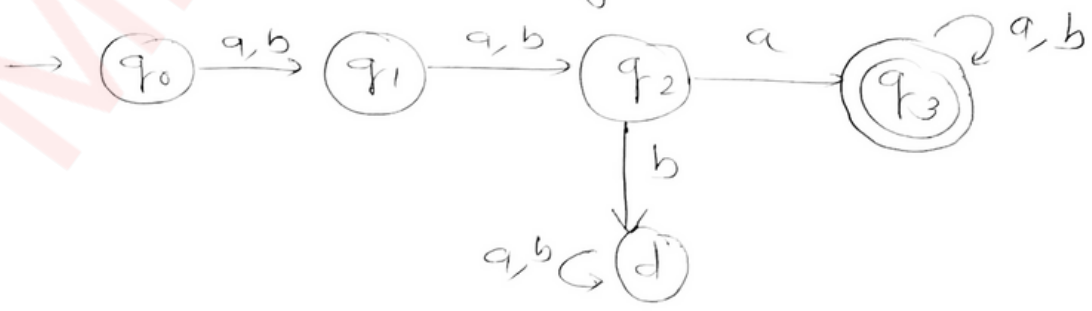


Q: $\Sigma = \{a, b\}$

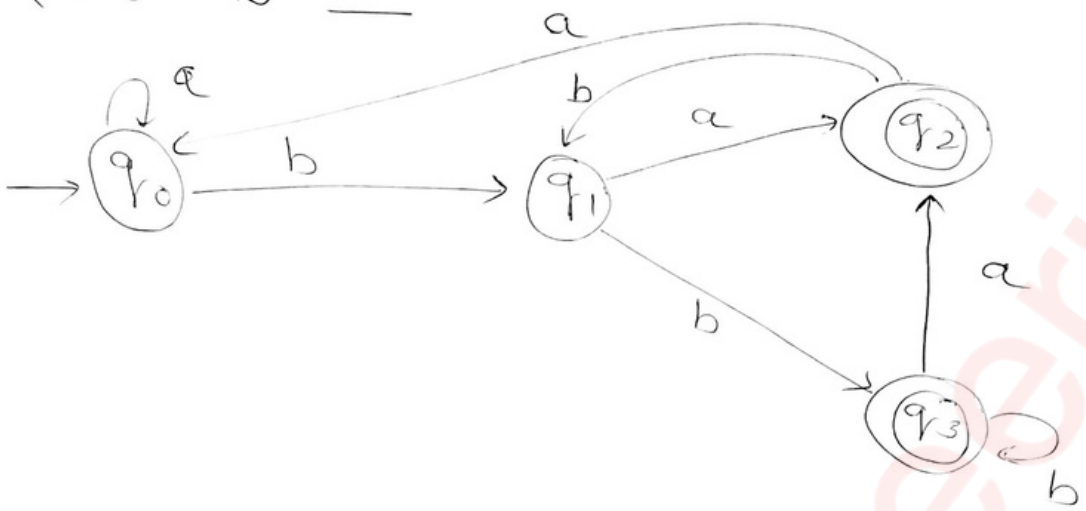
2nd is 'a' (from Left)



3rd is 'a' (from Left)



Q: 2nd last is 'b'

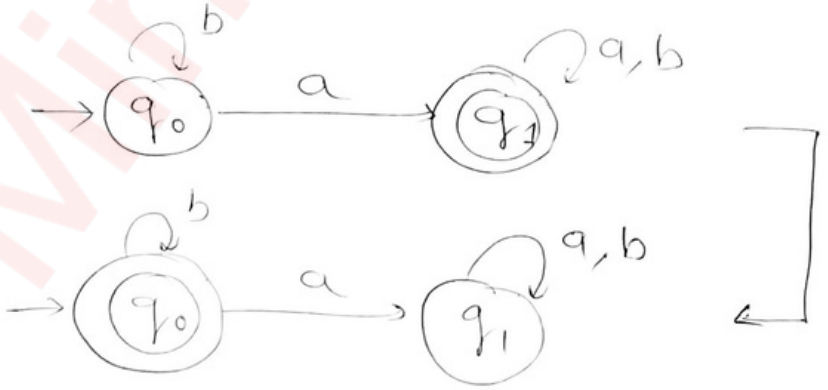


$2^2 = 4$ states

δ	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

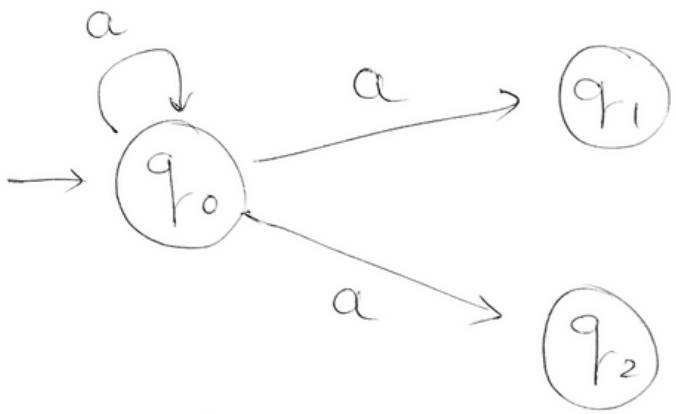
• Compliment of DFA.

→ does not contain 'a'



• NFA : Non Deterministic FA

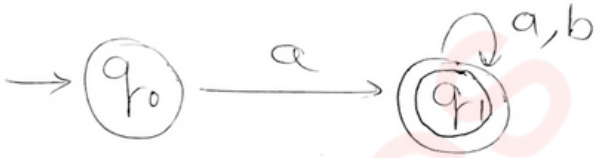
$\{Q, \Sigma, \delta, q_0, F\}$



} allowed

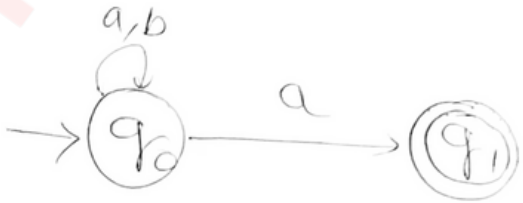
$\delta : Q \times \Sigma \rightarrow 2^Q$

Q: Starts with 'a'

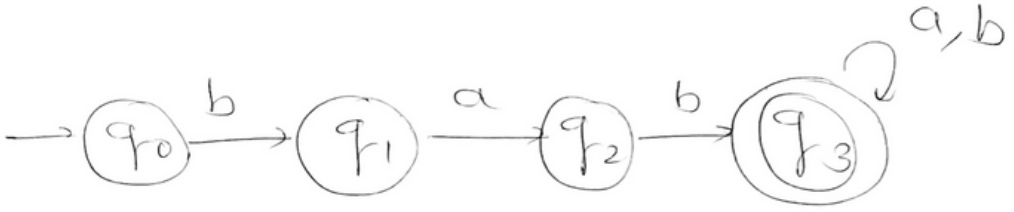


Q: Contains 'a'

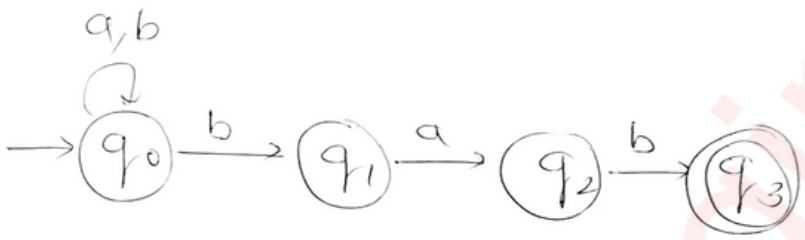
Q: Ends with 'a'



Q: Contains 'bab'



Q: Ends with 'bab'

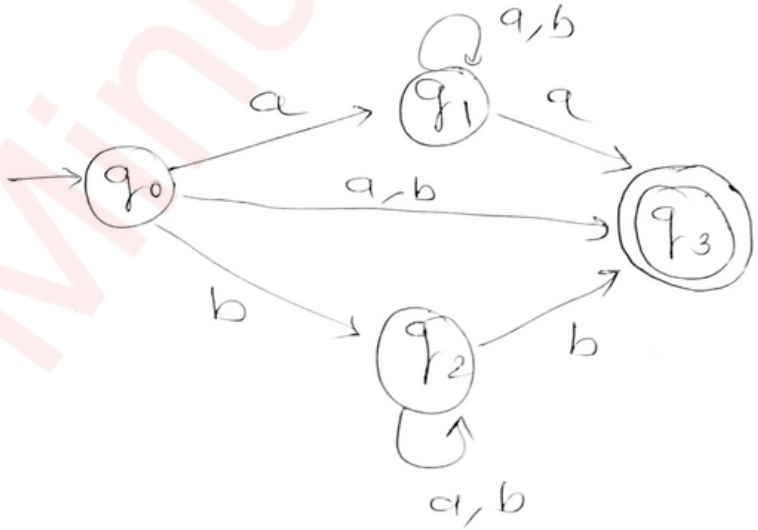


Invalid steing cannot be accepted.

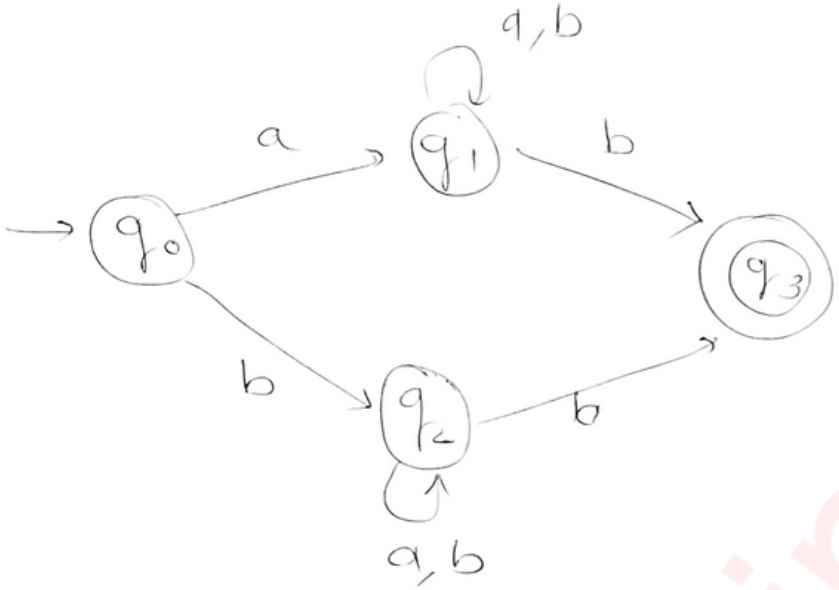
Valid steing → Can be Accepted
Not be Accepted.

eg: aabab it can go to q_3 or q_0

Q: Start end with same symbol.

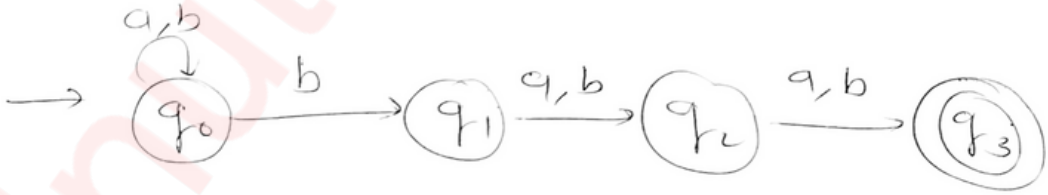


Q: Start & End with different symbol.

