TOC

S → Automata → L → Yes / No

FA

- FA with o/p
  - Mealy
  - Moore
- FA without o/p
  - DFA
  - NFA
  - ε-NFA

Automata	Def	Transition fn
DFA	$(Q, \Sigma, \delta, q_0, F)$	$\delta: Q \times \Sigma \to Q$
NFA	$(Q, \Sigma, \delta, q_0, F)$	$\delta: Q \times \Sigma \to 2^Q$
ε-NFA	$(Q, \Sigma, \delta, q_0, F)$	$\delta: Q \times \Sigma \cup \{\epsilon\} \to 2^Q$
Moore	$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$	$\delta: Q \times \Sigma \to Q$, $\lambda: Q \to \Delta$
Mealy	$(Q, \Sigma, \delta, q_0, \Delta, \lambda)$	$\delta: Q \times \Sigma \to Q$, $\lambda: Q \times \Sigma \to \Delta$
DPDA	$(Q, \Sigma, \delta, q_0, z_0, F, \Upsilon)$	$\delta: Q \times \Sigma \cup \{\epsilon\} \times \Upsilon \to Q \times \Upsilon^*$
NDPDA	$(Q, \Sigma, \delta, q_0, z_0, F, \Upsilon)$	$\delta: Q \times \Sigma \cup \{\epsilon\} \times \Upsilon \to 2^{Q \times \Upsilon^*}$
TM	$(Q, \Sigma, \Upsilon, \delta, q_0, B, F)$	$\delta: Q \times \Upsilon \to Q \times \Upsilon \times \{L, R\}$
NDTM	$(Q, \Sigma, \Upsilon, \delta, q_0, B, F)$	$\delta: Q \times \Upsilon \to 2^{Q \times \Upsilon \times \{L, R\}}$

*) Complimentation applies only to DFA (Not on NFA)

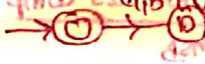
DFA: $(Q, \Sigma, \delta, q_0, F)$

$\overline{DFA}$: $(Q, \Sigma, \delta, q_0, Q-F)$

Reversal of a language applies for NFA

*) $L = \{\}$ $\Sigma = \{a, b\}$.

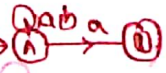
*) $L = \{\epsilon\}$ $\Sigma = \{a, b\}$



NFA to DFA Conversion:

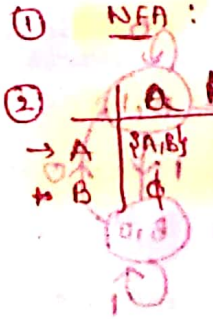
$\Sigma = \{a, b\}$ L = Ends with a

① NFA:



②

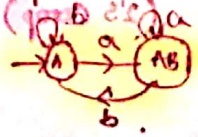
	a	b
→ A	{A,B}	A
*B	ϕ	ϕ



③

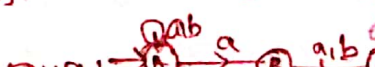
	a	b
→ A	[AB]	A
*[AB]	[AB]	A

④



$\Sigma = \{a, b\}$ L = second sym from RHS is a

①NFA:

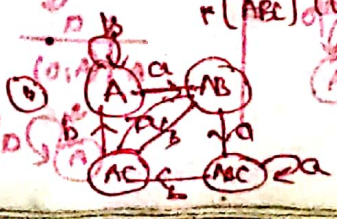


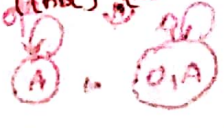
②

	a	b
→ A	{A,B}	A
B	C	
*C	ϕ	ϕ

③

	a	b
→ A	[AB]	A
*[AB]	[AB]	[AC]
*[AC]	[AB]	A
*[ABC]	[ABC]	[AC]





*) If NFA is formed with 'n' states then in worst case DFA will form with 2^n states.

ie Num of states (DFA) $\geq$ num of states (NFA)

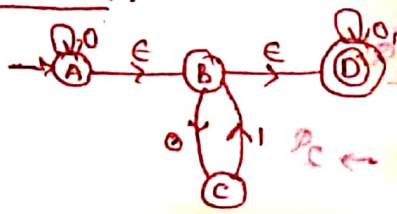
Minimization of DFA:

step(1): Remove the unreachable states from Initial state

step(2): Draw state transition table

step(3): Find n Equivalents until two equivalent groups become equal.

ε-NFA:



ϵ-closure(A) = {A, B, D}

ϵ-closure(B) = {B, D}

ϵ-closure(C) = {C}

ϵ-closure(D) = {D}

ε-NFA to NFA Conversion:

step(1): Find closures of all states

step(2): Draw state transition table.

step(3): Find ϵ^* & ϵ^*

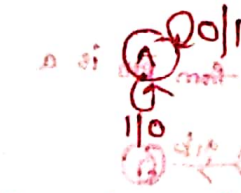
step(4): Draw NFA.

Moore Machine & Mealy Machine:

*) If there are n inputs in moore m/c, then n+1 o/p's will be produced. whereas if there are n inputs in mealy m/c then n o/p's are produced.

Note *) No Final state is present in moore and mealy m/c's.

Mealy m/c for 1's comp:

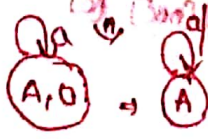


Mealy m/c for 2's comp

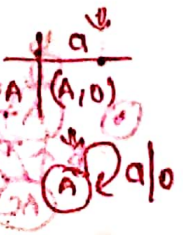
LSB to MSB

Moore to Mealy

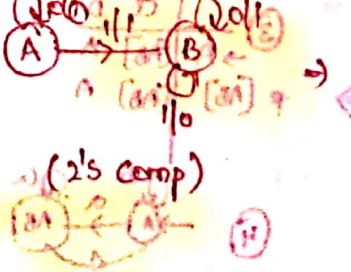
By Transition diagram

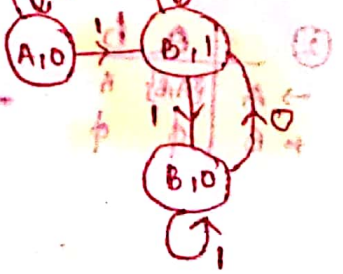


By transition table



Mealy to Moore



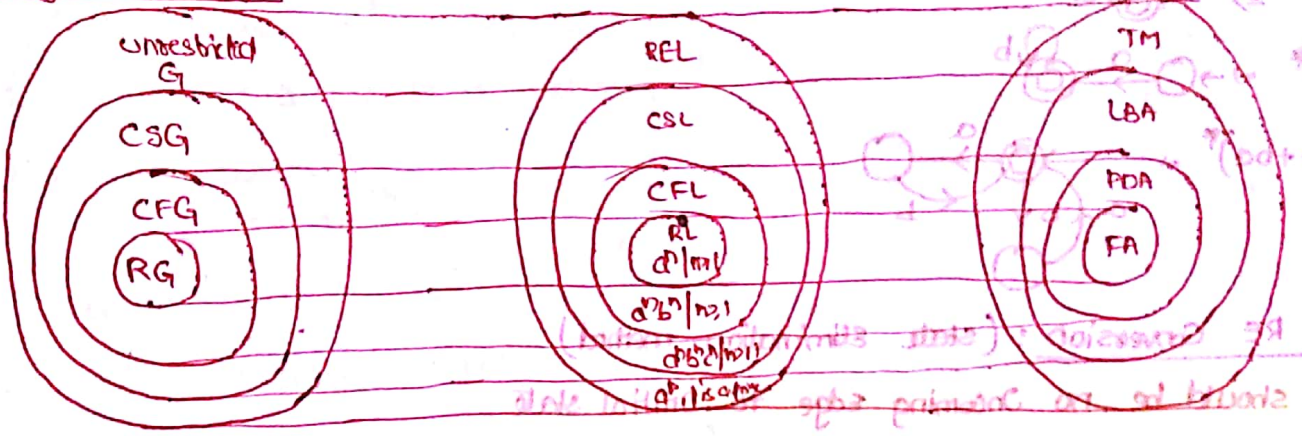


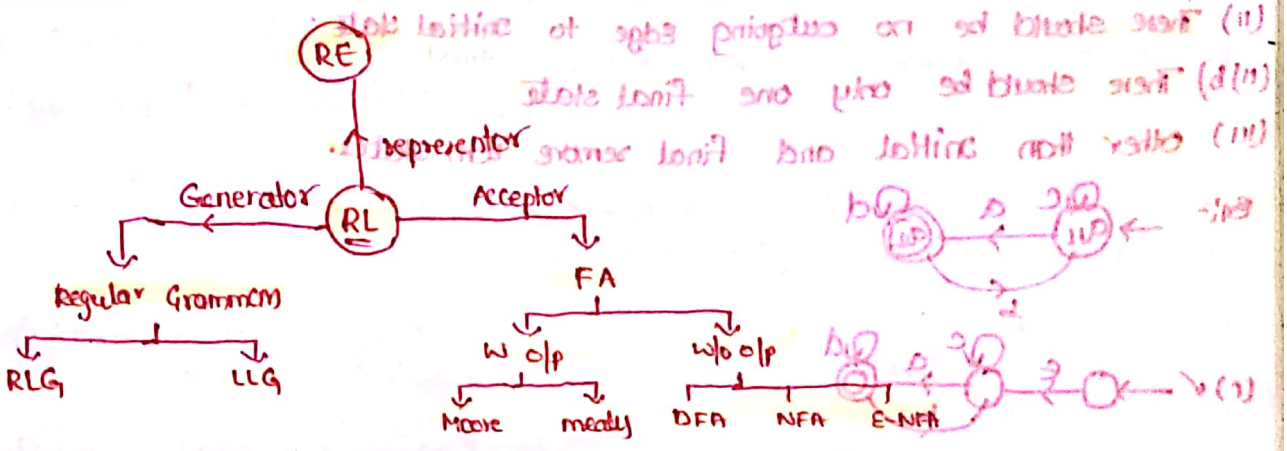
*) Moore → mealy → num of states doesn't change

Mealy → moore → num of states increases.

→ worst case: N states, M o/p's → Num of states in moore = N×M

Regular language :





Regular Expression :

Contains three types of operators. (i) + (union) (ii) · (concat) (iii) * (Kleen closure)

Ex:- All strings of length atmost 2

RE: $(a+b+\epsilon)(a+b+\epsilon)$

String when a's are even

RE: $(b^*ab^*ab^*)^* + b^*$

Starting and ending with same symbol

RE: $a(a+b)^*a + b(a+b)^*b + a + b + \epsilon$

2 a's should not come together.

RE: $(b+ab)^* + (b+ab)^*a$.

$: (b+ab)^*(\epsilon+a)$

Atmost 2 a's

RE: $b^*(\epsilon+a)b^*(\epsilon+a)b^*$

no 2 a's & no 2 b's should come together.

$L = \{\epsilon, a, b, ab, aba, ba, bab \dots\}$

RE: $(ab)^*a + (ab)^* + b(ab)^*a + b(ab)^*$

$: (ab)^*(a+\epsilon) + b(ab)^*(a+\epsilon)$

RE: $(a+\epsilon)(\epsilon+b)(ab)^*$

Identities of RE :

① $\phi + R = R + \phi$; $\phi \cdot R = R \cdot \phi = \phi$

② $\epsilon \cdot R = R \cdot \epsilon = R$

③ $\epsilon^* = \epsilon$; $\phi^* = \epsilon$; $\epsilon + RR^* = R^*R + \epsilon = R^*$

④ $(a+b)^* = (a^* + b^*)^*$

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

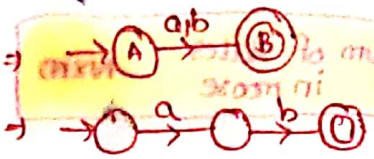
$= (a^* + b)^*$

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

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

RE to FA Conversion :

ϕ ⇒

a+b ⇒ 

a.b ⇒

a* ⇒ 

ab* ⇒ 

(ab+ba)* ⇒ 

FA to RE Conversion : (State Elimination method)

(i) There should be no incoming edge to initial state

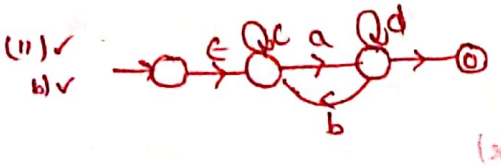
(ii) There should be no outgoing edge to ~~initial~~ final state

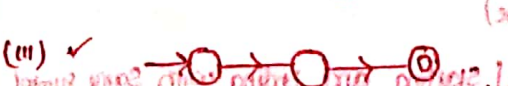
(iii)b) There should be only one final state

(iii) other than initial and final remove rem states.

