Unit-V - Turing Machines (TM)

★ Turing M/c Model

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

Q - Finite set of states.
Γ - Finite set of external symbols
Σ - Finite set of I/p symbols
B - Blank symbol majorly used as end marker for I/p.
δ - is a transition function.

★ Design of TM :-

1> The I/p tape having infinite number of cell each cell containing one I/p symbol and thus the I/p string can be placed on a tape. The empty tape is filled by blank symbol (B)

2> The finite control & the tape head which is responsible for reading the current I/p symbol. The tape head can move to left or right.

3> A Finite set of states through which m/c has to undergos.

4> Finite set of symbols called external symbol which are used in building the logic of Turing M/C.

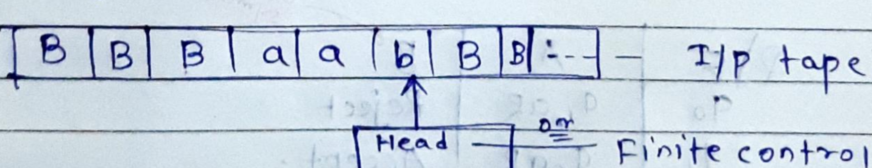


fig:- Turing M/C.

5> you can perform read & write operation on TM.

6> $\delta = Q * \Gamma \longrightarrow Q * \Gamma * (L, R)$ Change in cell

$(q_0, a) \longrightarrow (q_1, X, R)$

Ex 1) Design TM for a Lang over $\Sigma = \{a\}$ which contains the string of odd number of a's

Step 1:- Let TM is defined as

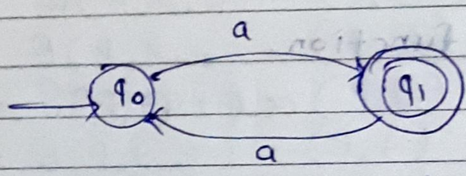
$M = (Q, \Sigma, \partial, q_0, F, \Gamma, B)$.

Step 2:- Find TM that accepts the string containing odd Number of a's

$L(M) = \{a\}$

similary aaa, aaaaa, ---

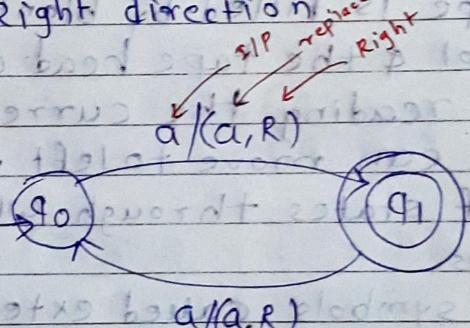
Now find transition Graph.



B	B	a	a	a	B

Step-3:- Transition dig for T.M.

It is very clear that we want to count number of a's, so we are read I/P only in Right direction.



Step-4:- Transition table.

Blank symbol

$Q \backslash \Sigma$	A	B
q_0	q_1, aR	Reject
q_1	q_0aR	Accept.

Step 5:- Hand Run

W = BaaaB

↑

q_0

$\partial(q_0, a) = q_1 aR$

BaaaB
↑
q_1

$\partial(q_1, a) = q_0 a R$

BaaaB
↑
q_0

$\partial(q_0, a) = q_1 a R$

BaaaB
↑
Accept

$\partial(q_1, B) = \text{Accept}$

$q_1 \in$ Accept.

Hence ω = aaa is Accepted.

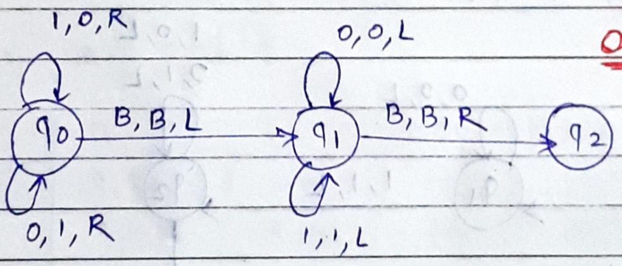
2) construct TM for 1's complement of binary Number.
[SPPU: May-18, Mark-6]

⟶

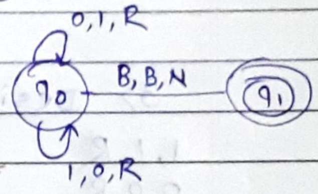
Step-I :-

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

Step 2 :-



OR



Step-3 :- Transition table.