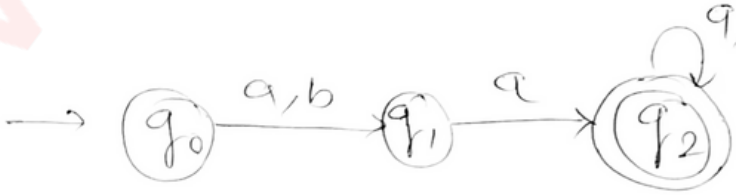


Q: 2nd Last is 'a'

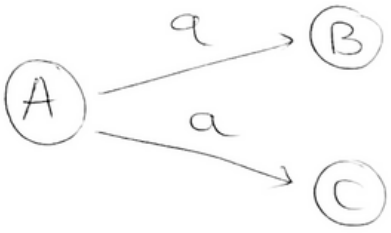
Q: 3rd Last is 'b'



Q: 2nd from Left 'a'



DFA

① 

(x)

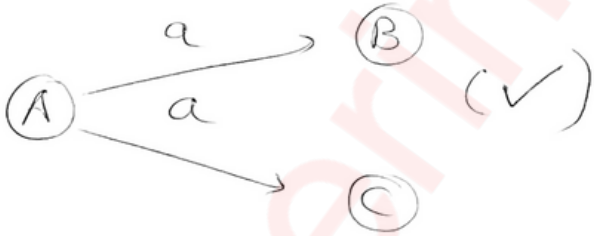
② Difficult to consteuct

③ All DFA are NFA

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

⑤ Need more space

NFA

① 

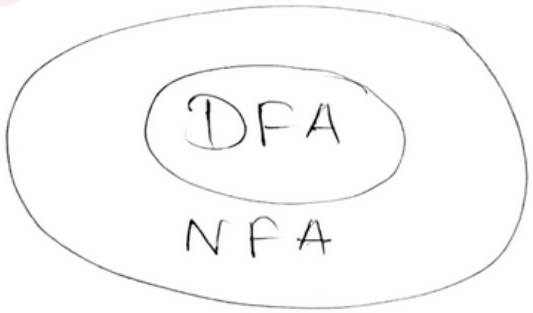
(✓)

② Easy to consteuct

③ Not All NFA are DFA

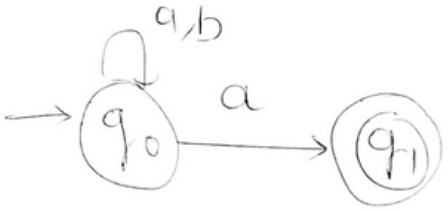
④ $Q \times \Sigma \rightarrow 2^Q$

⑤ Need Less spad



NFA to DFA Conversion

ends with 'a'

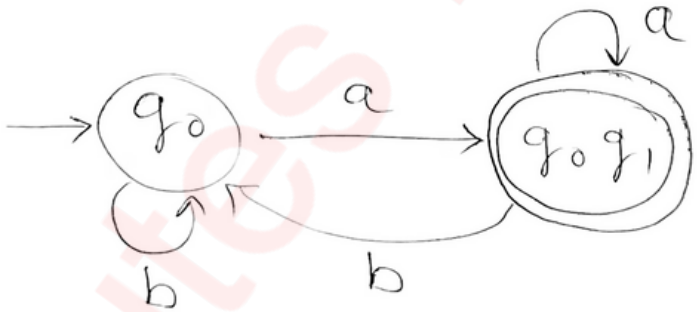


	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
* q_1	$\{\phi\}$	$\{\phi\}$

→

	a	b
q_0	$[q_0 q_1]$	$[q_0]$
* $q_0 q_1$	$[q_0 q_1]$	$[q_0]$

$\times [q_1]$
↳ not reachable.



NFA has 'n' states then.

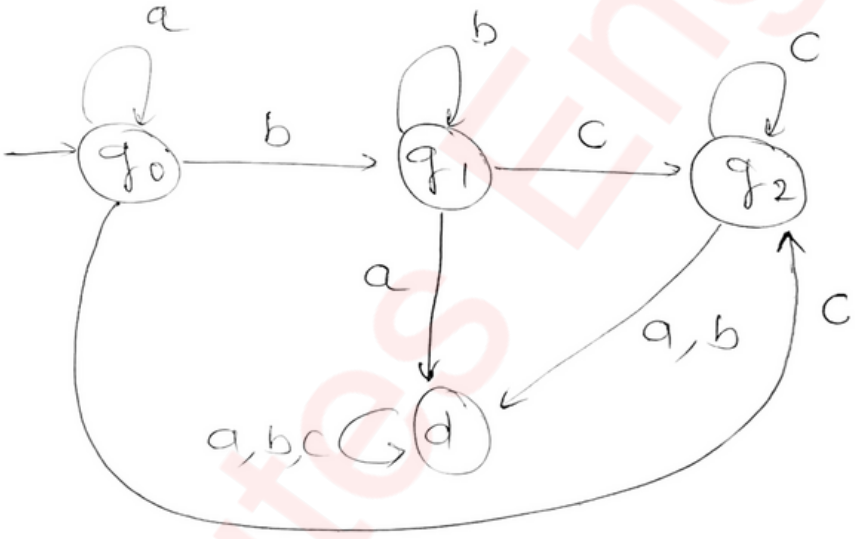
DFA has 'm' states $\quad 1 \leq m \leq 2^n$

ε-NFA (NFA with ε move)

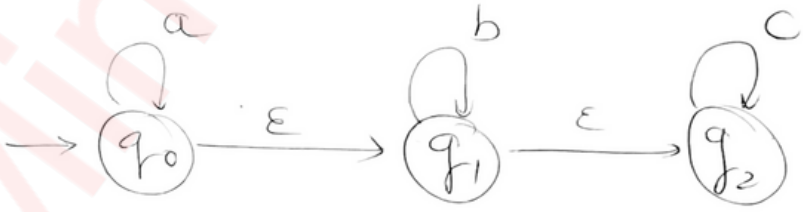
$$(Q, \Sigma, \delta, q_0, F)$$

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

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



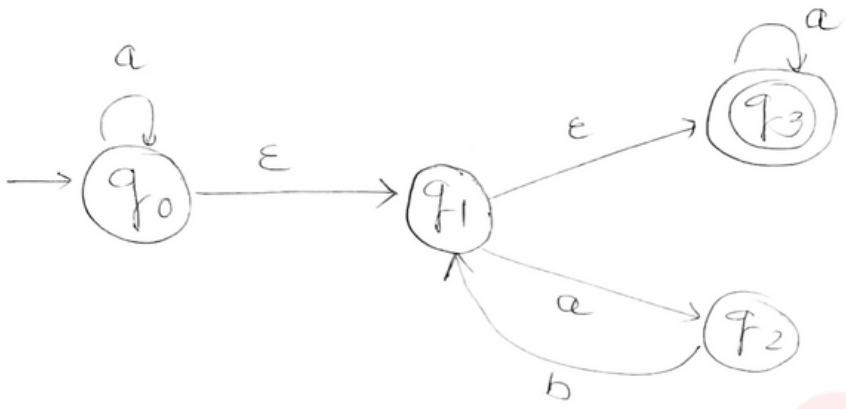
But ε-NFA



εclosure(q_i) → $q_0 = q_0, q_1, q_2$
→ $q_1 = q_1, q_2$
→ $q_2 = q_2$

ε-NFA to NFA conversion

Eg:-



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

$F = \{q_3\}$

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 $\delta[\varepsilon\text{-closure}(q_i), \alpha]$ & take again a ε-closure of that.

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

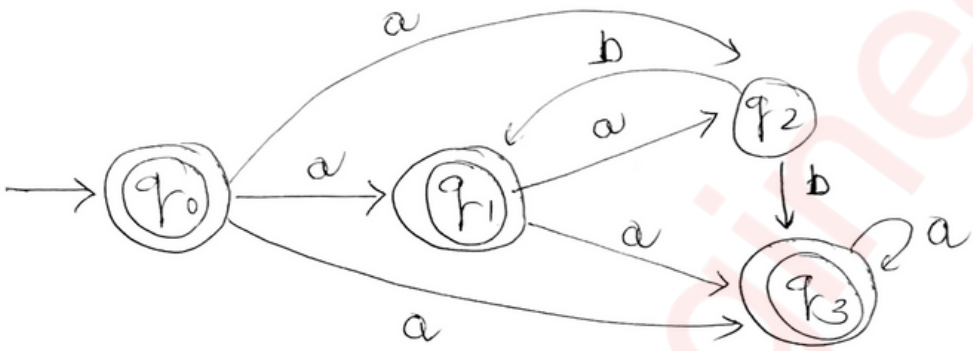
$\downarrow$

$\varepsilon\text{-closure}[q_0, q_2, q_3]$

$\downarrow$

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

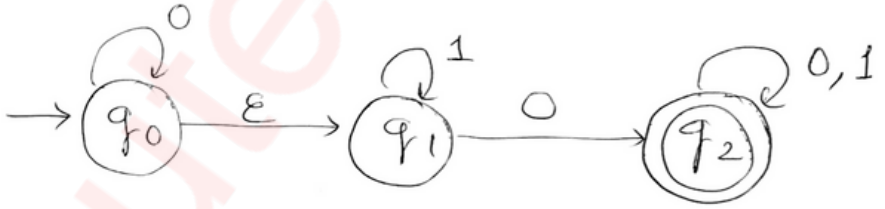
	a	b
q_0	$\{q_0, q_1, q_2, q_3\}$	
q_1	$\{q_2, q_3\}$	
q_2		$\{q_1, q_3\}$
q_3	$\{q_3\}$	



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

- Conversion of ε-NFA to DFA

Eg:



→ ε-closure of q_0 = $q_0 q_1$

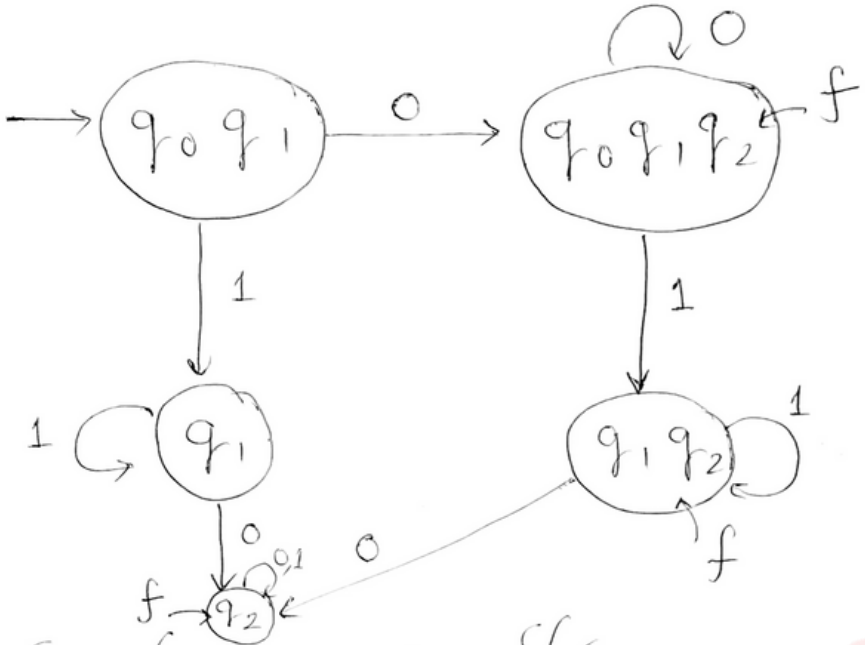
$\delta((q_0, q_1), 0)$

↓

ε-c ($q_0 q_2$) ⇒ q_0, q_1, q_2

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

↳ ε-c(q_1) → q_1



$\varepsilon\text{-}c(q_0, q_1 q) = \delta((q_0 q_1, q_2), 1))$

$\downarrow$

(q_1, q_2)

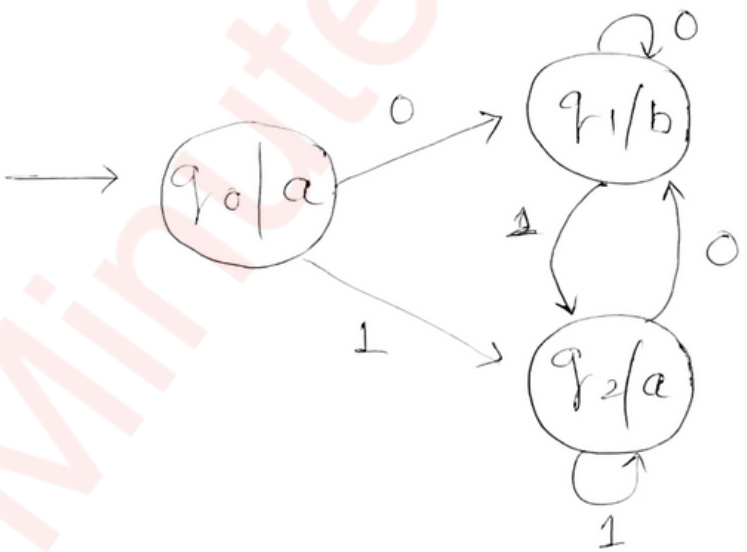
- Moore m/c

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

$\lambda : Q \times \Sigma \to Q$

Δ → o/p symbol

λ → o/p f^n → $Q \to \Delta$



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

passed string

$\downarrow$

1 1 0 1 0 1

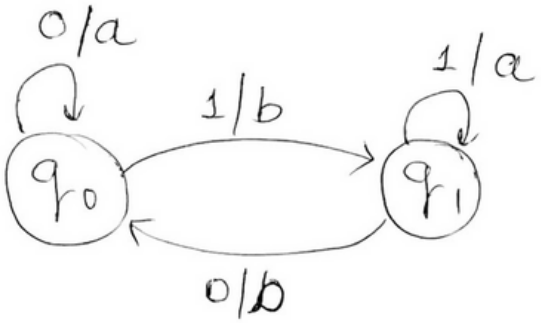
$\Rightarrow q_0 \xrightarrow{} q_2 \xrightarrow{} q_2 \xrightarrow{} q_1 \xrightarrow{} q_2 \xrightarrow{} q_1 \xrightarrow{} q_2$

a a a b a b a

$n+1$

- Mealy m/c
  - $\{Q, \Sigma, \Delta, \delta, \lambda, q_0\}$

Eg:



passing string: 00110 1

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

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

① O/P → PS

② I/P string length = n
so O/P = n+1

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

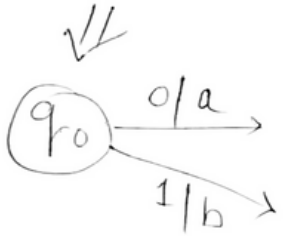
④ O/P is placed on State

⑤ I/P changes, O/P doesn't change

⑥ Easy to design

Mealy m/c

① O/P → PS & PI/P



② I/P string length = n
so O/P = n

③ $\lambda : Q \rightarrow \Delta$

④ O/P is placed on transition

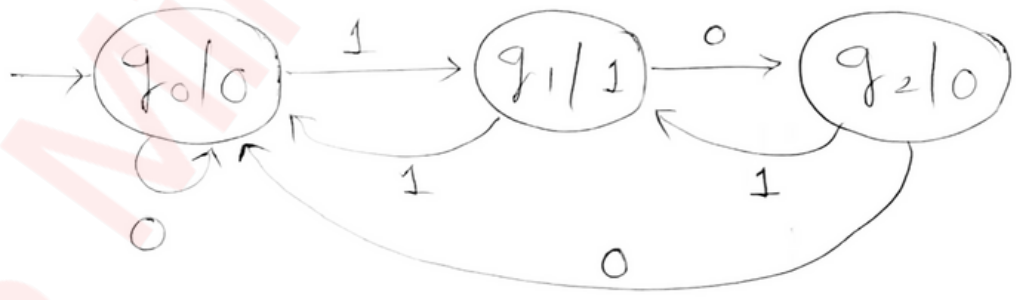
⑤ I/P change, O/P also change.

