MODULE NO : 4

CHAPTER NO : 1

CHAPTER NAME : TURING MACHINE

CHAPTER OBJECTIVES :

1. Definition of Turing machine.
2. Turing Machine Model.
3. Design of Turing Machine.
4. Computable functions.
5. Church's hypothesis.
6. Counter Machine.
7. Types of Turing M/c (Proofs not required)
8. Universal Turing M/c.
9. Halting Problem.

- **Introduction of Turing M/c :**
- Just as in physics no particle's speed can not be greater than light, similarly no computer's power is less than turing machine.
- Turing M/c is a mathematical model of computation. that defines abstract computer was invented in 1936 by Alan Turing.
- If a task or problem can be solved using Turing M/c then our computer can also solve it.
- If a problem is not solvable in the Turing M/c then it it also not solvable using ~~into~~ any computer.
- Turing M/c is a language generator, acceptor & transducer.

  Acceptor → Any language can be accepted by TM.

  Generator → Like Mealy Moore M/c TM can also generate language.

  Transducer → It can solve any mathematic function like +, −, /, %, etc.

(2) DEFINITION OF TURING MACHINE :

- It is defined by 7 tuples.

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

1. $\Sigma \rightarrow$ Set of i/p symbols.
2. $Q \rightarrow$ Set of finite set of states.
3. $q_0 \rightarrow$ Initial State. where $q_0 \in Q$.
4. $F \rightarrow$ Set of final State $[F \subseteq Q]$
5. $\Gamma \rightarrow$ tape alphabet
6. $B \rightarrow$ Blank symbol. $[B \in \Gamma]$
7. $\delta \rightarrow$ $\delta : Q \times \Sigma \longrightarrow Q \times \Gamma \times (L/R)$

- Tape is a two way infinite tape which is divided into cells & in each cell we can insert one i/p symbol.
- Unlike FA/PDA here we do not use any read head but we are using reading & writing head.
- Unlike PDA in TM we do not use any ~~read head~~ Stack. As because the transition function δ whose definition is ~~$Q \times \Sigma$~~ $\delta : Q \times \Sigma \times \Gamma \longrightarrow (Q \times \Gamma)^*$ (for DPDA)

become to complex as because we need to ~~perform~~ read the current state/PS ~~i.e~~ from Q & current i/p symbol from the tape $\in \Sigma$ & current stack " " " TOS $\in \Gamma$.

So, this operation becomes a bit Complex.

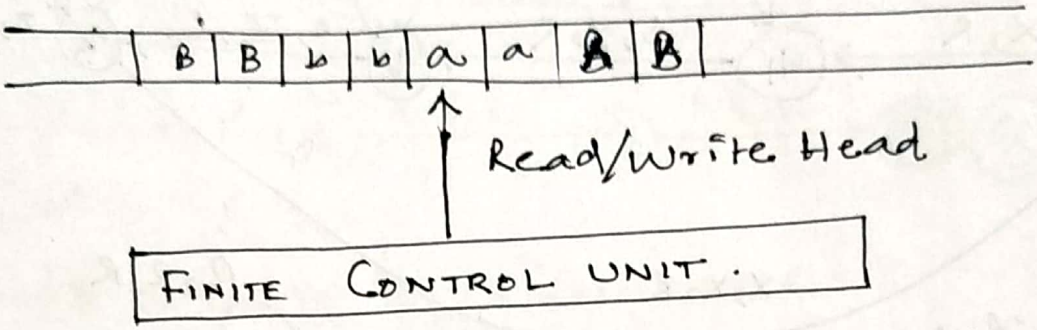
- $\Sigma \subseteq \Gamma$
- Empty cells are ~~filled~~ up by ~~black~~ blank symbol.
- Transition function δ is, Γ Capital (Sigma) Gamma.

on a present state, & taking i/p Symbol ■ we transit to ● a next state & we are generating an o/p symbol ● which belongs to Γ to be written

into the tape & L/R denotes which way we can move from the current position of the tape to the Left (L) or right (R).

