Theory of Computation
Chapter 3:Recursive enumerable Language

GATE Computer Science Lectures

By

Monalisa Pradhan

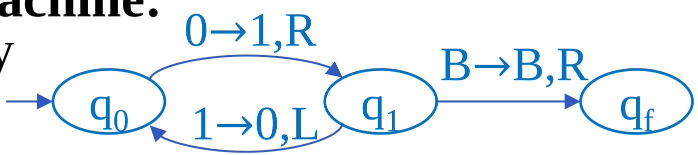
- **Section 6: Theory of Computation(≅ 10mark)**
  Regular expressions and finite automata. Context-free grammars and push-down automata. Regular and context free languages, pumping lemma. Turing machines and undecidability.

- Chapter 1:Regular Language [RL,FA,RE ,Pumping lemma]
- Chapter 2: Context free Language [Grammar(RG,CFG),CFL,PDA, Pumping lemma]
- Chapter 3: Recursive enumerable Language [CSL, LBA ,RS,RES,TM]
- Chapter 4: Undecidability

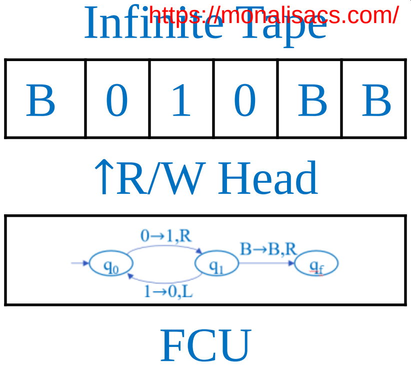
Turing Machine

- The Mathematical representation of Recursive Enumerable language is called TM.
- FA+2 stack=FA + Tape with read & write capability is called Turing Machine.
- TM has 7 tuple $M=(Q,\Sigma,\delta,\Gamma,B,q_0,F)$
- Q:Set of States.
- Σ:Set of input alphabet.
- **δ**:Transition function where $\delta: Q\times \Gamma \rightarrow Q\times \Gamma\times\{L,R\}$
- Γ:Set of Tape Symbol.
- B:Blank symbol.
- q_0:Initial state.
- F:Set of Final State.
- **Instantaneous Description(ID)**
- ID describe next move of TM.The next move of TM depends on two entry.
- 1.Current state
- 2.Current input symbol.
- **Representation of Turing Machine:**
- It can be represented in 2 way
- 1-Transition Diagram
- 2-Transition Table
- The State where transition is not defined is called as halt.

	0	**1**	**B**
q_0	q_1,1,R	H	H
q_1	H	q_0,0,L	q_f,B,R
q_f	H	H	H



- **Block diagram of Turing Machine:**
- TM consist of 3 component
- 1.Infinite Tape 2.R/W Header 3.Finite control Unit



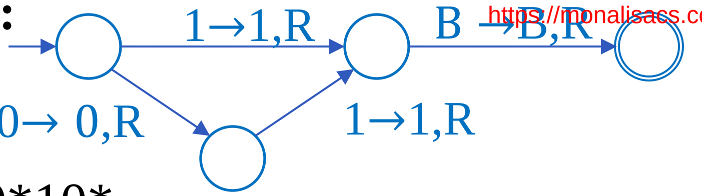
- **Infinite Tape:**
- The tape is collection of cell.
- Every cell can hold only one input symbol.
- The empty cells are filling with blank symbol B.
- The tape divided into two part.
- Blank region: Contain infinite number of blank symbol B.
- Non Blank region: contain finite number of input string.
- The tape can be one way infinite, Two way infinite.
- **R/W Header:**
- The header can read symbol from tape or can write or modify symbol over tape.
- After accessing the input symbol from the current cell the header can move to exactly one cell toward left or right.
- The header movement is bidirectional.
- **FCU:**
- FCU takes care of implementation of mathematical logic.
- FCU implement all the movement of TM defined by ID.

- TM is the abstract model of real computing model.
- The computing power of TM=computing power of RAM or Computer.
- Every mathematical function which is computable can be implement by TM.
- TM can be designed as:
- 1.language Accepter
- 2.Language Generator or enumerator
- 3.Input output device.
- TM can be constructed in two mode:1.DTM, 2.NTM.
- E(DTM)=E(NTM)
- DTM is more efficient than NTM.
- **TM as Language Accepter:**
- TM accept every Language accepted by FA & PDA also accept some language which are not accepted by PDA.
- TM is more powerful than FA & PDA.
- E(TM)=4(RL,CFL,CSL,REL)
- The language accepted by TM is called as REL.

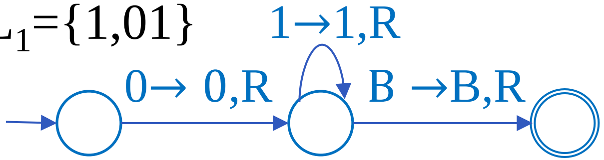
- **Acceptance by TM:**
- After taking input string there are 3 possibility of TM
- 1.May go to final halt.
- 2.May go to non final halt.
- 3.May go to loop.
- After reading input string if TM goes to final halt then input string is accepted by TM.If TM reaches nonfinal halt then input string is rejected by TM.
- On input string if TM goes to infinite loop then acceptance is undecidable.
- The halting problem of TM is undecidable.
- If TM accept the RL then header movement is always right .
- If TM constructed for accepting non RL then header movement is both left & right.
- Hence halt not garneted may go to loop.
- The TM goes to loop iff the header movement is both left & right.
- The transition function can't be defined for ∈.
- Since it can't be placed on tape.
- If q_f is final halt then no transition allowed at q_f.

Construct Turing Machine for Regular Language:

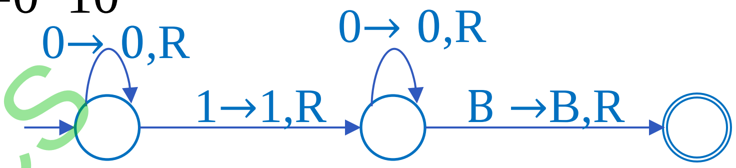
- $\Sigma=\{0,1\}, L_1=\{1,01\}$



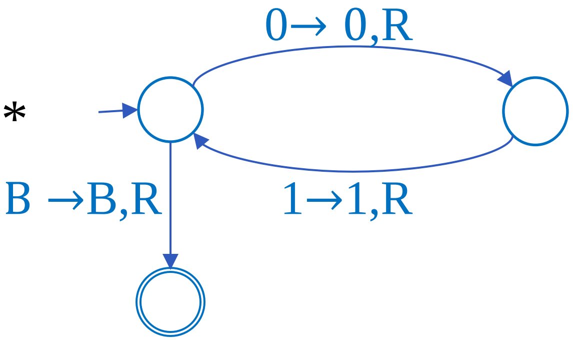
- $L_2=01^*$



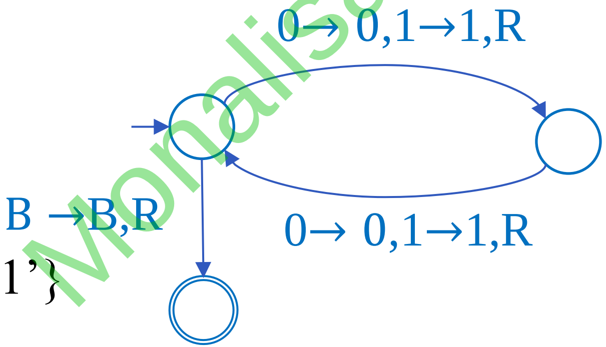
- $L_3=0^*10^*$



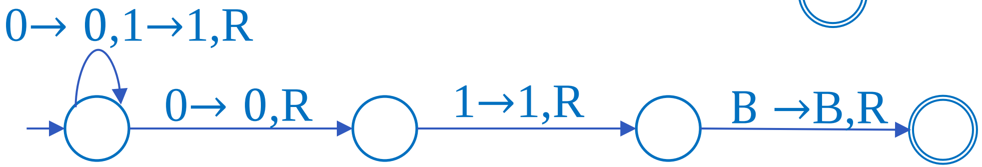
- $L_4=(01)^*$



- $L_5=\{|w|=\text{even} \mid w\in(0+1)^*\}$



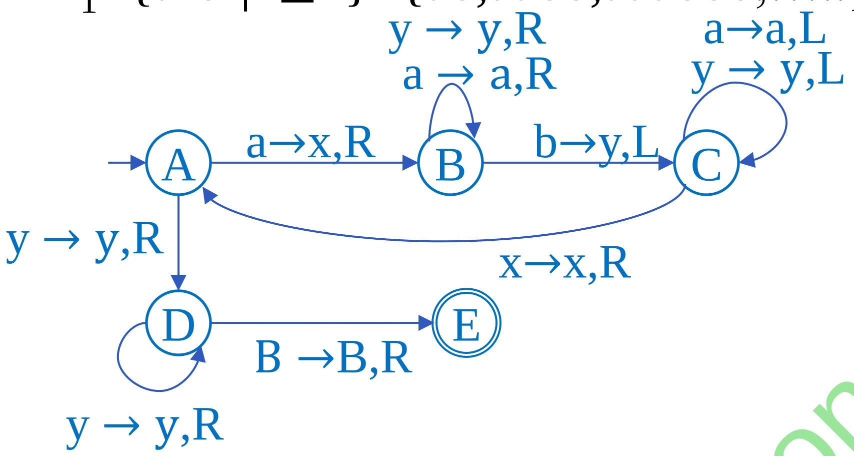
- L_6={Every string ends with '01'}



- TM require minimum 2 state.

Construct Turing Machine for Context Free Language:

- $\Sigma=\{a,b\}$
- $L_1=\{a^nb^n|n\geq1\}=\{ab,aabb,aaabbb,.....\}$

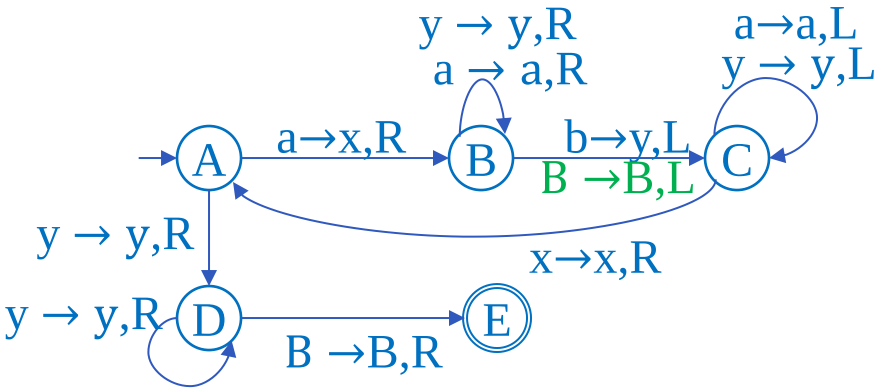


	a	b	x	y	B
A	(B,x,R)			(D,y,R)	
B	(B,a,R)	(C,y,L)		(B,y,R)	
C	(C,a,L)		(A,x,R)	(C,y,L)	
D				(D,y,R)	(E,B,R)
E					

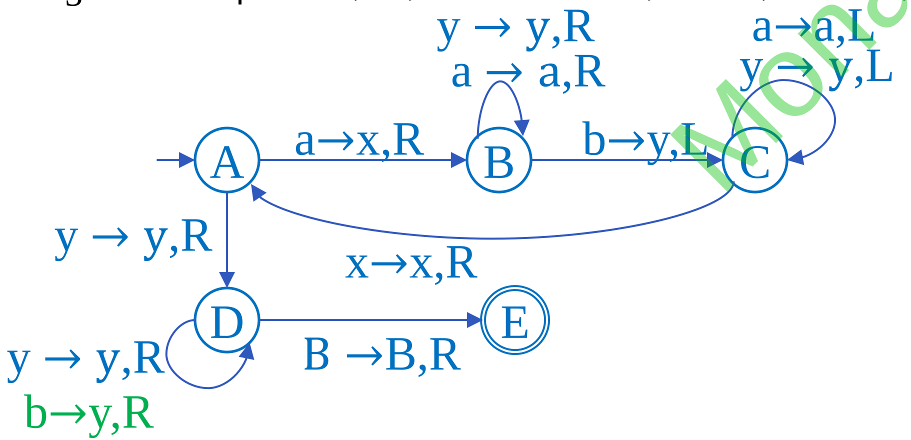
- $Q=\{A,B,C,D,E\}$
- $\Sigma=\{a,b\}$
- $\Gamma=\{a,b,x,y,B\}$
- B
- $q_0=A$
- F=E

- $\delta(A,a)=(B,x,R)$
- $\delta(A,y)=(D,y,R)$
- $\delta(B,a)=(B,a,R)$
- $\delta(B,y)=(B,y,R)$
- $\delta(B,b)=(C,y,L)$
- $\delta(C,a)=(C,a,L)$
- $\delta(C,y)=(C,y,L)$
- $\delta(C,x)=(A,x,R)$
- $\delta(D,y)=(D,y,R)$
- $\delta(D,B)=(E,B,R)$

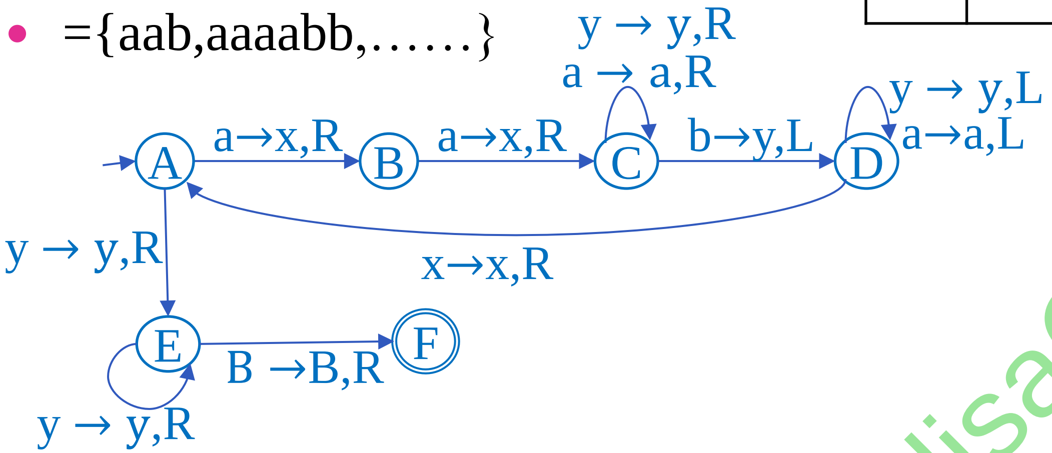
- $L_2=\{a^m b^n | m \geq n, m,n \geq 1\}$
- $=\{ab, aabb, aab, aaaabbb, \ldots\ldots\}$



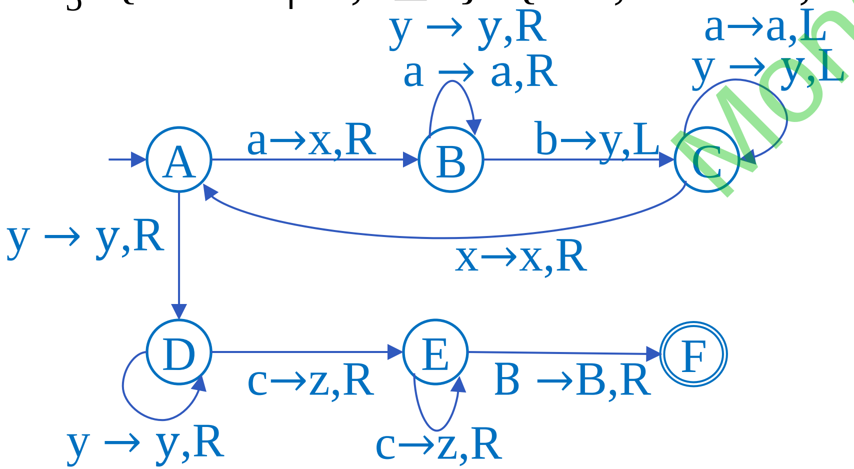
- $L_3=\{a^m b^n | m \leq n, m,n \geq 1\}=\{ab, aabb, aabbb, aaabbbbb, \ldots\ldots\}$