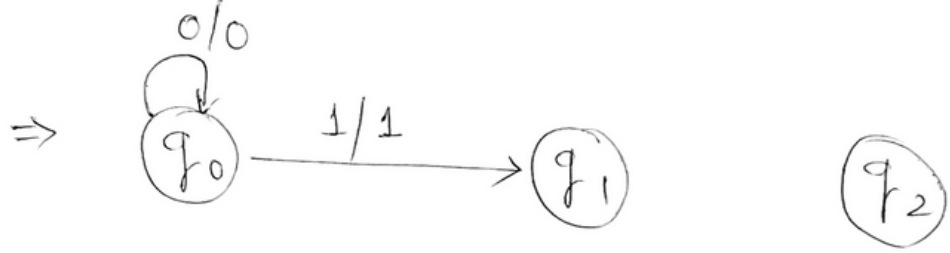
⑥ Difficult to design

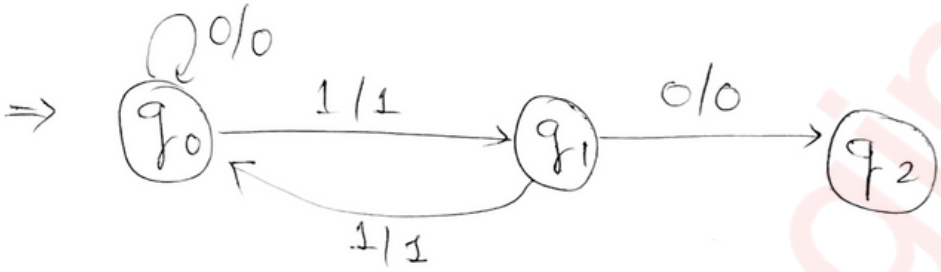
Moore to mealy conversion

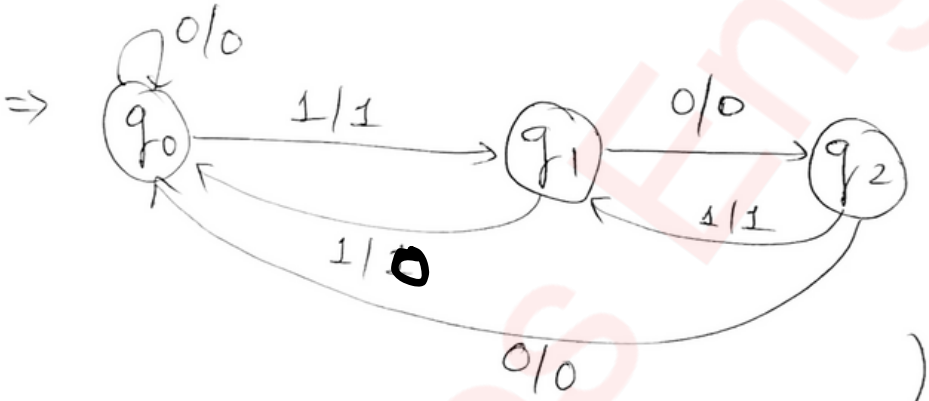
↓

Given





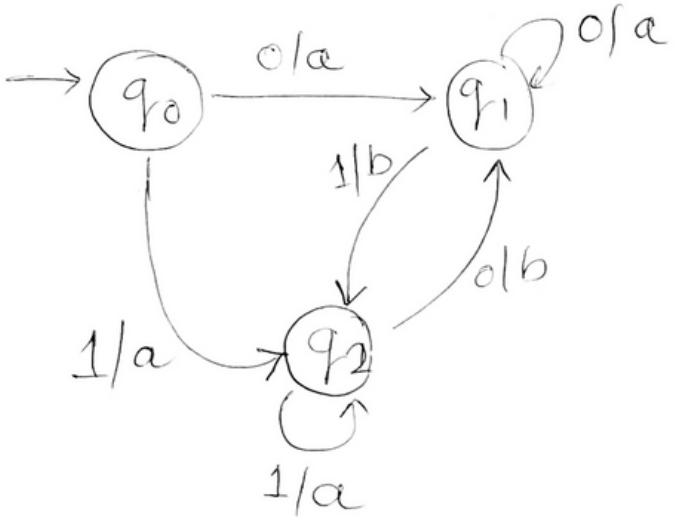


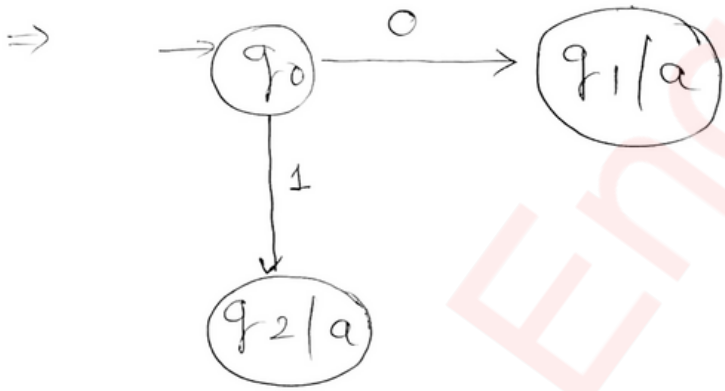


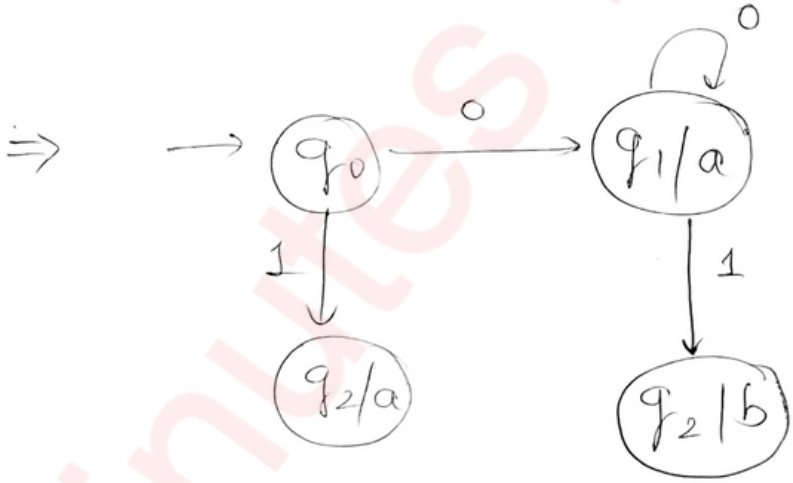
CS	NS		O/P
	0	1	
q_0	q_0	q_1	0
q_1	q_2	q_0	1
q_2	q_0	q_1	0

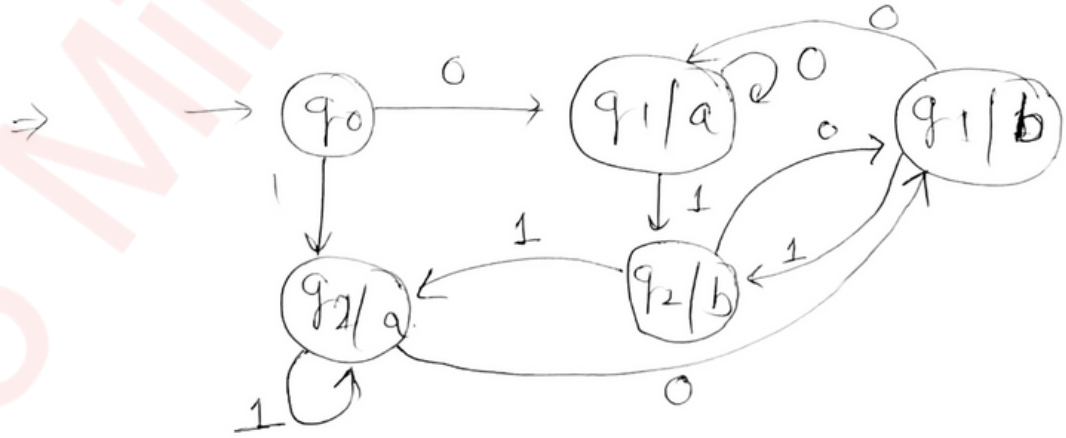
CS	NS			
	I/p = 0		I/p = 1	
q_0	q_0	0	q_1	1
q_1	q_2	0	q_0	0
q_2	q_0	0	q_1	1

Mealy to Moore Conversion







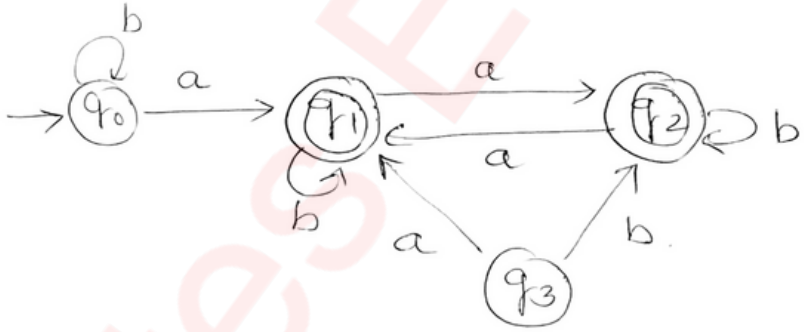


- Minimization of finite Automata
  - → MDFA
  - → minimum no of states

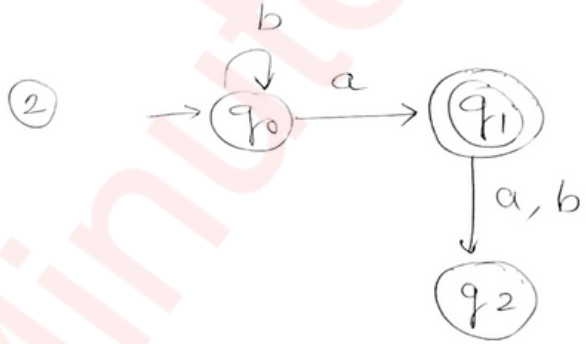
ε-NFA – NFA – DFA – MDFA

- Productivity
  - → Productive states (फायदेमंद)
  - → Non-productive states (NO effect)
    - → Unreachable
    - → dead
    - → Same / Equal.

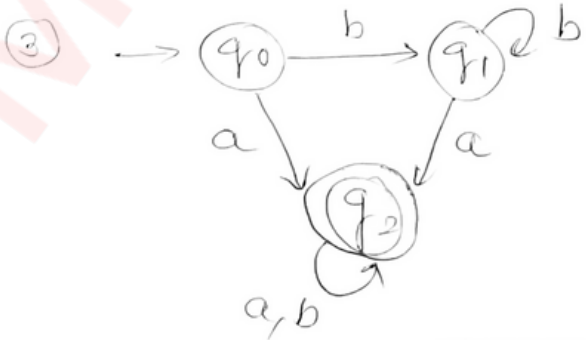
Eg: ①



②



③



- **Regular Expressions**
  - ↳ used to denote RL (describe / represent)
- **Regular Grammer**
  - ↳ Generates RL
- Regular Language is accepted by FA

$\Rightarrow \{\Sigma, \varepsilon, \phi\}$

eg:-

RE	RL
a	$\{a\}$
b	$\{b\}$
$a \cdot b$	$\{ab\}$ no choice
$a+b$	$\{a, b\}$ choice
a^*	$\{\varepsilon, a, aa, aaa, aaaa, \ldots\}$
a^+	$\{a, aa, aaa, \ldots\}$
$a+b+c$	$\{a, b, c\}$

RL → RE1

RL → RE2 → RL1 (✓)

RE2 → RL2 (✗)

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.

