Unit-V

TURING MACHINE

Turing machine is a mathematical model that is designed to accept the Strings of languages namely recursive enumerable languages.

[These languages are Decidable, Undecidable, Complete, P-Complete and so on]. The Turing machine can be represented in a 7-tuple

$$M : < Q, \Sigma, \delta, T, q_0, F, B >$$

where Q - finite set of States

Σ - Input Symbols over alphabet

T - Infinite Size Input tape

q_0 - Initial State | $q_0 \in Q$

F - Final State | $F \in Q$

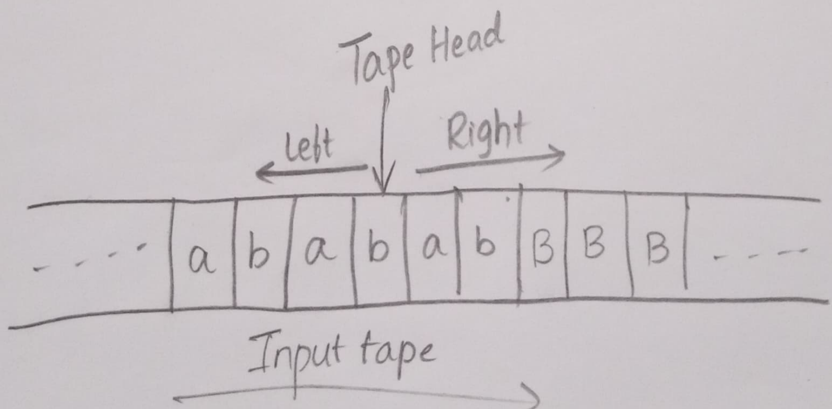
B - Blank Symbol. where $B \notin \Sigma$

$$\delta : (Q \times \Sigma) \longrightarrow (Q \times \Sigma \times R/L)$$

where $R \rightarrow$ Right move

$L \rightarrow$ left move

Turing Machine Model:-



1) The Read/write tape head of the turing machine can have the read operation on input symbol and moving one position to right.

2) The Read/write tape head of the turing machine can have the read operation on input symbol & moving one position to left.

3) The Read/write tape head reads a blank symbol & may write any symbol including blank by moving one position to left/right.

* Language accepted by turing machine can be represented by L(M)

$L(M) = \{ w / w \in \Sigma^* \text{ and } \underbrace{q_0 w \vdash^* \alpha_1 p \alpha_2}_{\text{Instantaneous description}} \}$

where q_0 - Initial State

w - Input String

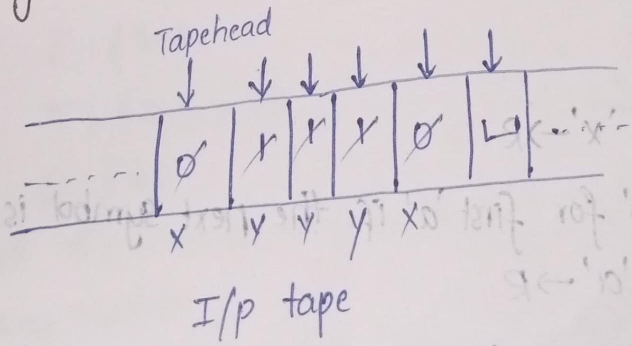
Σ^* - Input alphabet(s)

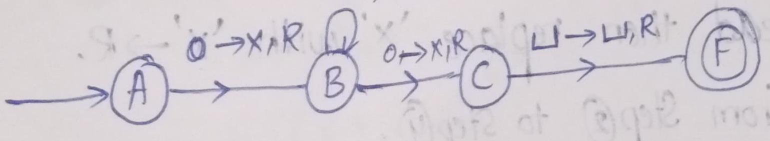
$\alpha_1 \alpha_2$ - T^* (Input Tape)

$p \in F$ (Final State).

Eg: Design a turing machine for $L = \{01^*0\}$.

Sol:-





$\delta(A, 0) = (B, X, R)$

$\delta(B, 1) = (B, Y, R)$

$\delta(B, 0) = (C, X, R)$

$\delta(C, \sqcup) = (E, \sqcup, R)$

$M: \langle Q, \Sigma, \delta, q_0, F, T, B \rangle$

$Q - \{A, B, C, F\}$

$\Sigma - \{0, 1\}$

$q_0 - \{A\}$

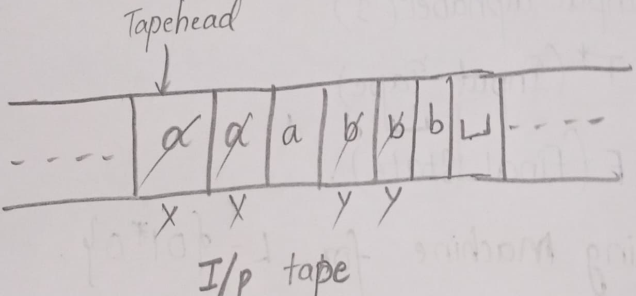
$T - \{0, 1, X, Y\}$

$F - \{F\}$.