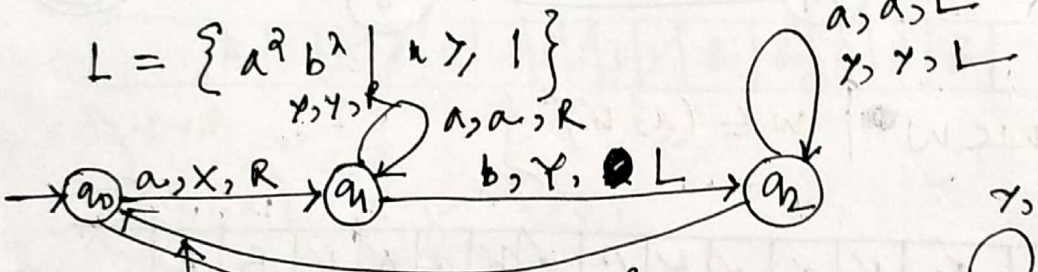
- Power of a M/c : A language accepted by a M/c /generated. or mathematical function computed by a M/c

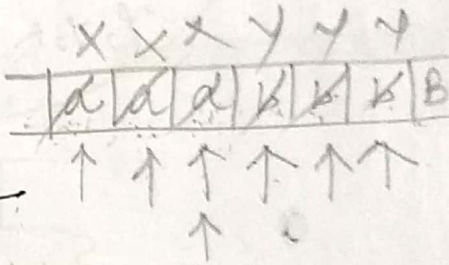
(2) TURING MACHINE MODEL :



(3) DESIGN OF TURING MACHINE :

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





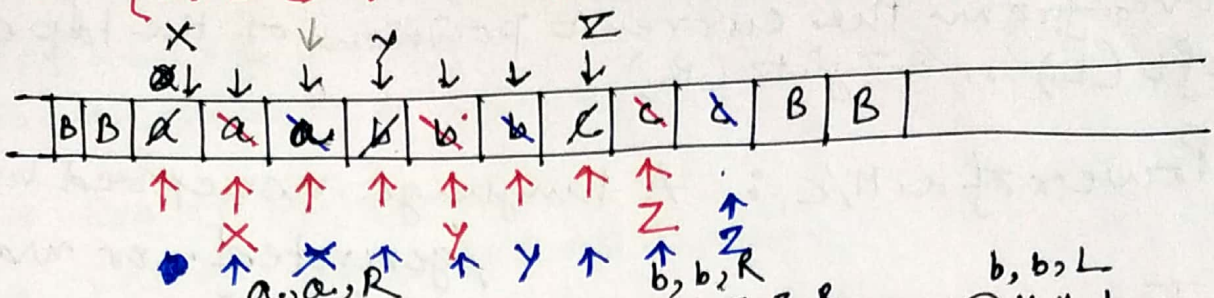
[δ maps q_0 by reading i/p symbol $a \in \Sigma$ we write the position of a by $X \in \Gamma$ & we are moving towards Right.]

