Name :- Prof. Anand Ghosu
College : PVGCOE, Nasik
Class : TE COMPUTER
Subject : TOC

4. TURING MACHINE

* Introduction to Turing Machine :-

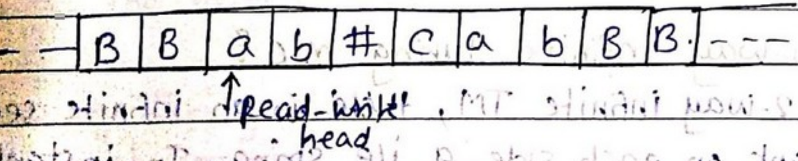
→ " Turing m/c is mathematical model which consist of an infinite length tape divided into cells on which input is given. It consist of head which reads the i/p tape."

- A state register stores the state turing m/c. After reading an i/p symbol, it is replaced with another symbol, it's internal state is changed and it moves from one cell to the right or left.
- If TM m/c reaches to final state, the i/p string is accepted otherwise rejected.

A TM m/c can be described as 7-tuple $(Q, X, \Sigma, \delta, q_0, B, F)$

where, Q is finite set of state.
X is tape alphabet
Σ is i/p alphabet
δ is transition function
q_0 is initial state.
B is Blank symbol
F is set of final state.

e.g.



Example of Turing Machine.

- Turing m/c is more powerful than PDA.
- Turing m/c is capable of performing computation on i/p and producing a new result.

* Applications g Turing Machine :

→ 1) To read or write infinite tape.

2) to solve problem in computer science & testing limit g Computation.

3) it is used to simulate other turing machine.

4) Turing m/c is used to reverse string g any character.

5) it is used in algorithmic information theory

6) Turing m/c is used for high performance Computing, m/c learning, S/w engg and computer network.

7) Turing m/c is used to perform computation for computer system.

8) ~~It~~ Turing m/c is used ~~to~~ in theory g computation.

* Different way g Extension g TM :

→ In std TM, the tape is semi-infinite. It is bounded on the left and unbounded on the right side.

Some of the Extension g TM :

1) Tape is g infinite length in both the direction.

2) Multiple heads and single tapes.

3) Multiple tape with each tape having it's own independent head.

4) K-dimensional tape.

5) Non-deterministic turing m/c.

1) Two-way infinite turing m/c :

— In 2-way infinite TM, there is an infinite sequence g blank on each side g i/p string. In instantaneous description, these blocks are never shown.

2) A turing m/c with multiple head -

→ A TM with single tape can have multiple heads. let's consider TM with two head H1 & H2. Each head is capable g performing read/write operation

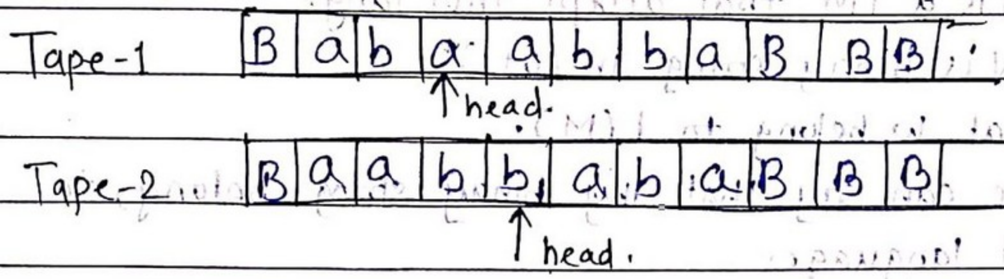
B aabaaba BB

↑H1 ↑H2. ←TM with two head

3] Multi-tape Turing Machine :-

→ - Multi-tape turing m/c has multiple ~~tape~~ tuples with each tape having it's own independent head.

→ Let's consider case g two tape turing m/c as shown in fig.



∴ Two Tape Turing Machine

- The transition behaviour g a two-tape turing m/c can be defined as given below:

$$\delta(q_1, a_1, a_2) = (q_2, (S_1, M_1), (S_2, M_2))$$

where,

q_1 is current state.

q_2 is next state.

a_1 symbol head on tape 1

a_2 symbol under head on tape 2.

$S1$ is symbol written in current cell on tape 1.

$S2$ is symbol written In current cell on tape 2.

$M1$ is the movement (L, R, N) g head on tape 1

$M2$ is the movement (L, R, N) g head on tape 2.

* limitation g Turing Machine :-

→ 1) Computational Complexity theory.

- limitation is that they do not model strength g particular arrangement

2) Concurrency :-

limitation is that they do not model Concurrency well.

3) There are alway halting Concurrent system with no i/p.

4) If two CFGs are given G1 & G2 then $L(G1) \cap L(G2) = \phi$ is undecidable

5) Recursively Enumerable lang. and the halting problem.