Ex:- 

(i) ✓ 

(ii) ✓
b) ✓ 

(iii) ✓ 

$c^*a(d+ba)^*$

RE : $c^*a(d+ba)^*$

Testing whether a language is regular or not

Lang

- Finite
  - ∴ it is RL
  - Ex: {ab, ba, aa, bb}
- Infinite.
  - Pattern ✓ → ∴ it is RL
  - Pattern ✗ → ∴ it is not RL
    - $L=\{a^p \mid p \text{ is prime}\}$
    - $L=\{ab, aabb, \ldots\}$

Ex:- $L=\{a^n b^n \mid n \le 10^{10^{10}}\}$ ⇒ RL

$L=\{a^n b^n \mid n \ge 1\}$ ⇒ Not RL

① $ww^R \mid w \in (a+b)^*$ ⇒ Not RL

② $ww \mid w \in (a+b)^*$ ⇒ not RL

③ $wxw^R \mid w \in (a+b)^*$ ⇒ not RL

⑤ $wxw^R \mid w \in (a+b)^+$ & $|x|=5$ ⇒ not RL

⑥ $xwwy \mid w \in (0+1)^+$ x, y are fixed ⇒ not RL

⑦ $xww^R \mid w \in (0+1)^+$ ⇒ Not RL

⑧ $wwy \mid w \in (0+1)^+$ ⇒ not RL

*) Modular Counting is finite ⇒ RL

$w \mid w \in (a+b)^*$ & $n_a(w) \bmod 3 = n_b(w) \bmod 3$

*) $w \mid w \in (a+b)^*$ & $n_a(w) = n_b(w)$ ⇒ Not RL

*) $L=\{a^n \mid n$ is even / odd $\}$ ⇒ (AP) Pattern ⇒ RL

$L=\{a^n \mid n$ is prime $\}$ ⇒ (Not AP) ⇒ Not RL

$L=\{a^i b^{j^2} \mid i,j \ge 1\}$ ⇒ (Not AP) ⇒ Not RL

(i) $wxw^R \mid w,x \in (0+1)^+$ ⇒ RL

(ii) $xwwy \mid w,x,y \in (0+1)^+$ ⇒ RL

Regular grammar

$G=(V, T, P, S)$

Lang ↓ RE ↓ RG

(i) starts & ends with same sym.

RE: $a(a+b)^*a + b(a+b)^*b + a + b + \epsilon$

P: $S \to aAa \mid bAb \mid a \mid b \mid \epsilon$

$A \to aA \mid bA \mid \epsilon$

$L=\{a^n b^n \mid n \ge 1\}$

Not RL

$S \to aSb \mid ab$.

$L=\{a^n b^m c^n d^n, m,n \ge 1\}$

Not possible.

$L=\{a^n b^n c^n \mid n \ge 1\}$

Not possible.

$L=\{a^n c^m b^n \mid n,m \ge 1\}$

$S \to aAb \mid abb \mid \epsilon$

$A \to cA \mid \epsilon$

Chomsky classification

- Type-3 (RG)
  - RLG: $A \to \alpha B \mid \beta$, $A, B \in V$, $\alpha, \beta \in T$
  - LLG: $A \to B\alpha \mid \beta$, $A, B \in V$, $\alpha, \beta \in T$
- Type-2 (CFG): $A \to \alpha$, $A \in V$, $\alpha \in (VUT)^*$
- Type-1 (CSG): $A \to B$, $A, B \in (VUT)^*$, $|A| \leq |B|$
- Type-0 (UR G): $A \to B$, $A \in (VUT)^*$, $B \in (VUT)^*$

FA to RG Conversion :

$$FA \xrightarrow{} RLG \xrightarrow{R} LLG$$
$$(L) \qquad (L) \qquad (L^R)$$

$$FA \xrightarrow{R} FA \xrightarrow{} RLG \xrightarrow{R} LLG$$
$$(L) \qquad (L^R) \qquad (L^R) \qquad (L)$$

RG to FA Conversion :

RLG to FA:

$$RLG \xrightarrow{} FA$$
$$(L) \qquad (L)$$

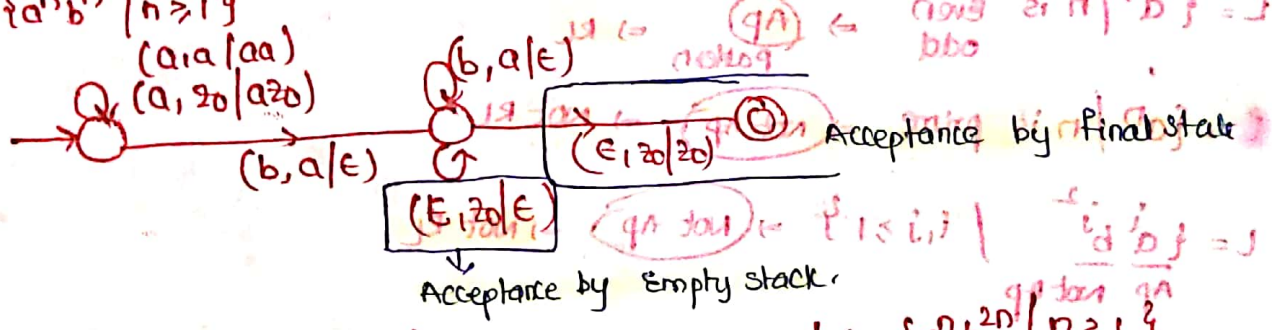
LLG to FA:

$$LLG \xrightarrow{R} RLG \xrightarrow{} FA \xrightarrow{R} FA$$
$$(L) \qquad (L^R) \qquad (L^R) \qquad L$$

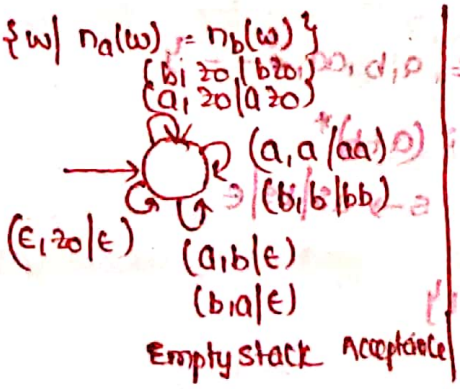
How ??

Push down Automata : (FA + stack)

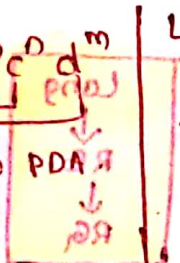
$L = \{a^n b^n \mid n \geq 1\}$



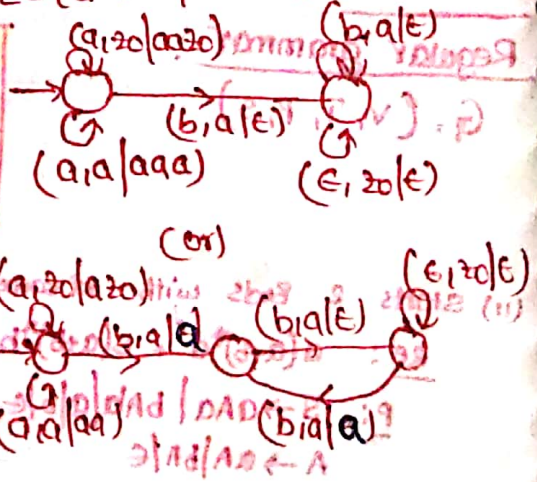
$L = \{w \mid n_a(w) = n_b(w)\}$



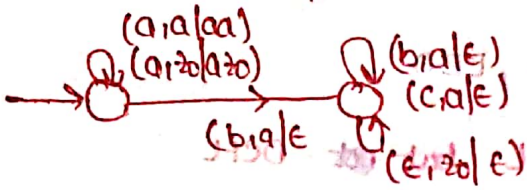
$L \subseteq \{a^n b^m c^n d^m\}$



$L = \{a^n b^{2n} \mid n \geq 1\}$



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



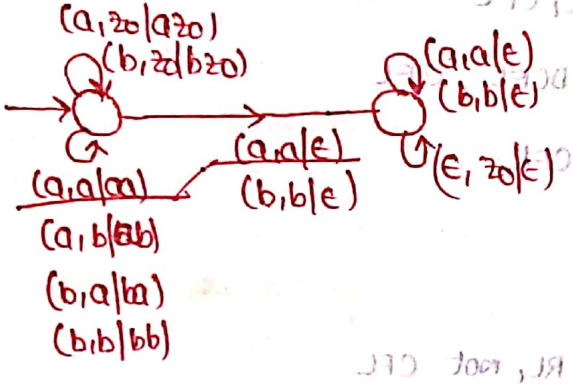
It also accepts aabbcc, $\notin L$

∴ No PDA but TM possible

Non deterministic PDA:

$L = \{ww^R \mid w \in (a,b)^+\}$

It is difficult to find centre, but there may be a chance of two symbols are same.



Working:

aaaa

$(q_1, aaa, a z_0)$

↓ No centre → | centre.

$(q_1, aa, aa z_0)$ | (q_2, aa, z_0) ×

↓ NC | C

$(q_1, a, aaa z_0)$ | $(q_2, a, a z_0)$

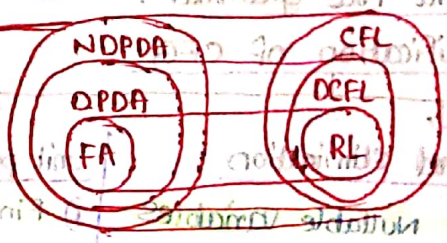
↓ NC | (q_2, ϵ, z_0) ✓

$(q_1, \epsilon, aaa z_0)$ × | $(q_2, \epsilon, aa z_0)$ ×

Note:

NDPDA is more powerful than DPDA

NDPDA ⊃ DPDA



Testing: CFL or not:

① $a^n b^n / n \geq 1$ ⇒ Not RL, Not CFL

② $a^n b^2 / n \geq 1$ ⇒ Not RL, Not CFL

③ $ww^R \mid w \in (a,b)^*$ ⇒ Not RL, Not DCFL but CFL

④ $ww \mid w \in (a,b)^*$ ⇒ Not RL, Not CFL.

⑤ $a^n b^n c^m / n > m$ ⇒ Not RL, Not CFL.

⑥ $a^n b^n c^n d^n / n \leq 10^{10}$ ⇒ RL, DCFL, CFL

⑦ $xcy / x,y \in (0,1)^*$ ⇒ RL, DCFL, CFL

(8) $xx^R \mid x \in (0+1)^* \& |x| = l$ ⇒ RL, DCFL, CFL

(9) $ww w^R \mid w \in (0+1)^*$ ⇒ Not RL, Not CFL, LBA

(10) $a^m b^n \mid m \neq n$ ⇒ Not RL but CFL, but DCFL

(11) $a^m b^n \mid m = 2n+1$ ⇒ Not RL, DCFL, CFL

(12) $a^i b^j \mid i \neq 2j+1$ ⇒ Not RL, DCFL, CFL

(13) $a^k \mid k$ is even ⇒ RL, DCFL, CFL

(14) $a^i b^j c^k \mid i > j > k$ ⇒ Not RL, Not CFL

(15) $a^i b^j c^k \mid j = i+k$ ⇒ Not RL, DCFL, CFL

(16) $a^i b^j c^k d^l \mid i = k$ or $j = l$ ⇒ Not RL, not DCFL, CFL

(17) $a^i b^j c^k d^l \mid i = k$ and $j = l$ ⇒ Not RL, Not CFL

(18) $w \mid w \in (a,b)^*, |w| \geq 100$ ⇒ RL, CFL

(19) $w \mid w \in (a,b,c)^*, n_a(w) = n_b(w) = n_c(w)$ ⇒ Not RL, not CFL

(20) $w \mid w \in (a,b)^*, n_a(w) \geq n_b(w) + 1$ ⇒ Not RL but CFL.

Content Free grammar:

Simplification of CFG

ε-prod Elimination:
1) Find Nullable variables
2) Replace in production
3) Eliminate the terminals which are Nullable

Unit prod Elimination:
1) Find unit prod
2) Replace all unit prod with its equivalent terminals

Useless symbols Elimination:
1) Find useful symbols
2) See if RHS is made of useful symbols
3) Eliminate the useless production
4) Check if the productions are reachable

*) If ε belongs to L(G) then All ε's can't be Eliminated.
ε doesn't belongs to L(G) then All ε's can be Eliminated.

CFG

Chomsky Normal form (CNF)

$A \to BC$
$A \to a$

$A, B, C \in V$
$a \in T$

Easy to Apply CYK Algo using CNF

☺ • length of Each prod is restricted
• Parse tree is always binary tree
• Num of steps reqd to derive a string of length $|w|$ is $2|w|-1$

Greibach Normal form (GNF)

$A \to a\alpha$
$\alpha \in V^*$

☺ Num of steps reqd to generate a string of length $|w|$ is $= |w|$

Useful to Convert CFG to PDA

CYK Algorithm : $O(n^3)$

Ex:- $S \to AB$

$A \to BB|a$ → CNF

$B \to AB|b$ ↓ Apply CYK.

"aabbb" (1 2 3 4 5)

↳ accepted.

1–5 contains 'S'

	5	4	3	2	1
1	S, B	A	S, B	ϕ	A
2	S, B	A	S, B	A	
3	B, S	A	B		
4	A	B			
5	B				

Sub strings → aab, abbb, ab, bbb.

CFG to PDA Conversion :

① push the start symbol onto the stack.

$\delta(q_0, \epsilon, z_0) = (q_1, A z_0)$

② push RHS as follows

$\delta(q_1, a, A) = (q_1, \alpha)$ ∵ $A \to a\alpha$

$\delta(q_1, b, A) = (q_1, \alpha)$ ∵ $A \to b\alpha$

③ Add final state with

$\delta(q_1, \epsilon, z_0) = (q_f, \epsilon)$

Ex:- $S \to aSb|ab$.

