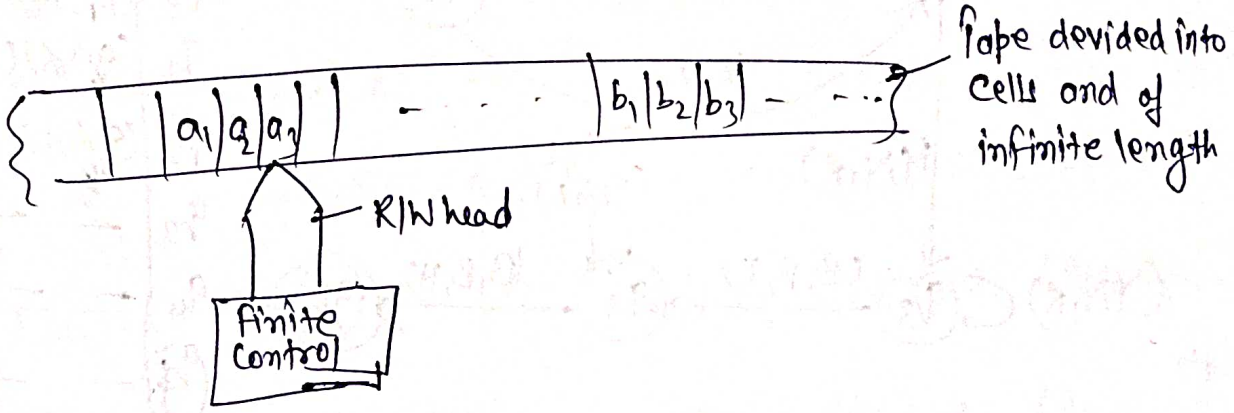
UNIT-05 Turing Machine (Lecture-01)

The turing machine can be through of as finite control connected to a R/W head. It has on tape which is devided into no. of cells. it's Represented by Fig.1



Each cell can store only one symbol the I/P to and the O/P from the finite state automaton are effected by the R/W head which can examine one cell at a time.

In one move, the machine examine the present symbol under the R/W head on the tape and the present state of an automaton to determine

(i) A new symbol to be written on the tape in the cell under the R/W head.

(ii) A motion of R/W head along the tape, L, R

(iii) The next state of the automaton

(iv) Whether to halt or not.

The above Model define as follows

TM-7 tuples $(Q, \Sigma, \Gamma, \delta, q_0, B, q_f)$

$Q \rightarrow$ Non empty set of states

$\Sigma \rightarrow$ Non empty set of I/P alphabate & is subset of Γ & $B \notin \Sigma$

$\delta \rightarrow$ Transition function $Q \times \Gamma = Q \times \Gamma \times (R, L, N)$

$\Gamma \rightarrow$ Tape alphabate Non-empty set

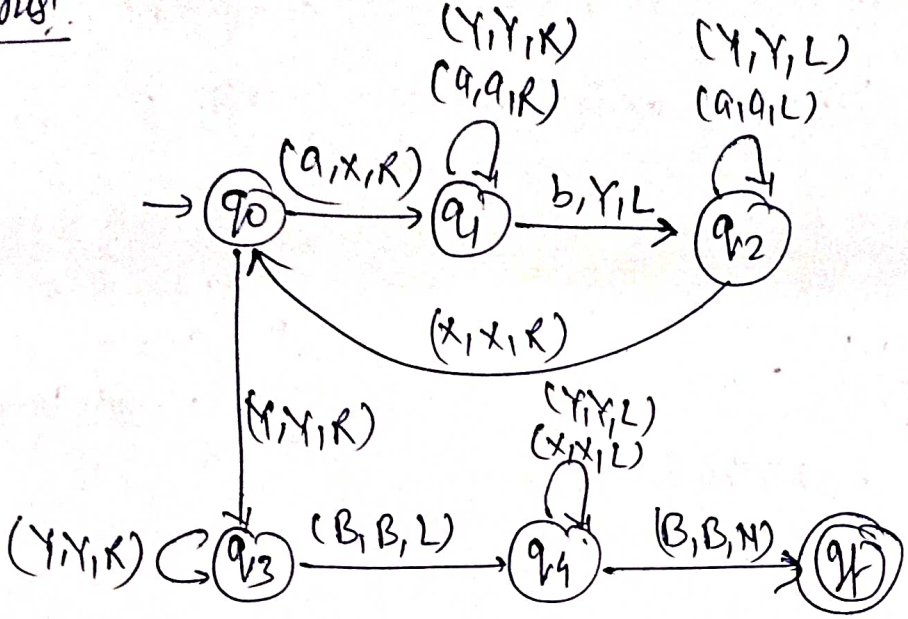
$q_0 \in Q$ initial state

$B \in \Gamma$ tape symbol

$q_f \subseteq Q$ final state

Q. Design a TM for language $L = \{a^n b^n \mid n \geq 1\}$

Ans:

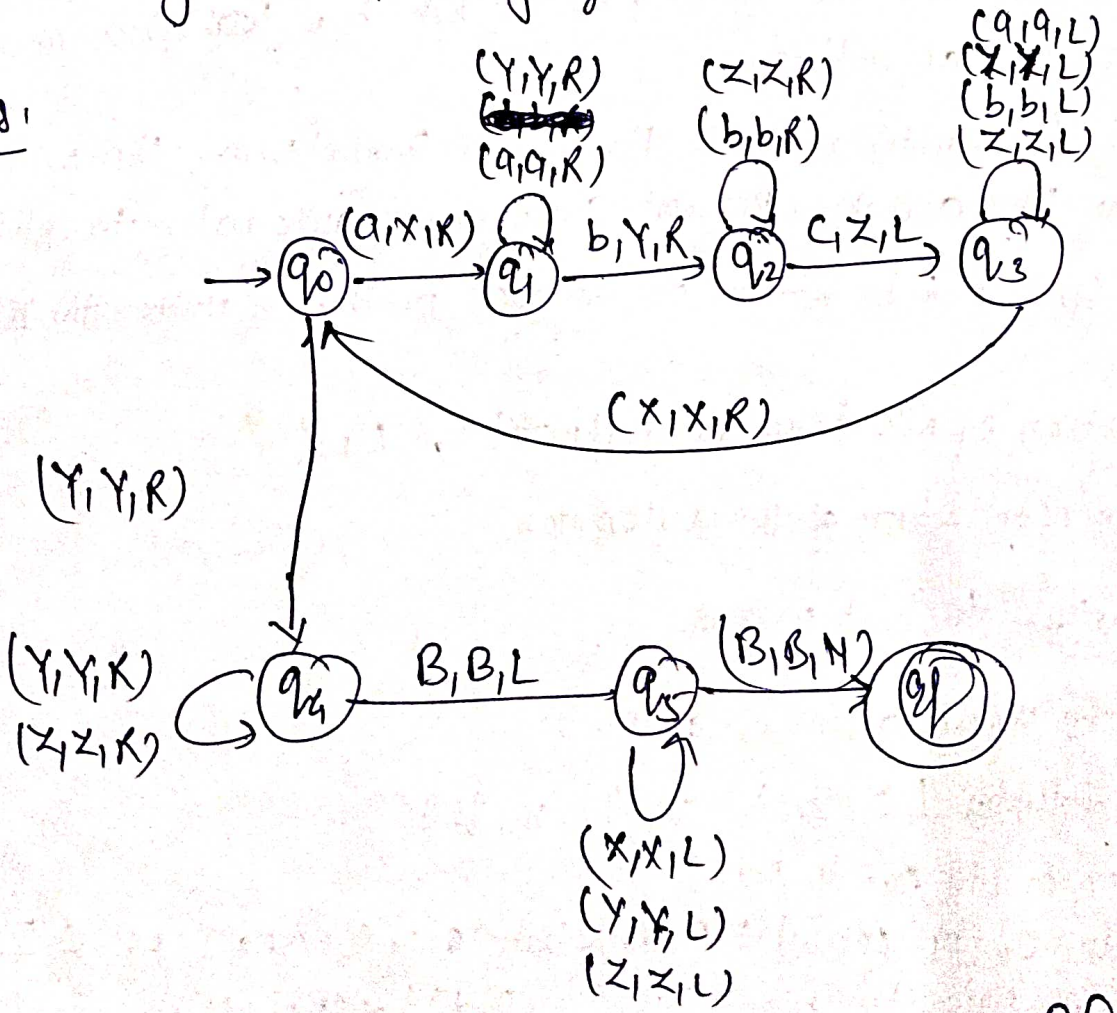


Transition Table

P.S	a	b	X	Y	B
→q_0	(q_1,X,R)	—	—	(q_3,Y,R)	—
q_1	(q_1,a,R)	(q_2,Y,L)		(q_2,Y,R)	—
q_2	(q_2,a,L)		(q_0,X,R)	(q_0,Y,L)	
q_3	—	—	—	(q_3,Y,R)	$(q_4,B,$
q_4	—	—	(q_4,X,L)	(q_4,Y,L)	$(q_4,B,$
q_f	—	—	—	—	—

Q. Design a TM for language $L = \{a^n b^n c^n \mid n \geq 1\}$

Ans:



P.T.O.