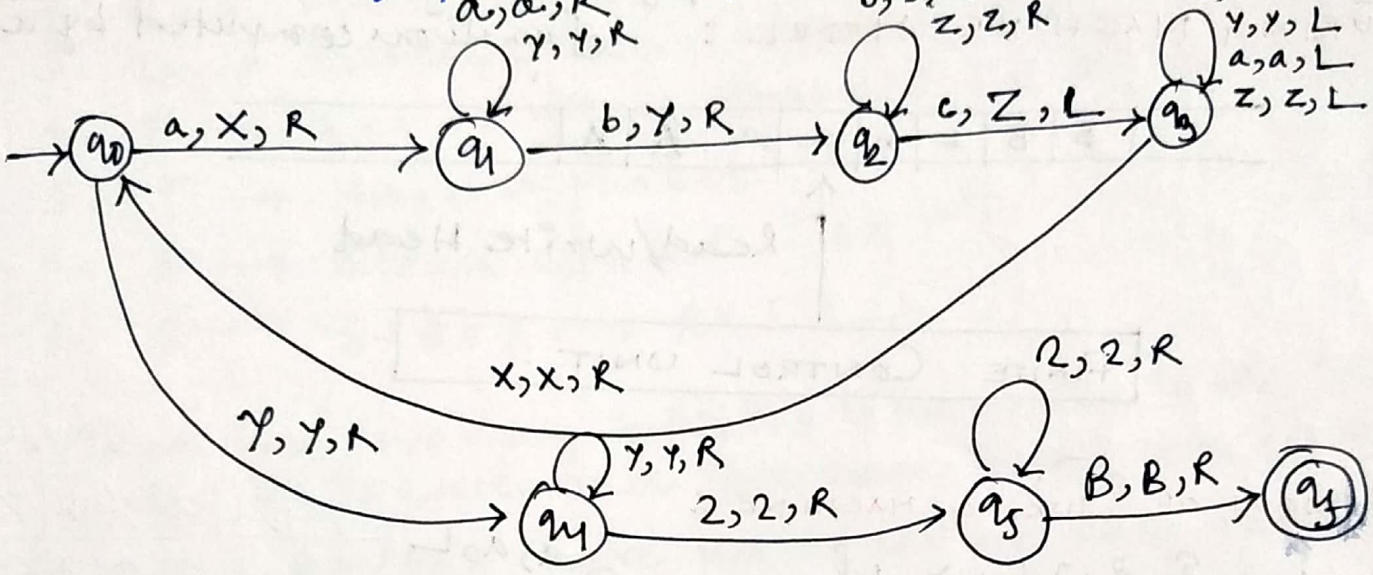
$\delta(q_0, a) \rightarrow (q_1, X, R)$
$\delta(q_1, Y) \rightarrow (q_1, Y, R)$
$\delta(q_1, a) \rightarrow (q_1, a, R)$
$\delta(q_1, b) \rightarrow (q_2, Y, L)$
$\delta(q_2, a) \rightarrow (q_2, a, L)$
$\delta(q_2, Y) \rightarrow (q_2, Y, L)$
$\delta(q_2, X) \rightarrow (q_0, X, R)$
$\delta(q_0, Y) \rightarrow (q_3, Y, R)$
$\delta(q_3, Y) \rightarrow (q_3, Y, R)$
$\delta(q_3, B) \rightarrow (q_f, B, R)$

This m/c is deterministic because in any state we do not take multiple move.

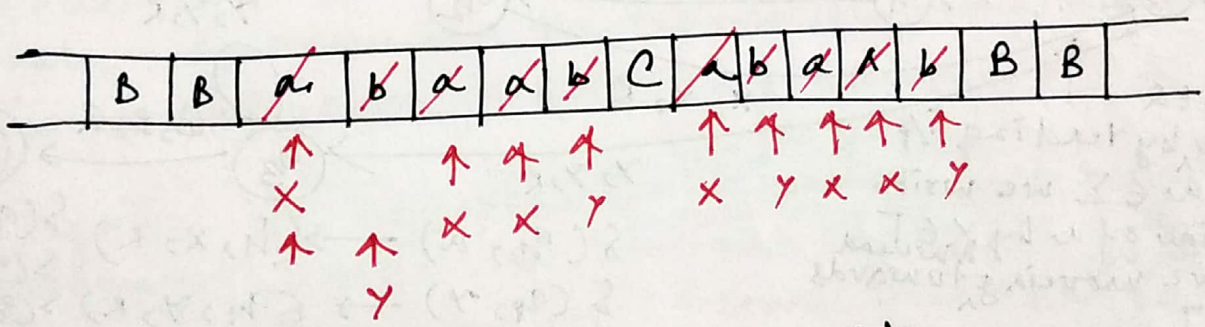
	a	b	X	Y	B
→ q_0	q_1, X, R	H	H	q_3, Y, R	H
q_1	q_1, a, R	q_2, Y, L	H	q_1, Y, R	H
q_2	q_2, a, L	H	q_0, X, R	q_2, Y, L	H
q_3	H	H	H	q_3, Y, R	q_f, B, R
q_f	H	H	H	H	H