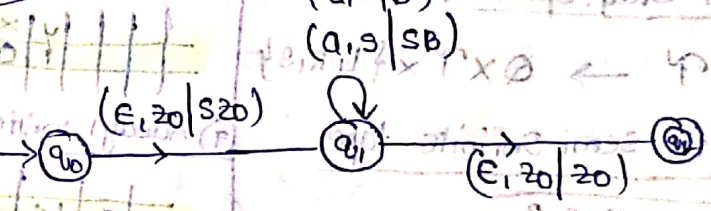
⇓

$S \to aSB$

$S \to aB$ → GNF

$B \to b$

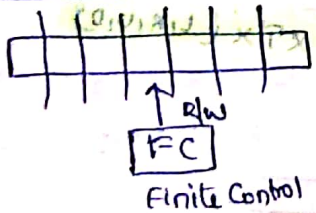
→



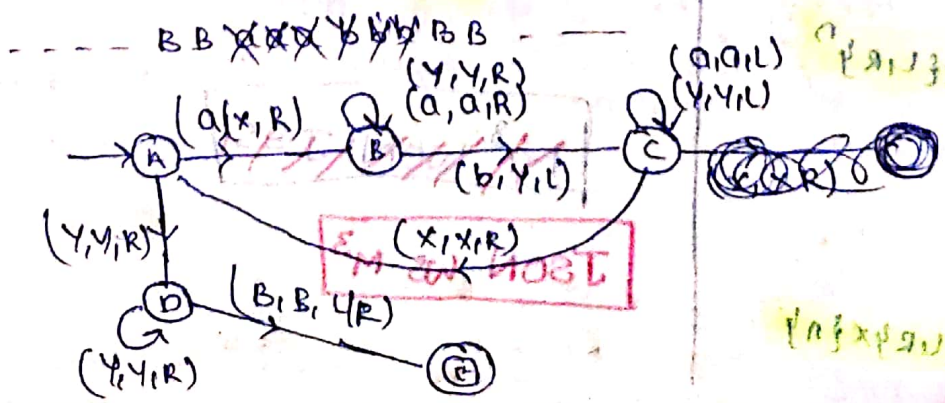
Turing Machine : (Transducer) (Acceptor)

↳ No final state

↳ Final state present



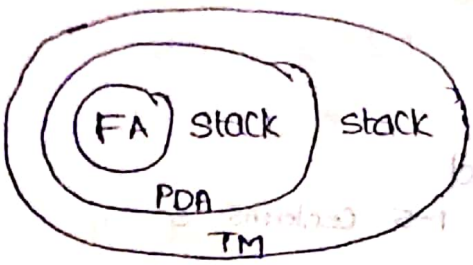
Ex:- $L = \{a^n b^n / n \geq 1\}$



aabb → 'E' state

aabbb → 'D' state

aaabb → 'B' state



TM → Copier
→ 1's comp
→ 2's comp
→ Comparator
→ Adder
→ subtractor
} → mathematically Complete

Turing Machine :

→ Tape is unbounded, so any num of left and right moves possible.
→ It is deterministic ie atmost one move for each configuration.
→ No special i/p or o/p file.

Turing Thesis :

Any Computation that can be done by mechanical means can also be done by TM.

TM is accepted as def of mechanical Computation / computer

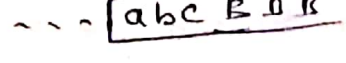
- Because
  - → Any thing that can be done on Computer can be done on TM
  - → No one yet suggested a problem ie solvable by Algo and unsolvable by TM
  - → Alternative models are also proposed for def of mech computation but none of them are as poweful as TM

Modifications to TM : (All modifications doesn't increase power of TM)

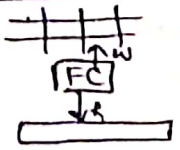
① TM with stay option :

$Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$

② TM with semi Infinite tape



③ offline TM : Read only o/p file



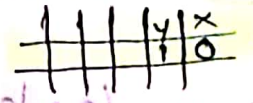
④ Multitape TM :

$\delta : Q \times \Gamma^n \rightarrow Q \times \Gamma^n \times \{L, R\}^n$

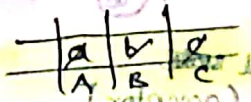
⑤ Jumping TM :

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\} \times \{n\}$

⑥ Non-erasing TM



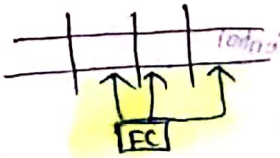
⑦ Always writing TM



⑧ Multidimensional TM :

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D\}$

⑨ Multihead TM :



M³ WON JTS

JSON WS M³

*) Automata with a queue ⇒ TM

*) TM with only 3 states

*) Multitape TM with stay option & almost 2 states.

*) Non deterministic TM

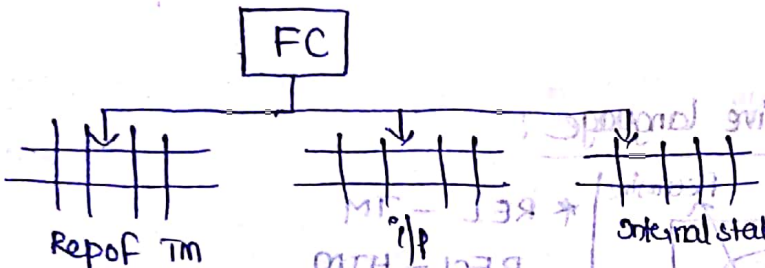
$Q \times T \longrightarrow 2^{Q \times T \times \{L,R,S\}}$, DET TM ≡ NDET TM

These modifications are as good as Standard TM.

*) A Non deterministic PDA with two independent stacks

$\delta : Q \times \Sigma \cup \{\in\} \times T \times T \longrightarrow 2^{Q \times T^* \times T^*}$

Universal TM :



Encoding

$Q = \{q_1, q_2 \cdots q_n\}$

$T = \{a_1, a_2 \cdots a_n\}$

$q_1 - 1$ $a_1 - 1$

$q_2 - 11$ $a_2 - 11$

$q_3 - 111$ $a_3 - 111$

'0' separator.

L - 1

R - 11

$\in$

$\delta : (q_1, a_1) \longrightarrow (q_2, a_2, R)$

0101011011011 0 → rep in Tape ①

∴ Every TM can be rep as a string of 0's and 1's. $\Sigma = \{0,1\}$ Σ^* contains TM

→ Not every string of 0's and 1's may be a Turing machine.

Linear Bounded Automata :

TM + Tape (Finite) → FA

TM + Tape (stack) → PDA

TM + Tape ([I/p restricted]) ⇒ LBA

Note :

FA < PDA < LBA < TM

(DFA ≅ NFA) (DPDA ≇ NDPDA) (DLBA ≅ NLBA) (DTM ≅ NTM)

undecidable

Languages accepted by LBA :