(Lecture-02)

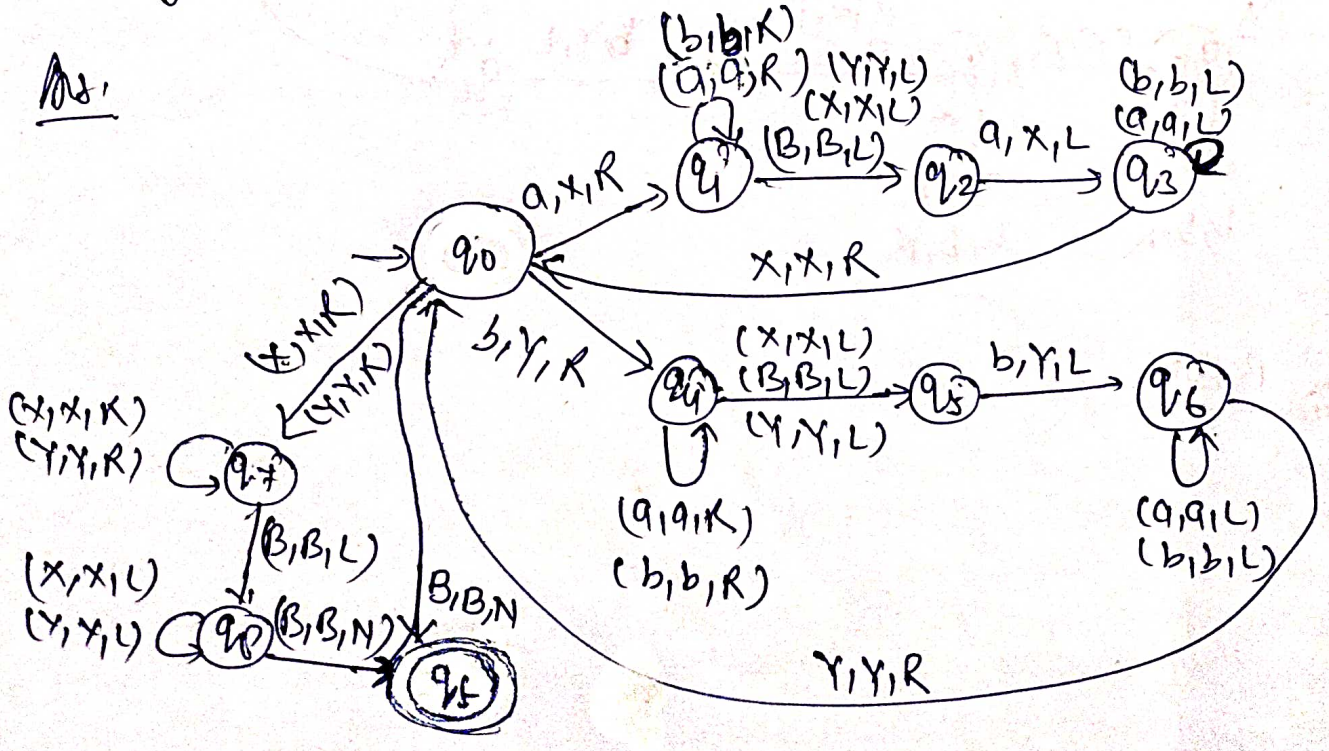
Transition Table:

Γ

P.S	a	b	c	X	Y	Z	B
→ q_0	(q_1, X, R)	—	—	—	(q_4, Y, R)	—	—
q_1	(q_1, a, R)	(q_2, Y, R)	—	—	(q_1, Y, R)	—	—
q_2	—	(q_2, b, R)	(q_3, Z, L)	—	—	(q_2, Z, R)	—
q_3	(q_3, a, L)	(q_3, b, L)	—	(q_0, X, R)	(q_3, Y, L)	(q_3, Z, L)	—
q_4	—	—	—	—	(q_4, Y, R)	(q_4, Z, R)	(q_5, B, R)
q_5	—	—	—	(q_5, X, L)	(q_5, Y, L)	(q_5, Z, L)	(q_f, B, R)
q_f	—	—	—	—	—	—	—

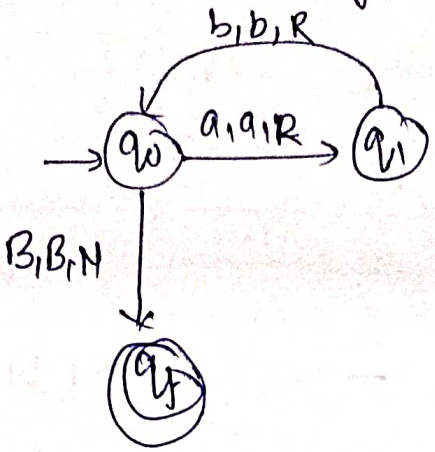
Q. Design a TM for the language $L = \{ WW^R \mid W \in (a,b)^* \}$

Ans:



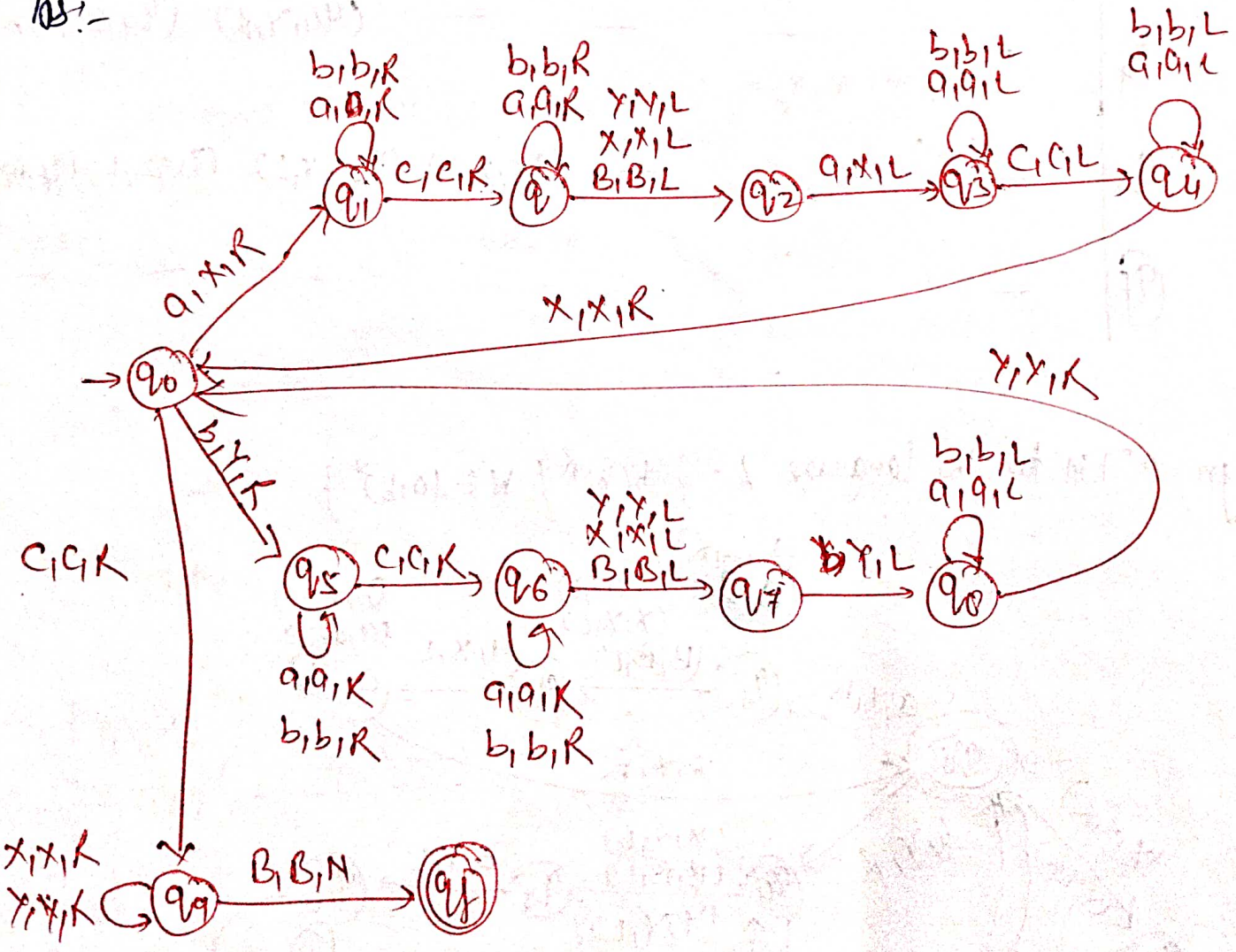
Q. Design a TM for the language $L = \{(ab)^n \mid n \geq 0\}$

Ans:



Q. Design a TM for language $L = \{ wcw^R \mid w \in (a,b)^+ \}$

Ans:-



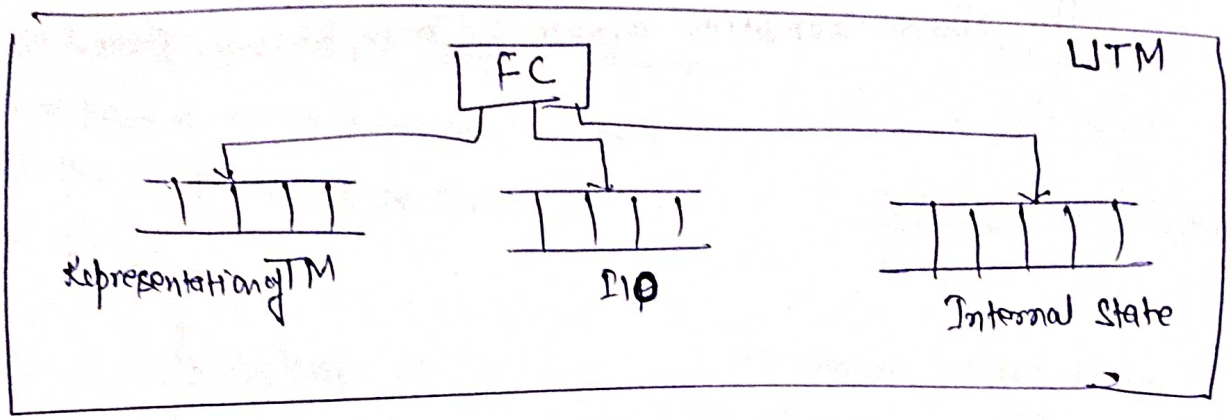
TM With Modifications

1) TM With stay option
2) TM With Semi infinite tape
3) offline TM
4) Multidiamentional TM
5) NDTM
6) UTM (Universal TM)
7) Multitape TM
8) Multihead TM
9) TM With 3-state
10) Jumping TM
11) Non Erasing TM
12) Always Writing TM

1) TM Without Writing Capacity
2) TM With I/p size tape
3) TM With tape used as stack (NDTM)
4) TM With finite ~~state~~ tape.