States	0	1	B
q_0	$1Rq_0$	$0Rq_0$	BLq_1
q_1	$0Lq_1$	$1Lq_1$	BRq_2
q_2	—	—	

Step-4

B10B
↑

$\partial(q_0, 1) = 0Rq_0$

B10B
↑

$\partial(q_0, 0) = 1Rq_0$

B 1 0 B $\qquad \delta(q_0, B) = BLq_1$

B 1 0 B $\qquad \delta(q_1, 0) = 0Lq_1$

B 1 0 B $\qquad \delta(q_1, 1) = 1Lq_1$

B 1 0 B $\qquad \delta(q_1, B) = BRq_2$ — Accept.

3> Design TM to find 2's complement.

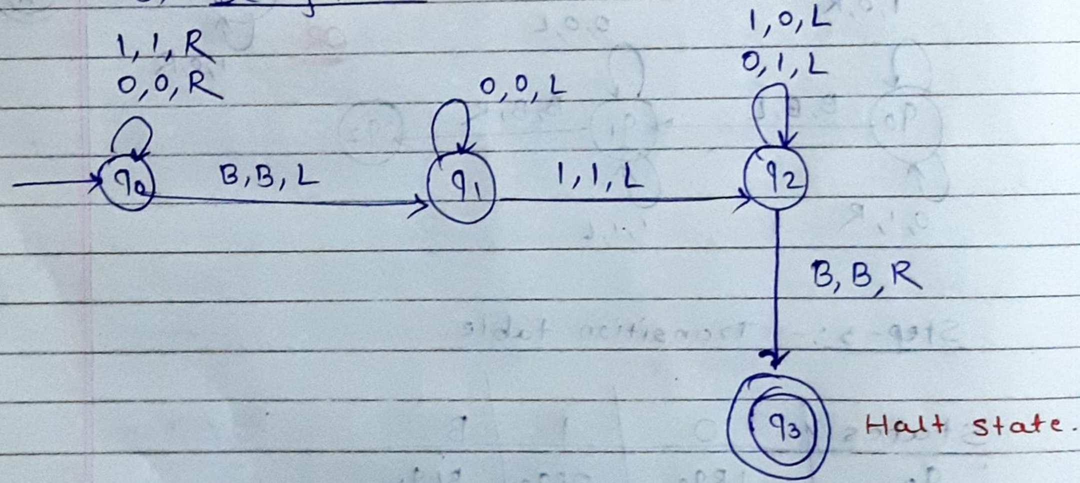
1> | B | B | 1 | 0 | 0 | 1 | 1 | 0 | 0 | B | B | I/P

2> <u>Logic</u>:-

1 0 0 1 1 0 0

0 0 1 1 0 1 0 0

3> <u>Design T.M</u>:-



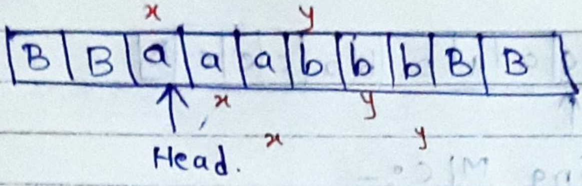
(4) B 1 0 0 1 1 0 0 B $\qquad$ | $\qquad$ B 1 0 0 1 B

B 0 1 1 0 1 0 0 B. $\qquad$ | $\qquad$ B 0 1 1 1 B

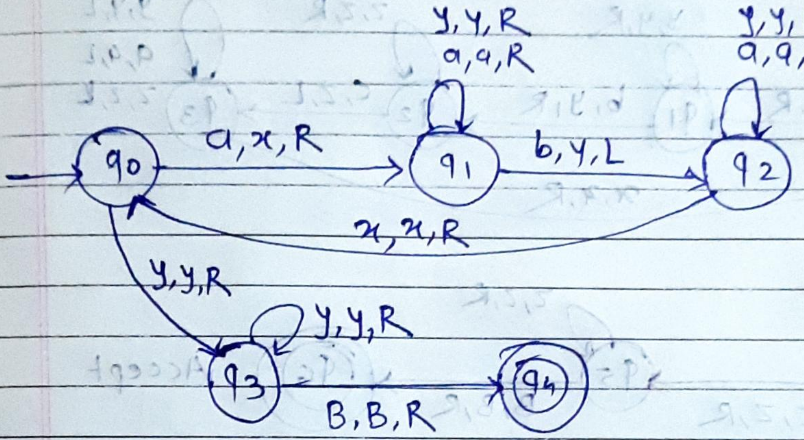
4> construct TM for the Lang $L = \{a^n b^n\}$ where $n \geq 1$