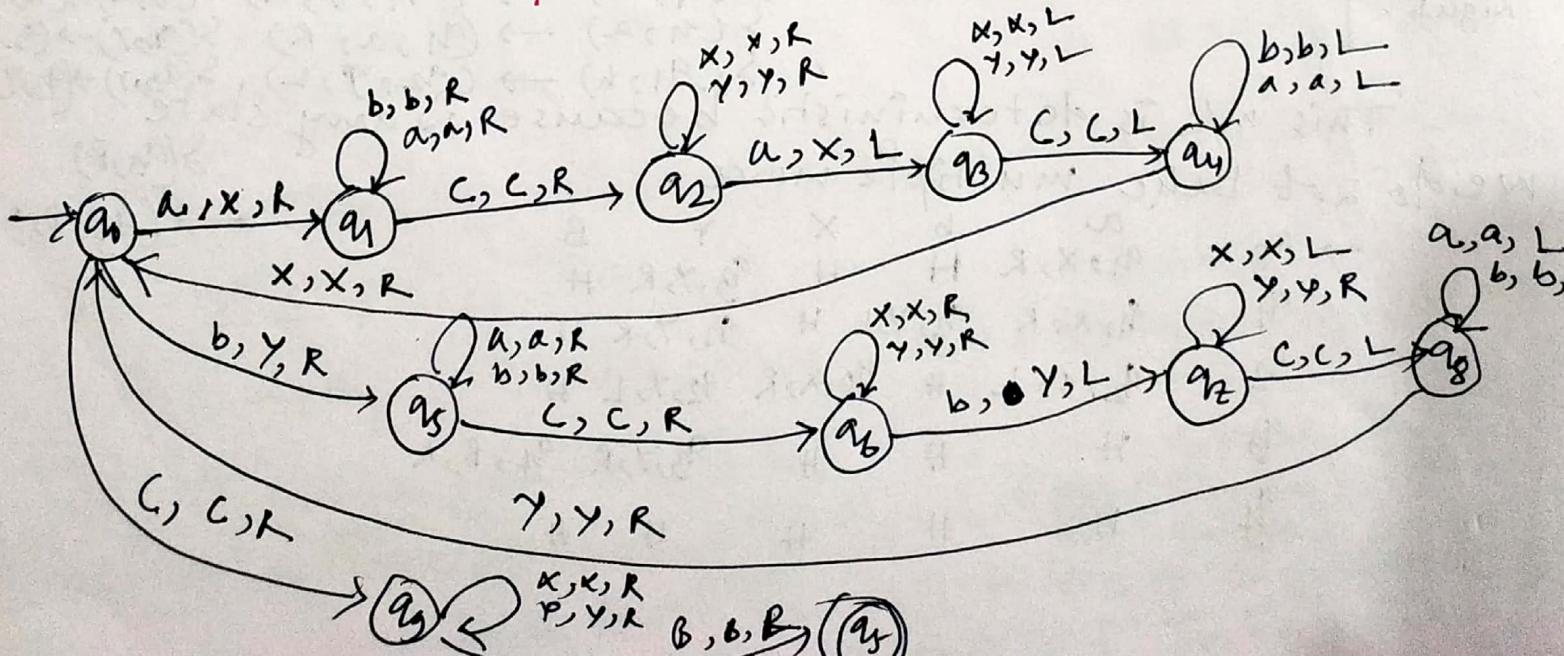
Q:2 $L = \{ a^z b^z c^z \mid z \geq 1 \}$





