(Rough Page) (5)

TURING MACHINE

(1)

* Turing Machine is a mathematical model of computation that defines abstract computer and was invented in "1936" by "Alan Turing".

* A Turing Machine M is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$

(1.) Q is a finite non-empty set of state

(2.) Γ is a finite non-empty set of tape symbols.

(3.) $B \in \Gamma$, is the blank.

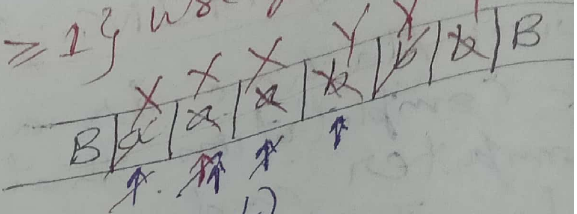
(4.) Σ is a non-empty set of input symbols and is a subset of Γ and $b \notin \Sigma$

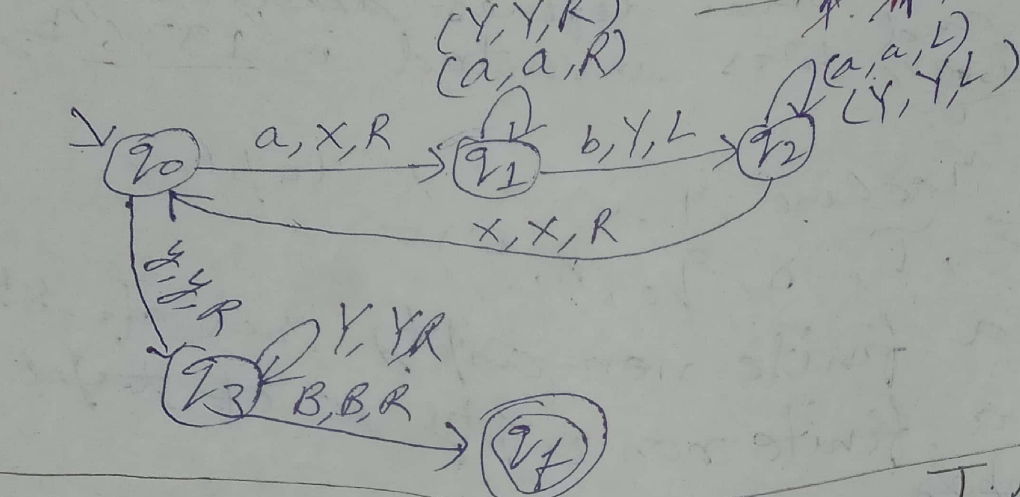
(5.) δ is the transition function mapping the states of finite automation and tape symbols to states, tape symbols and movement of the head, i.e. $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$

(6.) $q_0 \in Q$ is the initial state

(7.) $F \subseteq Q$ is the set of final states.

② Ex-(1.) $L = \{a^n b^n \mid n \geq 1\}$ using T.M.





Transition Table for above T.M.

	a	b	X	Y	B
q_0	q_1, X, R	Halt	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

Transition function for above T.M.