→

L = aaabbb You can Read single 'a' then 'b'



If 'a' is read then it is Replace by 'x';
if 'b' is read then it is Replace by 'y';



Transition Table

	a	b	x	y	B
q_0	$q_1 x R$	—	—	$q_3 y R$	—
q_1	$q_1 a R$	$q_2 y L$	—	$q_1 y R$	—
q_2	$q_2 a L$	—	$q_0 x R$	$q_2 y L$	—
q_3	—	—	—	$q_3 y R$	$q_4 B R$
q_4^*	—	—	—	—	—

5> L = BaabbB
↑

$\delta(q_0, a) = q_1 x R$
$\delta(q_1, a) = q_1 a R$
$\delta(q_1, b) = q_2 y L$
$\delta(q_2, a) = q_2 a L$
$\delta(q_2, x) = q_0 x R$
$\delta(q_0, a) = q_1 x R$
$\delta(q_1, y) = q_1 y R$
$\delta(q_1, b) = q_2 y L$

$\delta(q_2, y) = q_2 y L$
$\delta(q_2, x) = q_0 x R$
$\delta(q_0, y) = q_3 y R$
$\delta(q_3, y) = q_3 y R$
$\delta(q_3, B) = q_4 B R$
↑
Accepted.

5) construct TM for $a^n b^n c^n \mid n \geq 1$

$n = 3$

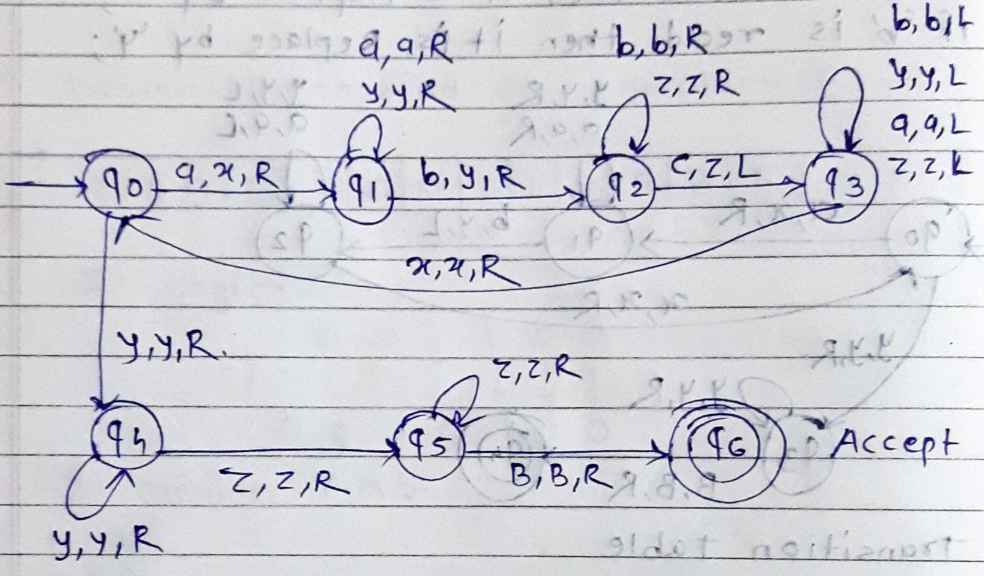
1) L = aaa bbb ccc

2)

B	B	a	a	a	b	b	b	c	c	c	B

↑

3) Turing M/C :-



4) L = B aabbccB

↑ ↑ ↑

B x a y b z c B

B x x y y z z B

6) Draw a transition table for turing machine accepting each of the following Lang :