$B - \{\sqcup\}$

Eg:- Design Turing machine for $L = \{a^n b^n / n \geq 1\}$

Sol:-



Algorithm:-

1) Starts with 'a'.

2) Replace 'a' with 'x' → R

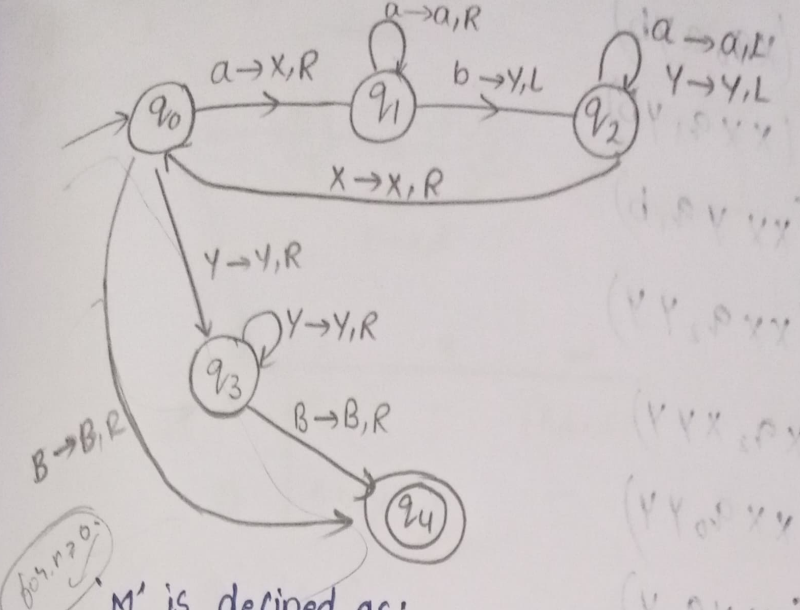
3) Read the first 'b' for first 'a' if the next symbol is 'a' replace a with 'a' → R

4) If 'b' for 'a' is read then replace 'b' with 'y' → L.

5) If 'x' is read then replace 'x' with 'x' → R.

6) Repeat from Step ② to Step ④.

7) If 'Y' is read replace 'Y' with 'Y' → L.



M' is defined as:

Q : $\{q_0, q_1, q_2, q_3, q_4\}$

Σ : $\{a, b\}$

δ	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	–	–	–	–	–

q_0 : $\{q_0\}$

F : $\{q_4\}$

T : $\{a, b, X, Y\}$

B : $\{B\}$

Eg: Check the acceptance of string $w = aabb$ for $L = \{a^n b^n / n \geq 1\}$ in Turing machine.

Sol: $(q_0, aabb) \vdash (X q_1 abb)$

$\vdash (Xa q_1 bb)$

$\vdash (X q_2 aYb)$

$\vdash (q_2 XaYb)$

$\vdash (x q_0 a y b)$

$\vdash (x x q_1 y b)$

$\vdash (x x y q_1 b)$

$\vdash (x x q_2 y y)$

$\vdash (x q_2 x y y)$

$\vdash (x x q_0 y y)$

$\vdash (x x y q_3 y)$

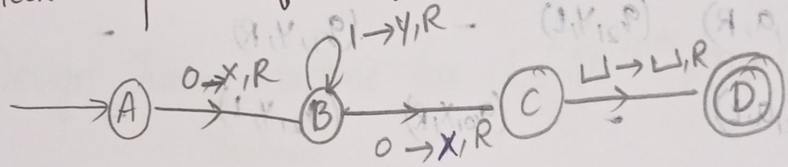
$\vdash (x x y y q_3 B)$

$\vdash (x x y y q_4) \in F$

$\therefore$ w = aabb is ~~accepting~~ accepted by turing machine.

Eg: Check acceptance of string w = 01110 for $L = \{0 1^* 0\}$ in TM.

Sol:



$(A, 01110) \vdash (XB1110)$

$\vdash (XYB110)$

$\vdash (XYYB10)$

$\vdash (XYY\cancel{B}YB0)$

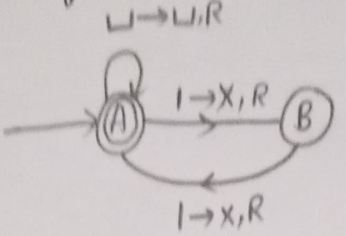
$\vdash (XYYYX\sqcup)$

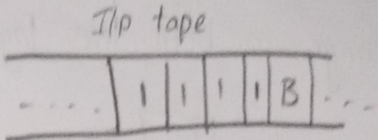
$\vdash (XYYYXD) \in F$

$\therefore$ w = 01110 is accepted by TM.