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

$L = \{a^{n!} \mid n \geq 0\}$

$L = \{a^n \mid n = m^2 \mid m \geq 1\}$

$L = \{a^n \mid n \text{ is prime}\}$

$L = \{a^n \mid n \text{ is not prime}\}$

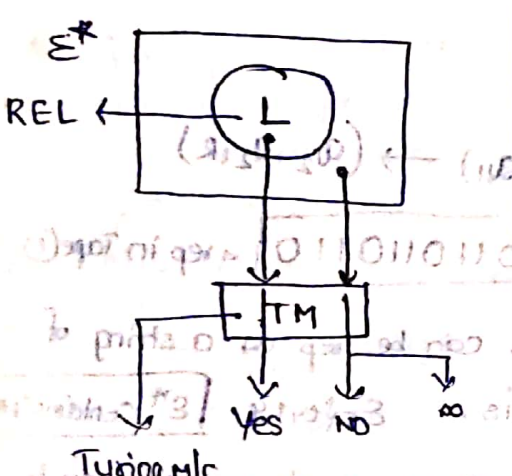
$L = \{ww \mid w \in (a|b)^+\}$

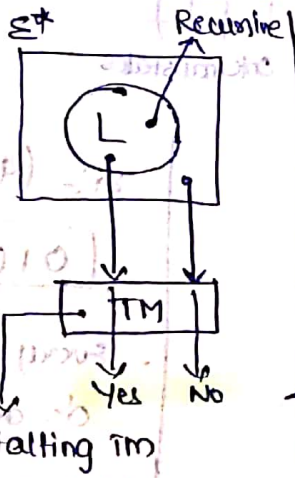
$L = \{w^n \mid n \geq 1, w \in (a,b)^+\}$

$L = \{wwwR \mid w \in (a|b)^+\}$

→ LBA

Recursively Enumerable and Recursive language :





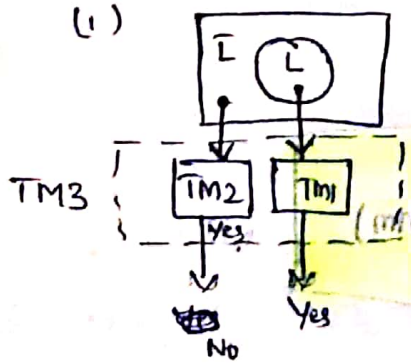
* REL – TM
RECL – HTM
CSL – LBA (HTM + ∃ p restricted)
CFL – PDA
RL – FA

*) REL – No membership Algo
RECL – Membership Algo exist
CSL –
CFL –
RL –

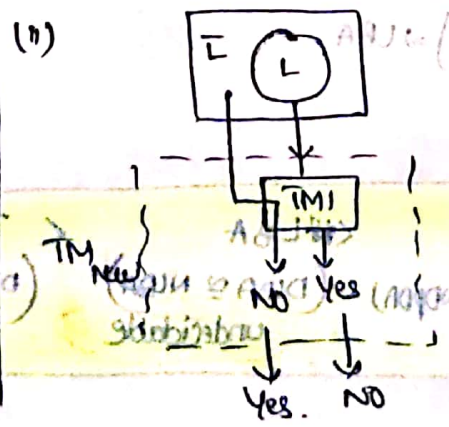
Theorem :

(i) If a language 'L' and its compliment $\bar{L}$ are both REL then both languages are recursive, (ii) If 'L' is recursive then $\bar{L}$ is also recursive and consequently both are REL

(i)



(ii)



Countable sets :

*) Sets

- Finite : $\{n \mid n < 10^{10^{10}}\}$
- Infinite
  - Countable
  - uncountable

Countable : A set 's' is said to be Countable, if the elements of the set can be put into one-to-one correspondance with the set of Natural numbers.

(or)

A set 's' is said to be countable if there exists a enumeration method using which all the elements of the set can be generated.

→ For any particular element, it takes finite num of steps to generate (used as index).

uncountable : Infinite + Not Countable

Ex:- $S = \{0, 2, 4 - - - - - -\}$ $2i$ (↓ $i+1$)

$N = \{1, 2, 3 - - - - - -\}$ $i+1$

Countable

R: 0 —— 1 (Infinite)

N: {1 2 3 - - - }

→ Def ①

uncountable -

Ex:- Set of all Even number

for i = 0 to ∞
  generate (2i)

Enumeration

0	2	4	6	8	- - -	$2i$	→ Def ②
↓	↓	↓	↓	↓		↓	
steps 1	2	3	4	5	- - -	$i+1$	

Ex:- p/q such that $p, q \in Z^+$

$S = \{1/1, 1/2 - - - 2/1, 2/2 - - -\}$

for i = 1 to ∞
  for j = 1 to ∞
    generate (i/j)

Enumeration

Index not possible because of no correspondance with Natural numbers.

Another Approach :

Sum : 2 → (1) ; 3 → $1/2$ (2), $2/1$ (3) ; 4 → $1/3$ (4), $2/2$ (5), $3/1$ (6) ; 5 → $1/4$ (7), $2/3$ (8), $3/2$ (9), $4/1$ (10) → Index

→ Countable

Ex:- $\Sigma = \{a, b\}$ $\Sigma^* = \{\epsilon, a, b, aa - - -\}$

(i) Dictionary order : a aa aaa - - - 'b's can't be generated

(ii) proper ordering :

Length	0	1	2	3	→ Index :
	ε	a, b	aa, ab, ba, bb		Countable
	↓	↓	↓		
	(1)	(2) (3)			

Σ^* is Countable

*) Every set of Countable set is either finite or Countable

$L \subseteq \Sigma^*$ ⇒ Every L is Countable.

Points to remember:

Countable sets:

① Σ^* is Countable

② Every lang is Countable

③ All TM's are Countable

TM ⟶ Countable ⟵ REL
HTM ⟶ Countable ⟵ RECL
LBA ⟶ Countable ⟵ CSL
PDA ⟶ Countable ⟵ CFL
FA ⟶ Countable ⟵ RL

④ S_1, S_2 are Countable then

$S_1 \cup S_2$ ⟶ Countable
$S_1 \times S_2$ ⟶ Countable.

⇓
Proved by Enumeration method

Uncountable sets:

① set of all lang are uncountable.
ie L^* is uncountable
2^{Σ^*} is uncountable.

②

REL	Not REL
Countable	uncountable

2^{Σ^*} ⇓ uncountable

⇓
Proved by diagonalization method

Diagonalization method:

Cantor stmt: If S is Countably infinite then 2^S is Uncountable.

Proof: Assume $S = \Sigma^*$

Assume 2^{Σ^*} is Countable

Σ^*	∈	a	b	aa	ab	ba	bb	aaa	–
1	0	1	1	0	1	1	0	1	–
2	1	0	1	1	0	1	0	1	–
3	1	0	0	1	1	0	1	1	–
4	1	1	1	1	1	0	0	0	–
5	1	1	1	1	1	0	1	0	–
.									
.									
.									

Diagonal : 0 0 0 1 1 – – – –

(Diagonal)' : 1 1 0 0 – – – – ∈ Σ^*

⇓
but not present in 2^{Σ^*}

∴ 2^{Σ^*} is uncountable

Computability & Decidability :

Algorithm → HTM → Countable

Computability :

$f(n) = n^2 + 1$

↓

There Exist Algo ⇒ HTM

D = Natural num

R = Natural num

Decidability : A stmt which may be T or F

Decidable

Ex:- If 'n' is a prime ?

D = Natural num

R = {T, F}

Undecidable

Ex:- A Given grammar 'G' is ambigues ?

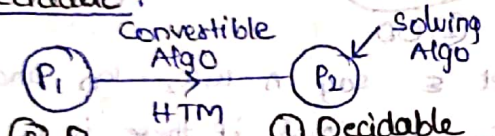
D = { set of all CFGs }

Note :

Any particular instance of a problem is always decidable.

To Check whether a problem is decidable or undecidable :

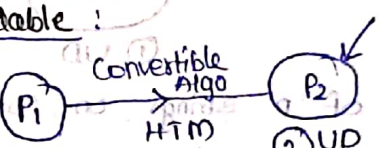
Decidable :



If P_2 is decidable then P_1 is decidable.

↓ Reducability

Undecidable :



If P_1 is proven to be undecidable then P_2 is undecidable.

Halting problem :

Given a TM 'M' and i/p 'w', does 'M' when started with 'w' as its i/p Eventually halts ?

Theorem : If the Halting problem is decidable

↓

Yes, No

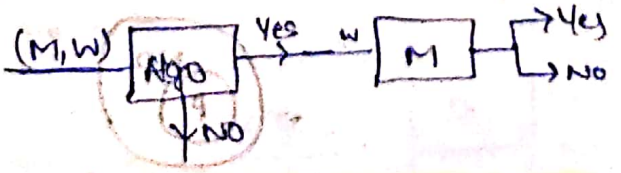
Halting TM ⇒ Rec L ≡ REL

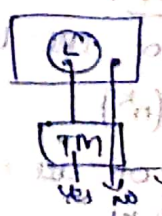
But REL ≠ RecL

So, our Assumption is wrong

∴ Halting problem is undecidable.

Proof : Assume Lang 'L' is REL (or) ↓ TM → M





Algo : Checks whether TM halts or not

↓

doesn't Exist bcoz REL has no membership Algo

∴ Halting prob is undecidable.

State entry problem is undecidable :

Given a TM a state $q \in Q$ and $w \in \Sigma^*$, decide whether or not the state q is ever entered when M is applied to W.

This is undecidable.

Blank Tape TM problem is undecidable :

Given a TM M, whether or not M halts if started with a blank tape

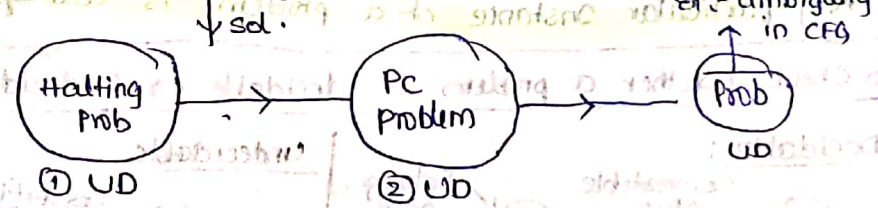
This is undecidable.

Post Correspondance problem is undecidable

Halting problem ⟶ easy to apply for REL, Rec lang

difficult to apply for CFL

↓ sol.



Given two sequences of 'n' strings on some alphabet Σ say $A = w_1 w_2 \cdots w_n$ and $B = v_1 v_2 \cdots v_n$, we say there exist a PC-solution for pair (A,B) if there is a non empty sequence of integers $i, j \cdots k$ such that

$$w_i w_j \cdots w_k = v_i v_j \cdots v_k$$

PC problem is to device an algorithm that will tell us for any (A,B) whether there exist a PC solution or not ?

∴ This is undecidable.

Ex:-

A:	w_1	w_2	w_3
	a	ab	bba

B:	v_1	v_2	v_3
	baa	aa	bb

PC solution ⟶ **undecidable**

$w_3 w_2 w_3 w_1 \rightarrow$ bbaabbbaa

$v_3 v_2 v_3 v_1 \rightarrow$ bbaabbbaa

PC sol → (3,2,3,1)

Ambiguity in CFG problem is undecidable.

Proof by PC problem

∴ This is undecidable.

Complexity classes :

P class : set of all lang that are accepted by some DTM in polynomial time ie $O(n^k)$

NP class : set of all lang that are accepted by some NDTM in polynomial time ie $O(n^k)$

→ NP Hard

→ NP Complete

Note : is **P = NP** ?

Not known, No proofs.

Note:

If NTM solves a problem in polynomial time then DTM will solve it Exponential time

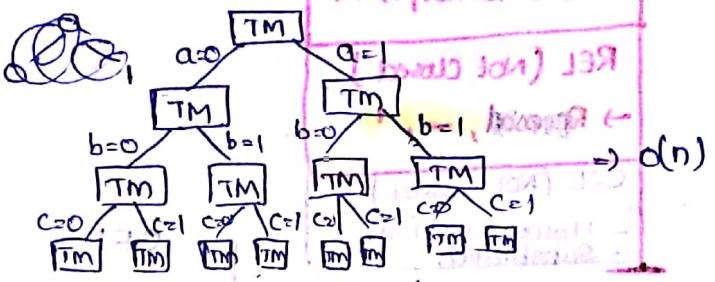
i.e NTM $\rightarrow$ DTM

$O(n^k)$ $O(2^n)$

Ex:- $a \vee b \vee c$ $\quad a,b,c \in \{0,1\}$

DTM : $2 \times 2 \times 2 \Rightarrow 2^n \Rightarrow$ only one TM $\Rightarrow O(2^n)$

NTM :



Refer Algorithm Notes

Updated table :-

1D 5UD $\rightarrow$ S1

2D 4UD $\rightarrow$ $C \xrightarrow{*} R$

3D 3UD $\rightarrow$ EF

5D 1D $\rightarrow$ M

D D UD $\rightarrow$ C^1

CFL X $\rightarrow$ I^2

lock

Decidability table : MECE^{8}S IFC^{1}I^{2}R

Problem	RL	DCFL	CFL	CSL	Rec Lang	REL
(1) Does $w \in L$? (Membership problem)	D	D	D	D	D	UD
(2) Is $L = \phi$? (Emptiness problem)	D	D	D	UD	UD	UD
(3) Is $L = \Sigma^*$? (Completeness problem)	D	UD	UD	UD	UD	UD
(4) Is $L_1 = L_2$? (Equality problem)	D	UD	UD	UD	UD	UD
(5) Is $L_1 \subseteq L_2$? (subset problem)	D	UD	UD	UD	UD	UD
(6) Is $L_1 \cap L_2 = \phi$?	D	UD	UD	UD	UD	UD
(7) Is L is finite or not? (Finiteness problem)	D	D	D	UD	UD	UD
(8) Is Compliment of a lang 'L' is sametype or not?	D	D	UD	D	D	UD
(9) Is intersection of two lang are of same type?	D	UD	UD	UD	UD	UD
(10) Is L regular language?	D	D	UD	UD	UD	UD

L_1, L_2 are of same type

*) All prob related to REL are undecidable and all problems related to RL are decidable

*) 1D 5UD → CES1|²
*) 3D 3UD → E F
*) 5D 1UD → M
*) 2D 4UD → R
*) 2(DD UD) → C'

Regular (Not Closed)
→ (∞) Union (All)

CFL (Not Closed)
→ ∩, −, ¹ (F∞) (F∞)

Recursive (Not Closed)
→ substitution
subset (F∞)
Homomorphism

REL (Not Closed)
→ [redacted], −, ¹

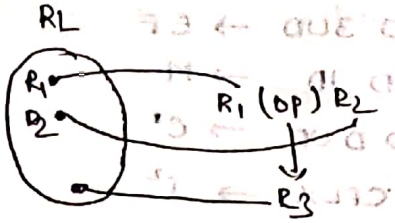
CSL (Not Closed)
- Homomorphism
- Substitution.