$a^i b^j \mid i < j$

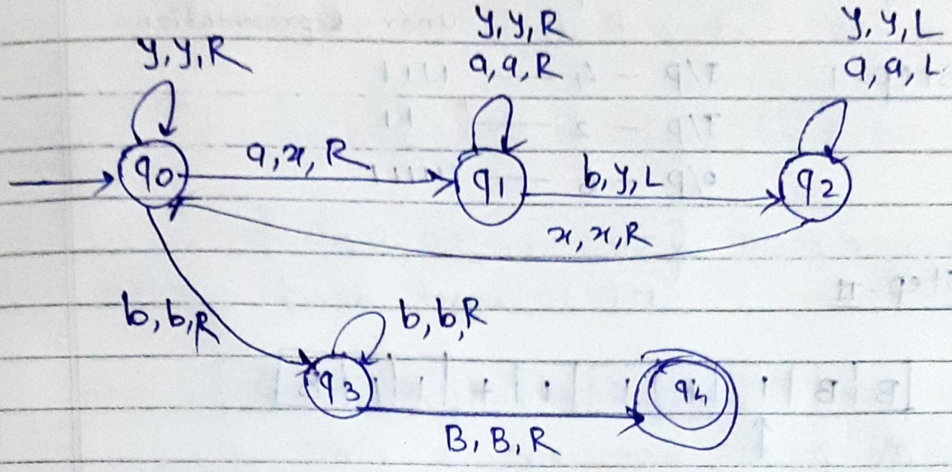
Step-I :- L = aabbb

Step II :-

B	B	a	a	b	b	b	B	B

↑

Step-III



Step-IV

B a a b bb B

x ← y

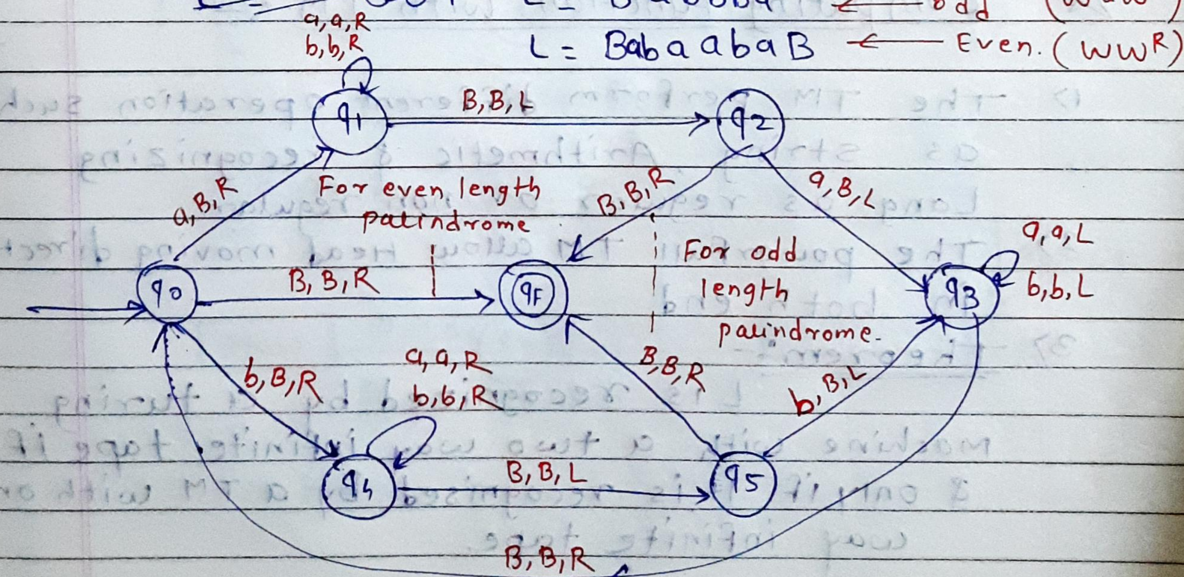
B x x y y b B

7) construct TM for all palindrome of a string for $\Sigma = \{0, 1\}$. or $\Sigma = \{a, b\}$

→

~~L = 001~~ L = Babbba ← odd (waw^R)

L = BabaabaB ← Even. (ww^R)



8) construct TM for the addition function for the unary number system.

unary Representation

step-I

I/P – 4 — 1111

\+

I/P – 2 — 11

O/P – 6 — 111111

Step-II

B	B	1	1	1	1	+	1	1	B

↑

Step-III 1) Logic if '+' sign is obtain then it is Replace by '1'

2) Last '1' Replace by 'B'

