UNIT - IV TURING MACHINE :

Definitions of Turing Machine - Models - Computable Languages & Function - Techniques For Turing Machine - Construction - Multi head & Multi Tape Turing Machine - The Halting problem - Partial solvability - Problems about Turing Machine - Chomskian Hierarchy of Languages.

INTRODUCTION - TURING MACHINE (TM):

- During the year 1936, Alan Turing introduced a new mathematical model called Turing Machine
- Turing Machine is an abstract machine (or) mathematical model to represent a real computer.
- Turing Machine is a tool, for studying the computability of mathematical function.
- Turing Hypothesis believed that a function is computable if and only if it can be computed by turing machine.
- Turing machine can solve any problem that a modern computer can solve.
- Turing machine is used to define the language and to compile the integer functions.
- Turing machine accepts recursive language or recursive enumerable language.
- Turing machine differs from PDA and FA.
- FA has finite memory and PDA has infinite memory and access in LIFO order
- But TM has both infinite memory and no restriction in accessing the input.

• TM has infinite tape memory & the tape head can move either left or right to access the input

MODEL OF TURING MACHINE:

Turing Machine has

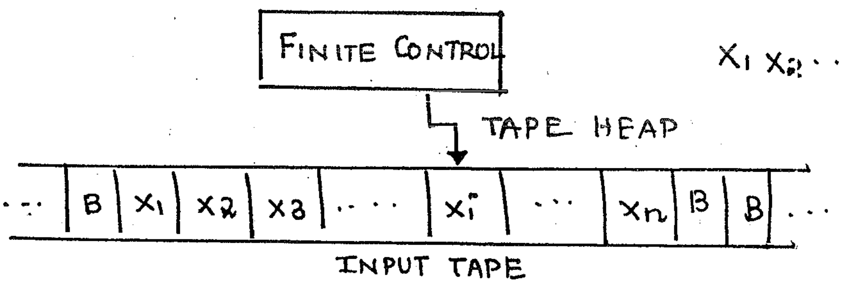
1. Finite control - which contains set of states and transitions between the states.

2. Turing Machine has an input tape (i.e) divided into cells & each cell can hold any one of the finite number of symbols over alphabet.

• It has a tape head that scans one cell on the input tape at a time.

B → BLANK SYMBOL

$X_1 X_2 \cdots X_n$ → INPUT SYMBOL



WORKING OF TURING MACHINE:

• The Turing Machine, the input initially consists of a finite length string of symbols chosen from the I/p alphabet & the i/p is placed on the input tape.

• All other tape cells extending infinitely into the left & right of the input tape contains the spirial symbol called "Blank symbol"

• The tape head is positioned at one of the tape cells for scanning the input symbol from the input tape.

• Initially the tape head points at the left most cell of the input tape

FORMAL NOTATION | DEFINITION OF A TURING MACHINE:

Turing Machine has 7-tuple:

$\boxed{M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)}$ where

$Q \rightarrow$ The Finite set of states of the Finite control.

$\Sigma \rightarrow$ The Finite set of input symbols.

$\Gamma \rightarrow$ The complete set of tape symbols, Σ is always a subset of Γ.
($\Sigma \subseteq \Gamma$)

$\delta \rightarrow$ The Transition Function $\boxed{\delta(q, x) = (P, Y, D)}$.

where $q \rightarrow$ a state, $x \rightarrow$ a tape symbol, $P \rightarrow$ new state | same state in Q, $Y \rightarrow$ symbol in Γ, written in the cell being scanned, replacing whatever symbol was there.

$D \rightarrow$ Direction, either Left or Right and telling us the direction in which the head moves.

$q_0 \rightarrow$ The start state, a member in Q, in which the Finite control is found initially.

$B \rightarrow$ The blank symbol. This symbol is in Γ but not in Σ

$F \rightarrow$ The set of Final / Accepting states i.e $F \subseteq Q$.

PROCESSING OF MOVE IN A TURING MACHINE:

- The single move of a Turing Machine depends on the current state of Finite control and the tape symbol present in the input tape.
- The Following changes happen in one ~~one~~ move of a TM.

$\rightarrow$ Changes the state after consuming an i/p symbol. It may also be in the same state or transfer to any new state

$\rightarrow$ The Tape symbol to be replaced for the scanned i/p tape symbol

→ Deciding the move of the tape head to left or right of i/p tape

→ Whether to halt the TM or not.

INSTANTANEOUS DESCRIPTIONS OF A TM: (ID).

- The execution sequence of an i/p string is represented by the ID of a TM.
- Each move of TM is represented by the ID.
- ID of a TM describes the current configuration and it can be of following types

✓	Accepting configuration
✓	Rejecting Configuration

- A move of TM can be represented as a pair of ID separated by the symbol $\vdash$:
- Each move is represented by $\alpha_1 q \alpha_2$ where

  α_1 & α_2 are the strings from T^* and q is the state of TM

- The move can be of single move or zero or more moves as

  $\vdash_m$ = single move $\vdash_m^*$ = zero/more moves.

Let us use the string

$x_1 x_2 \ldots x_{i-1}\, q\, x_i\, x_{i+1} \ldots x_n$ to represent ID.

where 1. q is the state of TM.

2. The Tape head is scanning the ith symbol from left.

3. $x_1 x_2 \ldots x_n$ is the portion of the tape between the leftmost & rightmost non-Blank.

If the transition function of TM is

CASE 1: $\boxed{\delta(q, x_i) = (P, Y, L)}$

i.e. the next move is leftward. Then

$$x_1 x_2 \ldots x_{i-1}\, q\, x_i\, x_{i+1} \ldots x_n \vdash_m x_1\, x_2 \ldots x_{i-2}\, P\, x_{i-1}\, Y\, x_{i+1} \ldots x_n$$

NOTE: This move reflects the change to state P and the fact that the tape head is now positioned at cell $i-1$.

There are 2 important exceptions

1. If $i=1$, then M moves to the blank to the left of $x_1 \cdots x_n$. In that case, $x_1 x_2 \cdots x_{i-1}\, q\, x_i\, x_{i+1} \cdots x_n \vdash_m PBYx_2 \cdots x_n$.
2. If $i=n$, then the symbol B written over x_n joins the infinite sequence of trailing blanks and doesn't appear in next ID.

$$x_1 x_2 \cdots x_{n-1}\, q\, x_i \cdots x_n \vdash_m x_1 x_2 \cdots x_{n-2}\, P\, x_{n-1} Y$$

CASE 2: $\boxed{\delta(q, x_i) = (P, Y, R)}$ i.e, the next move is Rightward, then $x_1 x_2 \cdots x_{i-1}\, q\, x_i\, x_{i+1} \cdots x_n \vdash_m x_1 x_2 \cdots x_{i-1} Y P x_{i+1}$ · Here the move reflects the fact that the head is $\cdots x_n$ moved to cell etc.