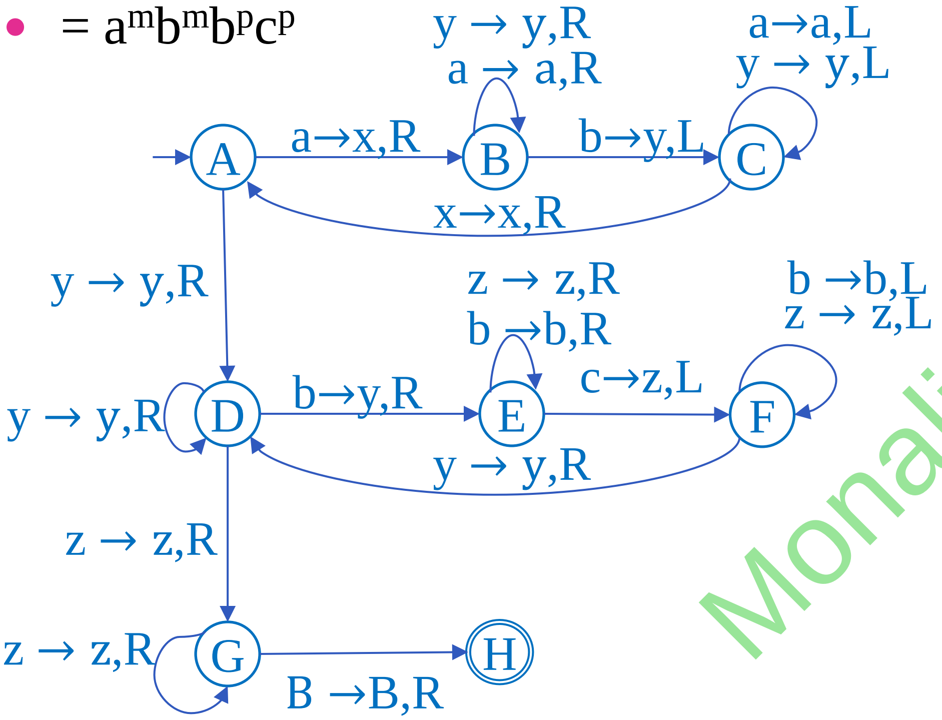
- $L_4=\{a^m b^n | m=2n, m,n \geq 1\} = a^{2n}b^n$
- $=\{aab, aaaabb, \ldots\ldots\}$



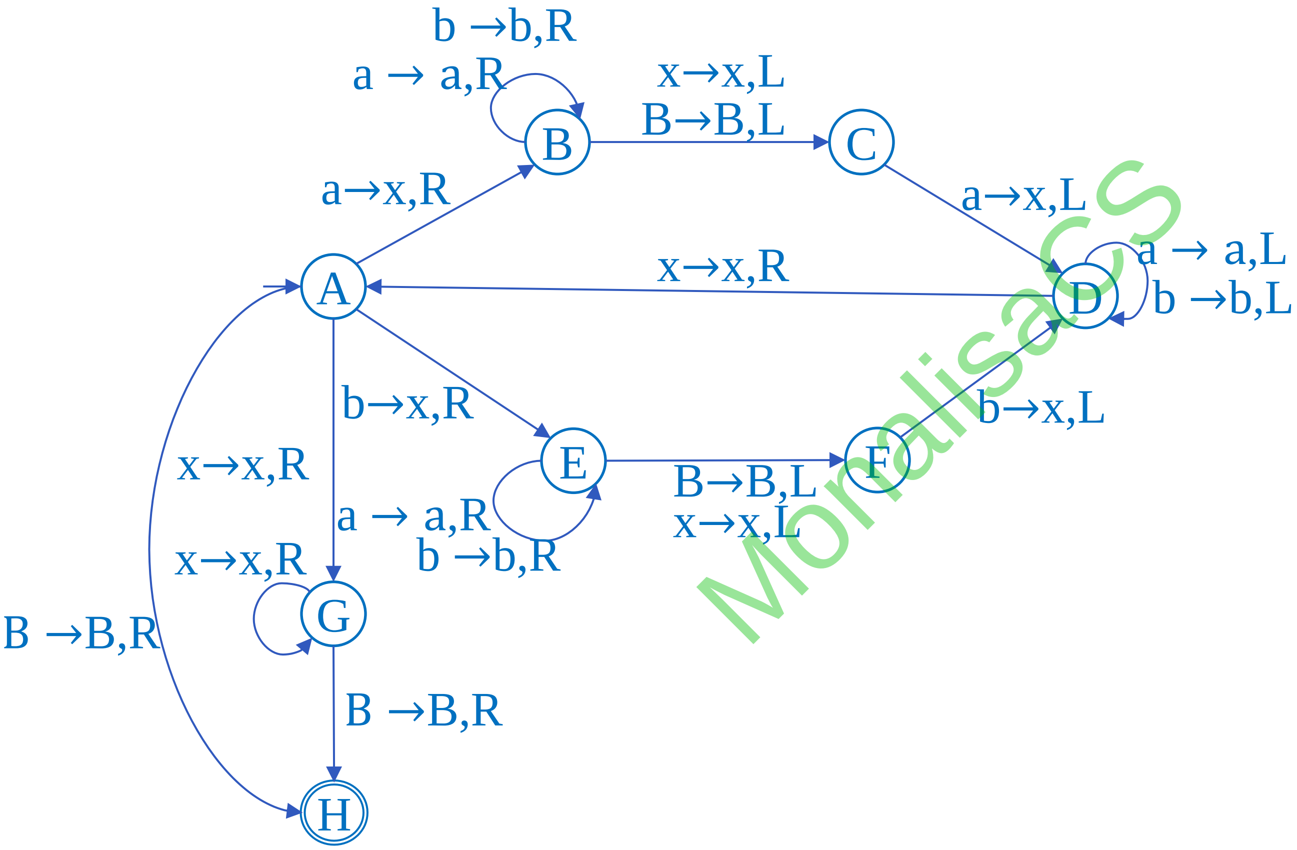
- $L_5=\{a^m b^m c^n | m,n \geq 1\}=\{abc, aabbccc, aaabbbcc, \ldots\ldots\}$