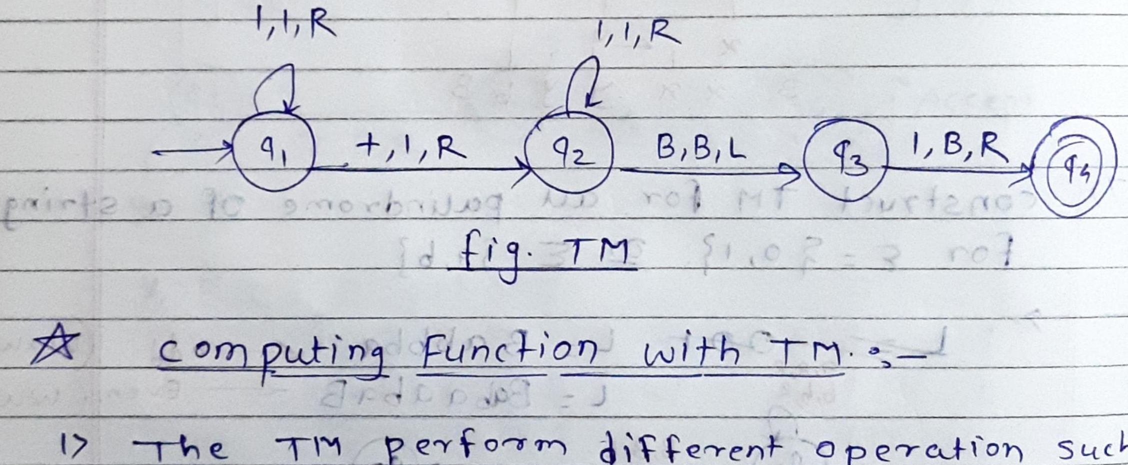


fig. TM

☆ computing Function with TM :-

1) The TM perform different operation such as string, Arithmetic & recognizing Lang as regular or non regular.
2) The powerfull TM allow Head moving direction in both end.
3) Theorem:-
L is recognized by a turing machine with a two way infinite tape if & only if it is recognised by a TM with one way infinite tape.

→ consider two TM

$M_1 = (Q_1, \Sigma_1, \Gamma_1, \delta_1, q_1, B, F_1)$ one way infinite

$M_2 = (Q_2, \Sigma_2, \Gamma_2, \delta_2, q_2, B, F_2)$ two way infinite

$L = \{a_0, a_1, a_2, a_3, a_4, a_5\}$ It is represent in the both type of TM.

#	a_5	a_4	a_3	a_2	a_1	a_0	B	---

This a one way TM & '#' symbol placed in the very first cell from left side. This symbol is used as indicator for the left side termination.

---	B	B	a_5	a_4	a_3	a_2	a_1	a_0	B	B	---

This a two way TM.

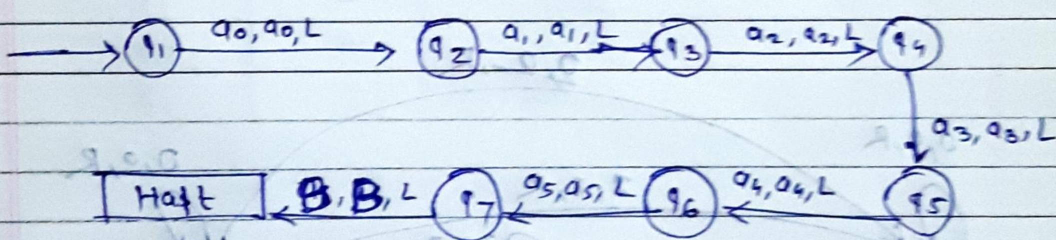


Fig:- TM for M_2

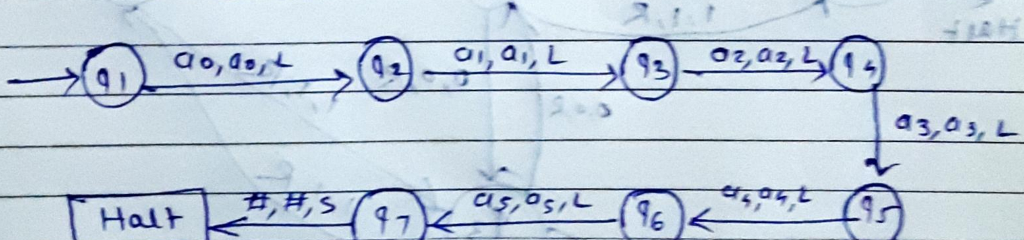


Fig TM for M_1.

∴ Hence proved.

Page No:
Date: \ \

Ex 1> Design TM to recognize an arbitary string divisible by 4 from $\Sigma = 0,1,2$

1> As per the I/P value we will obtain ternary number.

2> This ternary number divisible by 4 or not it check.

3> Remider value as follow 0,1,2,3 & each state q_0, q_1, q_2, q_3 denote reminder respecti-vely

4> How to conver ternary number into decimal number.

$(121)_3$
$= 1 * 3^2 + 2 * 3^1 + 1 * 3^0$
$= 1 * 9 + 2 * 3 + 1 * 1$
$= 9 + 6 + 1$
$= (16)_{10}$

5> I/P = 121

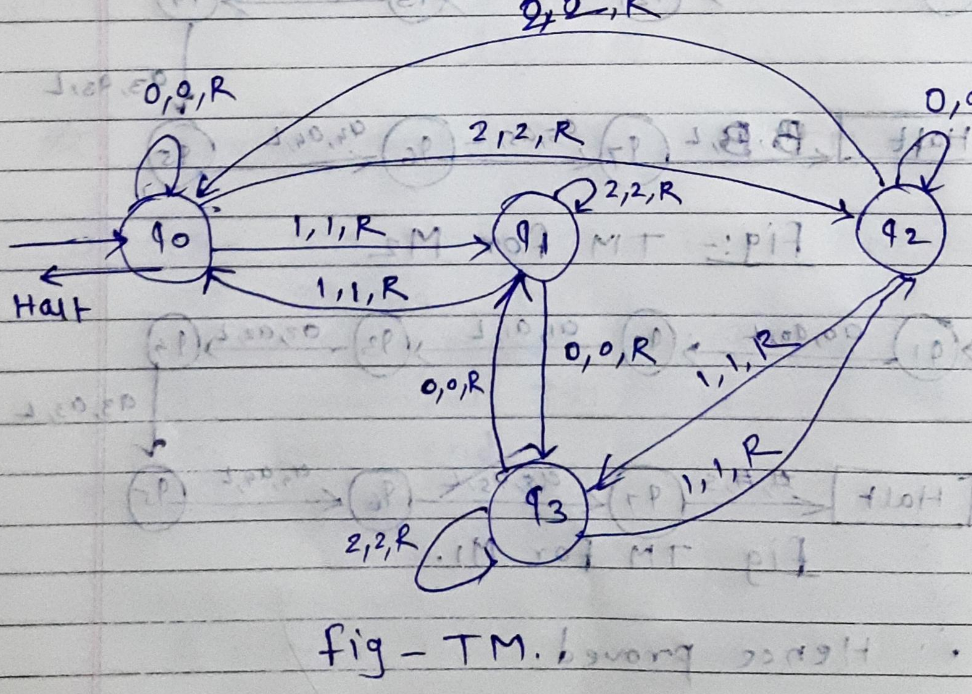


fig - TM.

2) Construct TM for the function $f(x) = x+3$

→ We will represent number x using unary number. Hence the I/p $\Sigma = \{1\}$,

If we assume $x = 4$ then sample I/p tape can be represent as.

1	1	1	1	B	B	B

~~Logic~~ Logic - Last 3 'B' symbol Replace by '1'

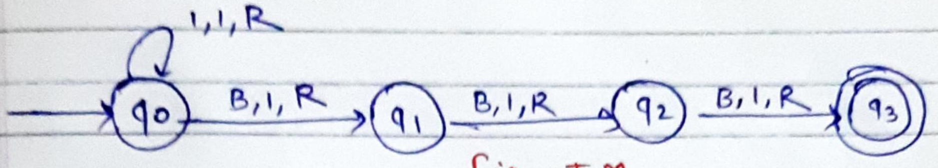


fig. TM

Transition table :-

I/p / State	1	B
q_0	$1, q_0, R$	$1, q_1, R$
q_1	—	$1, q_2, R$
q_2	—	$1, q_3, R$
q_3	—	—

2) Construct TM for the function $f(x) = x+3$

→ We will represent number x using unary number. Hence the I/P $\Sigma = \{1\}$.
If we assume $x = 4$ then sample I/p tape can be represent as.