$\delta(q_0, a) = (q_1, X, R)$

$\delta(q_0, Y) = (q_3, Y, R)$

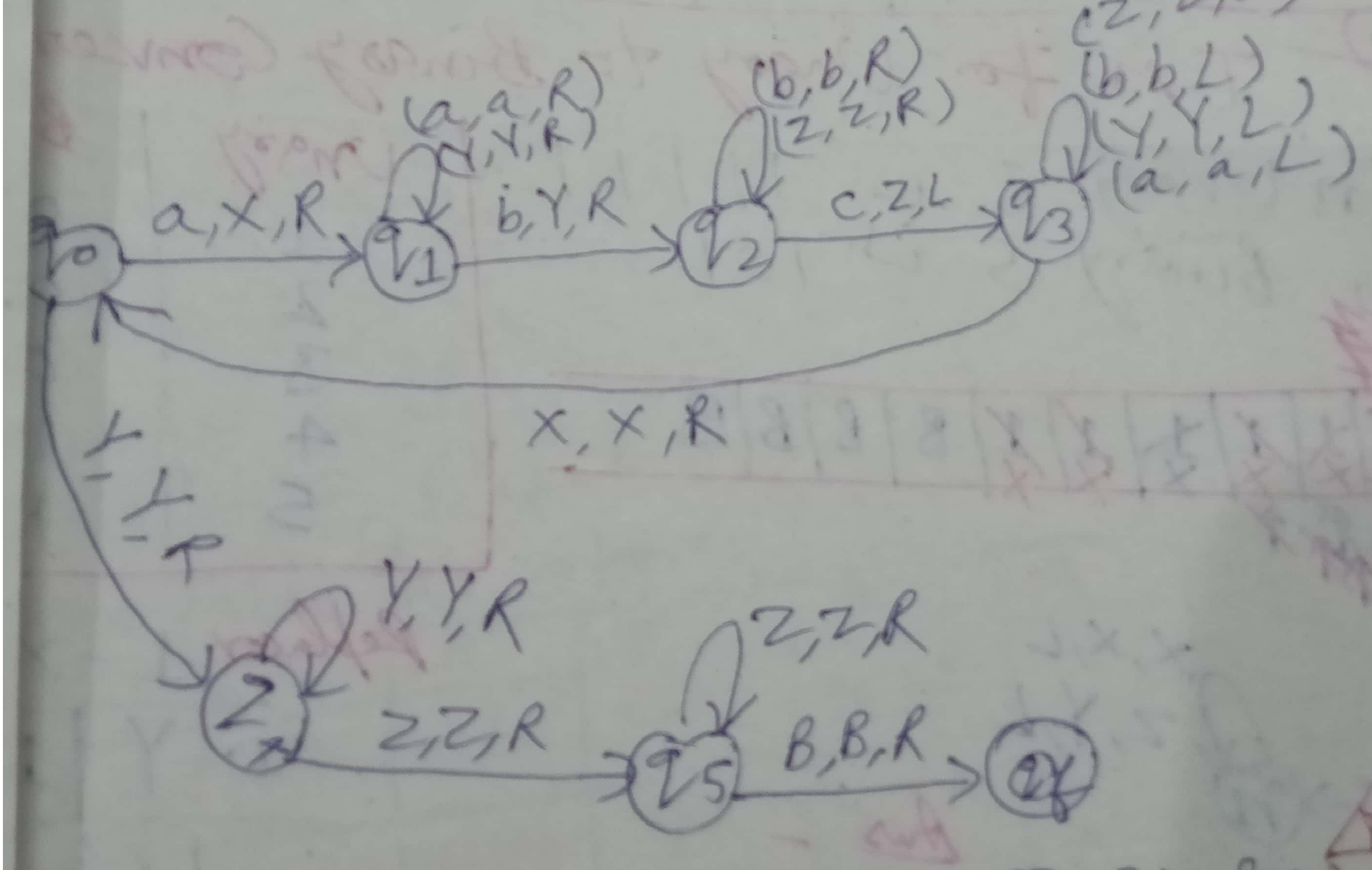
$\delta(q_1, a) = (q_1, a, R)$

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

Ans.

2.) $L = \{a^n b^n c^n \mid n \geq 1\}$ using T.M.

B	a	~~a~~	~~a~~	~~b~~	~~b~~	~~b~~	~~c~~	~~c~~	~~c~~	B
	X	X	X	Y	Y	Y	Z	Z	Z	

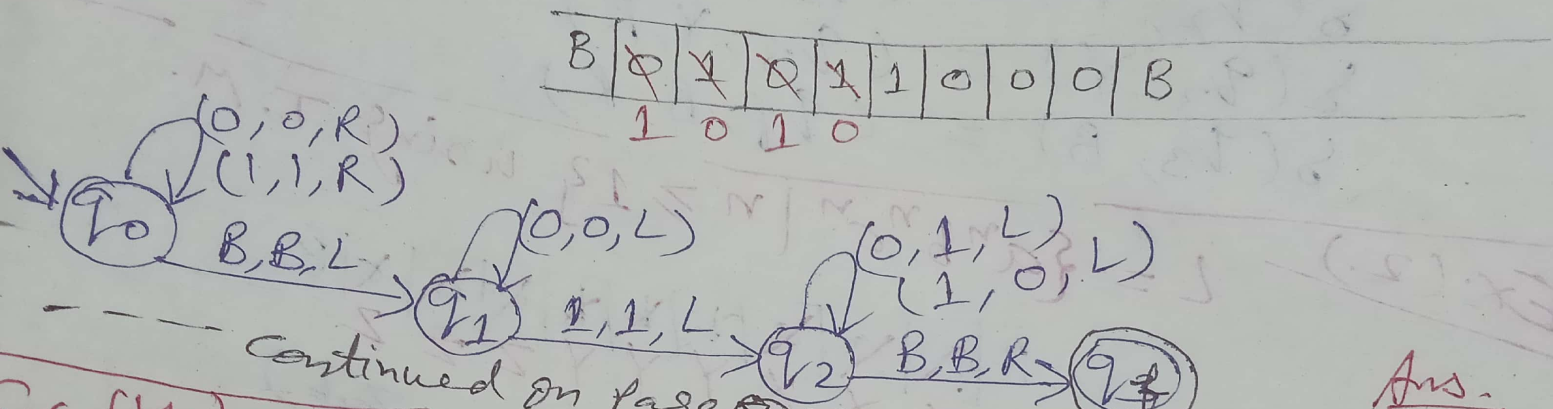


Ans.

--- Continued on page ⑤ (Fair Page) for Transition function & Table

for 2's Complement

01011000	
10100111	1's Comp.
+1	
10101000	2's Comp.