Q:3 $L = \{ wcw \mid w \in (a, b)^* \}$

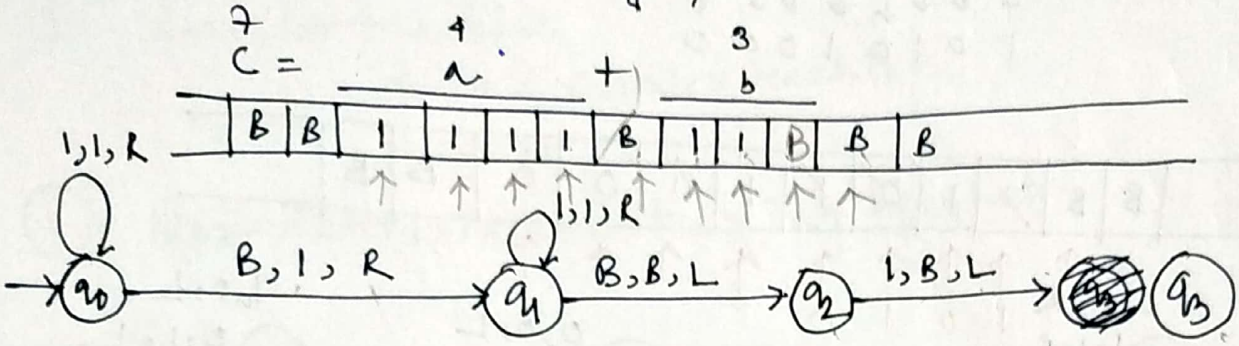




• COMPUTABLE FUNCTIONS:

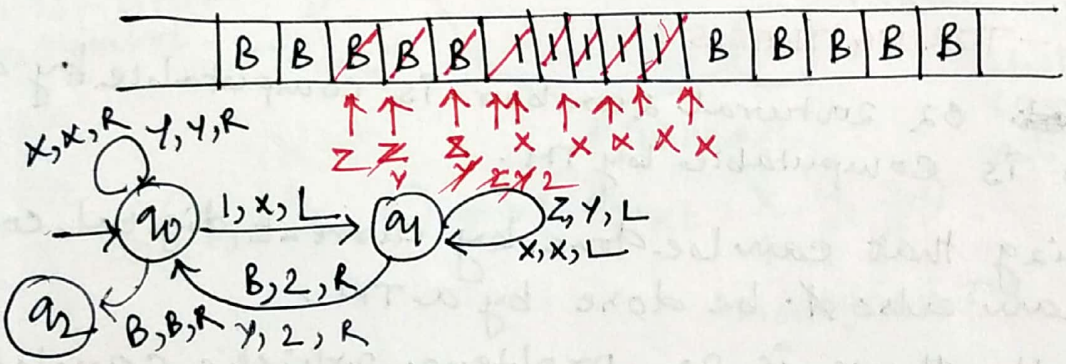
Q: TURING MACHINE AS ADDER. TURING MACHINE AS TRANSDUCER.

A: TM works on Unary System.



Q: TM AS 1's UNARY TO BINAY CONVERTER.

Answer: Let us assume we convert 1 to X. 0 to Y & 1 to 2.



Result will be ZYZ

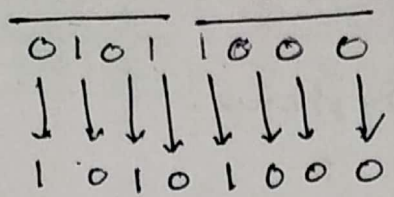
	UNARY	BINARY
1)	1	1
2)	11	10
3)	111	11
4)	1111	100
5)	11111	101