6) TM m/c is weak to describe the property the internet, evolution or robotics bcz It is closed model.

* Universal Turing Machine :-

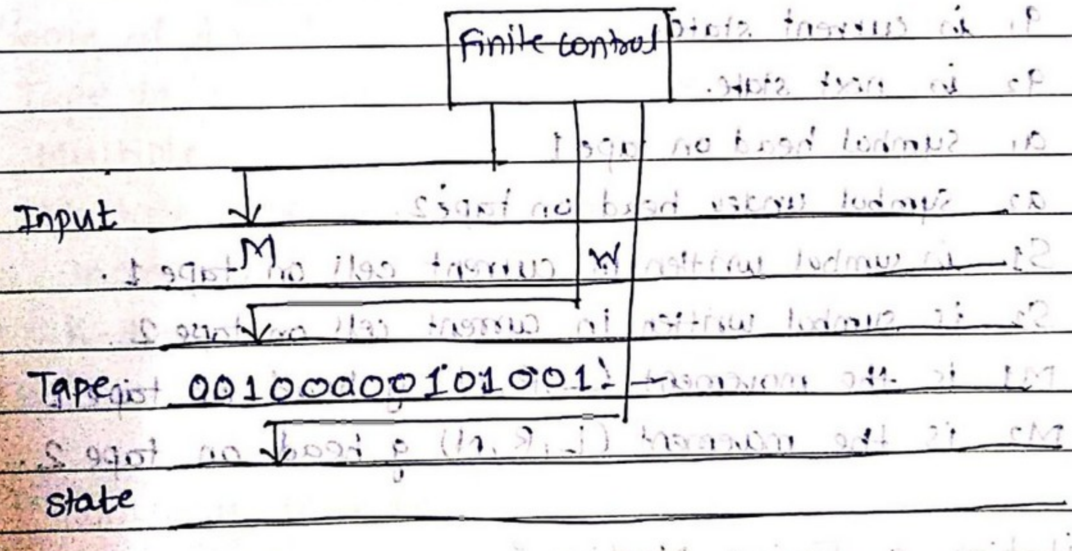
1) The universal lang. Lu is the set g binary string which can be modeled by turing m/c.

2) The universal lang is can be represented by pair (M, w) where M is a TM that accept this lang.
W is binay string in $(0+1)^*$
such that w belong to L(M).
Thus, we can say that any binay string belongs to universal language.

3) The universal lang. can be represented by
Lu = L(U) where U is universal Turing m/c.

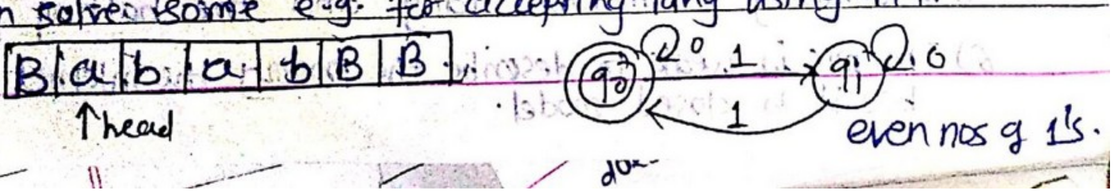
4) In fact, U is binay string. This binay string represent varias codes g many turing machine.

5) Thus, the universal turing m/c is a turing m/c. which accepts many turing m/c.



* Language acceptabiliy by turing machine :-

→ TM accepts all lang. eventhough there are recursively enumerable.
- Recursive means repeating the same set g rule for any nos g times.
& enumerable means list g elements.
- TM also accepts computable function, such as Addition, multiplication, Substruction, division, power fn, square fn & logarathmic function.
- we can solve some e.g. for accepting lang using TM.

e.g.


even nos g 1's.

* TM's Halting Problem :-

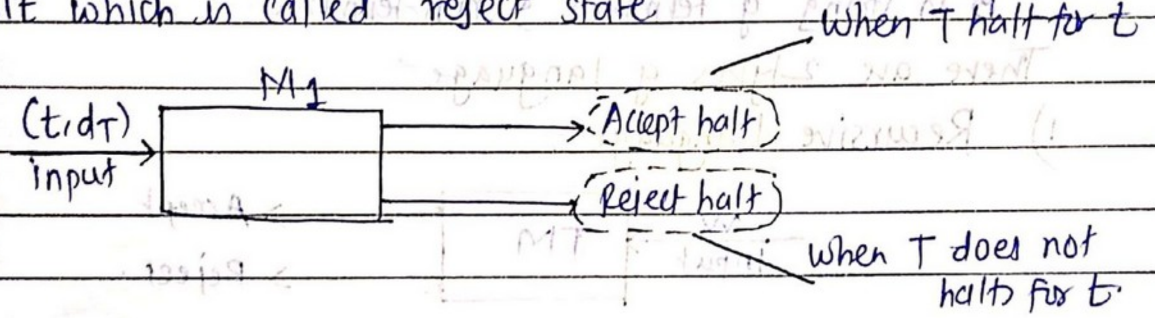
" Halting Problem is unsolvable "

proof :-

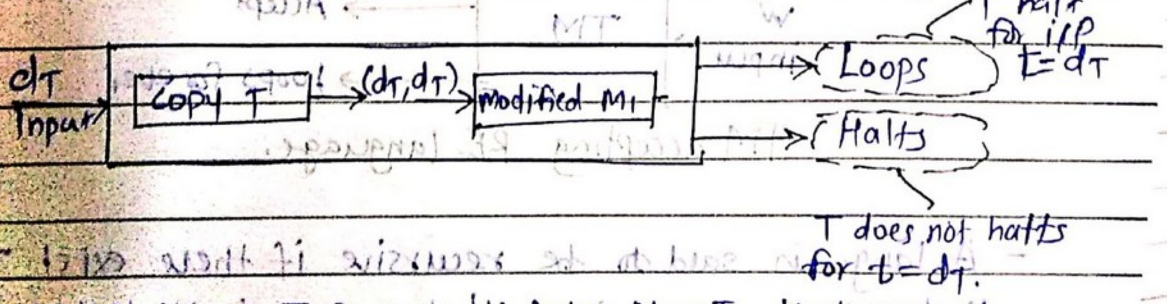
- Let there exist, a TM M1 which decides whether or not any computation by a TM T will ever halt when a description dt g T and tape t g T is given.

Then for every i/P (t, dT) to M1 if T halt for i/P t.

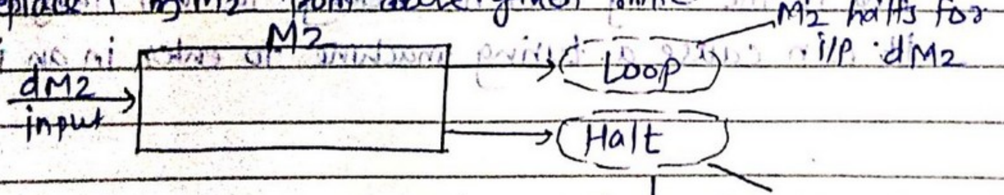
- M1 also halts which is called accept halt.
- similarly if T does not halt for i/P t then the M1 will halt which is called reject state



when we will consider another TM M2 which takes i/P dT. ~~It~~ 1st copy dT and duplicate dT on it's tape and then this duplicate tape infor? is given as i/P m/c M1. But M1 is modified ~~m/c~~ m/c with the modification that whenever M1 is supposed to reach as accept halt, M2 loop forever. Hence behaviour g M2 is given. It loops if T halts for i/P t=dT and halts if it does not halt for T=dT. The T is any arbitrary m/c.



As M2 itself is one TM, we will take M2 = T, that means we will replace T by M2 from above given m/c.



This is contradiction, that means M1 which can tell whether any other m/c TM will halt on particular i/P does not exits. Hence halting problem is solvable.

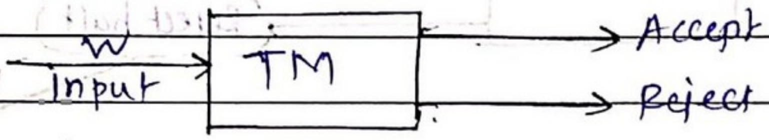
* TM and Type 0 Grammers.

→ TM ~~is~~ can accept Type 0 grammar.

- The type 0 grammar is said to be unrestricted grammar
  e.g. g restricted lang are almost all natural lang.
- The clas g lang. accepted by type 0 grammare are called recursively enumerable language.
- The prod^n can be g form $\alpha \rightarrow \beta$
  where ~~β~~ α is string g terminal and non-terminal with atleast one non-terminal and α cannot be null.
  β is string g terminal & non-terminal.

There are 2 types g language :-

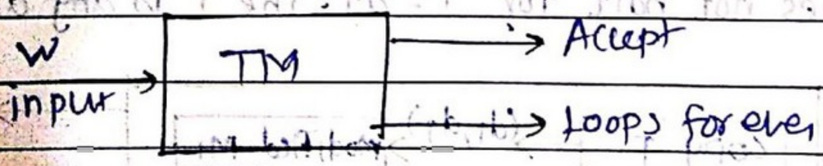
1) Recursive Language :-



TM Accepting Recursive Language

A lang is said to be recursive if there exist a TM that accept every string g lang. and every string is rejected if it is not belonging to that lang.

2) Recursively Enumerable Language :-



TM accepting RE language.

- A lang is said to be recursive if there exist TM that accepts every string belonging to that language. And if the string does not belong to that language then it can cause a turing machine to enter in an infinite loop.