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

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.

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

Q: Ends with substring aba or bab

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

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

- $|s| = 2$
  $\Rightarrow (a+b)^2$
- $|s| \leq 2$
  $\varepsilon +$
  $(a+b)$
  $(a+b)^2$
  } choice
- $|s| \geq 2$
  $(a+b)^2.(a+b)^*$

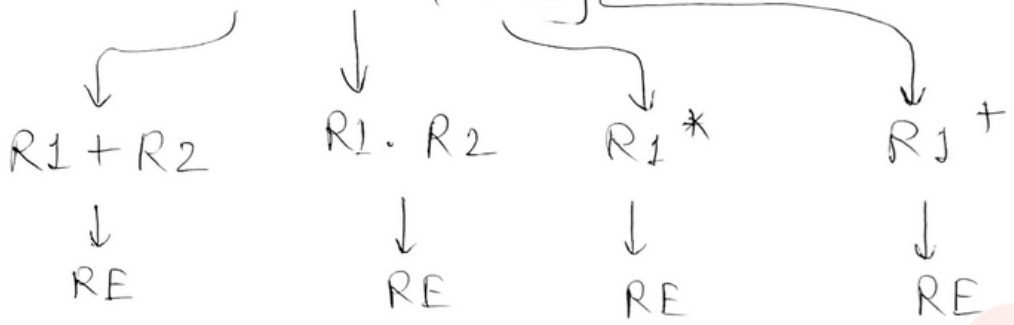
Q: $\Sigma = \{a, b\}$
2nd last is 'a'

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

Q: 2nd from Left is 'a'

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

- Closure Property



- 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

$$R \cdot \varepsilon = R \qquad R + \phi = R$$

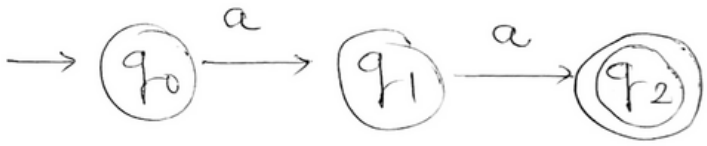
- Idempotent Property

$$R1 + R1 = R1 \qquad R1 \cdot R1 \neq R1$$

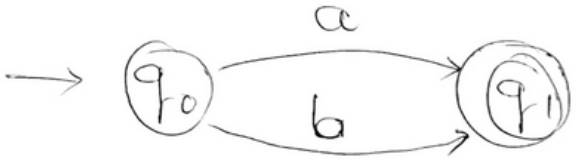
◦ Convert FA → to → RE

↑

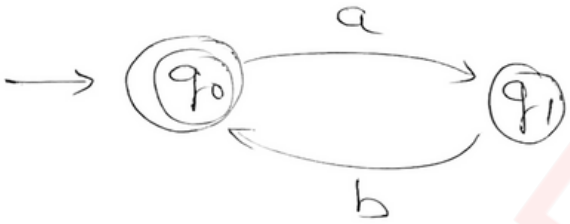
Given



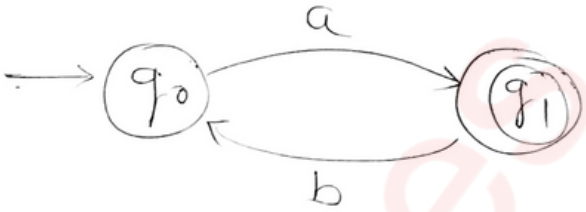
$\Rightarrow a \cdot a$



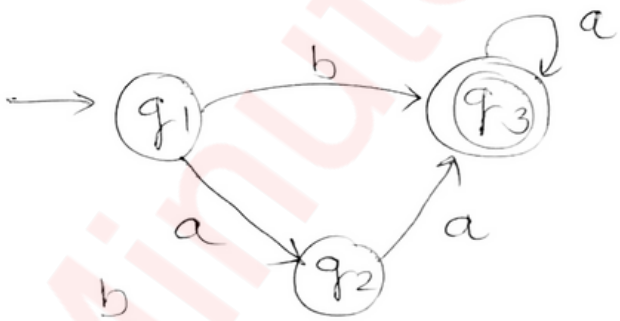
$\Rightarrow a + b$



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

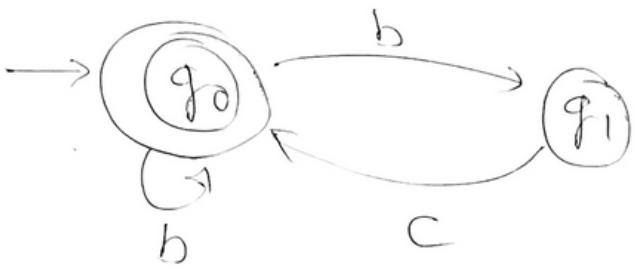


$\Rightarrow a \cdot (b.a)^*$

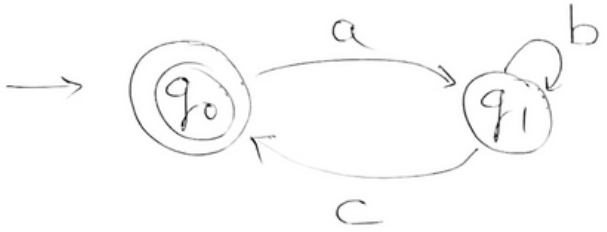


$\Rightarrow (b + a.a).a^*$

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

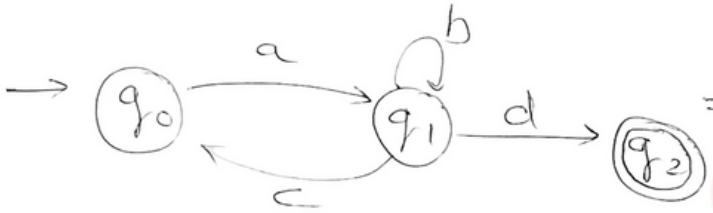


$\Rightarrow (a+bc)^*$

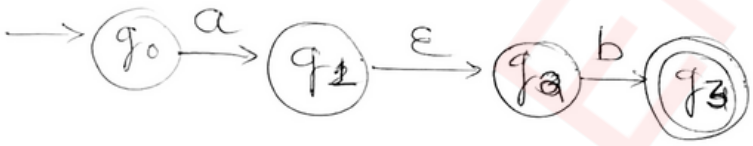


$\Rightarrow a\cdot(b+ca)^*\cdot c$ (X)

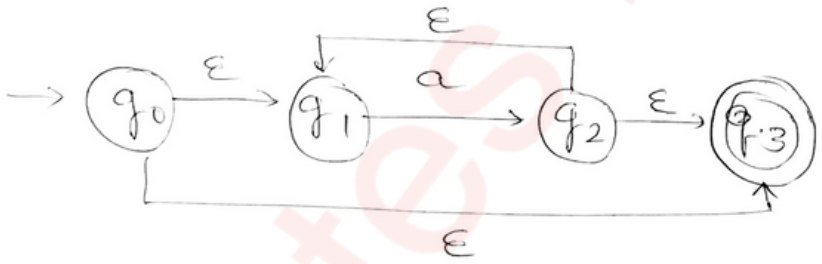
$(a\cdot b^*\cdot c)^*$



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



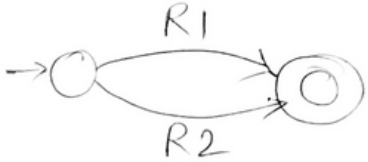
$\Rightarrow a\cdot b$



$\Rightarrow a^*$

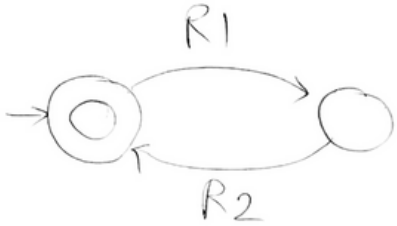
- Convert RE → FA

$R1+R2$

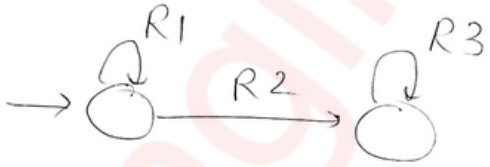


$R1\cdot R2$

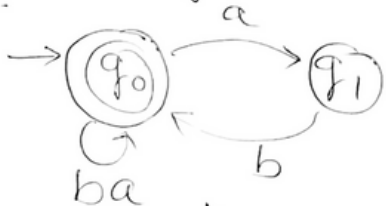
R^* $\Rightarrow$ 

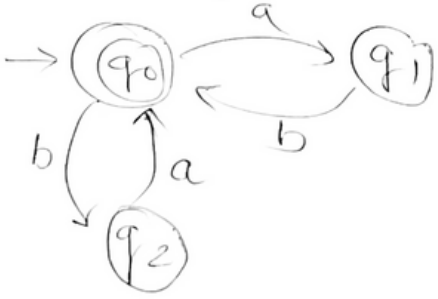
$(R_1 \cdot R_2)^*$ $\Rightarrow$ 

$(R_1 + R_2)^*$ $\Rightarrow$ 

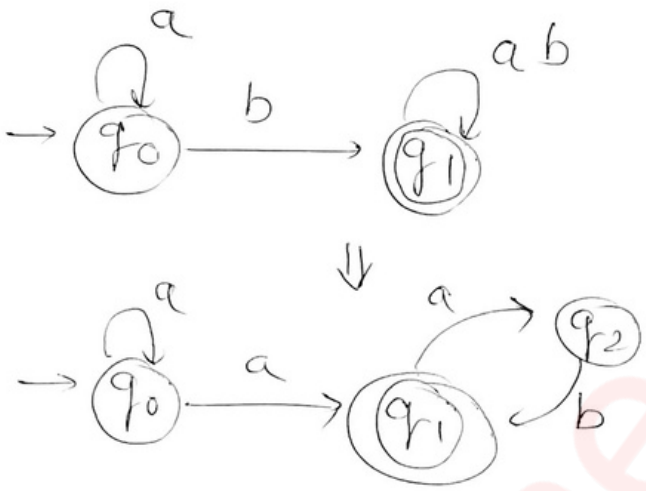
$(R_1^* \cdot R_2 \cdot R_3^*)$ $\Rightarrow$ 

R=
$(ab + ba)^*$ $\Rightarrow$ 

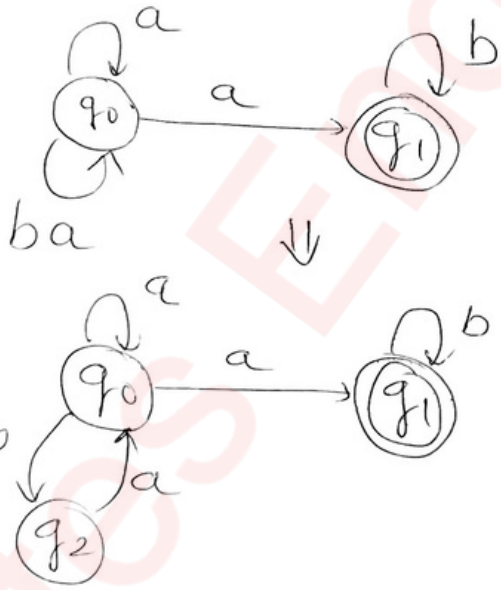




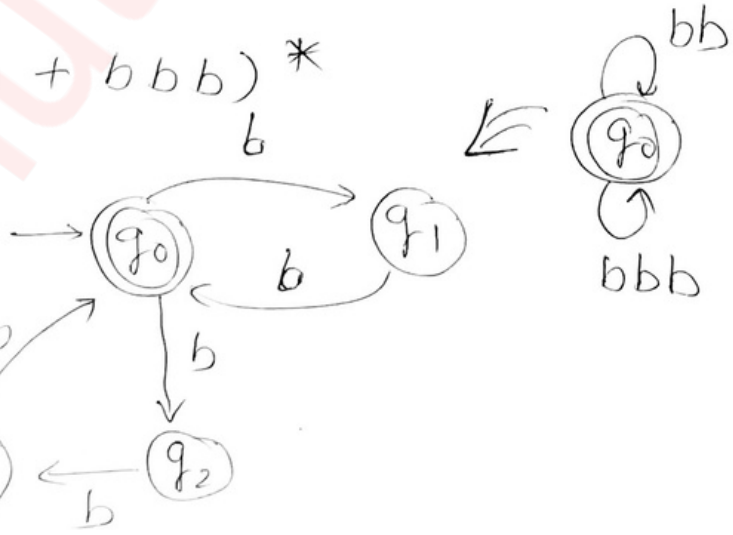
$R = a^* b (ab)^*$



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

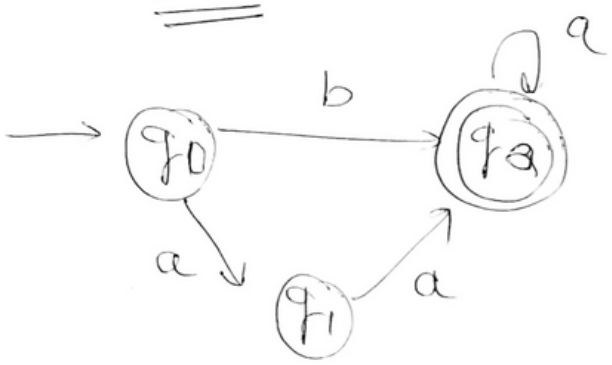


$R = (bb + bbb)^*$



- Arden's theorem: RE ⟵ FA
  - on DFA

eg:



form eqn's:

$$q_0 = \varepsilon \longrightarrow (1)$$

$$q_1 = q_0 \cdot a \longrightarrow (2)$$

$$q_2 = q_0 \cdot b + q_1 \cdot a + q_2 \cdot a \longrightarrow (3)$$

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

$$\downarrow$$

$(R = QP^*)$ soln

$$q_1 = \varepsilon \cdot a = a \longrightarrow (4)$$

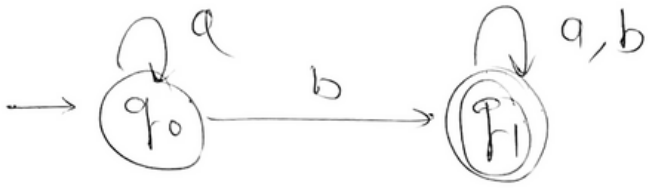
Use (1), (4) in (3)

$$q_2 = \varepsilon \cdot b + a \cdot a + q_2 \cdot a$$

$$= (b + a \cdot a) + q_2 \cdot a \Rightarrow (Q + RP)$$

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

Q:



$$q_0 = \varepsilon + q_0 \cdot a \longrightarrow (1)$$

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

final state in $R = Q + RP$ (in (1)) $\rightarrow$ soln $\rightarrow R = QP^*$

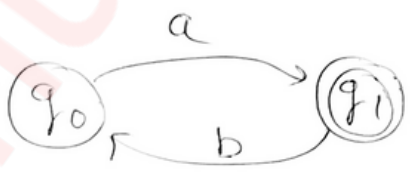
$$q_0 = \varepsilon \cdot a^* = a^* \longrightarrow (3)$$

use (3) in (2).

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

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

Q:



$$q_0 = \varepsilon + q_1 \cdot b \longrightarrow (1)$$

$$q_1 = q_0 \cdot a \longrightarrow (2)$$

use (1) in (2).

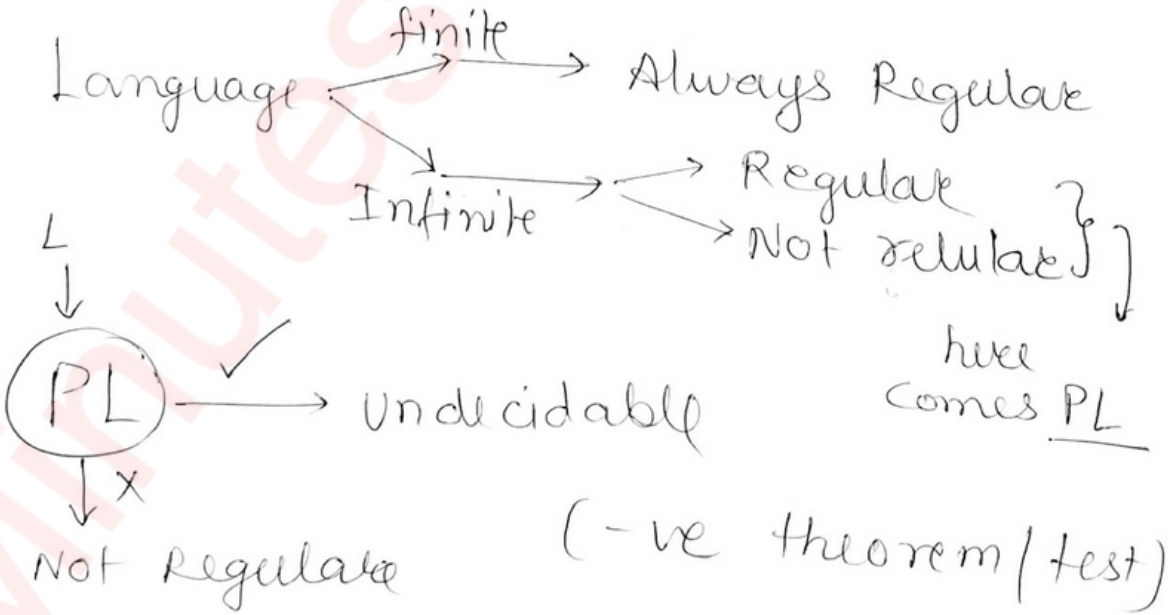
$$q_1 = (\varepsilon + q_1 \cdot b) \cdot a$$

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

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

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

- Pumping Lemma:
  - Check if 'L' is RL or not



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

ω : $\underbrace{aa}_{x} \underbrace{bb}_{y} \underbrace{cc}_{z}$

$\downarrow$ pump $y \rightarrow y^i$ $(bb)^1$ $\downarrow$ $(bb)^2$

$\underbrace{aa}_{x} \underbrace{bbbb}_{y} \underbrace{cc}_{z} \notin L$

So, Its not Regular.

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

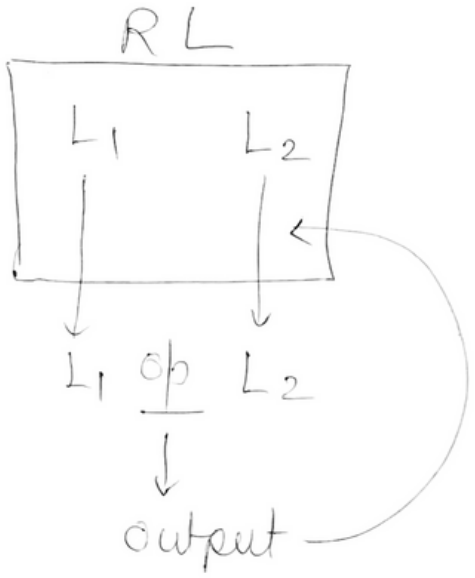
ω : $\underbrace{a}_{x} \underbrace{a\,b}_{y} \underbrace{b\,b}_{z}$

$\downarrow$ pump y

$\underbrace{a}_{x} \underbrace{ab\,ab}_{y} \underbrace{bb}_{z} \notin L$

So, Its not Regular.

Closure Properties of RL



op is closed
Then ∧

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. Complimentation ($\overline{L}$)
6. Difference ($L_1 - L_2$)
7. Reversal (L^R)
8. Homomorphism
9. Reverse Homomorphism
10. Substitution.
11. Infinite Union
12. INIT Operation.

	Regular L	DCFL	CFL	CSL	REC	RE
$\cup$		X				
$\cap$		X	X			
L^C			X			X
$-$		X	X			X
$\cdot$		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	✓	×	×	×	×	×

Grammer

→ Rules for generating correct sentences.

→ Terminal Symbol → Generate (a, b, x, y..)

or

Non Terminal Symbol → helps to Generate (A, B, X, Y..)

→ 4 Tuples N, T, P, S

N: Non-Terminal Symbols (F.set of)

T: Finite set of Terminal Symbols.

P: Production Rules

S: Start symbol.

Types

Type of 'P'

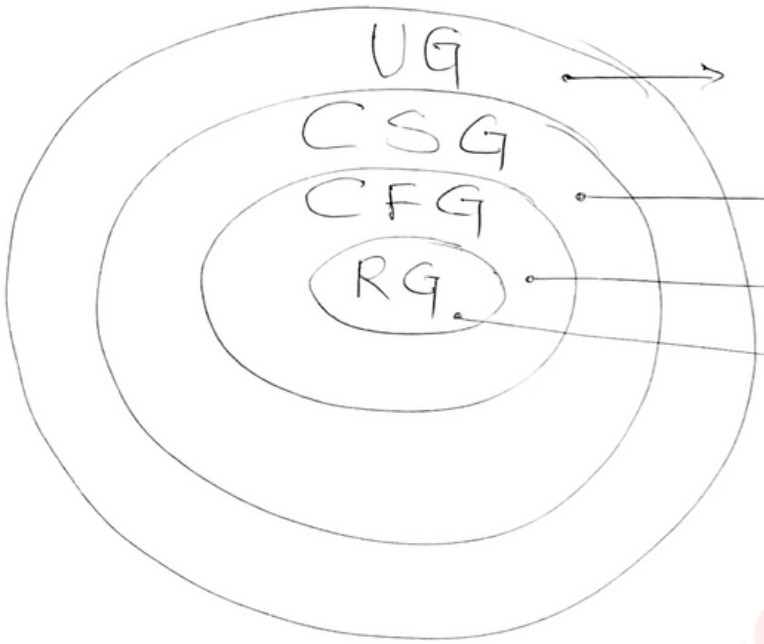
- Type-0
- Type-1
- Type-2
- Type-3

D | T

- Ambigaus
- Unambigous

No. of steings

- Recursive
- Non-Recursive



RG → Accepted by → FA

CfG → Accepted by → Push down Automata

CSG → Accepted by → Linear Bound Automata

UR → Ricognized by → Turning Machine.

Grammer	Language	Automata	'P'
Type 0	Recursively Enumirable	TM	$Sxy \to yx$ $X \to s$
Type 1	CSL	LBA	$S \to XY$ $XY \to xyz$
Type 2	CFL	PDA	$S \to XY$ $X \to x$ $Y \to y$
Type 3	RL	FA	$S \to x$

- Type-0 Grammer (Unrestricted)

→ NO Restriction on Production Rule.

$$\alpha \longrightarrow \beta$$

$\alpha \in \{T \cup V\}^* V \{T \cup V\}^*$ (at least '1' variable)

$\beta \in \{T \cup V\}^*$

- Type-1 Grammer (CSG)

↳ Non-Contracting grammer

$$\alpha \longrightarrow \beta$$

$\alpha \in \{T \cup V\}^* V \{T \cup V\}^*$

$\beta \in \{T \cup V\}^*$

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

- Generates Context sensitive Language.

CFG : (V, T, PS)

Type-2

$$\alpha \longrightarrow \beta$$

Only one Variable

(β Variable as well as Terminal)

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

eg:-

$S \rightarrow XY$

$X \rightarrow x$

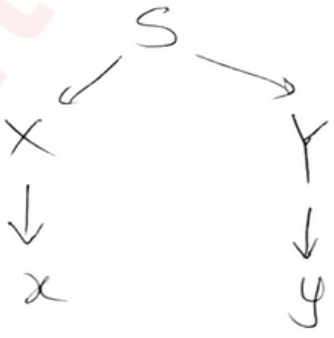
$Y \rightarrow y$

} 'P'

Start symbol : {S}

Variable : {S, X, Y}

Terminal : {x, y}



"xy"

- Type-3 Grammer (Regular)

→ Most Restriction.

→ Left Regular

$X \longrightarrow x \mid Yx$

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

→ Right Regular

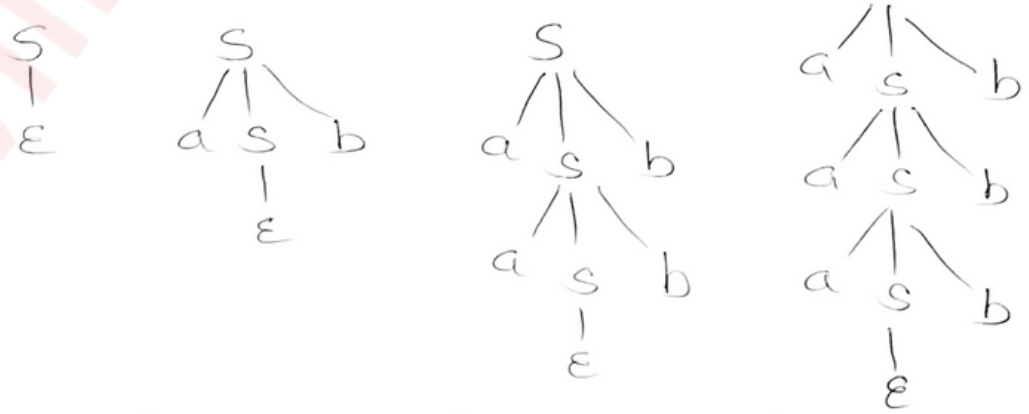
$X \longrightarrow x \mid xY$

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

- Convert CFL ⟶ CFG

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

$S \longrightarrow aSb \mid \varepsilon$



Q: $L = a^m b^n$, $m > n$

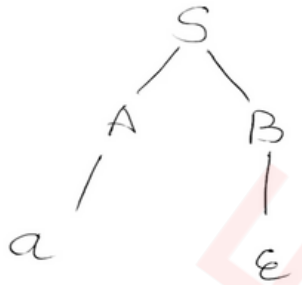
↑

DCFL

$\Rightarrow S \longrightarrow AB$

$B \longrightarrow aBb \mid \varepsilon$

$A \longrightarrow aA \mid a$



$\Rightarrow a^1 b^0 = a$

$m > n$

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

↑

DCFL

$\Rightarrow S \longrightarrow AB$

$B \rightarrow aBb \mid \varepsilon$

$A \rightarrow aA \mid \varepsilon$

Q: $L = a^m b^n$, $m \leq n$

$\Rightarrow$ $S \rightarrow AB$

$A \rightarrow aAb \mid \varepsilon$

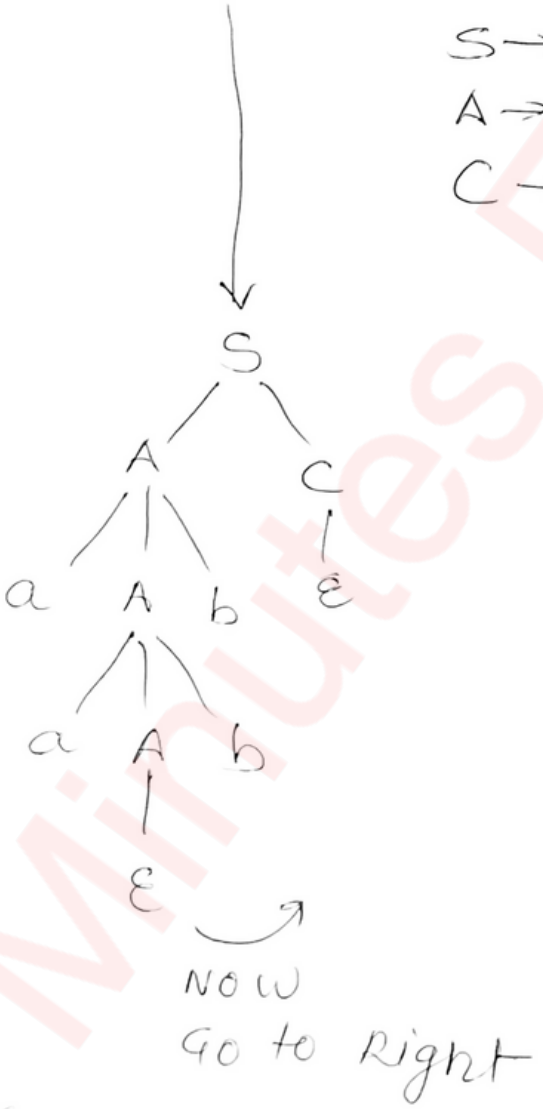
$B \rightarrow bB \mid \varepsilon$

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

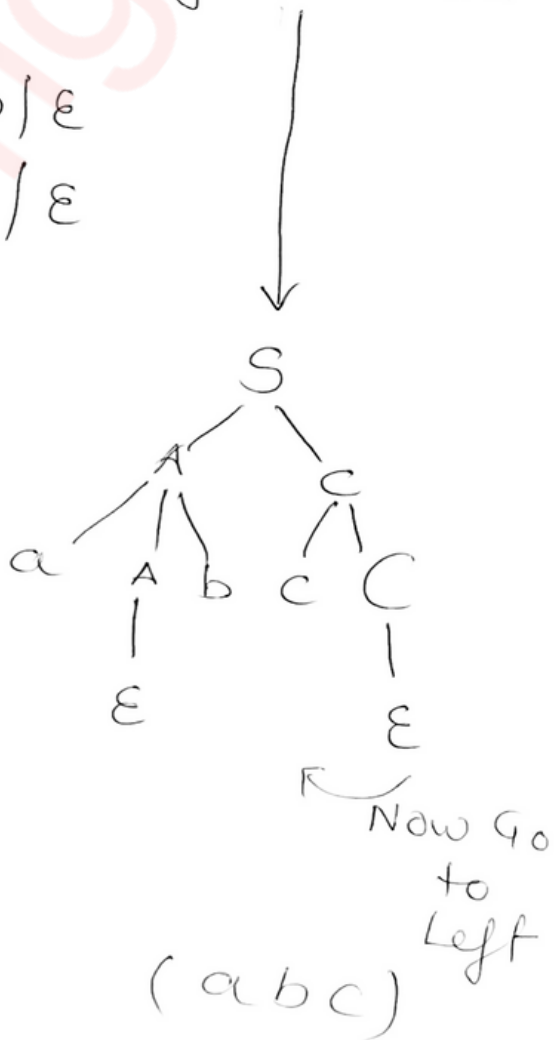
$S \rightarrow AC$
$A \rightarrow aAb \mid \varepsilon$
$C \rightarrow cC \mid \varepsilon$

- Left Most Derivation & Right Most D

$S \rightarrow AC$
$A \rightarrow aAb \mid \varepsilon$
$C \rightarrow cC \mid \varepsilon$



(a a b b)

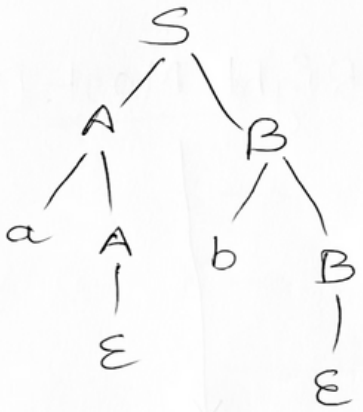


(a b c)

◦ Ambiguous Grammer

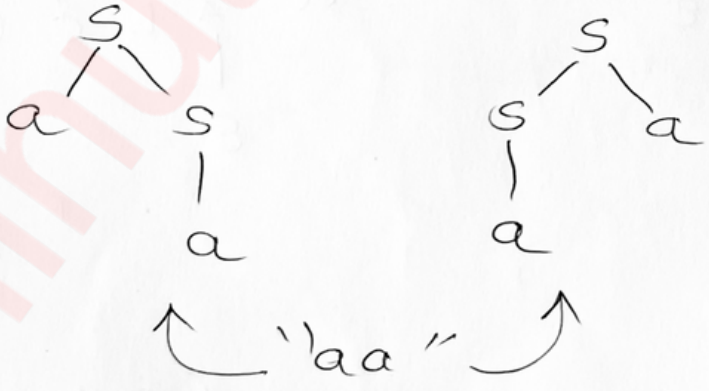
↳ for a string if there are more than one DT (LMDT or RMDT)

eg: $S \rightarrow AB$
$A \rightarrow aA \mid \varepsilon$
$B \rightarrow bB \mid \varepsilon$



"ab" Unambiguous

eg: $S \rightarrow aS \mid Sa \mid a \Rightarrow a^+$



But ↱ at
$S \rightarrow aS \mid a$
Unambiguous

• Minimization of CFG

- → ε / Null
- → unit
- → useless

} productions elimination

eg: $S \rightarrow XxY$

$X \rightarrow x \mid \epsilon$

$Y \rightarrow y \mid \epsilon$ ← $Y \rightarrow \epsilon$ remove

$X \rightarrow \epsilon$ remove

⇓

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

$X \rightarrow x$

$Y \rightarrow y$

eg: $S \rightarrow A$

$A \rightarrow B$

$B \rightarrow C$

$C \rightarrow x$

} Removing unit production

⇒ $S \rightarrow x$

eg:

$S \rightarrow Aa$

$A \rightarrow a/B$

$B \rightarrow d$

So here $A \rightarrow B$ is Unit P so remove it.

$\Downarrow$

$S \rightarrow Aa$

$A \rightarrow a/d$

$B \rightarrow d$ ⟶ now here $B \rightarrow d$ is never used / reach-able its useless so remove it.

$\Downarrow$

$S \rightarrow Aa$

$A \rightarrow a/d$

eg:

$S \rightarrow xX/yY$

$X \rightarrow y$

So here $S \rightarrow y$  its dead here. So remove it

$\Downarrow$

$S \rightarrow xX$

$X \rightarrow y$

- Chomsky Normal form (CNF)
  - ↳ Its a way of rewriting a CFG to simplify parsing.

Production form

$$X \rightarrow YZ \mid x$$

$X \rightarrow V$, $YZ \rightarrow V$, $x \rightarrow T$

eg: $S \rightarrow xSy \mid xy$

$X \rightarrow x \qquad Y \rightarrow x$

So

$S \rightarrow XSY \mid XY$

$S1 \rightarrow XS$

So finally:

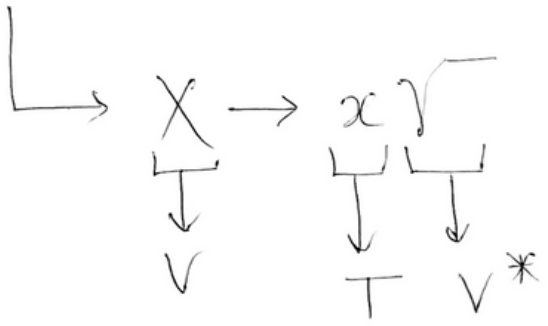
$S \rightarrow S1Y \mid XY$

$S1 \rightarrow XS$

$X \rightarrow x$

$Y \rightarrow y$

Greibach Normal form (GNF)



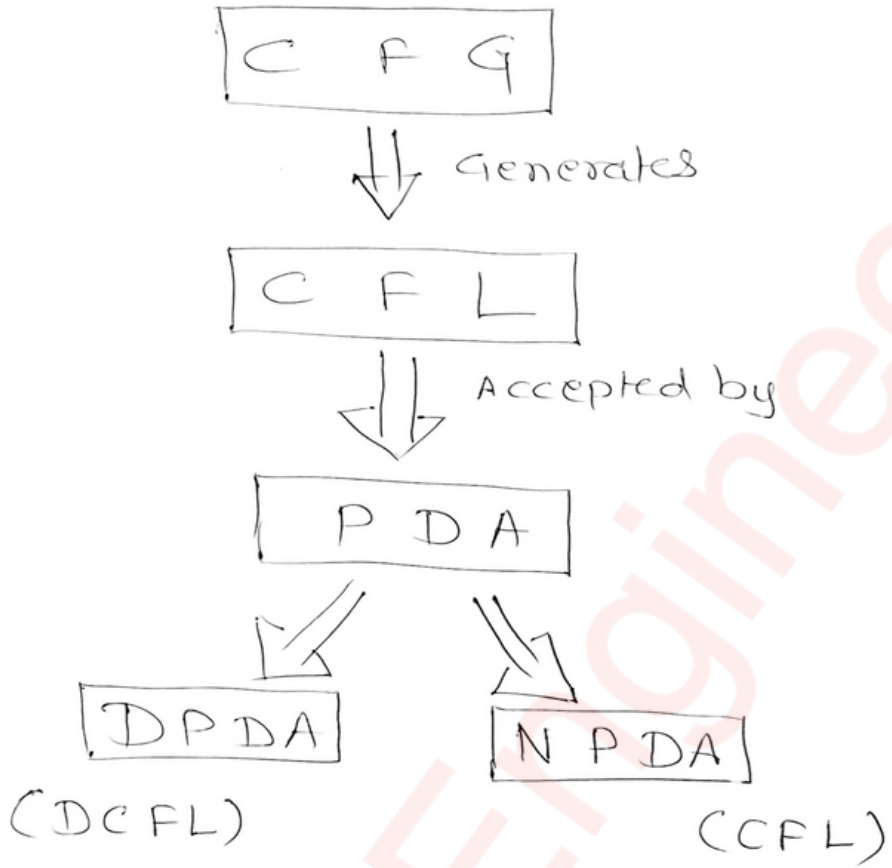
eg: $S \rightarrow xSy \mid xy$

$Y \rightarrow y$ } just add this PR

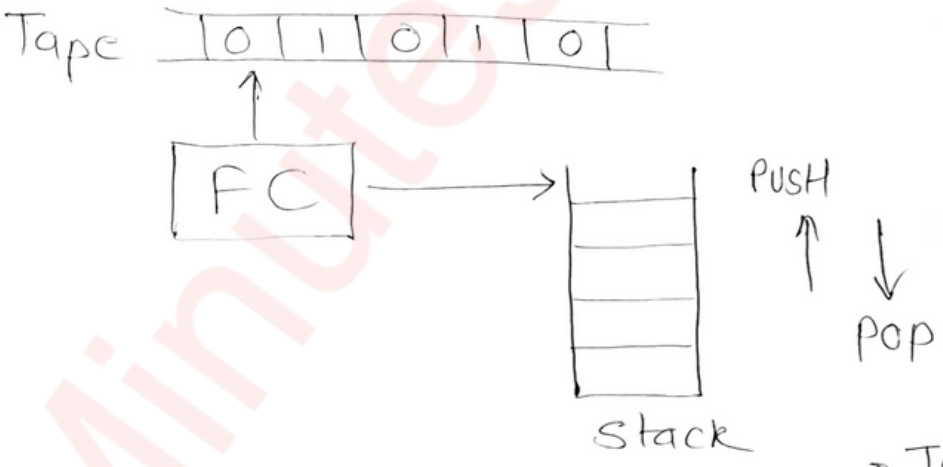
So, $S \rightarrow xSY \mid xY$
$Y \rightarrow y$
} In GNF

eg:- $S \rightarrow aAB \mid aB$
$A \rightarrow aA \mid a$
$B \rightarrow bB \mid b$
} Already In GNF.