AGAIN THERE ARE 2 IMPORTANT EXCEPTIONS:

1. If $i=n$, then the $i+1^{st}$ cell holds a blank and that cell was not part of the previous ID. Thus we insert,

$$x_1 x_2 \cdots x_{i-1}\, q\, x_i\, x_{i+1} \cdots x_n \vdash_m x_1 x_2 \cdots x_{n-1} Y P B$$

2. If $i=1$ & $Y=B$, then the symbol B written over x_1 joins the infinite sequence of leading blanks & doesn't appear in next ID i.e $x_1 x_2 \cdots q\, x_i \cdots x_n \vdash_m YPx_2 \cdots x_n$.

LANGUAGE OF A TM:

- The set of languages accepted by TM is recursively enumerable language.
- The input string is placed on the input tape & the tape head begins at the leftmost input symbol

If the TM enters an accepting state, then i/p is accepted else the i/p string is not accepted.

The Languages accepted by TM M is defined as L(M) and it is denoted by $L(M) = \{ w \mid w \text{ is in } \Sigma^* \, q_0 w \vdash_{m}^{*} \alpha_1 P \alpha_2$ for some state P in F & α_1 and α_2 is in $\Gamma^* \}$.

HALTING OF TM:

- There is another notion of "acceptance" i.e commonly used for TM: acceptance by halting.
- We say a TM halts if it enters a state q, scanning a i/p tape symbol X, and there is no move in this situation (i.e) $\delta(q, X)$ is undefined.
- TM always halts when it is an accepting state. Unfortunately, it is not always possible to require that a TM halts even if it doesn't accept.
- Those lang with TM that donot halt eventually, regardless of whether or not they accept are called recursive.
- TM that always halt, regardless of whether or not they accept, are a good model of an "algorithm". If an algorithm to solve a given problem exists, then we say the problem is "decidable". So TM's that always halt.

COMPUTABLE LANGUAGE AND FUNCTIONS:

DESIGN A TM FOR COMPUTABLE FUNCTIONS

PROBLEMS:

1. DESIGN a TM to process zero function such that $f(x) = 0$. where x is input.

SOLUTION :

STEP I : IDEA OF CREATION :

The idea to design this TM is that X is the i/p, if $X=5$, then i/p tape contains 5 no. of 1's in the input and steps are as follows.

(i) The TM initially in the state q_0 and if it reads '1' as the left most symbol, it replaces '1' to 'B' & moves to right without changing the state.

(ii) The TM remains in the same state q_0 and replaces all 1's to 'B' until it sees 'B'.

(iii) At state q_0, if it finds 'B' it enters the final state q_1, then halt the TM.

STEP 2 : DIAGRAMMATIC REPRESENTATION.

EXAMPLE $X=3$.

INPUT TAPE

1	1	1	B	..
↑ q_0				

$(q_0, 1) = (q_0, B, R)$.

B	1	1	B
	$q_0 \rightarrow$		

$(q_0, 1) = (q_0, B, R)$

B	B	1	B
		↑ q_0	

$(q_0, 1) = (q_0, B, R)$.

B	B	B	B
			↑ q_0

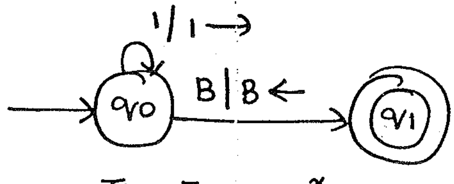
$(q_0, B) = (q_1, B, L)$ halts.

B	B	B	B
	$\leftarrow q_1$		

STEP 3 : TRANSITION TABLE

STATE	1	B
$\rightarrow q_0$	(q_0, B, R)	(q_1, B, L)
$* q_1$	–	–

STEP 4 : TRANSITION DIAGRAM



TM FOR $f(x) = 0$

STEP 5 : TM DEFINITION IS $M = (\{q_0, q_1\}, \{1\}, \{1, B\}, \delta, q_0, B, \{q_1\})$

δ : $\delta(q_0, 1) = (q_0, B, R)$

$\delta(q_0, B) = (q_1, B, L)$

STEP 6 : INSTANTANEOUS DESCRIPTION :

EXAMPLE $x = 2$ $\delta(q_0, 11B) \vdash_m (Bq_0 1B) \vdash_m (BBq_0 B) \vdash_m (Bq_1 BB)$

String accepted and all 1's changed to Blank and the zero function is implemented.

---

2. Design a TM to implement the Function $f(n) = x+1$.

SOLUTION : If $x = 3$ then

Input Tape

1	1	1	B	...

output Tape

1	1	1	1	B	...

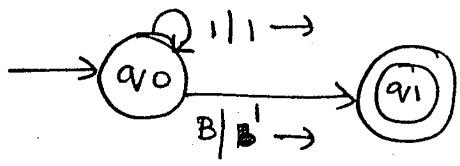
STEP 1 :

1. TM is initially in the state q_0 and it reads '1' in the leftmost input tape.
2. At state q_0 when it reads '1' it remains in the same state, without changing '1' and just move the tape head to right
3. At state q_0, it skips all 1's and searches for the 1st blank symbol B.
4. At state q_0, when it finds 1st 'B', it enters the final state q_1 & changes 'B' to '1'.

STEP 2: TRANSITION TABLE

	1	B
$\rightarrow q_0$	$(q_0,1,R)$	$(q_1,1,R)$
$* q_1$	–	–

STEP 3: TRANSITION DIAGRAM.



TM for $f(x) = x+1$.

STEP 4: TM Definition $M = (\{q_0,q_1\},\{1\},\{1,B\},\delta,q_0,B,\{q_1\})$

δ: $\delta(q_0,1) = (q_0,1,R)$

$\delta(q_0,B) = (q_1,1,R)$.

STEP 4: INSTANTANEOUS DESCRIPTION: $x=3$

$\delta(q_0,111B) \vdash_m (1q_011B) \vdash_m (11q_01B) \vdash_m (111q_0B) \vdash_m (1111q_1B)$

String is accepted.