---- Continued on page 7 (Fair Page) for Transition Fun? & Table.

Ans.

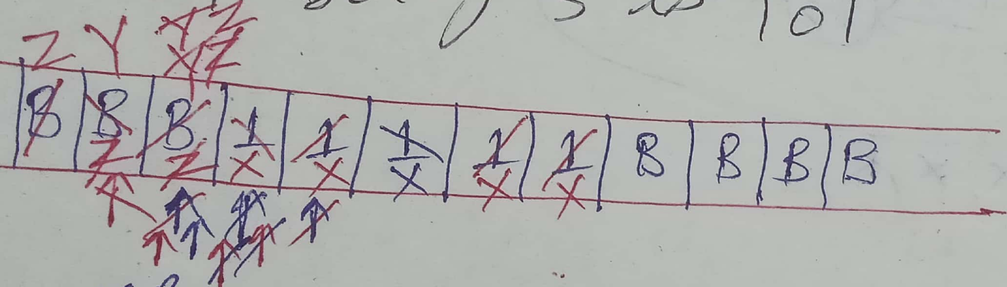
Ex.-(4.)

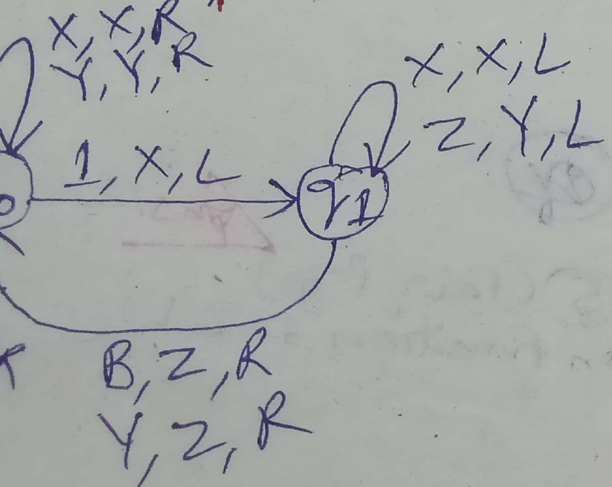
T.M. for Unary to Binary Converter.

In Unary 5 is 11111

In Binary 5 is 101

Unary	Binary
1	001
2	010
3	011
4	100
5	101





Ans -

Replacer 1 - X

0-Y	1-Z

101

O/P ZYZ

100 +1 101 ZYZ	11 +1 100 Z YY	1 Z +1 Z 10 ZY	0 Y +1 Z 1 Z

Ex.(2.) continued Page (3) (Rough Page)
Rough Page (3.)

Sol: $L = \{a^n b^n c^n \mid n \geq 1\}$ using T.M.

Transition Function for above T.M.

$\delta(q_0, a) = (q_1, X, R)$
$\delta(q_0, Y) = (q_4, Y, R)$
$\delta(q_1, a) = (q_1, a, R)$
$\delta(q_1, Y) = (q_1, Y, R)$
$\delta(q_1, b) = (q_2, Y, R)$
$\delta(q_2, b) = (q_2, b, R)$
$\delta(q_2, Z) = (q_2, Z, R)$
$\delta(q_2, c) = (q_3, Z, L)$
$\delta(q_3, b) = (q_3, b, L)$
$\delta(q_3, Y) = (q_3, Y, L)$
$\delta(q_3, a) = (q_3, a, L)$
$\delta(q_3, Z) = (q_3, Z, L)$
$\delta(q_3, X) = (q_0, X, R)$
$\delta(q_4, Y) = (q_4, Y, R)$
$\delta(q_4, Z) = (q_5, Z, R)$
$\delta(q_5, Z) = (q_5, Z, R)$
$\delta(q_5, B) = (q_f, B, R)$

Transition Table for above T.M.

	a	b	c	X	Y	Z	B
q_0	q_1, X, R	Halt	H	H	q_4, Y, R	H	H
q_1	q_1, a, R	q_2, Y, R	Halt	H	q_1, Y, R	H	H
q_2	Halt	q_2, b, R	q_3, Z, L	H	H	q_2, Z, R	H
q_3	q_3, a, L	q_3, b, L	Halt	q_0, X, R	q_3, Y, L	q_3, Z, L	H
q_4	Halt	H	H	H	q_4, Y, R	q_5, Z, R	H

P.T.O.

Page - (6)

a	b	c	X	Y	Z	B
Halt	H	H	H	H	q_5, Z, R	q_f, B, R
Halt	H	H	H	H	H	H

continued Page (4) (Rough Page) Ex. (3.) Rough Page (4.)

T.M. for 2's Complement.

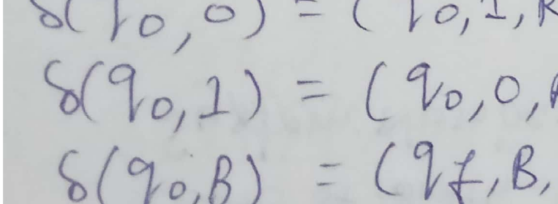
Transition Function for above T.M.

$\delta(q_0, 0) = (q_0, 0, R)$
$\delta(q_0, 1) = (q_0, 1, R)$
$\delta(q_0, B) = (q_1, B, L)$
$\delta(q_1, 0) = (q_1, 0, L)$
$\delta(q_1, 1) = (q_2, 1, L)$
$\delta(q_2, 0) = (q_2, 1, L)$
$\delta(q_2, 1) = (q_2, 0, L)$
$\delta(q_2, B) = (q_f, B, R)$

Transition Table for above T.M.

	0	1	B
q_0	$q_0, 0, R$	$q_0, 1, R$	q_1, B, L
q_1	$q_1, 0, L$	$q_2, 1, L$	Halt
q_2	$q_2, 1, L$	$q_2, 0, L$	q_f, B, R
q_f	Halt	Halt	Halt

Ex. (4.) T.M. for 1's Complement




```
  01011000
  10100111   1's
```

B	0	1	0	1	1	0	0	0	B

Transition Function for T.M.

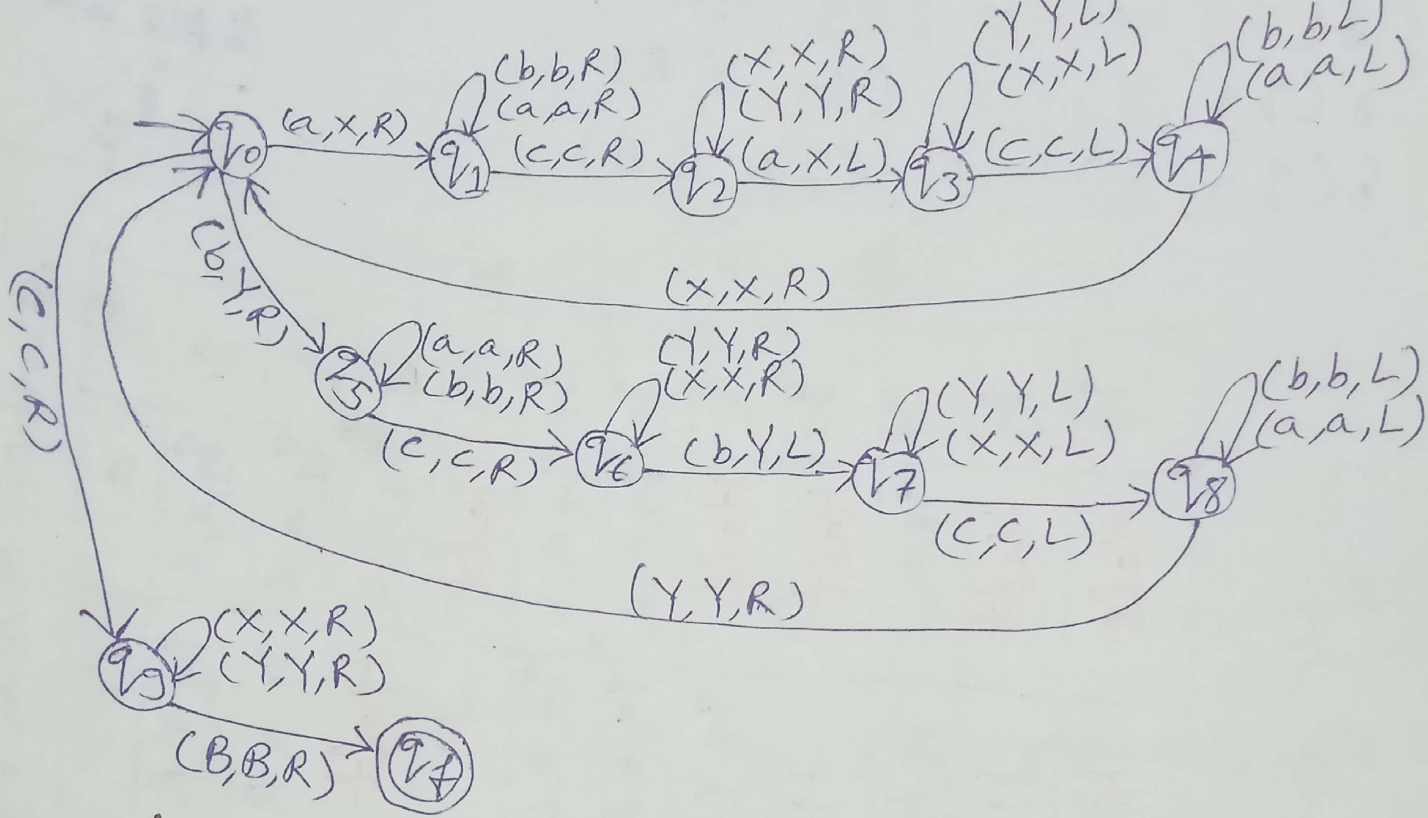
$\delta(q_0, 0) = (q_0, 1, R)$
$\delta(q_0, 1) = (q_0, 0, R)$
$\delta(q_0, B) = (q_f, B, R)$

Transition Table for T.M.

	0	1	B
q_0	$q_0, 1, R$	$q_0, 0, R$	q_f, B, R
q_f	Halt	H	H

Soln- Design a T.M. for $L = \{wcw \mid w \in$

B	~~a~~	~~b~~	~~a~~	~~a~~	~~b~~	c	~~a~~	~~b~~	~~a~~	~~a~~	~~b~~	B
	X	Y	X	X	Y		X	Y	X	X	Y	



Transition Function for above T.M.

$\delta(q_0, a) = (q_1, X, R)$
$\delta(q_0, b) = (q_5, Y, R)$
$\delta(q_0, c) = (q_9, c, R)$
$\delta(q_1, b) = (q_1, b, R)$
$\delta(q_1, a) = (q_1, a, R)$
$\delta(q_1, c) = (q_2, c, R)$
$\delta(q_2, X) = (q_2, X, R)$
$\delta(q_2, Y) = (q_2, Y, R)$
$\delta(q_2, a) = (q_3, X, L)$
$\delta(q_3, X) = (q_3, X, L)$
$\delta(q_3, Y) = (q_3, Y, L)$

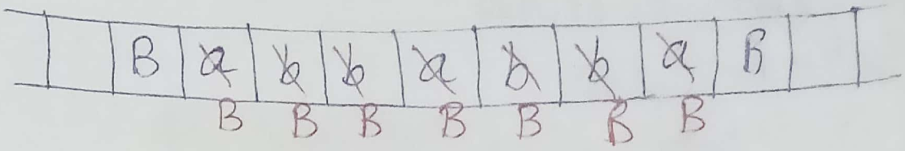
$\delta(q_3, c) = (q_4, c, L)$
$\delta(q_4, a) = (q_4, a, L)$
$\delta(q_4, b) = (q_4, b, L)$
$\delta(q_4, X) = (q_0, X, R)$
$\delta(q_5, a) = (q_5, a, R)$
$\delta(q_5, b) = (q_5, b, R)$
$\delta(q_5, c) = (q_6, c, R)$
$\delta(q_6, X) = (q_6, X, R)$
$\delta(q_6, Y) = (q_6, Y, R)$
$\delta(q_6, b) = (q_7, Y, L)$
$\delta(q_7, X) = (q_7, X, L)$
$\delta(q_7, Y) = (q_7, Y, L)$

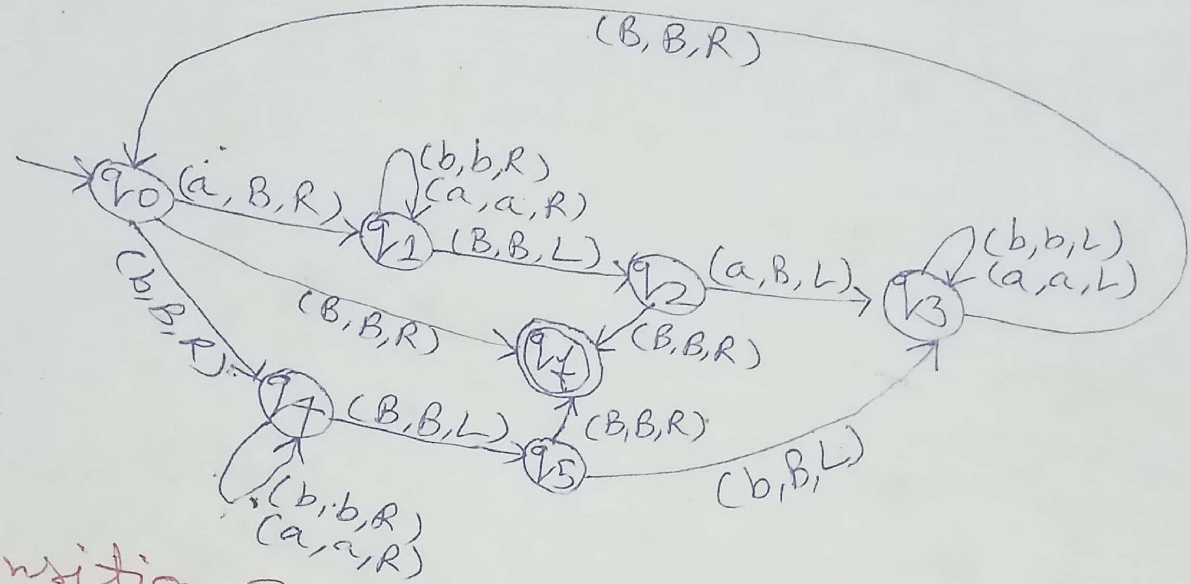
$\delta(q_7, c) = (q_8, c, L)$
$\delta(q_8, a) = (q_8, a, L)$
$\delta(q_8, b) = (q_8, b, L)$
$\delta(q_8, Y) = (q_0, Y, R)$
$\delta(q_9, X) = (q_9, X, R)$
$\delta(q_9, Y) = (q_9, Y, R)$
$\delta(q_9, B) = (q_f, B, R)$

Transition Table for above. T.M.

	a	b	c	X	Y	B
q_0	q_1,X,R	q_5,Y,R	q_9,c,R	Halt	H	H
q_1	q_1,a,R	q_1,b,R	q_2,c,R	Halt	H	H
q_2	q_3,X,L	Halt	H	q_2,X,R	q_2,Y,R	H
q_3	Halt	H	q_4,c,L	q_3,X,L	q_3,Y,L	H
q_4	q_4,a,L	q_4,b,L	Halt	q_0,X,R	H	H
q_5	q_5,a,R	q_5,b,R	q_6,c,R	Halt	H	H
q_6	Halt	q_7,Y,L	H	q_6,X,R	q_6,Y,R	H
q_7	Halt	H	q_8,c,L	q_7,X,L	q_7,Y,L	H
q_8	q_8,a,L	q_8,b,L	Halt	H	q_0,Y,R	H
q_9	Halt	H	H	q_9,X,R	q_9,Y,R	q_f,B,R
q_f	Halt	H	H	H	H	H

even and odd Palindrome





Transition Function for above T.M.

$\delta(q_0, a) = (q_1, B, R)$
$\delta(q_0, b) = (q_4, B, R)$
$\delta(q_0, B) = (q_f, B, R)$
$\delta(q_1, a) = (q_1, a, R)$
$\delta(q_1, b) = (q_1, b, R)$
$\delta(q_1, B) = (q_2, B, L)$
$\delta(q_2, a) = (q_3, B, L)$
$\delta(q_2, B) = (q_f, B, R)$
$\delta(q_3, a) = (q_3, a, L)$
$\delta(q_3, b) = (q_3, b, L)$
$\delta(q_3, B) = (q_0, B, R)$
$\delta(q_4, a) = (q_4, a, R)$
$\delta(q_4, b) = (q_4, b, R)$
$\delta(q_4, B) = (q_5, B, L)$
$\delta(q_5, b) = (q_3, B, L)$
$\delta(q_5, B) = (q_f, B, R)$

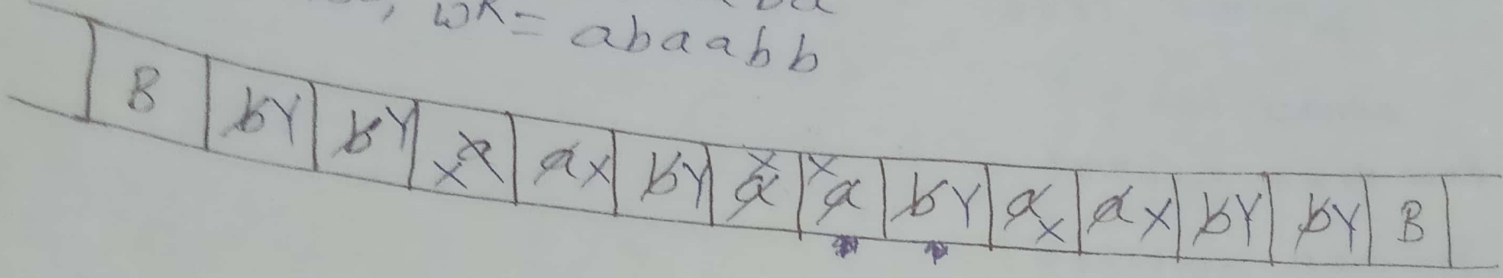
Transition Table for above T.M.

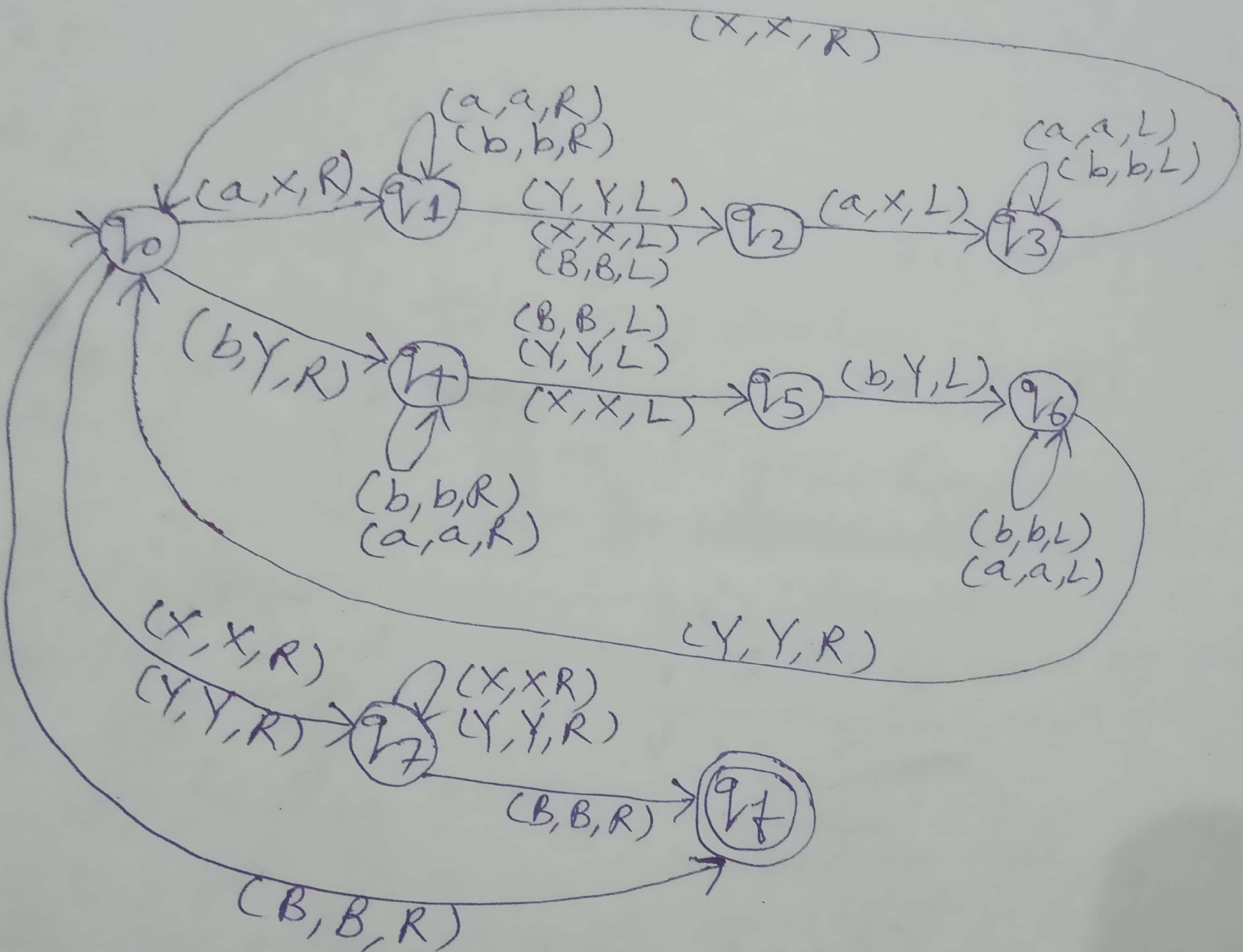
	a	b	B
q_0	q_1, B, R	q_4, B, R	q_f, B, R
q_1	q_1, a, R	q_1, b, R	q_2, B, L
q_2	q_3, B, L	Halt	q_f, B, R
q_3	q_3, a, L	q_3, b, L	q_0, B, R
q_4	q_4, a, R	q_4, b, R	q_5, B, L
q_5	Halt	q_3, B, L	q_f, B, R
q_f	Halt	H	H

gn a T.M. for $L = \{ww^R \mid w \in \{a,b\}^*\}$

Let, $w = bbaaba$

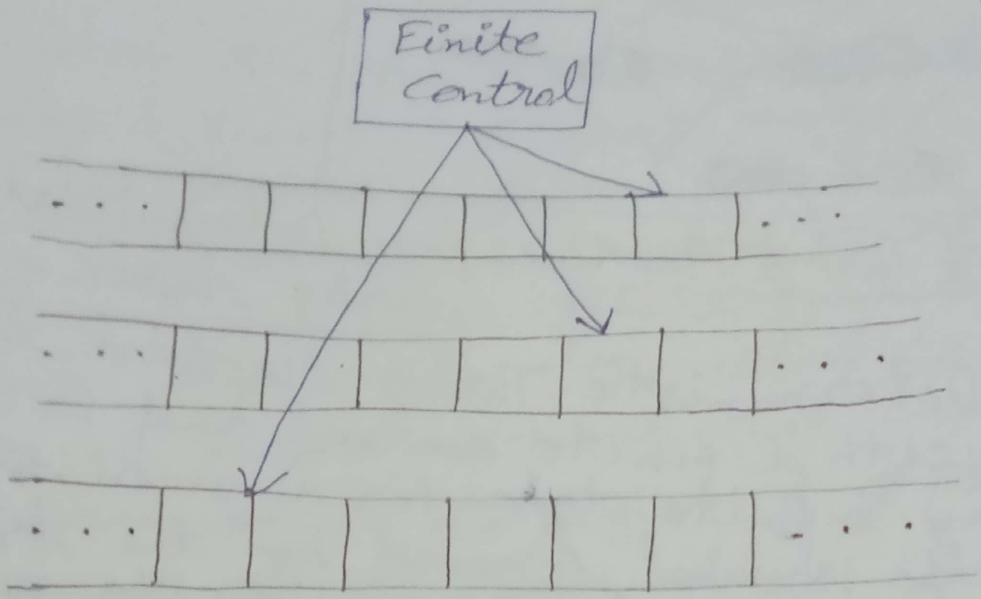
so, $w^R = abaabb$





①

** Variants of Turing Machine : Multitape Turing Machine —



E.g;- Multitape Turing Machine

Multitape Turing Machine consists of a finite control with k tape heads; each tape is infinite in both directions. On a single more, depending on the state of the finite control and the symbol scanned by each of the tape heads, the machine can:

(1.) change state;
(2.) print a new symbol on each of the cells scanned by its tape heads;
(3.) more each of its tape heads, indepen-dently, one cell to the left or right, or peep it stationary.

Initially, the input appears on the first tape, and the other tapes are blank.

*Variants of Turing Machine: Non-Deterministic Turing machine -

A Non-Deterministic Turing Machine is a device with a finite control and a single one-way infinite tape. For a given state and tape symbol scanned by the tape head, the machine has a finite number of choices for the next move. Each choice consists of a new state, a tape symbol to print, and a direction of head motion. The Non-Deterministic T.M. is not permitt-ed to make a move in which the next state is selected from one choice, and the symbol printed and the dir-ection of head motion are selected from ~~one~~ other choices. The Non-Deterministic T.M. accepts its input if any sequence of choices of moves leads to an accepting state.

Variants of Turing Machine: Universal Turing Machine –

Consider the following argument against Turing's thesis: "A Turing machine is a special purpose computer. Once δ is defined, the machine is restricted to carrying out one particular type of computation. Digital computers, on the other hand, are general purpose machines that can be programmed to do different jobs at different times. Consequently, T.Ms can not be considered equivalent to general purpose digital computers."

This objection can be overcome by designing a reprogrammable Turing machine called a Universal T.M.. A Universal T.M. M_u is an automation that, given as input the description of any T.M. M and a string w, can simulate the computation of M on w.

* Halting Problem of Turing Machine:-

- Given a Program; will it Halt?
- Given a Turing Machine, will it halt when run on some particular given input string?
- Given some program written in some language, will it ever get into an infinite loop or will it always terminate?

Answer:-

- In general, we can not always know.
- The best we can do is, run the program and see whether it halts?
- For many programs, we can see that it will always halt or sometimes loop?

* Church – Turing Thesis – A mathematical model, ~~Turing~~ Turing machine that can carry out complex tasks such as acceptance of language, computing functions and general purpose computations. At the begining of 20th Century, the mathematician D Hilbert asked, "Whether there exists an algo that can prove any well stated mathematical formula".
In 1931, K. Godel showed that such an algo cannot exist. In 1936, A.M. Turing proposed Turing machine as a computational model and suggested that the defination of an algorithm can be based on this model.
The mathematician and logician, Alonzo Church proposed an alternative formalization for the notion of algo in 1936, known as Church Turing thesis.
This conjecture is stated in number of ways by different writers.

~~Some~~

Some of the equivalent statements of Church turing thesis are as follows:-

① Any computation that can be carried out by mechanical means can be performed by some Turing machine.

② Anything that is initiitively computable can be computed by a Turing machine.

③ The turing machine that halts on all inputs is the precise formal notion of an algorithm.

④ Given any problem which can be solved with an effective algorithm, there is a TM that can solve this problem.

⑤ Any general way to compute is to compute only the partial-recursive functions or equivalently what TMs can compute.

⑥ There is no formalism to model any mechanical calculus that is more powerful than TM and equivalent formalisms

⑦ A number-theoretic function is computable by an algo if and only if it is Turing computable.

The word "THESIS" is used instead of the word "THEOREM" as it is not a mathematical result. It is based on the intuitive notion of the what "mechanical computations" are and equates it with a mathematical idea ie "algorithm". In fact Church Turing thesis is non provable. It is supported by previous experience and by intuitive evidences given as follows:-

(i) So far whatever alternative models have been proposed for mechanical computations are not found to be more powerful than The Turing machine

(ii) There exists no problem which is solvable by an effective algorithm but cannot be solved by TM.

(iii) All known formalisms to model discrete computing devices have at most the power of TMs (or anything that can be done on any existing digital computer can also be done by a TM).