Universal Turing Machine:-

Universal Turing Machine follow the Property of turing thesis. A TM is work like a digital Computer, Means all the work done by digital computer can be done by TM.



The above Multitape TM is work like a digital Computer, it having three tape one for representation of TM which can used to store TM corresponding to the problem, 2nd is used for I/O string storage and 3rd is used to represent the states of the TM.

Let if we design a TM for a^n b^n then lets input strings is

BB 11 0 111 BB is store in I/O tape., Let initially M/C is on q_0 state and transect to some other states that is stored in Internal state. Binary representation of transition function is store in Representation of TM.

Let's $Q = \{q_0, q_1, q_2, \ldots\}$

$\Gamma = \{a_0, a_1, a_2, a_3, \ldots\}$

Transition function

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

$$=$$

Let's $q_0 = 1$ $q_1 = 11$ $q_2 = 111$ ⋮

$a_0 = 1$ $a_1 = 11$ $a_3 = 111$ ⋮

L R

1 11

for Transition

$(q_0, q_0) \vdash (q_1, q_1, R)$

0 1 0 1 0 11 0 11 0 11 0

Seperator

The above transition represented in Binary stored in TM tape.

Halting Problem

Halting Problem is Undecidability, It is not a problem, it just asks question:. " is it possible to tell whether a given machine will halt for some given I/P "

Exp:. I/P → A TM & i/p string W

Problem → Does the TM finish Computing of the string W in a finite no. of steps.

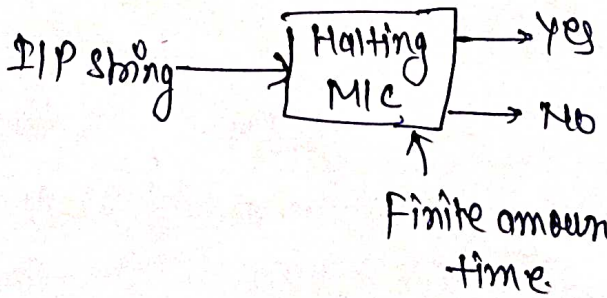
Note: The Answer must be yes / No

Proof:. Assum TM exists to solve this problem & then we will show it is Contradiction itself.

We will Call this T.M. as a Halting M/c that produce a 'yes' or 'No' in a finite amount of time.

⇒ if the halting m/c finishes in a finite amount of time, o/p comes as 'yes' otherwise as 'No'

The block diagram of Halting M/c



(Halting M/c Halts on i/p w)

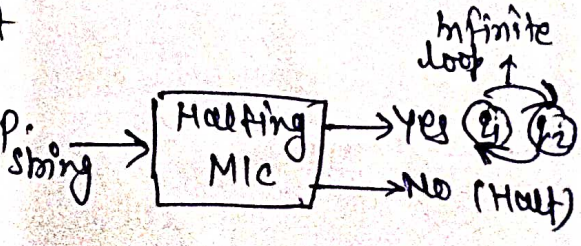
(Halting M/c does not Halt on i/p w)

Note:

Now we will design an inverted halting M/c as

- if H return yes, then loop forever
- if H return No, then Halt

The diagram of Inverted Halting M/c :