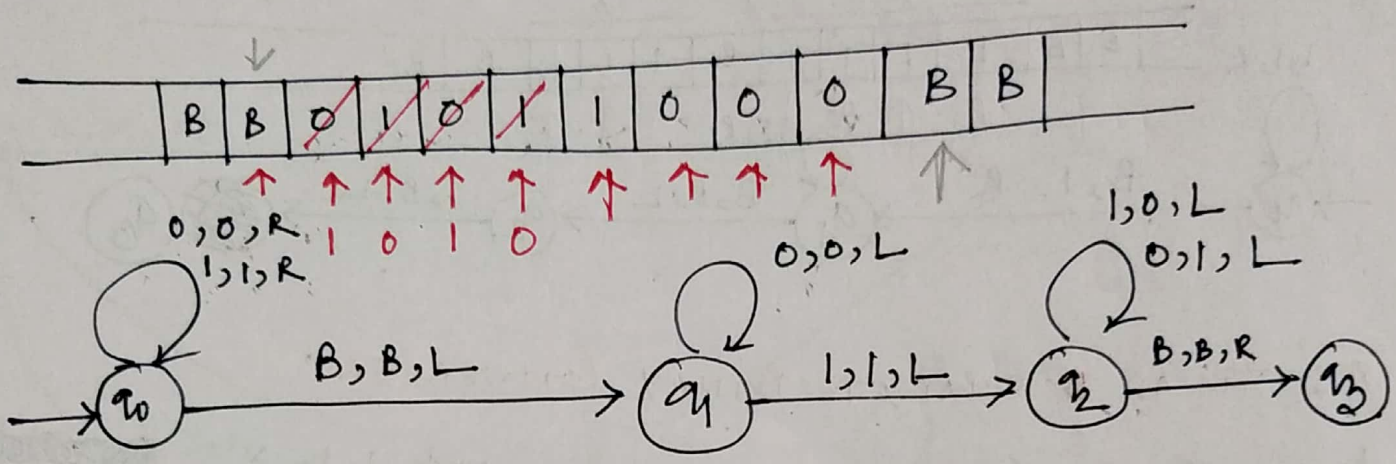
read 1 from unary & add 1 to binary.

(6)

Q: TURING MACHINE AS 2'S COMPLEMENT :

ANSWER:





(5) OBJECTIVES NO: 5

- CHURCH – TURING THESIS :

A function or a natural number is computable by an algo iff it is computable by TM.

→ anything that can be done by current digital computer can also be done by a TM.

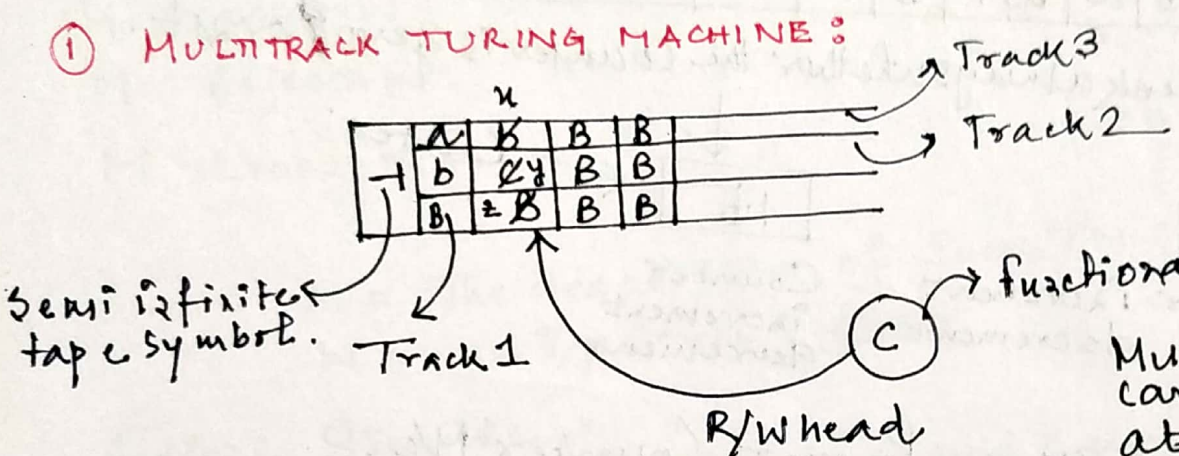
→ Currently there is no problem which can be solved by a digital computer and can not be solved by a Turing Machine.