DCFL (Closed)
- Compliment
- Inverse homomorphism
- finite subset
- Intersection with RL

Properties of RL (12) and CFL (5)

(1)

Closed under operation



① Regular languages are closed under union
② Regular languages are closed under Concatenation
③ Regular languages are closed under Kleen closure
④ Regular languages are closed under complimentation
⑤ Regular languages are closed under Intersection, difference.
⑥ Regular languages are closed under reversal.
⑦ Regular languages are closed under Homomorphism
⑧ Regular languages are closed under Inverse Homomorphism
⑨ Regular languages are closed under right quotient
⑩ Regular languages are closed under INIT operation
⑪ Regular languages are closed under substitution
⑫ Regular languages are not closed under infinite union

CFL
union
Concatenation
Kleen Closure
not set difference
not Complimentation
not Intersection.

Rec: union
Concatenation
Kleen Closure
Intersection
Complement
set difference

REL: union, Intersection
Concatenation
Kleen Closure.
not Complement
not set difference.

Homomorphism:

The function $h: \Sigma \to T^*$ is called homomorphism

Ex:- $\Sigma = \{a, b\}$ $T = \{0, 1, 2\}$

$h(a) = 01$

$h(b) = 112$

Now $h(abd) = 01112\,01$

Ex:- $L = a^*b$

$= (01)^* 112$

↳ regular.

Inverse homomorphism :

The inverse of homomorphic of language L is.

$$h^{-1}(L) = \{ x \mid h(x) \in L \}$$

For a string w

$$h^{-1}(w) = \{ x \mid h(x) \in L \}$$

Ex:- $\Sigma = \{0,1,2\}$ $\Delta = \{a,b\}$

$h(0) = a$ $h(1) = ab$ $h(2) = ba$

$L = a(ba)^*$ = { a, aba, ababa, abababa - - - }

$h^{-1}(L)$ = { 0, 10, 02, 022, 110, 102, 0222, 1110, 1022, 1102 - - - }

$= 02^* + 1^*0 \Rightarrow 1^*02^*$ → Regular.

Ex:- $\Sigma = \{0,1\}$ $\Delta = \{a,b\}$

$h(0) = aa$ $h(1) = aba$

$L = (ab+ba)^* a$ - { a, aba, baa, ababa, abbaa, baaba, babaa - - - }

$h^{-1}\{L\}$ = (a ✗, aba 1, baa ✗, ababa ✗, abbaa ✗, baaba ✗, babaa ✗)

$h^{-1}\{L\} = \{1\}$

$h(h^{-1}(L)) = h(1) \Rightarrow aba \in L$ / $\notin L$

Note : $h(h^{-1}(L)) \neq L$

$h(h^{-1}(L)) \subseteq L$

Right Quoitent :

Let L_1 and L_2 be languages on the same alphabet then the right quotient of L_1 with L_2 is defined as

$$L_1/L_2 \{ x \mid xy \in L_1 \text{ for some } y \in L_2 \}$$

Ex:- $L_1 = 10^*1$ $L_2 = 0^*1$

$L_1/L_2 \Rightarrow$ $L_1 = \{11, 101, 1001, 10001 - - - \}$

$L_2 = \{1, 01, 001 - - - \}$

$L_1/L_2 \Rightarrow \{1, 10, 100, 1000 - - - \} \Rightarrow 10^*$

INIT operation :

set of all prefines

Ex:- $L = \{a, ab, aabb\}$

$INIT(L) = \{\epsilon, a, \epsilon, a, ab, \epsilon, a, aa, aab, aabb\}$

$= \{\epsilon, a, aa, ab, aab, aabb\}$

Points to remember :

① L is CFL , $\bar{L}$ is CFL (x)

② L is CFL , $\bar{L}$ is Rec (✓)

③ L is CFL , $\bar{L}$ is REL (✓)

x — The End — x

En :

*) $|w| = n$ | Num of states = $n+2$

*) $|w| \geq n$ | Atleast $n+1$ states

*) $|w| \leq n$ | Atmost $n+2$ states.

*) If $|w| \bmod n = 0|1|\ldots n-1$ then num of states = n

*) If $w \in \{a|b\}^*$ and $n(a)(w) \cong 0 \bmod n$ & $n(b)(w) \cong 0 \bmod m$ then num of states = $m*n$

*) Divisible by 4 $\Sigma = \{a|b\}$

	a	b
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_1
q_3	q_2	q_3

* Fortran - } ∈ CSL bcoz of Typechecking variable declaration

C - } ∈ CSL

*) TM doesn't accept '∈'

— $L = \{\}$ is Recursive

— $L = \{a^m b^n / m \neq n\}$

$L' = \{a^m b^n / m = n\} \cup \{(a+b)^* ba (a+b)^*\}$

$(m=n)$ lang — b followed by a.

Pumping lemma : (negative test)

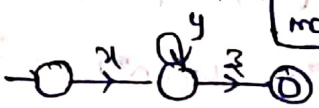
If L be an Infinite lang and $w \in L$ such that $|w| \geq n$ for some +ve Int n, then w can be decomposed into three string $w = xyz$ such that

(i) $y \neq e$

(ii) $|xy| \leq n$

(iii) $\forall k \geq 0$ the string $xy^k z$ is also in L

Finding y is main focus



Note :

1) Every RL satisfies PL property

2) If a lang L doesn't satisfy PL Property then L is non regular

3) If a lang L satisfies PL property then L need not be regular

4) PL is used to prove some of the languages are not regular

Ex :- $L = \{a^n b^n / n \geq 1\}$

$a\underset{y}{aa}aa\,bbbb$ | $aaa\underset{y}{ab}bbb$

$a\underset{y}{aa}\underset{y}{aa}a\,bbbb \notin L$ | $aaa\underset{y}{ab}\underset{y}{ab}bbb \notin L$

∴ Not regular

Imp Ex :- $L = \{a^n / n \geq 1\} \cup \{a^n b^n / n \geq 1\}$

$L = \{a, aa, aaa, aaaa, \ldots ab, aabb \ldots\}$

$a\underset{y}{a}a$

$a\underset{y}{aa}\underset{y}{a}a \in L$ but not regular