After that a M/c $(HM)_2$ Which i/p itself Contructed as

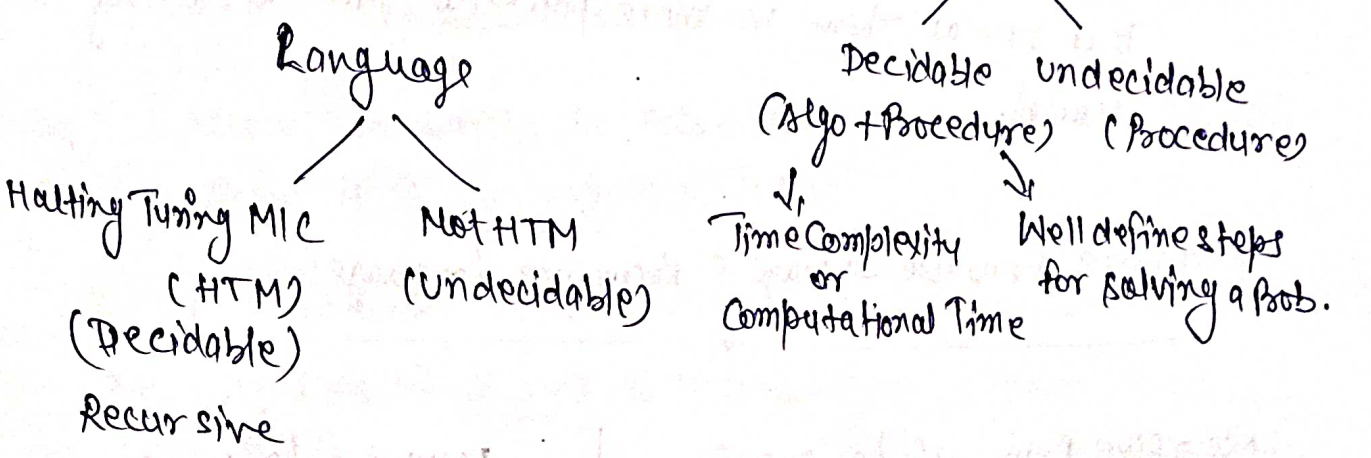
→ if $(HM)_2$ Halt on i/p, loop forever

→ else Halt

Decidability and Undecidability :-

If Any Problem that may/ solvable or not if solvable then it become decidable or un-decidable. In decidable there should be a Algorithm & Procedure but in undecidable there is only Procedure

In term of TM, if there is a Language and if we have a HTM corresponding to a language then it is decidable otherwise undecidable.



* The language corresponding HTL is recursive, if we prove language is recursive then it is HTL means it is decidable. & all lower language from Recursive like, Context sencitive, Context free, FA all are decidable.

* The set of all language is 2^{Σ^*} which is Uncountably Infinite (UCI) and set of all TM is Contable Infinite (CI), & set of all HTM is recursive, there are infinite language in UCI and finite no. of TM in CI, HTM is a recursive language that is subset of TM or Recursive enumerable language.

P.T.O

(Lecture-07)

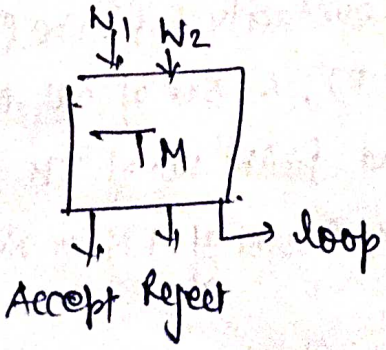
Turing Thesis: Turing thesis represented by or invented by L N Turing.
or
Church-Theorum ⇒ L N Turing is popular as Father of Computer Science.

1. Anything that can be done on any existing digital Computer can also done by Turing Machine.
2. No one has yet been able to suggest a problem, solvable by what we intuitively consider an algorithm, for which a TM Program can not be written
3. Alternative models have been proposed for mechanical Computation, but non of them is more powerful than the turing machine Model.

Recursive Enumrable language & Recursive Language:

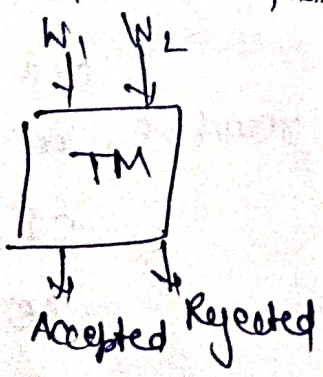
Recursive Enumrable Language

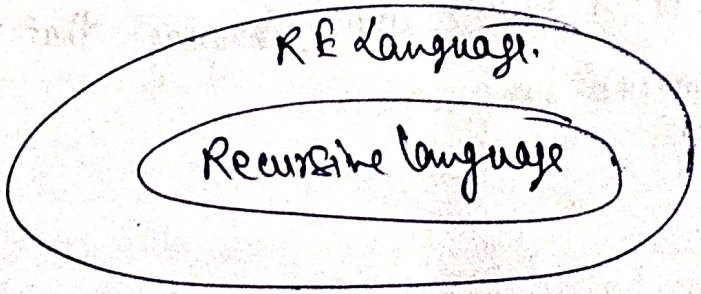
1. L is RE if there is TM
2. Three state :- Halt & Accepted
   Halt & Rejected
   Never Halt (loop)
3. closed under all except (set difference, Complement)



Recursive language

1. L is recursive if there is Halting / total TM
2. Two state :- Halt & Accepted
   Halt & Rejected
3. Closed under all except Homomorphism & Substitution





(Lecture-08)

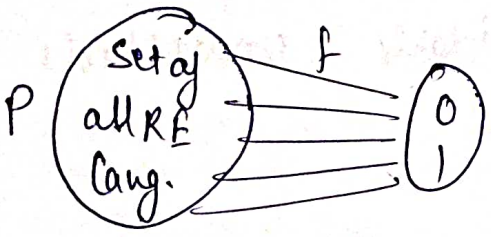
Rice's Theorem

Any non-Trivial property of Recursive Enumarable language is undecidable.