◦ Push down Automata



PDA = FA + Stack

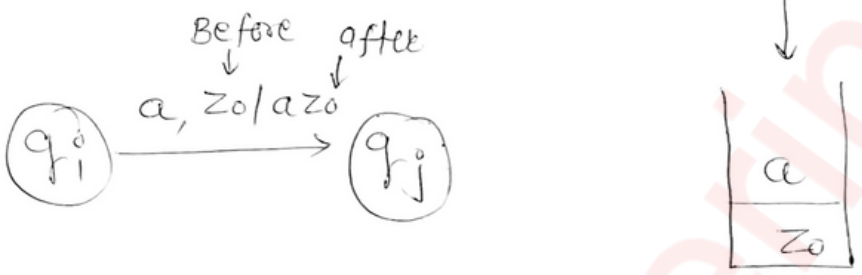


$P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ — Z_0: ToS

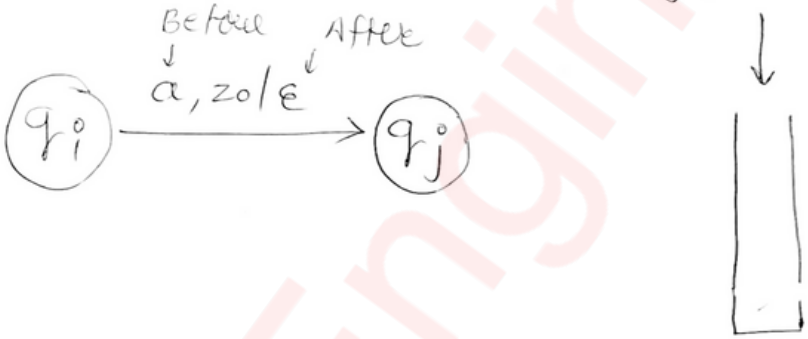
Γ: Stack Alphabet

$\delta : Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow Q \times \Gamma^*$

① PUSH : $\delta(q_i, a, z_0) \rightarrow (q_j, az_0)$



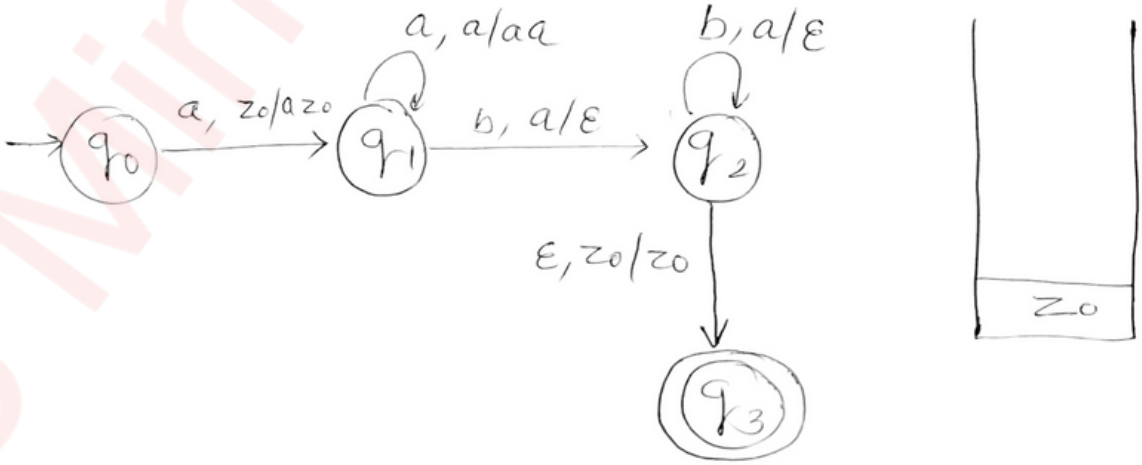
② POP : $\delta(q_i, a, z_0) \rightarrow (q_j, \varepsilon)$



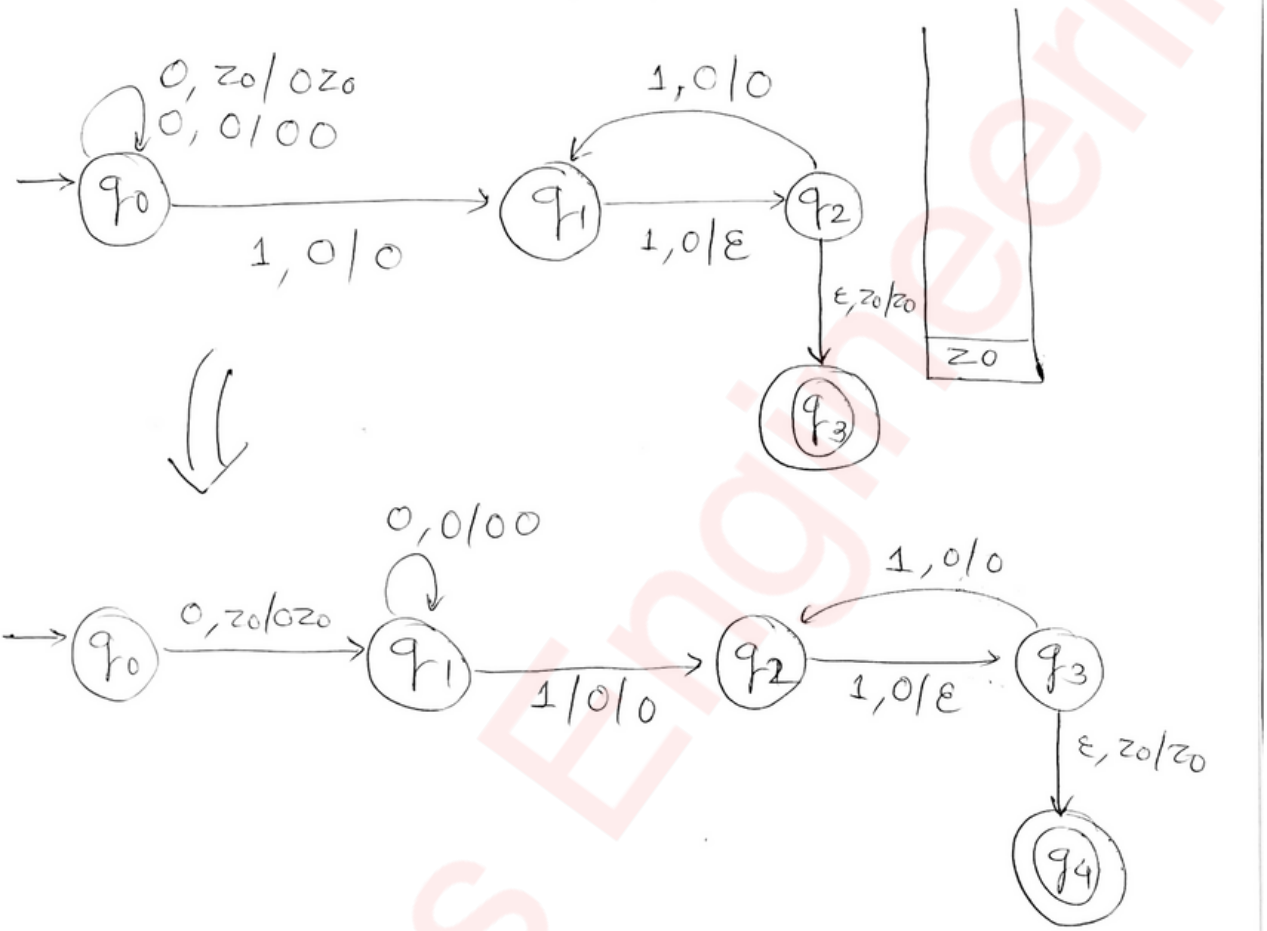
③ SKIP : $\delta(q_i, a, z_0) \rightarrow (q_j, z_0)$

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

a	a	a	b	b	b	ε



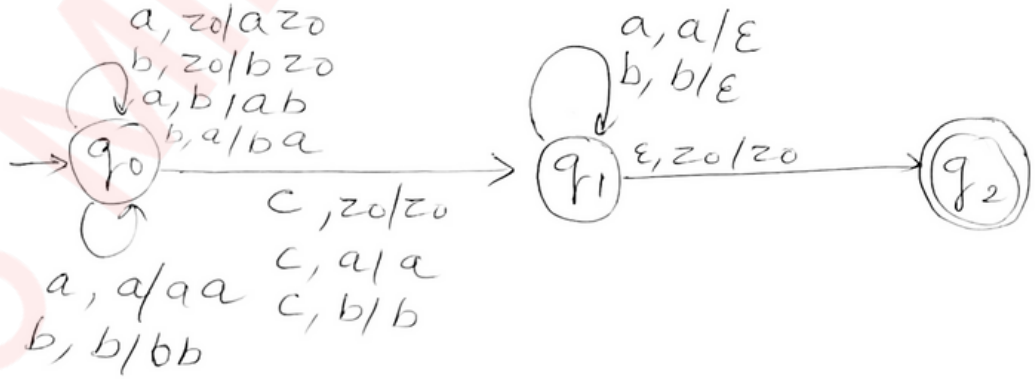
Q: $L = 0^n 1^{2n}$; $n > 0$



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

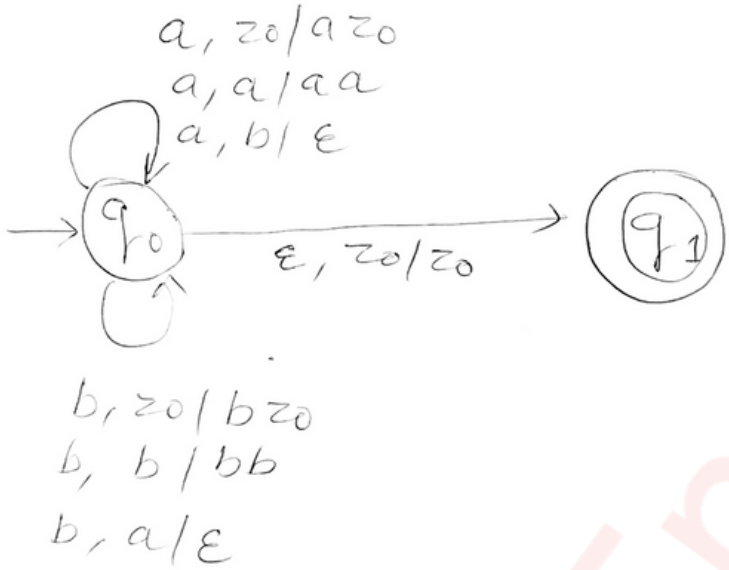
⇓

palindrome: a a b c b a a



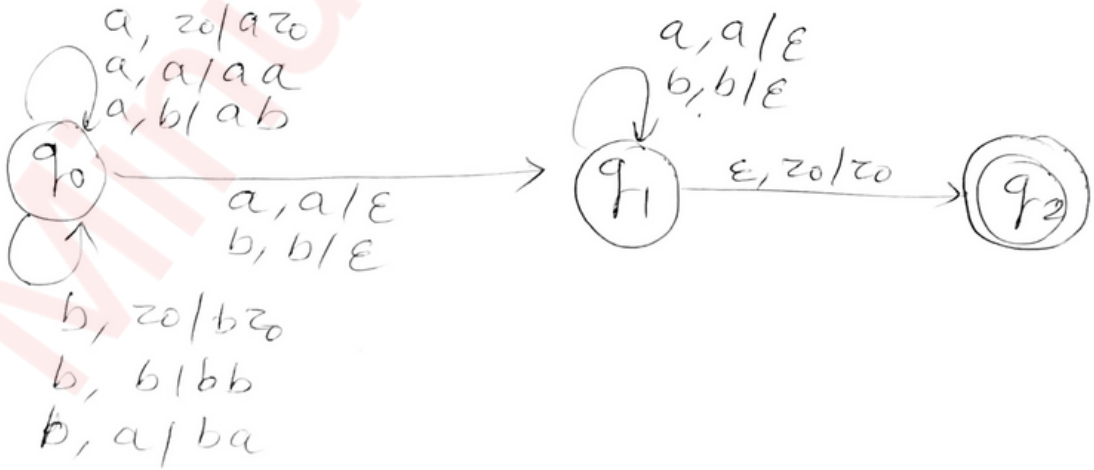
Q: $L = \{ \text{no. of } a = \text{no. of } b \}$ in any order