→ Many mathematical models are suggested but no one of them is more powerfull than TM.

- Types of Turing Machine & Counter Machine :
(Proofs not required)

1. Multitrack turing Machine.
2. Multitape turing Machine
3. 2-stack turing Machine.
4. Counter Machine
   - 2 Counter Machine
   - 4 " "
5. Non-deterministic turing Machine.

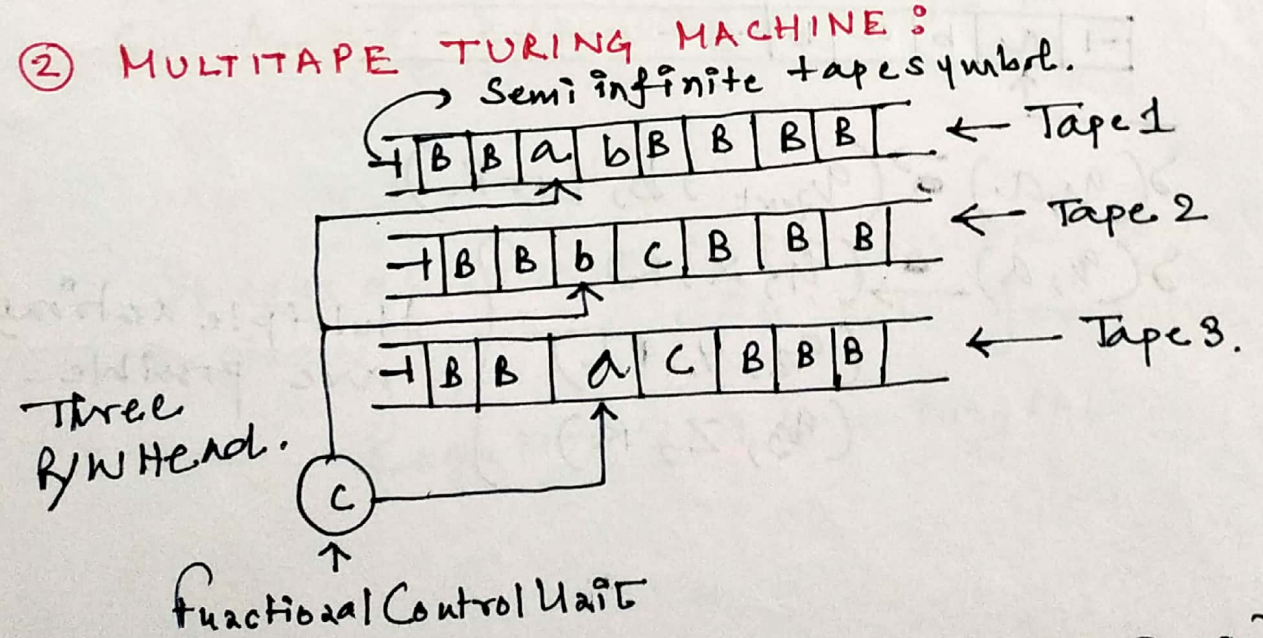
① MULTITRACK TURING MACHINE :