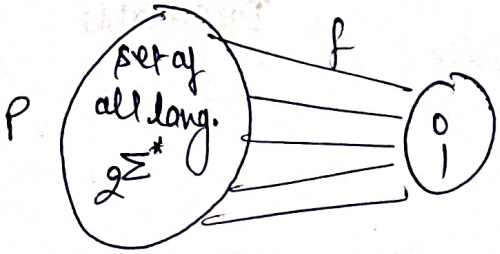
* The Property of language known as, the set of all language (2^{Σ^*}) mapped through function with the set of 0 (false) 1 (True) then

$P(L) = 1$ L Satisfies property P

$P(L) = 0$ L Not Satisfies Property P



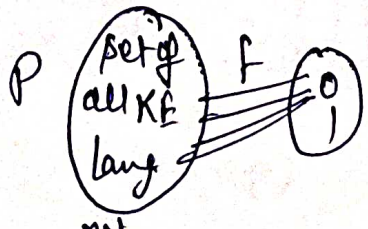
Property of RE language



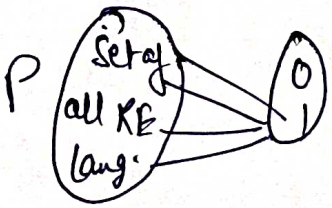
$P(L) = 0$ L doesnot stisfies Prop. P

$P(L) = 1$ L Satisfies Property P

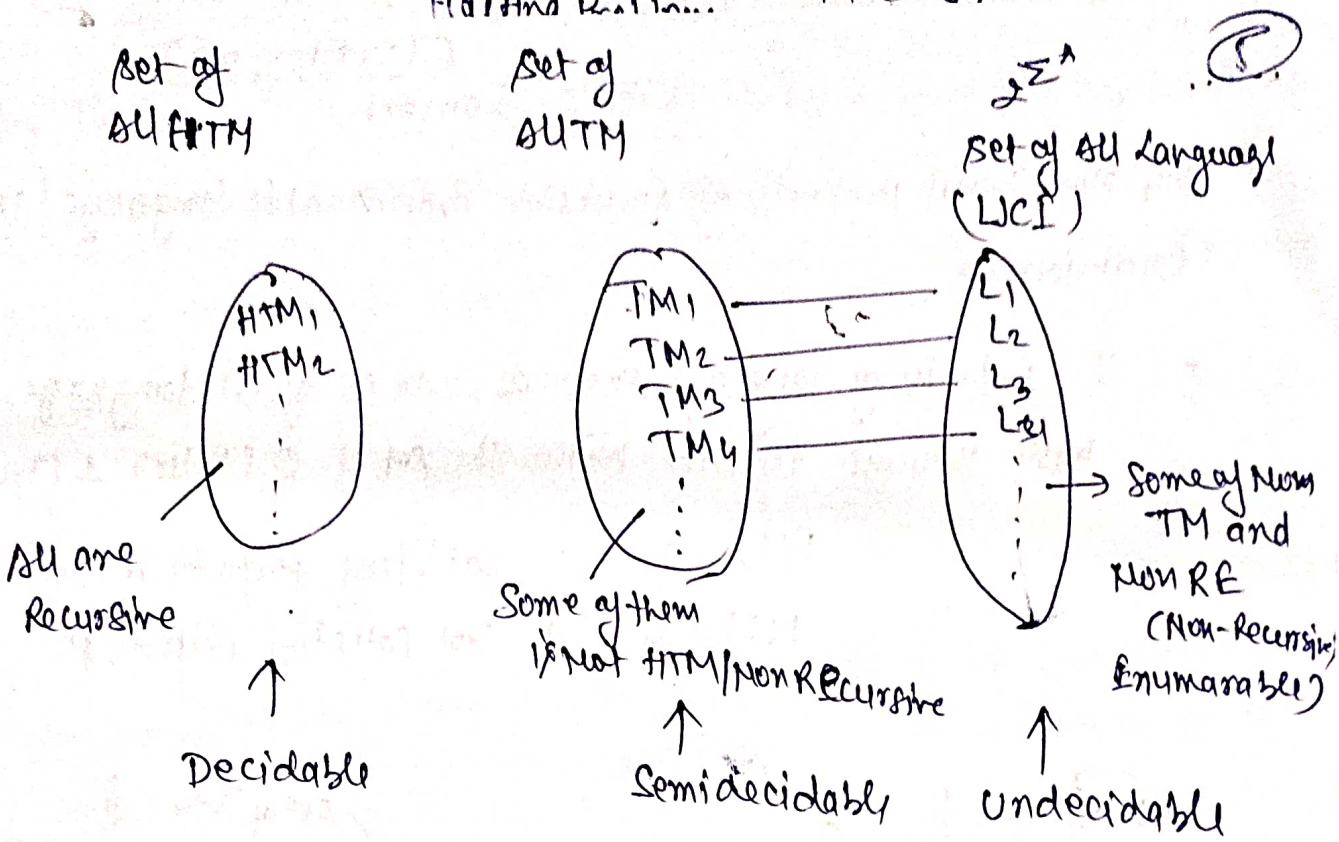
* if set of all RE language mapped only with 0 or 1, then it is called Trivial property of Recursive Enumarable (RE) language. if it is Partially mapped with 0 or 1 then it is called Non-Trival property of RE language.