1	1	1	1	B	B	B

~~Logic~~ Logic – Last 3 'B' symbol Replace by '1'

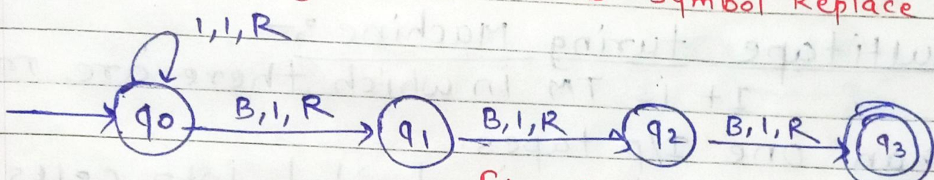


fig. TM

Transition table :-

State \ I/P	1	B
q_0	$1, q_0, R$	$1, q_1, R$
q_1	—	$1, q_2, R$
q_2	—	$1, q_3, R$
q_3	—	—

☆ Variants of Turing Machines :-

1) **Deterministic T.M :-**

It is a kind of turing machine in which the set of rules (I/P) denotes the single specific action on reading particular I/p in current specific state.
It is similar to DFA.

2) **Non Deterministic T.M (NTM) :-**

It is a kind of turing machine in which the set of rules denotes more than one specific action on reading

particular I/p in current specific state.
It is similar to NFA.

3) Universal TM:-

the TM to compute any computable fun it is called as universal TM.

The universal TM is TM which accepts many turing Machines.

4) Multitape turing Machine :-

It is TM in which there are more than one I/p tapes.

Each tape is devided into cells and each cell can hold any symbol of finite tape alphabet.

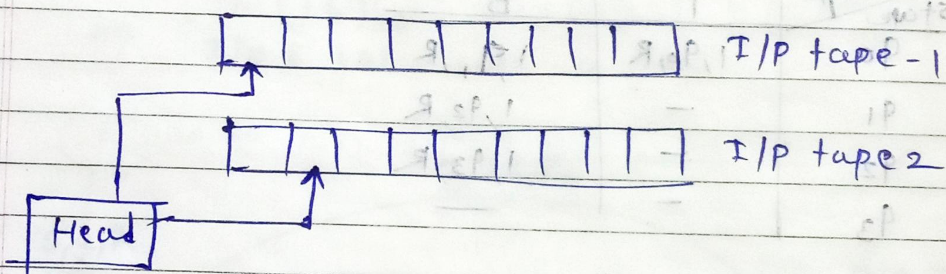


fig Multitaple TM.

Halting Vs looping

Halting :-

Halting means that the program on certain I/p will either accept it & halt or reject it & halt.

It would never go into an Infinite loop.

Halting means terminating.

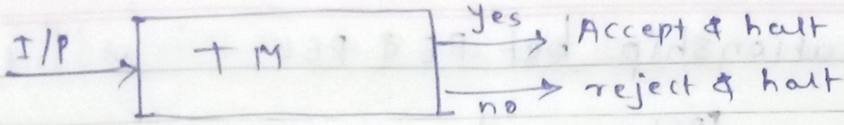


Fig Halting.

Looping:-

Looping means the program on certain I/p will either accept it & enter it in infinite loop or reject it & halt.

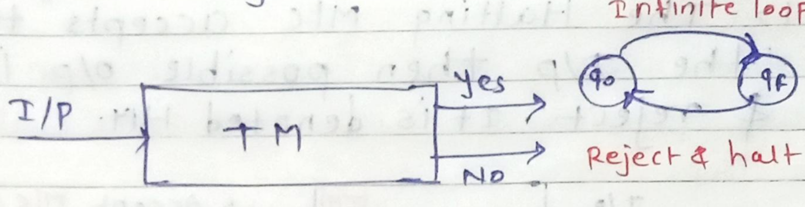


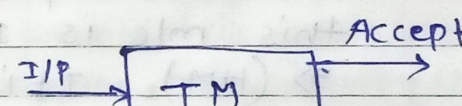
fig Looping

☆ A turing - unrecognizable Language:-

1> A TM can accept Type 0 Grammer.

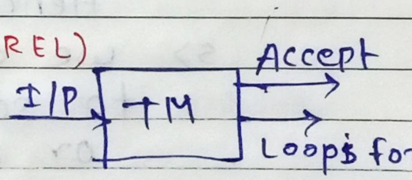
2> Type 0 Grammer are called recursively enumerable Lang.

3> Recursive Lang:- (RE)



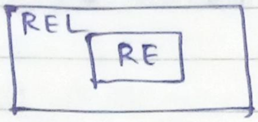
It is said to be recursive if there exists a TM that accepts every string of the Lang. & every string is rejected if it is a not belonging to that Lang.