3. Design a TM to implement the function $f(x) = x+2$.

SOLUTION: EXAMPLE: $X=3$

Input tape: | 1 | 1 | 1 | B | ... |

output tape: | 1 | 1 | 1 | 1 | 1 | B | .. |

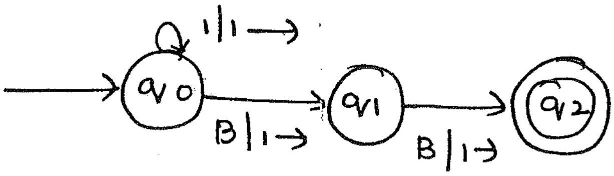
STEP 1:

1. At state q_0, the initial state of TM, it reads the leftmost 1, it skips 1 and searches for the 1st Blank symbol 'B' and moves to right
2. At state q_0, when it reads 1st B, it changes B to '1' and moves to right to see the next Blank symbol 'B' and changes to q_1
3. At state q_1, when it finds the 2nd 'B' blank symbol, it changes B to '1' and moves to right and enters the accepting state q_2.

STEP 2: TRANSITION TABLE:

	1	B
$\rightarrow q_0$	$(q_0, 1, R)$	$(q_1, 1, R)$
q_1	–	$(q_2, 1, R)$
$* q_2$	–	–

STEP 3 : TRANSITION DIAGRAM.



TM for $f(x) = x+2$.

STEP 4 : TM definition $M = (\{q_0, q_1, q_2\}, \{1\}, \{1, B\}, \delta, q_0, B, \{q_2\})$

δ: $\delta(q_0, 1) = (q_0, 1, R)$

$\delta(q_0, B) = (q_1, 1, R)$

$\delta(q_1, B) = (q_2, 1, R)$.

STEP 5: ID $x = 3$.

$\delta(q_0, 111B) \vdash_m (q_0 111B) \vdash_m (1 q_0 11B) \vdash_m (11 q_0 1B) \vdash_m (111 q_0 B) \vdash_m (1111 q_1 B) \vdash_m (11111 q_2 B)$

String is accepted.

4. Design a TM to implement the concatenation function $f(x, y) = xy$ (or) to implement addition function $f(x, y) = x+y$

SOLUTION:

STEP1:

Let us assume that x is represented by the 1^x and y is represented by 1^y in the input tape. The 1^x and 1^y is separated by the separator symbol '#' and is shown below.

$x = 2 \quad y = 3$

Input :

1	1	#	1	1	1	B	...

output :

1	1	1	1	1	B	...

$x+y = 2+3 = 5$

The sum of 2 values are performed by replacing the last '1' by Blank symbol and the steps are as follows :

a. At initial state q0, when it reads '1', it skips the 1's and remain in the same state.

b. At state q0, when it reads '#' it reaches the state q1 and changes '#' to '1' and moves right

c. At state q1, it skips all 1's and searches for 'B' by moving righ

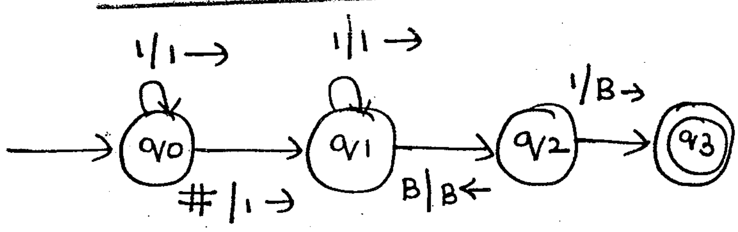
d. At state q1, when it sees blank symbol, it moves left and changes state to q2.

e. At state q2, when it finds '1' it replaces '1' to B and enters the Final state q3.

STEP 2: TRANSITION TABLE

State	1	#	B
→ q0	(q0,1,R)	(q1,1,R)	–
q1	(q1,1,R)	–	(q2,B,L)
q2	(q3,B,R)	–	–
* q3	–	–	–

STEP3: TRANSITION DIAGRAM.



TM for $f(x,y) = x+y$.

STEP 4: TM definition $M = (\{q0,q1,q2,q3\}, \{1\}, \{1,\#,B\}, \delta, q0, B, \{q3\})$,

STEP 5: ID EXAMPLE $x=2$ $y=3$

$\delta(q0, 11\#111B) \vdash_m (q0 11\#111B) \vdash_m (1q01\#111B) \vdash_m (11q0\#111B)$

$\vdash_m (111q1 111B) \vdash_m (1111 q1 11B) \vdash_m (11111q1 1B) \vdash_m (111111q_1 B)$

$\vdash_m (11111 q_2 1) \vdash_m (11111 B q_3 B)$

String accepted. The function $f(x,y) = x+y$ is implemented.

5. Design a TM to perform subtraction $f(x,y) = \begin{cases} x-y & \text{if } x>y \\ 0 & \text{if } x \le y \end{cases}$

SOLUTION :

The idea to create a TM to perform subtraction is, the i/p is represented as $1^m \# 1^n$. The value 1^m and 1^n is separated by a separator symbol '#' and $1^m \# 1^n$ is surrounded by B.

This proper subtraction function say that

$$f(m,n) = \begin{cases} m-n, & \text{if } m>n \\ 0, & \text{if } m \le n \end{cases}$$

So we have to design a TM such that if $m>n$ the subtracted value that is $1^m - 1^n$ should be on the tape. And if $m \le n$, then tape should have only 'B'

If $m=4$, $n=2$ (i.e) $m>2$

Input:

1	1	1	1	#	1	1	B	...

output:

B	B	1	1	B	B	...

$m-n=2$

If $m=2$, $n=4$, $m \le n$.

Input:

1	1	#	1	1	1	1	B	...

output:

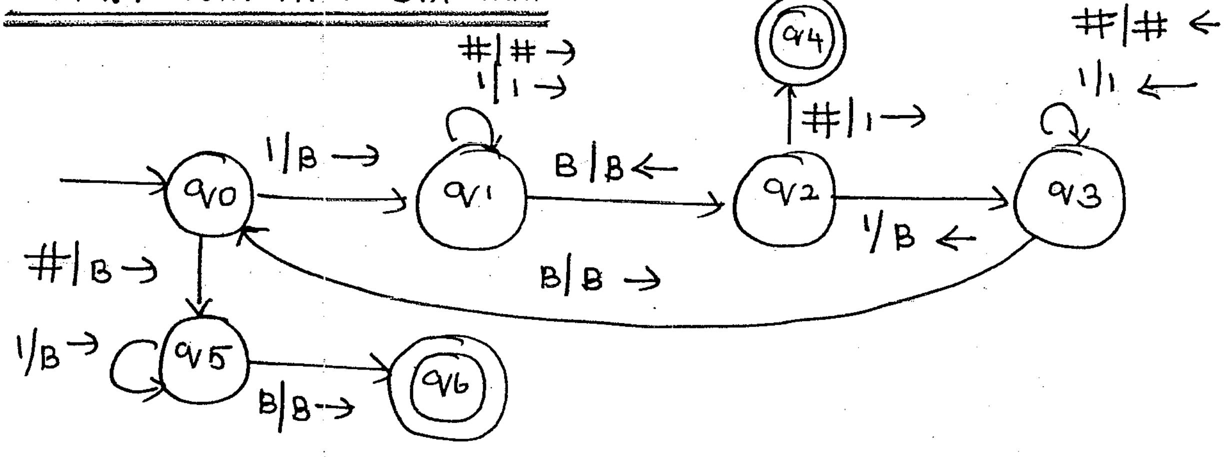
B	B	B	B	...

$m-n=0$

- The idea to design this TM is that the TM process in such a way that for each '1' on the Leftmost side, it replaces '1' on the right most side to 'B'. ['1' appearing before 'B']
- After replacing with 1's to the left and right when the m/c encounters separator symbol on right side, it is clear that n value ends.
- When 'n' value ends, it starts replacing '#' to '1' and enters final | accepting state.
- Similarly if $m \le n$, then m/c encounters the symbol '#'

from initial state then it starts replacing all 1's and '#' to Blank and enter the Final state.

STEP 2: TRANSITION DIAGRAM:



STEP 3: TRANSITION TABLE:

	1	#	B
→q0	(q1, B, R)	(q5, B, R)	–
q1	(q1, 1, R)	(q1, #, R)	(q2, B, L)
q2	(q3, B, L)	(q4, 1, R)	–
q3	(q3, 1, L)	(q3, #, L)	(q0, B, R)
* q4	–	–	–
q5	(q5, B, R)	–	(q6, B, R)
* q6	–	–	–

STEP 4: TM definition $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{1\}, \{1, \#, B\}, \delta, q_0, B, \{q_4, q_6\})$

STEP 5: ID: $m = 2$ $n = 1$

$\delta(q_0, 11\#1B) \vdash_m (q_0 11\#1B) \vdash_m (Bq_1 1\#1B) \vdash_m (B1q_1\#1B) \vdash_m (B1\#q_1 1B) \vdash_m (B1\#1q_1 B) \vdash_m (B1\#q_3 B) \vdash_m (B1q_3\#B) \vdash_m (Bq_3 1\#B \vdash_m (q_3 B1\#B) \vdash_m (Bq_0 1\#B) \vdash_m (BBq_1\#B) \vdash_m (BB\#q_1 B) \vdash_m (BBq_2\#B$

$\vdash (BB1q_4 B)$

String accepted and now the input tape contain one 1's and the function $f(m-n) = m-n$ is implemented.

Eg:2 $m=1, n=2$.

$\delta(q_0, 1\#11B) \vdash_m (q_0 1\#11B) \vdash_m (Bq_1\#11B) \vdash_m (B\#q_1 11B) \vdash_m (B\#1q_1 1$

$\vdash_m (B\#11q_1 B) \vdash_m (B\#1q_2 1B) \vdash_m (B\#q_3 1B) \vdash_m (Bq_3\#1B)$

$\vdash_m (q_3 B\#1B) \vdash_m (Bq_0\#1B) \vdash_m (BBq_5 1B) \vdash_m (BBBq_5 B) \vdash_m (BBBBq_6$

String accepted since m is less than n, then the i/p tape contains zero value.

6. Design a TM to implement multiplication function $f(x,y) = x*y$.

STEP 1:

The idea to design this TM is that we place the input as $1^x\#1^y\#$ on the TM. Now the multiplication is done by performing successive addition and it is shown below.

$x=2 \quad Y=3$

Input:

1	1	#	1	1	1	#	B	..

$x=2 \quad y=3$

$x*y = 2*3 = 6.$

output:

B	B	B	B	B	1	1	1	1	1	1	B	..

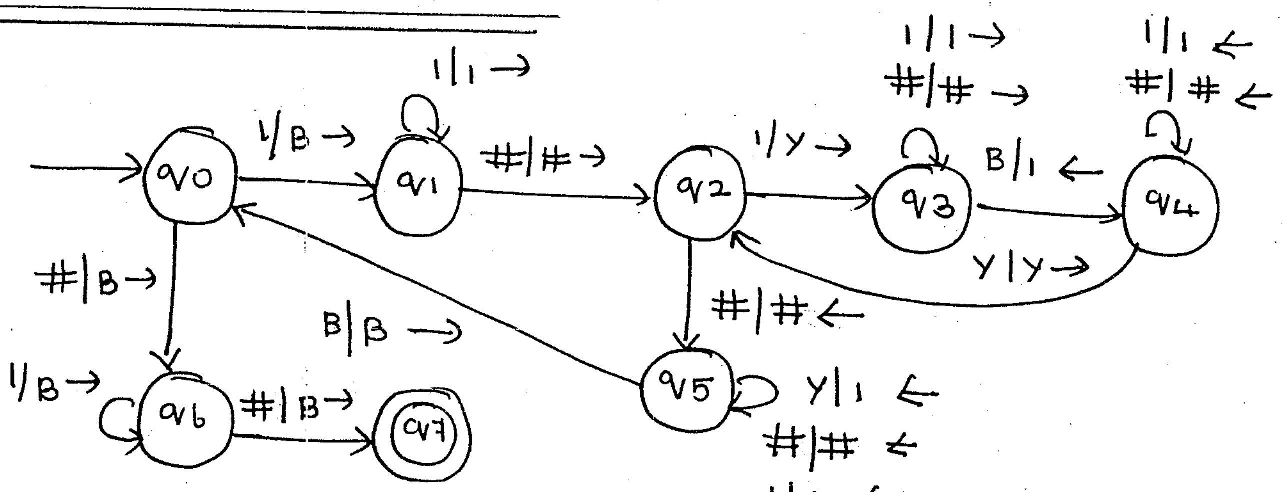
STEPS:

a. At initial state when '1' finds in the i/p, replace it to 'B' and move right for searching #

b. After Finding '#', copy the 'Y' no. of 1's for 'x' no. of times in B symbols

c. After performing 'x' no. of copy with 'y' no. of 1's we replac $\#1^y\#$ to 'B' then reach to final state and the tape contain 1^{xy}.