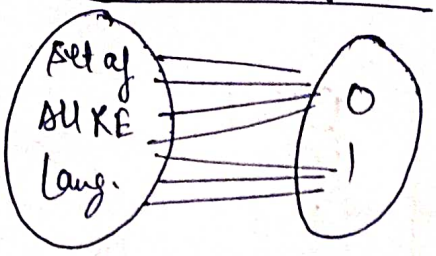
1. L is not RE language (Trivial Property)



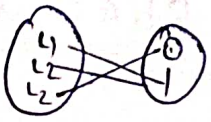
1. L is RE language (Trival Property
2. L is accepted by TM


Set of ALL HTM
Set of ALL TM
2^{Σ^*}
Set of All Language (LJCI)
HTM1
HTM2
TM1
TM2
TM3
TM4
L1
L2
L3
L4
Some of Non TM and Non RE (Non-Recursively Enumarable)
All are Recursive
Some of them is Not HTM/Non Recursive
Decidable
Semidecidable
Undecidable

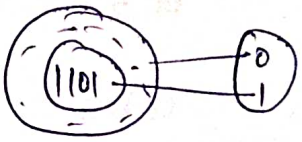
Non-Trivial Property

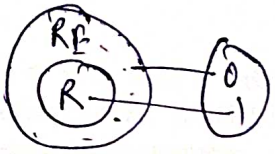


(1) L is finite language

(2) L is ϕ ⟶  Non-trivial

(3) L has 1101

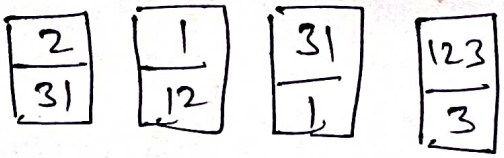
(4) L is Regular ⟶  Non-trivial

⟶  Non-trivial

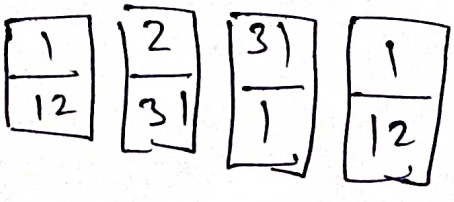
Post Correspondence Problem:-

The PCP introduced by EMIL POST in 1946 is an undecidable decision problem.

Eg 1 Dominos/Tiles



We need to find required set of dominos such that top & bottom string are same.

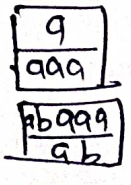


→ 123 11123

→ 123 11123 (decidable)

Eg 2

X (Top)	Y (Bottom)
a	aaa
abaaa	ab
ab	b



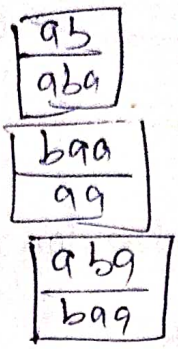
(decidable)

Top x: abaaa | a | a | ab → abaaaaaab

bottom Y: ab | aaa | aaa | b → abaaaaaab

Eg.

	X (TOP)	Y (Bottom)
1)	ab	aba
2)	baa	aa
3)	aba	baa



TOP x : ab aba aba aba - - - ∞

bottom Y : aba baa baa baa - - -

undecidable

UNIT V: (Important Questions)

1. What do you understand by Instantaneous Description of a Turing Machine ? Design a Turing machine that computes the integer function f which multiplies two given positive integers defined as follows : f(m, n) = m * n.
2. Let A = {001, 0011, 11, 101} and B = {01, 111, 111, 010}. Does the pair (A, B) have Post Correspondence (PC) solution ? Does the pair (A, B) have Modified Post Correspondence (MPC) solution ?(UPTU 2011-12)
3. Prove that recursively enumerable languages are closed under intersection.
4. What do you understand by undecidable problem ? State the Halting Problem and prove that Halting problem is undecidable.(UPTU 2013-14)
5. Design a TM for the following language:L= { $a^n b^n c^n$ |n≥1} (UPTU 2018-19)
6. Writeshortnoteon: (UPTU 2018-19)
   a. RecursiveLanguageand RecursivelyEnumerableLanguage.
   b. PCPproblemandModifiedPCPProblem.
7. Construct a turing machine for reversing a string.
8. Let T1 and T2 be two Turing machines; compute the functions f1 and f2 from N to N (where N is a natural number), respectively, construct a Turing machine that computes the function min (f1 , f2).(UPTU 2013-14)
9. Prove that every recursively enumerable language where complent is closed must be recursive.
10. Write a short note on the Multi Tape Turing Machine.
11. Design a Turing Machine that accept the language L = {ww^R | w is in (0+1)* and w^Rrepresents everse w}.(UPTU 2010-11)
12. Design a Turing Machine that can compute proper subtraction i.e. m$n, where m and n are positive integers, m$n is defined as m-n if m>n n.≤and 0 if m.