$$\delta(q, b, C, B) \Rightarrow (q_{next}, x, y, z, B)$$

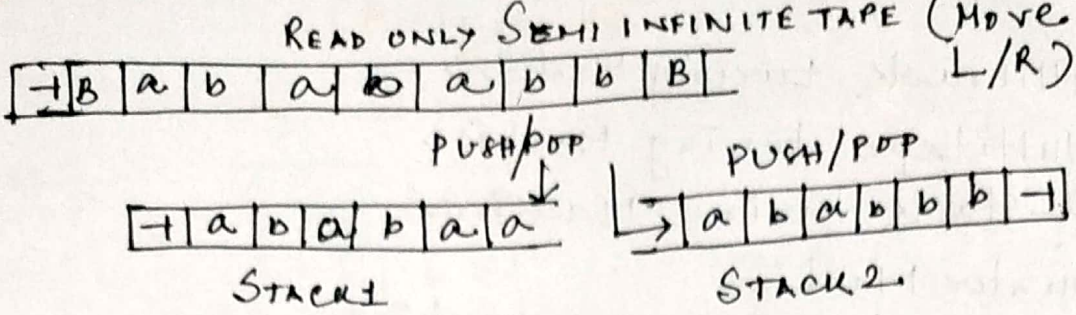
Modified TM.

② MULTITAPE TURING MACHINE :



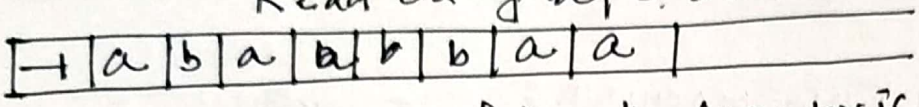
$$\delta(q, a, b, C) \longrightarrow (q_{next}, \underset{L, R, L}{x, y, z}, B)$$

(3) 2 – STACK MACHINE:

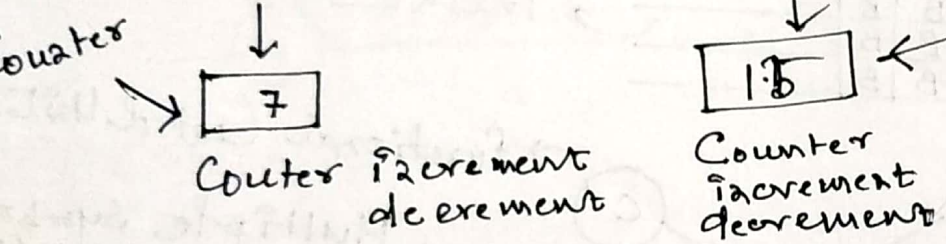


(4) COUNTER MACHINE:

Read Only Tape. Semi infinite tape.

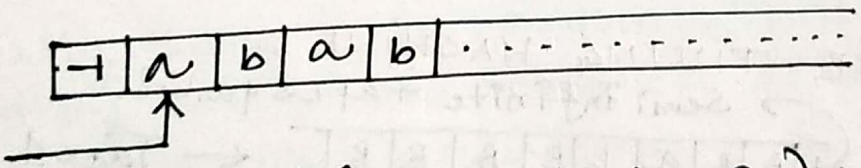


We check always whether the counter is zero or not.



Power of TM ≡ Power of Counter M/c.

(5) NON-DETERMINISTIC TURING MACHINE:



$\delta(q,a) \rightarrow (q_{next}, b, L \text{ or } R)$

$\delta(q,a) \rightarrow (q_1, X, R)$
(q_2, Y, L)
(q_3, Z, R)

} Multiple actions are possible.

• UNIVERSAL TURING MACHINE:

Let us consider a language,

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a Turing Machine and } M \text{ accepts } w \}$

is Turing Recognizable.

Q: Given, the description of a TM and some input, can we determine whether the m/c accepts it?

— Just simulate or Run the TM on the i/p.

M Accepts w : Our Algorithm will Halt & Accept.
M Rejects w : " " " " & Reject.
M Loops on w : " " " " not Halt.

Input: M = The description of Some TM.
w = an i/p string

Action: Simulate M
Behave just like M would (may accept, reject or loop)

The Universal TM is a recognizer (but not a decider) for $A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM \& } M \text{ accepts } w \}$

• HALTING PROBLEM:

- It asks question " is it possible to tell whether a given machine will halt for some given i/p "
- Proof: Page NO: 289 C.K. NAGPAL.