STEP 2: TRANSITION DIAGRAM:



TM for $f(x,y) = x * y$.

STEP 3: Transition Table.

States	1	#	B	Y
→ q0	(q1, B, R)	(q6, B, R)	–	–
q1	(q1, 1, R)	(q2, #, R)	–	–
q2	(q3, Y, R)	(q5, #, L)	–	–
q3	(q3, 1, R)	(q3, #, R)	(q4, 1, L)	–
q4	(q4, 1, L)	(q4, #, L)	–	(q2, Y, R)
q5	(q5, 1, L)	(q5, #, L)	(q0, B, R)	(q5, 1, L)
q6	(q6, B, R)	(q7, B, R)	–	–
q7	–	–	–	–

STEP 4: INSTANTANEOUS DESCRIPTION: $x = 2,\ y = 1$.

$\delta(q_0, 11\#1\#B) \vdash_m (q_0 11\#1\#B) \vdash_m (Bq_1 1\#1\#B) \vdash_m (B1q_1\#1\#B)$

$\vdash_m (B1\#q_2 1\#B) \vdash_m (B1\#Yq_3\#B) \vdash_m (B1\#Y\#q_3B) \vdash_m (B1\#Yq_4\#$

$\vdash_m (B1\#q_4Y\#1) \vdash_m (B1\#Yq_2\#1) \vdash_m (B1\#q_5Y\#1) \vdash_m (B1q_5\#1\#1)$

$\vdash_m (Bq_5 1\#1\#1) \vdash_m (q_5B1\#1\#1) \vdash_m (Bq_0 1\#1\#1) \vdash_m (BBq_1\#1\#1)$

$\vdash_m (BB\#q_2 1\#) \vdash_m (BB\#Yq_3\#1B) \vdash_m (BB\#Y\#q_3 1B)$

$\vdash_m (BB\#Y\#1q_3B) \vdash_m (BB\#Y\#q_4 11) \vdash_m (BB\#y\,q_4\#11)$

$\vdash_m (BB\#q_4 y\#11) \vdash_m (BB\#y q_2\#11) \vdash_m (BB\#q_5 y\#11) \vdash_m (BBq_5\#\#11)$

$\vdash_m (Bq_5 B\#1\#11) \vdash_m (BBq_0\#1\#11) \vdash_m (BBBq_6 1\#11) \vdash_m$

$(BBBBq_6\#11) \vdash_m (BBBBBq_2 11)$

String is accepted and the $f(x,y) = \cancel{f()}\; x * y$ is implemented.

7. Design a TM to perform 1's complement of a no. over $\Sigma = \{0,1\}$.

SOLUTION:

On Reading the i/p;

→ If the symbol = 0 replaces it by '1' & move right

→ If the symbol = 1 replace it by '0' & move right

→ Perform step 1 & 2 until the i/p symbols are processed from left to right

→ Halt the m/c when it encounters the 1st Blank symbol.

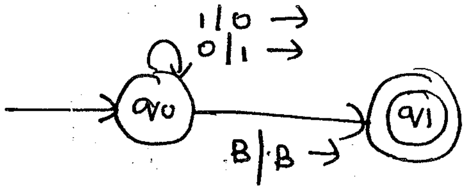
Example: 1011 → 0100

i/p o/p

STEP 2: TRANSITION TABLE:

	0	1	B
$\rightarrow q_0$	$(q_0,1,R)$	$(q_0,0,R)$	(q_1,B,R)
$* q_1$	(–	–	–

STEP 3: TRANSITION DIAGRAM.



STEP 4: TM Definition $M = (\{q_0,q_1\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_1\})$

STEP 5: I.D. $w = 101$

$\delta(q_0, 101B) \vdash_m (q_0 101B) \vdash_m (0q_0 01B) \vdash_m (01q_0 1B) \vdash_m (010q_0 B)$

$\vdash_m (010Bq_1)$

String accepted and 1's complement is implemented.

8. Design a TM to perform 2's complement of a no over $\Sigma = \{0,1\}$.

NOTE: Don't change the bits from the right towards left until the 1st 1 has been processed perform complementation to the rest of the bits from right to left [after 1st 1 is processed]

SOLUTION:

STEP 1:

a. Traverse Right & Locate Right most bit.

b. If the bit = 0, perform no replaces & move left.

c. If the bit = 1, perform no change & move left.

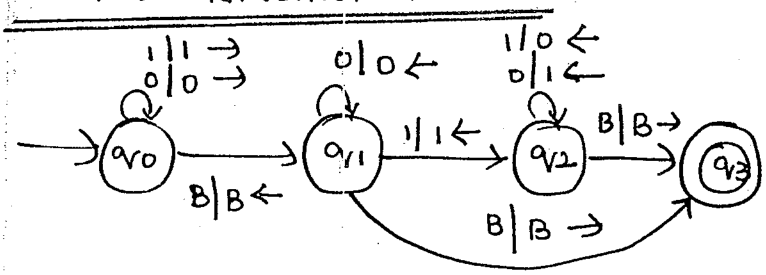
d. If the next bit symbol = '0' replace it by '1' and move left.

e. Else if the next bit symbol = '1' replace it by '0' & move left.

f. Perform steps until all the i/p symbols are processed [From Right to Left]

g. Halt the m/c.

STEP 2: TRANSITION DIAGRAM:



STEP 3: TRANSITION TABLE

	0	1	B
$\rightarrow q_0$	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_1, B$
q_1	$(q_1, 0, L)$	$(q_2, 1, L)$	$(q_3, B,$
q_2	$(q_2, 1, L)$	$(q_2, 0, L)$	$(q_3, B,$
$* q_3$	–	–	–

STEP 4: TM Definition:

$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_3\})$

STEP 5: ID. $w = 101$.

$\delta(q_0, 101B) \vdash_m (q_0 101B) \vdash_m (1 q_0 01B) \vdash_m (10 q_0 1B) \vdash_m (101 q_0 B)$

$\vdash_m (10 q_1 1B) \vdash_m (1 q_2 01B) \vdash_m (q_2 111B) \vdash_m (q_2 B011B) \vdash_m (B q_3 011B)$

String is accepted and function is implemented.

COMPUTABLE LANGUAGE

1. Design a TM that accepts the language $L = \{a^n b^n \mid n \geq 1\}$.

SOLUTION:

STEP 1: IDEA OF CREATION:

a. The idea to create this TM is to place $a^n b^n$ in the i/p tape

b. Let the TM initially be in the state q_0 (initial state).

c. while in q_0, the machine reads 'a' and changes ~~to~~ 'a' to X and moves to the right and changes its state to q_1 and starts scanning the next input

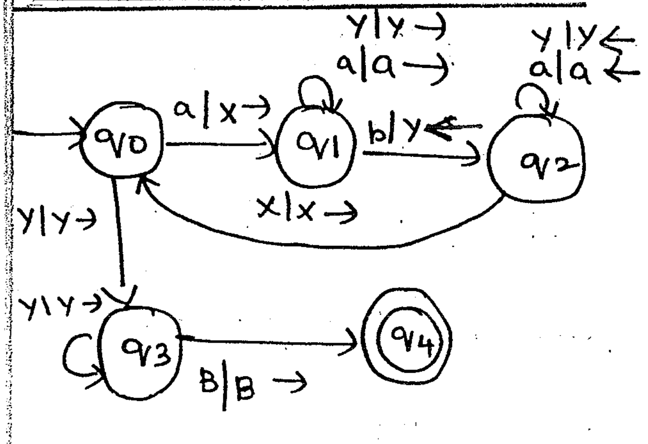
d. From the q_1, while reading 'a' it doesnot change state but simply moves to the right until seeing 1st 'b'

e. When seeing 'b' from state q_1, it reach the state q_2 and change 'b' to 'y' and moves to left to see 'X'

f. The From state q_2 when it sees X, it the state to q_0 and repeat the process.

g. The major idea is that for each 'a', we try to 'b' and alternatively, the process is repeated

STEP 2: TRANSITION DIAGRAM.



REJECTING STATE.

$(q_3, b) = (q_{reject}, b, R)$ [b>a]

$(q_3, a) = (q_{reject}, a, R)$ [ba]

$(q_3, b) = (q_{reject}, B, R)$ [a>b]

STEP 3: TRANSITION TABLE.

	a	b	Y	X	B
$\rightarrow q_0$	(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_2, Y, L)	(q_0, X, R)	-
q_3	-	-	(q_3, Y, R)	-	(q_4, B, R)
$* q_4$	-	-	-	-	-

STEP 4: TM definition $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \{a, b, X, Y, B\}, \delta, q_0, B, \{q_4\})$

STEP 5: ID $w_1 = aabb$.

$\delta(q_0, aabb) \vdash_m (q_0 aabb) \vdash_m (X q_1 abb) \vdash_m (Xa q_1 bb) \vdash_m (X q_2 a Y 1)$
$\vdash_m (q_2 X a Y b) \vdash_m (X q_0 a Y 1) \vdash_m (XX q_1 Y 1) \vdash_m (XXY q_1 1) \vdash_m (XX q_2 YY)$
$\vdash_m (X q_2 XYY) \vdash_m (XX q_0 YY) \vdash_m (XXY q_3 Y) \vdash_m (XXYY q_3 B) \vdash_m (XXYYB q_4)$

String "aabb" is accepted.

ID $w_2 = aab$.

$\delta(q_0, aab) \vdash_m (q_0 aab) \vdash_m (X q_1 ab) \vdash_m (Xa q_1 b) \vdash_m (X q_2 a Y)$
$\vdash_m (q_2 X a Y) \vdash_m (q_0 a Y) \vdash_m (XX q_1 Y) \vdash_m (XXY q_1 B)$

String "aab" is rejected.

2. Design a TM that accepts the language $L = \{a^n b^n c^n \mid n \geq 1\}$.

SOLUTION:

The Construction is similar to the design $a^n b^n$. Here we have to replace each 'a' by 'x' & 'b' by 'y' and 'c' by 'z' respectively.

IDEA :

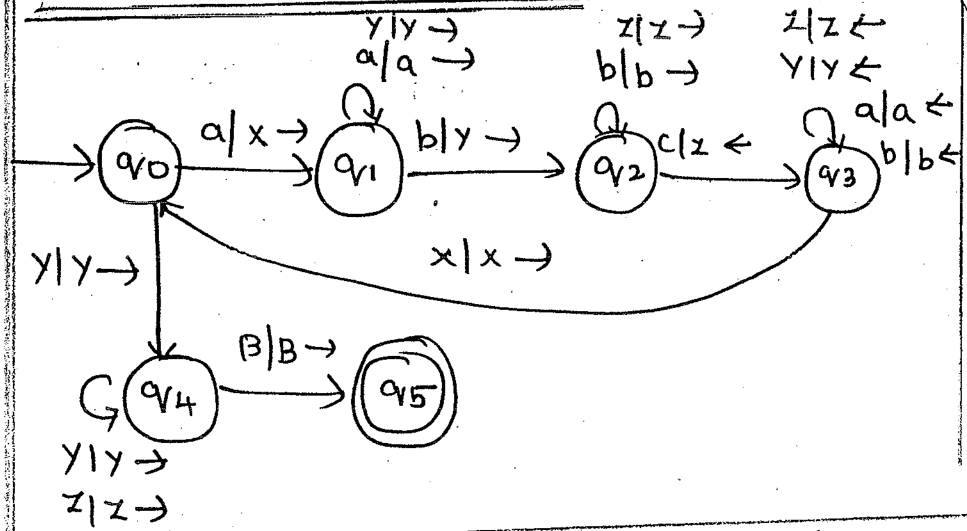
a. Initially the TM is at q0. At q0 if it finds a's replace it by x's and move right with state q1.

b. At q1, if it finds b's, replace it by y's and moves right with state q2.

c. At state q2, if it finds c's replace it by z and enters q3 by moving left

d. At q3, if it finds the leftmost x by skipping z by a then it goes to state q0. Repeat the process till at q0, if finds y.

STEP 2: TRANSITION DIAGRAM.



REJECTING STATE.

(q3, ~~b~~ c) = (qreject, ~~b~~ c, R)

(q3, a) = (qreject, a, R)

(q3, b) = (qreject, b, R)

~~(q3, b~~

STEP 3: TRANSITION TABLE.

	a	b	c	x	y	z	B
q0	(q1, y, R)	–	–	–	(q4, y, R)	–	–
q1	(q1, a, R)	(q2, y, R)	–	–	(q1, y, R)	–	–
q2	–	(q2, b, R)	(q3, z, L)	–	–	(q2, z, L)	–
q3	(q3, a, L)	(q3, b, L)	–	(q0, x, R)	(q3, y, L)	(q3, z, L)	–
q4	–	–	–	–	(q4, y, R)	(q4, y, R)	(q5, B, R)
q5	–	–	–	–	–	–	–

STEP 4: TM Definition $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a, b, c\}, \{a, b, c, x, y, z, B\}, \delta, q_0, B, \{q_5\})$.

STEP 5: ID $w_1 = aabbcc$.

$\delta(q_0, aabbcc) \vdash_m (q_0 aabbcc) \vdash_m (x q_1 abbcc) \vdash_m (xa q_1 bbcc)$
$\vdash_m (xay q_2 bcc) \vdash_m (xayb q_2 cc) \vdash_m (xay q_3 bzc) \vdash_m (xa q_3 ybzc)$
$\vdash_m (x q_3 aybzc) \vdash_m (q_3 xaybzc) \vdash_m (x q_0 aybzc) \vdash_m (xx q_1 ybzc)$
$\vdash_m (xxy q_1 bzc) \vdash_m (xxyy q_2 zc) \vdash_m (xxyyz q_2 c) \vdash_m (xxyy q_3 zz)$
$\vdash_m (xxy q_3 yzz) \vdash_m (xx q_3 yyzz) \vdash_m (x q_3 xyyzz) \vdash_m (xx q_0 yyzz)$
$\vdash_m (xxy q_4 yzz) \vdash_m (xxyy q_4 zz) \vdash_m (xxyyz q_4 z)$
$\vdash_m (xxyyzz q_4 B) \vdash_m (xxyyzzB q_5)$

string "aabbcc" is accepted.

3. Design a TM for language L. The set of strings with an equal no. of 0's and 1's.

SOLUTION:

Assume that the i/p string may start with either 0 or 1, but it should have equal no. of 0's and 1's.

For eg 0101, 0110, 1001...

a. Change all 0's to x's and all 1's to y's, whether the i/p may be in any position till reaches the blank symbol.

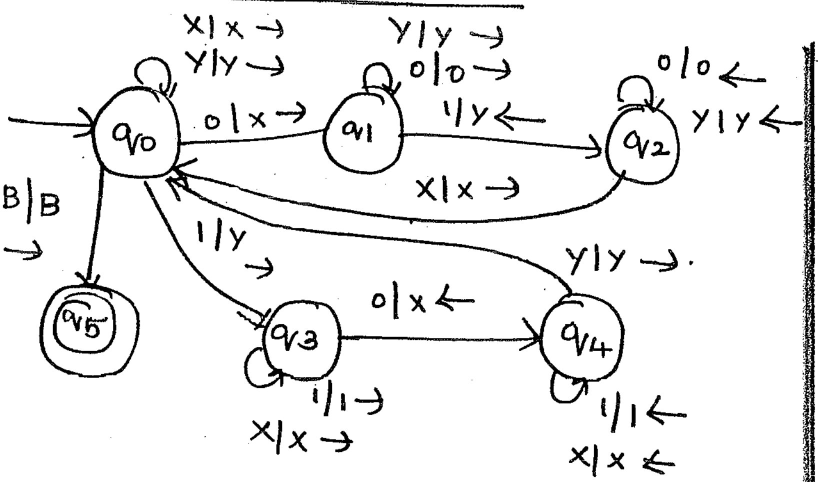
b. Initially, the TM is at state q_0. At q_0, if it finds the leftmost symbol as '0' change it to X and enters q_1

then moves right. If it finds 1 by skipping 0's y's at q_1, change it to y and enters state q_2. At state q_2, the TM searches for the leftmost x by skipping 0's and y's and enters q_0. Repeat the process till the TM finds blank symbol at q_0

c. At q_0, if it finds the leftmost symbol as 1, change it to y and enters state q_3. At q_3, if it finds 0's by skipping 1's and x's, change it to x and enters state q_4 by moving left. At q_4, it searches for the leftmost y. If it finds y at q_4, the TM enters state q_0. Repeat the process till it finds blank symbol.

d. For all other state changes, the input is rejected.

STEP 2: TRANSITION DIAGRAM:



REJECTING STATE.

$(q_3, 1) = (q_{reject}, B, R)$.

$(q_1, B) = (q_{reject}, B, R)$.

STEP 3: TABLE:

	0	1	X	Y	B
→ q_0	(q_1, x, R)	(q_3, Y, R)	(q_0, x, R)	(q_0, Y, R)	(q_5, B, R)
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	(q_4, X, L)	$(q_3, 1, R)$	(q_3, X, R)	–	–
q_4	–	$(q_4, 1, L)$	(q_4, X, L)	(q_0, Y, R)	–
q_5	–	–	–	–	–

STEP 4: TM definition $M = (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0,1\}, \{0,1,X,Y,B\}, \delta, q_0, B, \{q_5\})$

STEP 5: ID $w_1 = 1001$

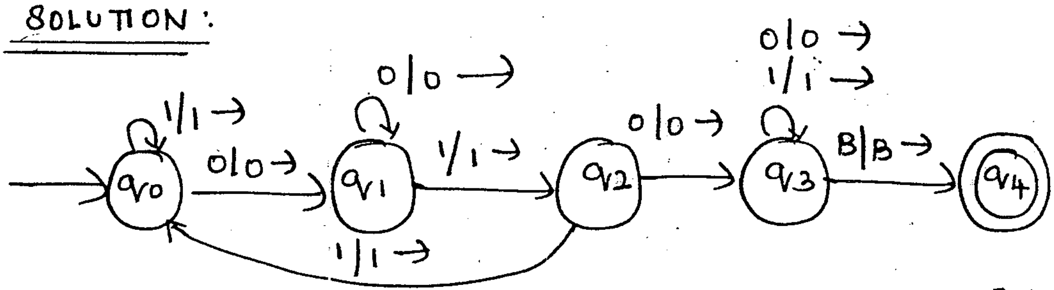
$\delta(q_0, 1001) \vdash_m (q_0 1001) \vdash_m (Y q_3 001) \vdash_m (q_4 YX01) \vdash_m (Y q_0 X01)$
$\vdash_m (YX q_6 01) \vdash_m (YXX q_1 1) \vdash_m (YX q_2 XY1) \vdash_m (YXX q_0 Y) \vdash_m (YXXY q_0 B)$
$\vdash_m (YXXYB q_5) \Rightarrow$ string is accepted.

$w_2 = 0100$

$\delta(q_0, 0100) \vdash_m (\cancel{q_0 0100}\ q_0 0100) \vdash_m (q_2 XY00) \vdash_m (X q_0 Y00) \vdash_m$
$(XY q_0 00) \vdash_m (XYX q_1 0) \vdash_m (XYX0 q_1 B) \Rightarrow$ Rejected [No Transition]

4. Design a TM to accept the language L contains a substring "010"

SOLUTION:



TM: $M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0,1,B\}, \delta, q_0, B, \{q_4\})$.

5. Design the TM to accept the language of palindromes over the alphabet $\{a,b\}$ or to accept the lang. $L = \{ww^R \mid w \in \{a,b\}\}$.

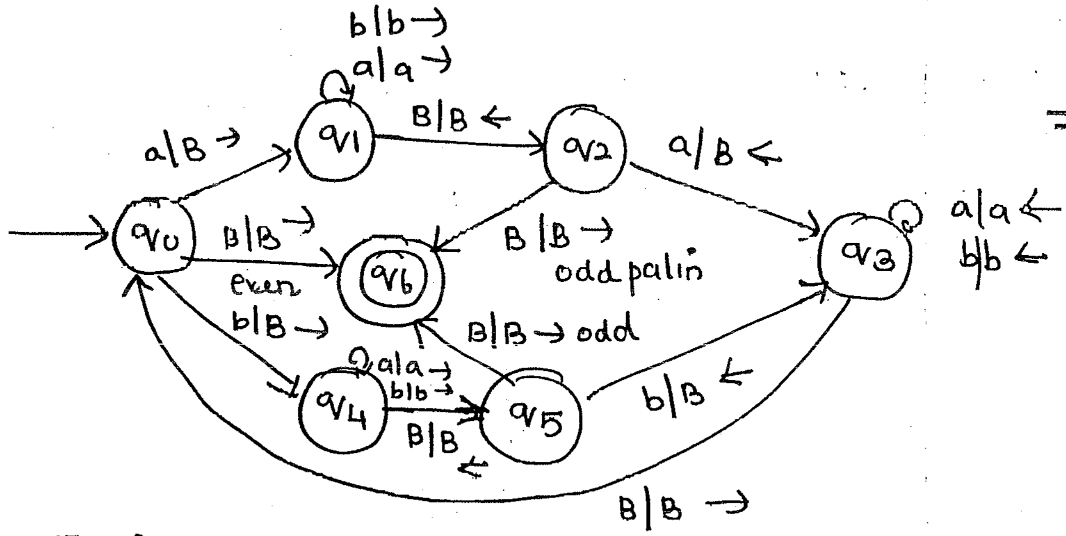
SOLUTION:

STEP 1: IDEA OF CREATION:

- The TM that we are designing ~~now~~ should accept the strings of palindromes such as ababa, abbbba .... The idea to design this TM, is that if the m/c reads 'a' on the left most symbol, replace 'a' to 'B' and move to right and changes last 'a' to B.

- Similarly if the m/c reads 'b' then it replaces b to B and moves to right by searching B and last b and replace b to B.
- So the overall idea is for each 'a' that is first 'a' on the left it matches the last 'a' on the rightmost side and for each b on the 1st time on the left, it matches last b on right si

STEP 3: TRANSITION DIAGRAM.



REJECTING STATE

$\delta(q_2, a) = (q_{reject}, a, L$

$\delta(q_5, b) = (q_{reject}, b, L)$

TM for $L = \{ww^R \mid w \in (a,b)^*\}$

STEP 4: $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a,b\}, \{a,b,B\}, \delta, q_0, B, \{q_6\})$.

6\. Design the TM to compute the fn $F(w) = w\,cw^R$. where w is any string of a's & b's.

SOLUTION:

STEP 1: IDEA OF CREATION.

- The idea to create this TM is that to read the string w and to create wcw^R.

→ Here we initially read all the symbols in the string w upto 'B' and then moves on the left one position and symbol.

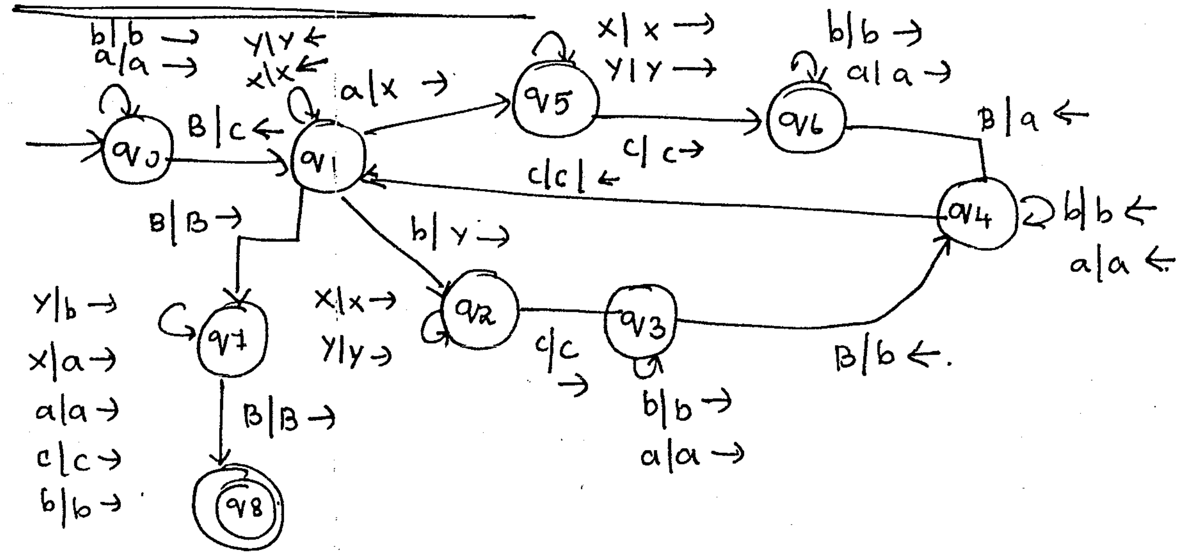
→ If the symbol is 'a', then we replace it by x and if the symbol is 'b', it is replaced by Y.

→ After replacing the symbol, we move to the right and replace B by 'a' or 'b' based on the symbol read before the B.

→ After processing all the strings w and we replace 'x' by 'a' and 'Y' by b.

→ After replacing the entire string symbol in 'w', we move to the right side until blank symbol.

STEP 2: Transition Diagram.



Rejecting state

$\delta(q_1, a) = (q_{reject}, a, L)$

$\delta(q_1, b) = (q_{reject}, b, L)$.

STEP 3: TRANSITION TABLE

STEP 5: ID - any string.

STEP 4: TM Definition

$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8\}, \{a, b, c\}, \{a, b, c, B\}, \delta, q_0, B, \{q_8\})$