- $L_6 = \{a^m b^n c^p \mid n = m+p, m,n,p \geq 1\}$
- $= \{abbc, aabbbbcc, \ldots\ldots\}$
- $= a^m b^m b^p c^p$



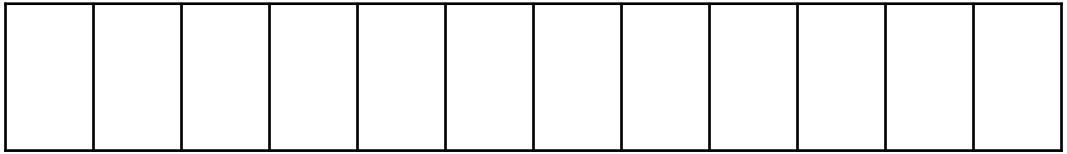
- $L_7=\{ww^R \mid w \in (a+b)^*\}$
- $=\{\in, aa, bb, abba, baaaab, ababbaba, \dots\}$

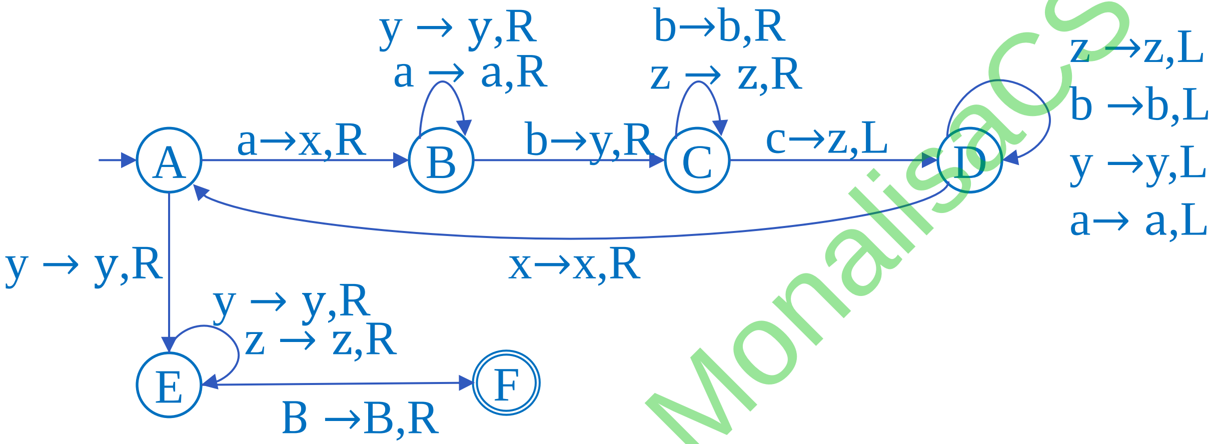


Construct Turing Machine for Recursive Enumerable Language:

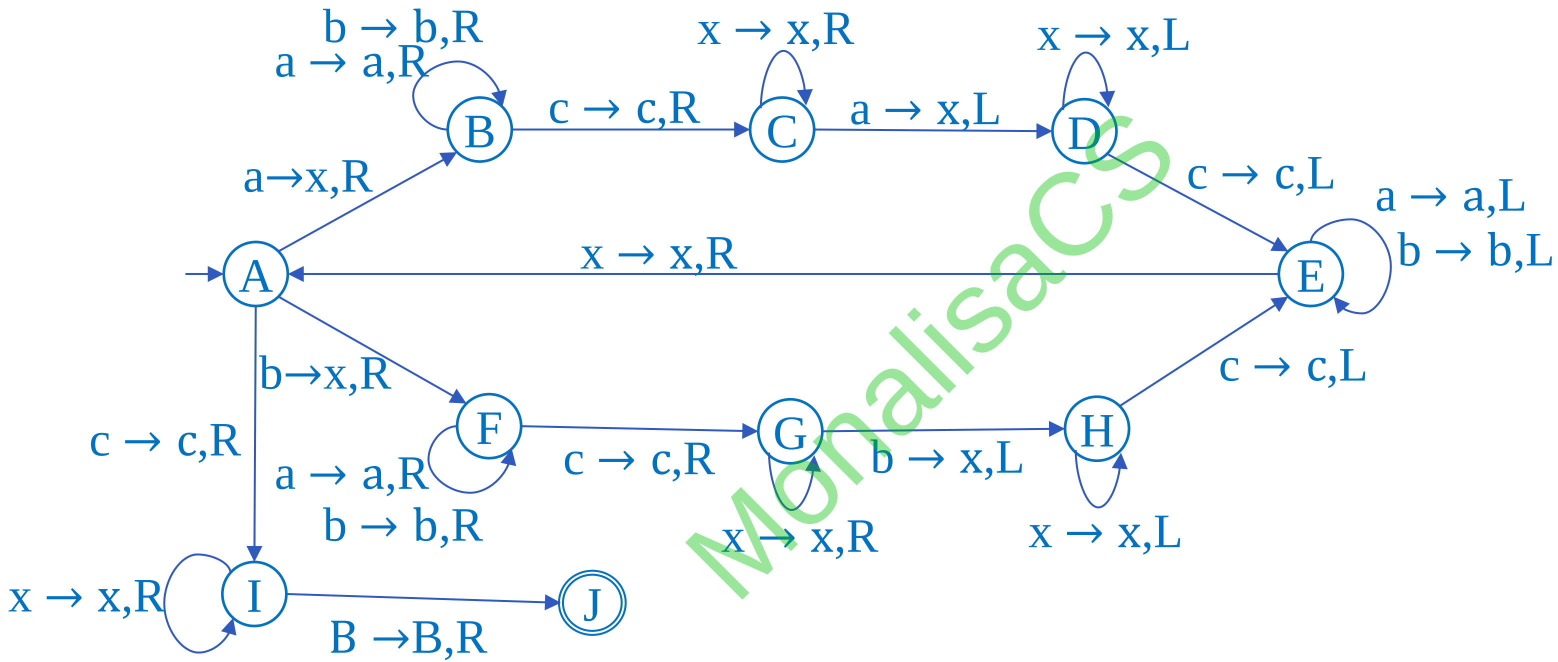
https://monalisacs.com/

- $L_1 = \{a^n b^n c^n \mid n \geq 1\}$
- $= \{abc, aabbcc, aaabbbccc, \ldots\}$





- $L_2=\{wcw \mid w \in (a+b)^*\}$
- $=\{c, aca, bcb, abcab, baacbaa\ldots\}$

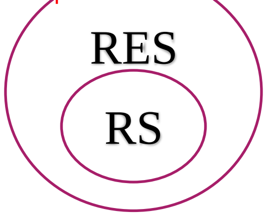


- **DTM** :The TM is said to be deterministic if every $\delta(q,x)$ has at most one outcome for all $x \in \Gamma$.
- **NTM**:The TM is said to be Non deterministic if every $\delta(q,x)$ has more than one outcome for all $x \in \Gamma$.

Recursive Set Vs Recursive Enumerable Set

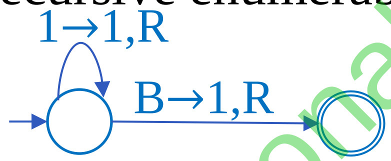
- The language accepted by TM is called as TM recognizable language.
- It is of two type
- 1.Recursive Set/Language
- 2.Recursive enumerable set/Language
- 1.The language L accepted by TM is said to be **Recursive** if for every x∈L TM goes to final halt & every x∉L TM goes to non final halt.
- The language accepted by TM for which membership property is <u>decidable</u> is called Recursive Set also called <u>Turing Decidable</u>.
- 2.The Language L accepted TM is said to be **Recursive Enumerable** if for every x∈L TM goes to final halt & every x∉L TM goes to non final halt or <u>loop</u>.
- The language accepted by TM for which membership property is <u>undecidable</u> is called Recursive Enumerable Set also called <u>Turing recognizable</u>.

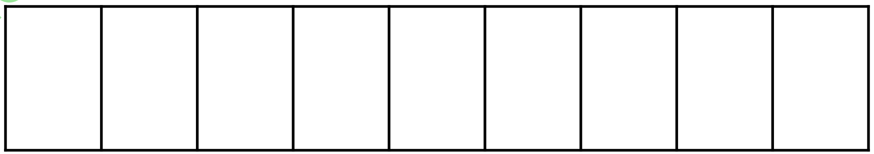
Language	**RS**	**RES**
The TM halt on every x∈ Σ^*	Yes	May or may not
The TM responds for every x∈ Σ^*	Yes	May or may not
If it can implemented by	an Algorithm.	Procedure



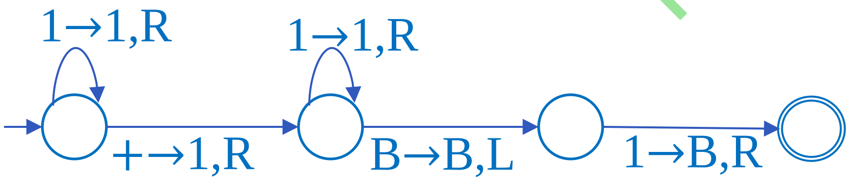
- Every RS is RES but every RES need not be RS
- RS=Enumeration Properties +Membership is decidable
- RES=Enumeration Properties +Membership is Undecidable
- **Closer Properties of Recursive Set**
- Recursive Set is closed under following operator
- Union ,Intersection ,Concatenation , Complement ,Kleene closer , Reversal ,Difference ,Inverse homomorphism ,Intersection with Regular set.
- Recursive Set is not closed under Homomorphism ,Quotient with Regular
- **Closer Properties of Recursive Enumerable Set**
- Recursive Enumerable Set is closed under following operator
- Union ,Intersection , Concatenation ,Kleene closer , Homomorphism ,Inverse Homomorphism ,Intersection with Regular Set, Quotient with Regular set
- Recursive Enumerable Set is not closed under Complement, Difference
- L=RES⇒ $\overline{L}$ =need not be RES
- Both L & $\overline{L}$ can be Recursive.
- Both L & $\overline{L}$ can not be Recursive Enumerable .