a	a	b	a	b	b	ε



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

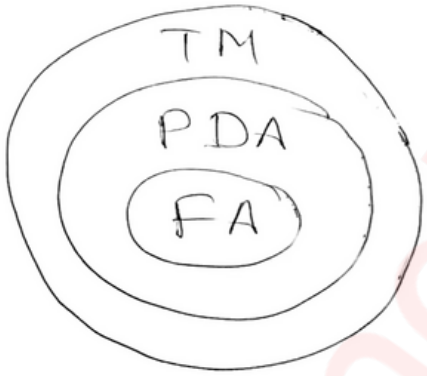
a	b	a	a	b	a	ε



Turing Machine

FA < PDA < TM

a^n $a^n b^n$ $a^n b^n c^n$

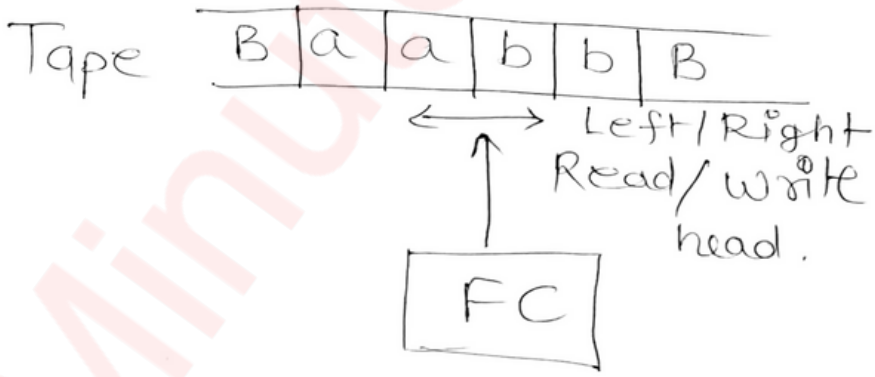


$(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

$\downarrow$

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

Left Right



Linear Bounded Automata (LBA)

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

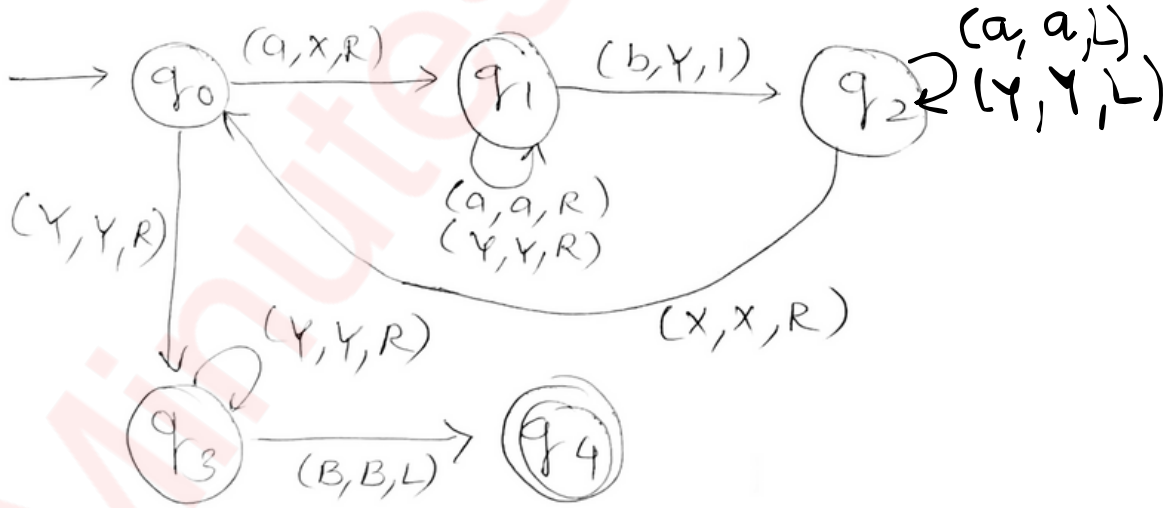
FA < PDA < LBA < TM

$a^n b^n c^n$, $n \geq 1$

a^n for → n prime
→ n non prime

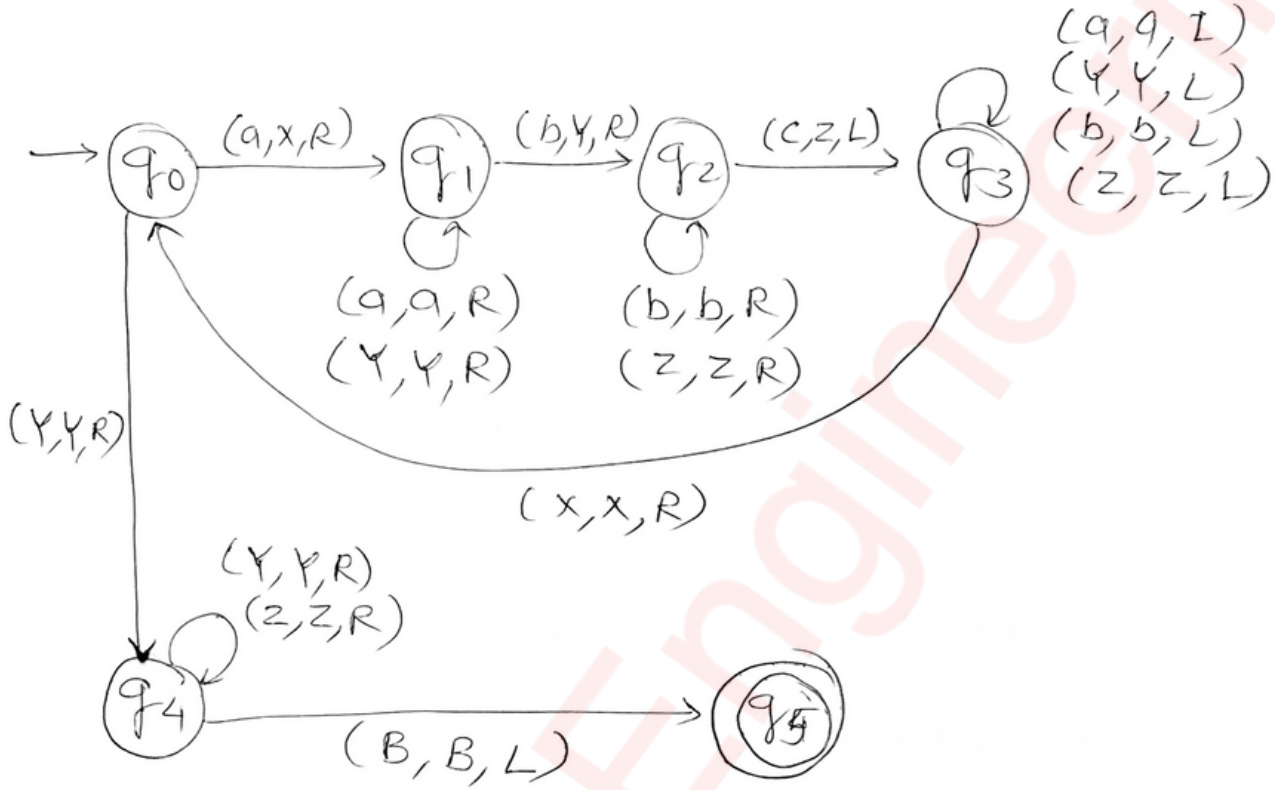
Q: $L = a^n b^n \mid n \geq 1$

B	B	a	a	b	b	B	B



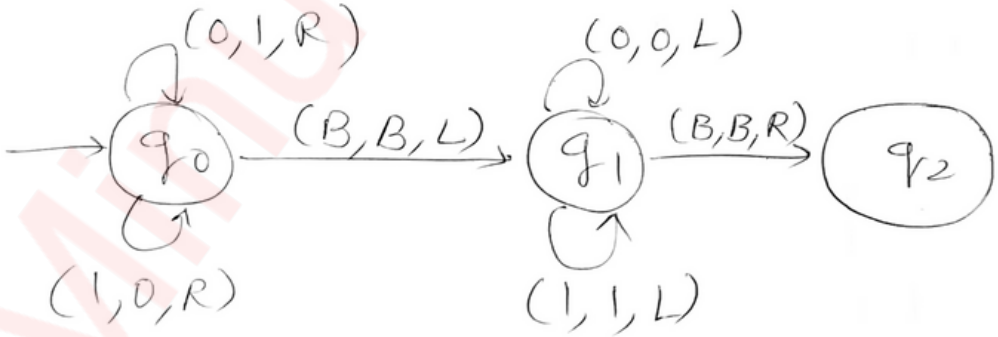
Q: $L = \{a^n b^n c^n \mid n \geq 1\}$

B	a	a	b	b	c	c	B

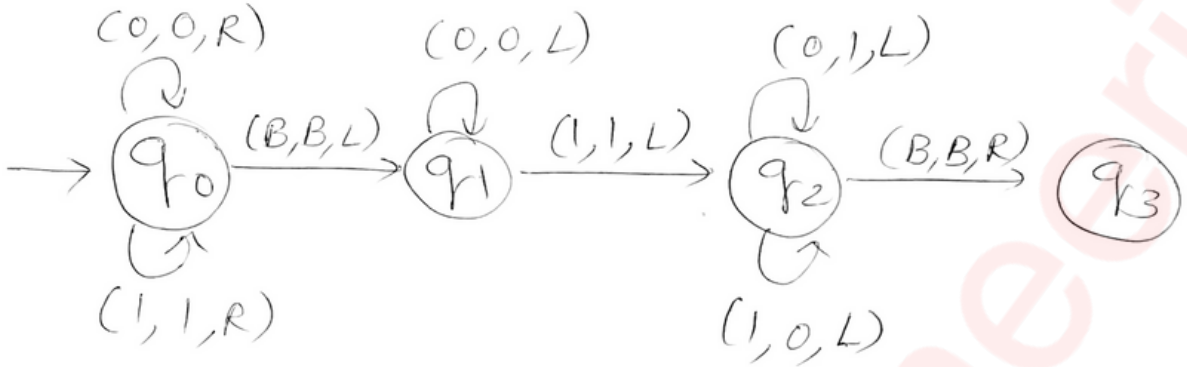


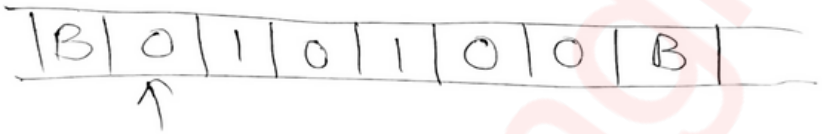
Q: 1's Complement

B	0	1	0	0	1	B



Q: 2's Complement





∘ Modification / variation / Versions of Turing machine.

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

→ Same power:

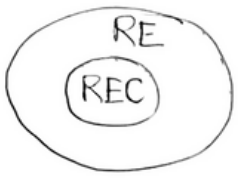
1. Stay option
2. Semi infinite tape
3. offline TM
4. Multidimentional TM
5. Multitape TM
6. Multihead TM
7. NDTM
8. Universal TM
9. Jumping TM
10. Non-Erasing TM
11. Always write TM

FA < DPDA < NPDA < LBA < TM

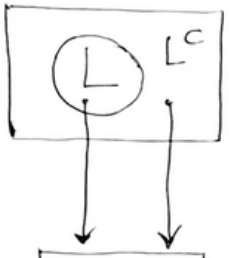
RE Language

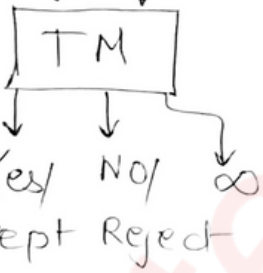
→ TM

→ Can go in ∞ Loop (Never halt) state

→ 

→ Σ^*





→ $\{-, L^c\}$ Not closed

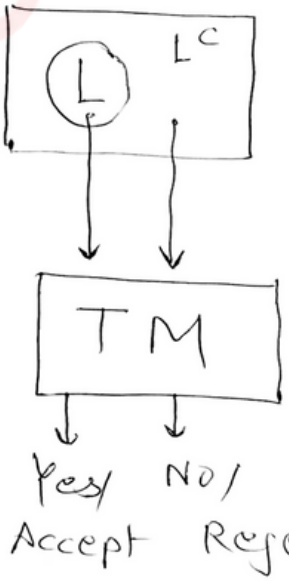
REC Language

→ Halting TM

→ No such state

→ Subset of RE

→ Σ^*



→ Homorphism & 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 problem → Problem
  - Algo (✓) (D)
  - Algo (x) (UD)

- In terms of Lang. → Language
  - Halting TM (D)
  - Non-Halting TM (UD)