Eg: Design TM for $L = \{x / x \in 1^*$, where length of x is even$\}$

Sol:





δ	1	X	⊔
→* A	(B,1,R)	–	(A,⊔,R)
B	(A,1,R)	–	–

Q : {A, B}

Σ : {1}

q_0 : {A}

F : {A}

T : {1, X}

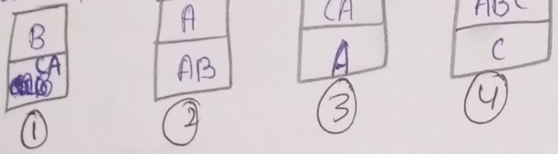
B : {⊔}

* Post correspondance Problem (PCP):-

PCP is post correspondance problem designed by Elen Post, a mathematician to give solution for a set of dominos by neutralizing the strings in upper domino as well as lower domino

Dominos/Tiles

Eg:



Sol:

Algorithm:-

1) For a given set of Dominos D : {D1, D2, ..., Dn} arrange them in an order by the following rules:

Rule 1:-

Choose the domino which has similar character or string in both the level, place it in Po.

Rule 2: Arrange the other dominos post to Po by analysing the lower domaino string (or) upper dominos string in Po.

2) After previous step the domino D should have the similar strings in both the layers.

Sol:

$\frac{w_1}{w_2}$ $w_1 \equiv w_2$

I)

②	①	③	②	④	
A	B	CA	A	ABC	→ ABCAAABC
AB	CA	A	AB	C	→ ABCAAABC

∴ Using pcp the domain sequence is 21324 & the string obta is 21324.

Eg:

	A	B
①	1	111
②	10111	10
③	10	0

Give the pcp sequence for dominos?

Sol:

	②	①	①	③
1011111110	10111	1	1	10
1011111110	10	111	111	0

Eg:

	A	B
①	10	101
②	011	11
③	101	011

Sol:

①	③	③	
10	101	101	. . . .
101	011	011	

The Sequence is 133, - - - -

This is an example of non-halting problem which is a special class of pcp.

Universal TM:-

The Universal TM (Tu) is an mathematical ~~capabili~~ model that has the capability to design the automata for all the Class of languages.

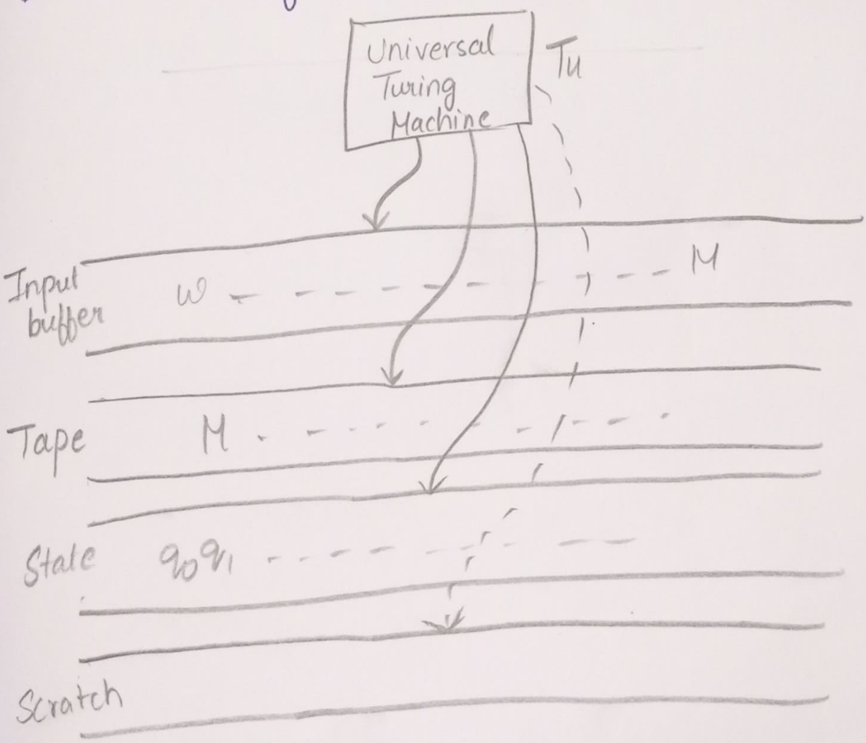
It has 3 major component-

1) ~~Buffer~~ Input buffer
2) Tape of Size m where m is of arbitary length.
3) State buffer

* Tu follows the 2 following properties:-

1) The language is accepted with the Strings of an arbitary length m only if that String is processed through the i/p ~~tape~~ buffer.
2) The String, if accepted produces an output over the 3 Components in the universal Shape.

* The model of universal TM is shown below:-



Types of Turing Machines: