MODULE - IV

TURING MACHINES.

1) → A turing machine is a mathematical model of algorithm or computation, which is widely accepted.

Church - Turing thesis:

→ "Any algorithmic procedure that can be carried out by human beings or computers can be carried out by a Turing machine."

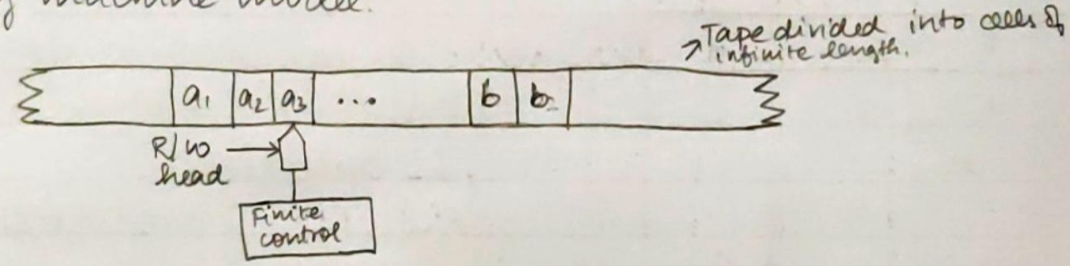
→ It's considered the ideal theoritical model for computers.

→ For formalizing computability, Turing assumed that while computing, a person writes symbols on a one-dimentional (1-D) tape which is divided into cells.

→ One scans one symbol at a time and usually performs one of the following three operations:-

* writing a new symbol into the cell being scanned
* moving to the cell left of the present cell
* moving to the cell right of the present cell.

Turing machine model.



→ It consists of an input tape that's divided into cells of infinite length.

→ A R/W head reads the input tape.

→ A state register stores the state of the Turing machine

→ In one move, the machine examines the present symbol under the R/W head & the present state of an automaton to determine:-

* a new symbol to be written on the cell under R/W head
* a motion of the R/W head along the tape - it moves one cell to the left or right.
* the next state of the automaton
* whether to halt or not

→ If the TM reaches the final state, the input string is accepted, otherwise its rejected.

Formal Definition of Turing Machine

A turing machine is a 7-tuple namely $(Q, \Sigma, \Gamma, \delta, q_0, b, F)$
where, Q - finite non-empty set of states.

Σ - finite non-empty set of input symbols (subset Γ & $b \notin \Sigma$)

Γ - finite non-empty set of tape symbols

δ - transition mapping function (q, x) onto (q', y, D)

D denotes direction of R/W head. $D = L / R$

$q_0 \in Q$ is the initial state

b - blank $b \notin \Sigma$

$F \subseteq Q$ - set of final states.

Representation of Turing Machine

A turing machine can be described by employing:-

i) Transition diagram / graph
ii) Instantaneous descriptions using move relations
iii) Transition table

2) Variants of Turing Machine

Several variants of TM are:-

i) Multiple track
ii) Shift over TM
iii) Non-deterministic
iv) Two way TM
v) Multitape TM
vi) Multi-dimensional TM
vii) Composite TM
viii) Universal

Multi-tape Turing Machine

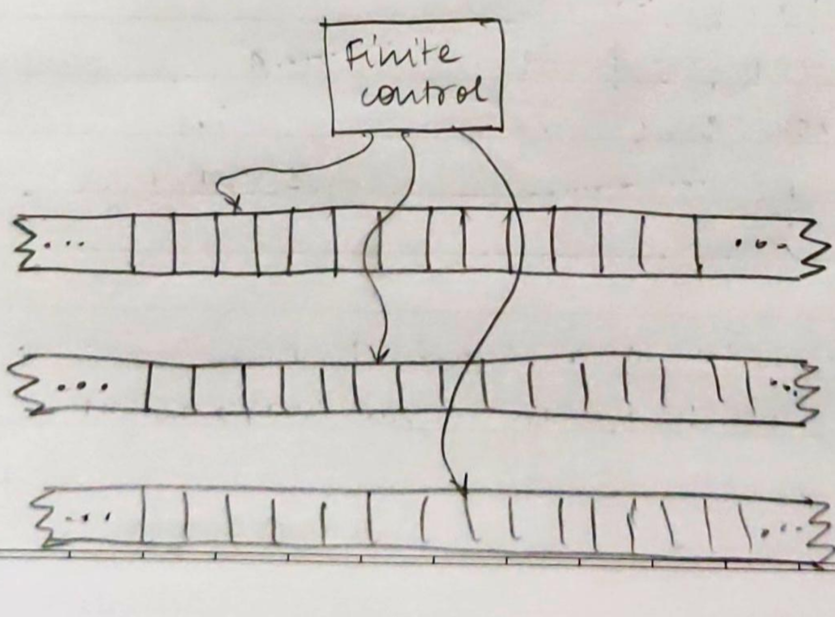
→ It's like a regular TM but with multiple tapes, each with its own R/W head.

→ There are k no. of tapes, each divided into cells.

→ First tape holds the input string w.

→ Initially all other cells are filled with blanks.

→ Initially the head of the first tape is at the left end of the input string w whereas the other heads can be placed anywhere



Formal definition

A k-tape TM is a 6 tuple namely $M(Q, \Gamma, q_0, b, F, \delta)$

where, Q - finite non-empty set of states

Γ - finite non-empty set of tape symbols

q_0 - initial state $q \subseteq Q$

b - blank symbol

F - finite set of final states

δ - transition mapping function

$Q \times \gamma^k$ into $(Q \times \gamma^k \times \{L, R, S\}^k)$

In a typical move:

i) M enters a new state

ii) On each tape, a new symbol is written in the cell under each head.

iii) Each tape head moves to the left or right or remain stationary. They move independently.

Non-deterministic Turing Machine

A non-deterministic turing machine M is a 7 tuple namely $(Q, \Sigma, \Gamma, \delta, q_0, b, F)$

where, Q - non-empty finite set of states

Σ - finite set of input symbols. subset of $b \times \Gamma$

Γ - non-empty finite set of tape symbols

$q_0 \subset Q$ - initial state

b - blank symbol $b \notin \Sigma$

$F \subseteq Q$ - finite set of final states

δ - transition mapping function / partial function

(q, x) onto (q', y, D)

D - direction of movement of R/W head - L, R, S

If $q \in Q$ & $x \in \Gamma$ and

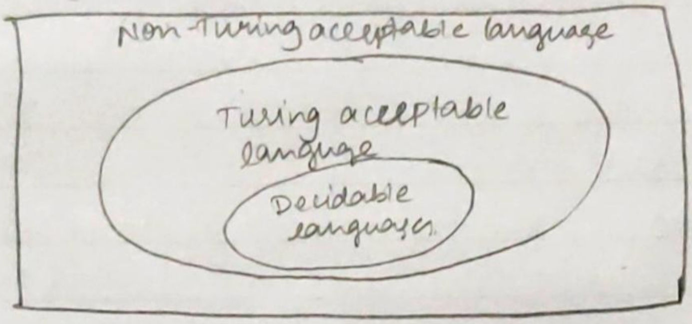
$\delta(q, x) = \{(q_1, y_1, D_1), (q_2, y_2, D_2), (q_3, y_3, D_3) \dots\}$

then NTM can choose any of the actions defined by (q_i, y_i, D_i) where $i = 1, 2, 3 \dots n$

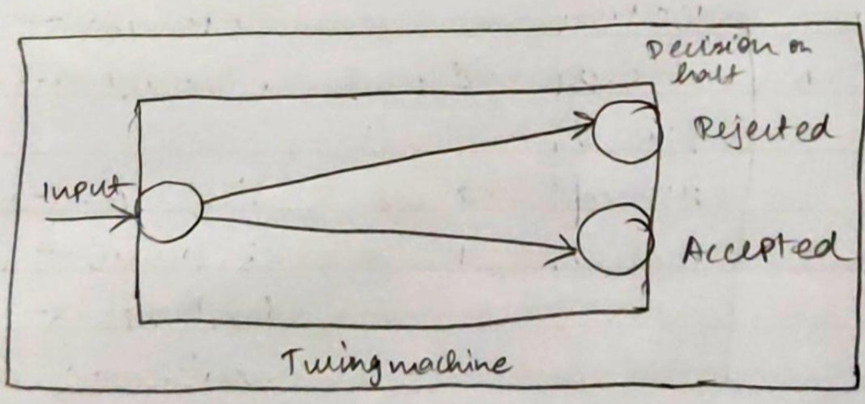
3) Language acceptability

A language is called decidable or recursive if there's a TM that accepts and halts on every input string w.

Every decidable language is Turing acceptable



For each decidable language, the TM halts either at accept or reject state.



Consider a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, b, F)$ and a string $w \in \Sigma^*$ is said to be accepted by M if $q_0 w \vdash \alpha_1 p \alpha_2$.

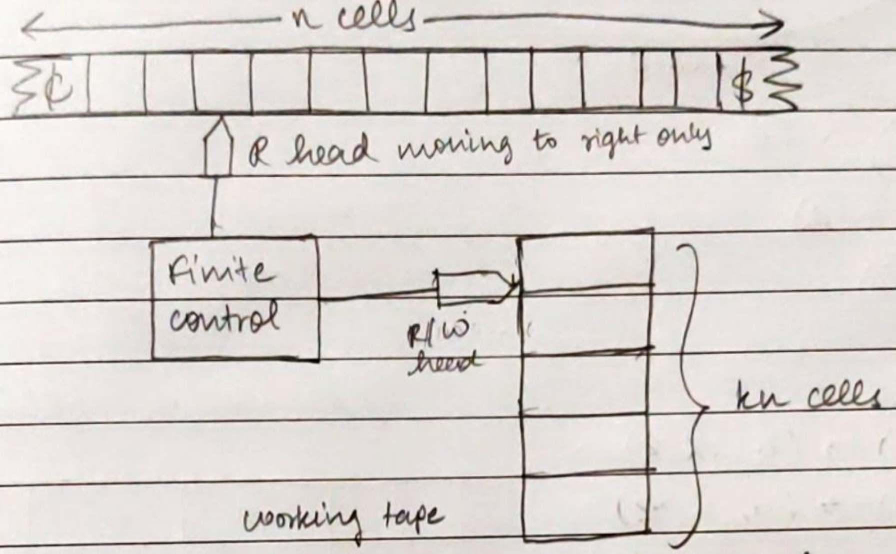
for, some $p \in F$ & $\alpha_1, \alpha_2 \in \Gamma^*$

A) Model of Linear Bound Automaton

→ The set of context-sensitive languages are accepted by the LBA

→ Its infinite storage is restricted in size but not in accessibility when compared to TM model.

→ It's called LBA because a linear function is used to restrict the length of the tape.



→ A LBA is a NDTM with a single tape whose length is not infinite but restricted by some linear function of the length of the input string.

→ LBA is a 9 tuple information $(Q, \Sigma, \Gamma, \delta, q_0, b, ¢, \$, F)$

¢ - left end marker of I/P tape to prevent R/W head from getting off the left side of the tape.

\$ - right end marker of I/P tape which prevents R/W head from getting off the right side of the tape.

¢ - rightmost cell of I/P tape

\$ - leftmost cell of I/P tape.

Teacher's Signature ______________