4> Recursively Enumerable Lang:- (REL)



A Lang is said to be recursive if there exists a turing M/c that accepts every string belonging to that lang. & if the string does not belong to that lang then It can cause a TM to enter in an infinite loop.

5> Relationship bet RE & REL



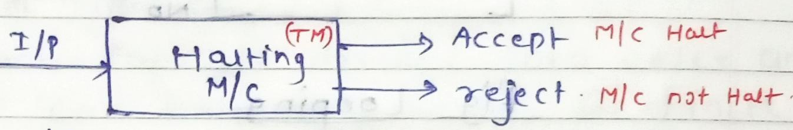
RE ∈ REL

proof :-

The Halting problem is not decidable problem.

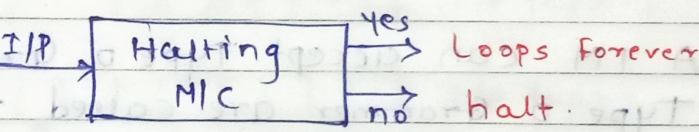
→ 1> The Halting M/C accepts the the I/p then possible o/p is Accept & reject. It is denoted HM.

1> HM



2> consider contradiction Halting M/c which is denoted by HM' (Inverted Halting M/c)

2> HM'



It is loop forever o/p is generated because this m/c is Inverted Halting M/c.

3> $(HM)_2$ It is second M/c.

which I/p itself is constructed.

condition i> IF $(HM)_2$ halt an I/p — Loop forever

ii> else — Halt.

4> so here we get contradiction

Hence. Halting problem is undecidable.

5> we can not design any generlize algo to solve the given problem is halting or not.

☆ Differentiate between FA & TM.

FA	TM:
1> It accepts Regular Lang.	1> It accept Recursive & Recursively enumerable Lang.
2> It can not be programmed	2> It can be programmed.
3> It accepts small class of Lang.	3> It accepts large class of Lang.

☆ Reducibility:-

1> It is a problem solving techniques in mathematics.

2> Let 'A' & 'B' are two problems & 'A' is reduced to 'B'. If we solve 'B', we solve 'A' as well. If we can't solve 'A', we can't solve 'B'.

3> to decide the given problem is solvable or not, we use mapping reducibility techniques.

4> If 'A' is reduced to 'B' & 'B' is decidable, then 'A' is decidable.

☆ Limitations of Turing Machines:-

1> The TM is that they do not model concurrency well.

2> The TM is that they do not model the strengths of a particular arrangment well.

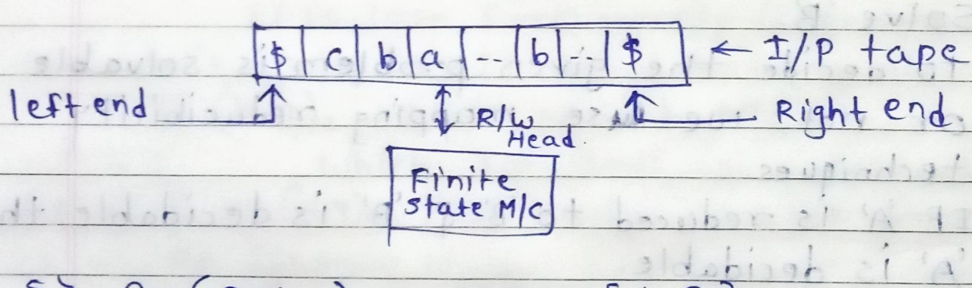
Linear Bounded Automata (LBA) :-

1) Mutitrack non deterministic TM with a tape of some bounded finite length.
2) I/p alphabet contains 2 special symbol which is end marker.
3) LBA is a 8 tuple

$$M = (Q, X, \Sigma, q_0, M_L, M_R, \delta, F)$$

Q = Finite set of state.
X = Tape Alphabet
Σ = I/p Alphabet
q_0 = Initial state.
M_L = Left end Marker
M_R = Right end Marker
δ = transition function
F = Final state.

4) <u>Model of LBA</u>



5) $\delta : (Q \times \Sigma) \longrightarrow Q \times \Sigma \times \{L, R\}$
6) LBA is combination of TM + I/p size tape
7) LBA accept context sensitive Lang.
8) FA < PDA < LBA < TM.
9) Ex) 1) $L = \{a^n, b^n, c^n \mid n \geq 1\}$ } EX of LBA

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