- **Turing Machine as Transducer:**
- TM can be design to implement basic mathematical operation +,-,×,÷
- The function which is implemented by TM is called as computable function.
- The function which can’t be implemented by TM is called uncomputable function.
- Ex: $\sqrt{x}$,log x
- Computable function are divided into two types
- 1.Total computable 2.Partial computable.
- Total computable =Recursive Set=Algorithm
- Partial computable=Recursive enumerable set=Procedure=Partial recursion
- Ex 1:f(x)=x+1,Σ={1}
- 2=11,f(2)=111

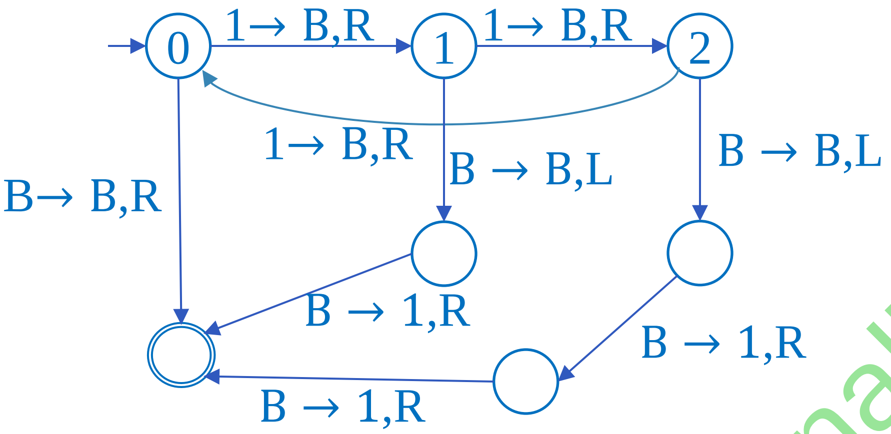




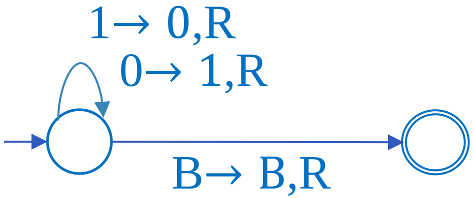
- Ex 2:f=x+y =111+11=11111



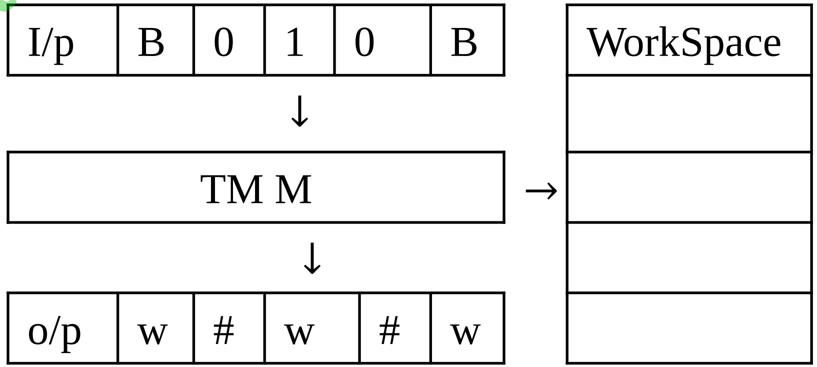
- Ex 3:f=r mod 3
- 11111 mod 3=11



- Ex 4:1s complement, $\Sigma=\{0,1\}$



- **Turing Machine as Enumerator:**
- Enumerator=System + Printer
- Enumeration is the process of generation and listing in a finite amount of time.
- Enumeration process can be implemented by TM. So it is called as Enumerator.
- The TM that implement enumerator consist of 3 tape
- Tape 1=Input , Tape 2=Workspace , Tape 3=Output
- The TM take input from input tape & generate the string that will return on output tape.
- The string written on output tape can't be modified . They are separated by # symbol.

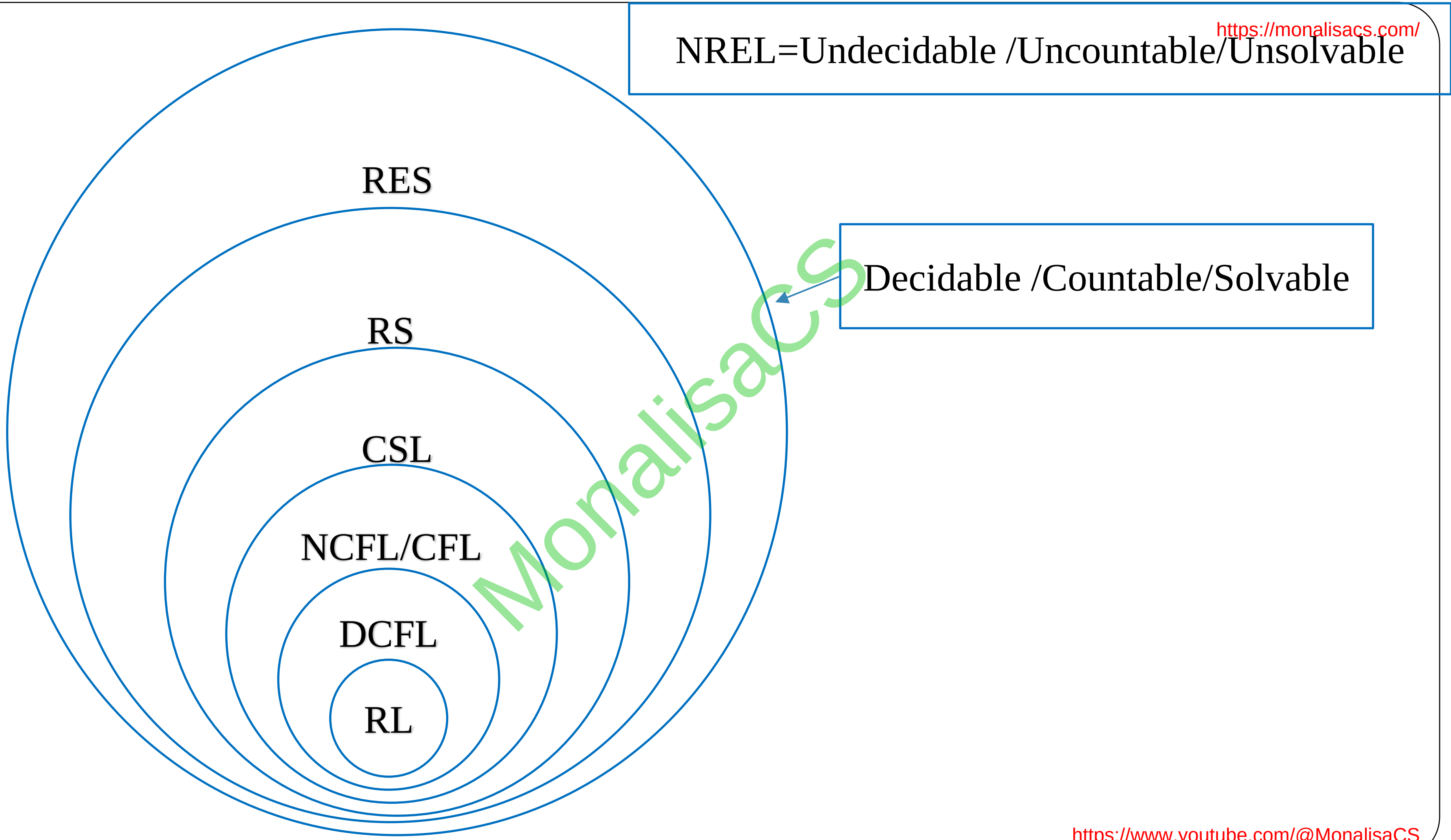


- **Universal Turing Machine:**
- Every TM accept only one language, single task oriented.
- RAM is multitask oriented system.
- RAM can be simulated by Universal Turing Machine(UTM).
- UTM is a multi tasking system.

- **Linear Bounded Automata:**
- The mathematical representation of CSL is called as LBA.
- LBA is NTM with following 2 condition
- 1.Include 2 special Symbol($ Left end Marker ,⊄Right end marker)
- 2.R/W header can't move left from $ & right from ⊄
- **Context Sensitive Language:**
- The language which is accepted by LBA or generated by CSG is called CSL.
- CSL is free from empty string (∈).
- **Context Sensitive Grammar**
- $L_1=\{a^n b^n c^n | n \geq 1\}$
- S →aSAc/abc
- cA →Ac
- bA →bb
- Let w=aabbcc
- S → aSAc
- → aabcAc
- → aabAcc
- → aabbcc

- **Example of CSL**

- $L_1=\{a^n b^n c^n \mid n \geq 1\}$
- $L_2=\{ww \mid w \in (a+b)^+\}$
- $L_3=\{a^m b^n c^m d^n \mid m,n \geq 1\}$
- $L_4=\{www \mid w \in (a+b)^+\}$
- $L_5=\{a^{n^2} \mid n \geq 1\}$
- $L_6=\{a^{2^n} \mid n \geq 1\}$
- $L_7=\{a^{n!} \mid n \geq 1\}$
- $L_8=\{a^p \mid p \text{ is a +ve prime number}\}$
- $L_9=\{a^p \mid p \text{ is not a prime number}\}$
- $L_{10}=\{a^i b^j c^k \mid j=i \times k, i,j,k \geq 1\}$
- **Closer Properties of CSL:**
- CSL is closed under following operation
- Union, Intersection, Complement ,Concatenation ,Kleene Closer , Reversal ,Inverse homomorphism ,Substitution ,Intersection with Regular Set.
- CSL is not closed under Homomorphism ,Quotient with Regular Set.
- Every CSL is recursive Set but Recursive need not be CSL


NREL=Undecidable /Uncountable/Unsolvable
RES
RS
Decidable /Countable/Solvable
CSL
NCFL/CFL
DCFL
RL