Turing Machine (TM)

It is defined by 7 Tuples

$$(q, \Sigma, \Gamma, \delta, q_0, B, F)$$

$q \rightarrow$ Finite set of states

$\Sigma \rightarrow$ Finite set of symbols

$\Gamma \rightarrow$ Tape (Infinite)

$\delta \rightarrow$ Transition function

$q_0 \rightarrow$ Initial state

$B \rightarrow$ Finite set of Blank symbol

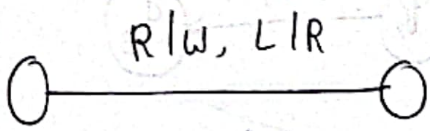
$F \rightarrow$ Finite set of Final State (Accept & Reject)

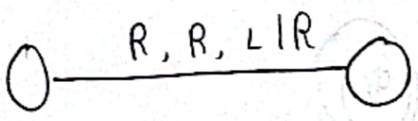
Infinite Tape:-

...	B	B	a	a	a	b	b	b	b	B	B	....

B → Blank Symbol

Operation:-

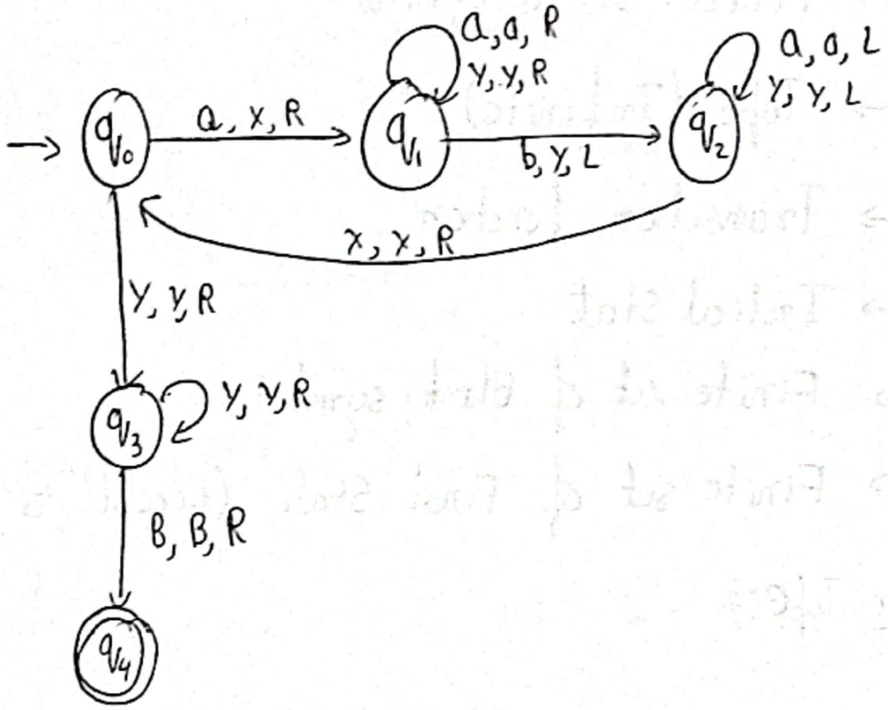




① Design a Turing Machine that recognizes the language

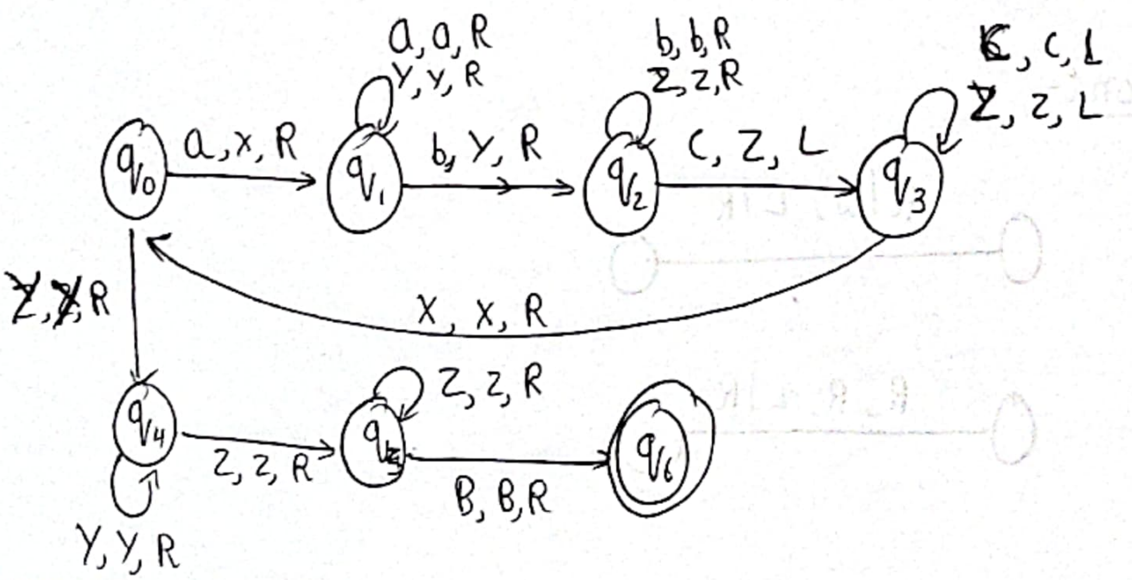
(i) $L = \{a^n b^n \mid n \geq 1\}$

…	B	a	a	a	b	b	b	B	…

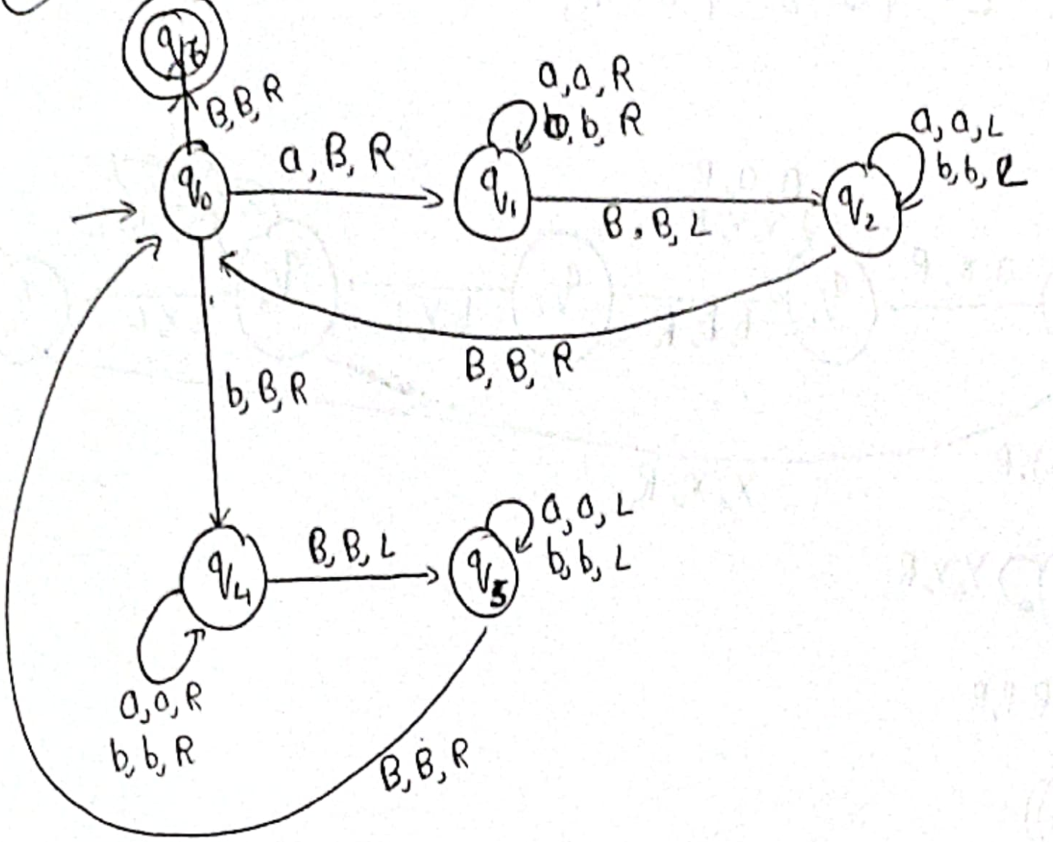


(ii) $L = \{a^n b^n c^n \mid n \geq 1\}$

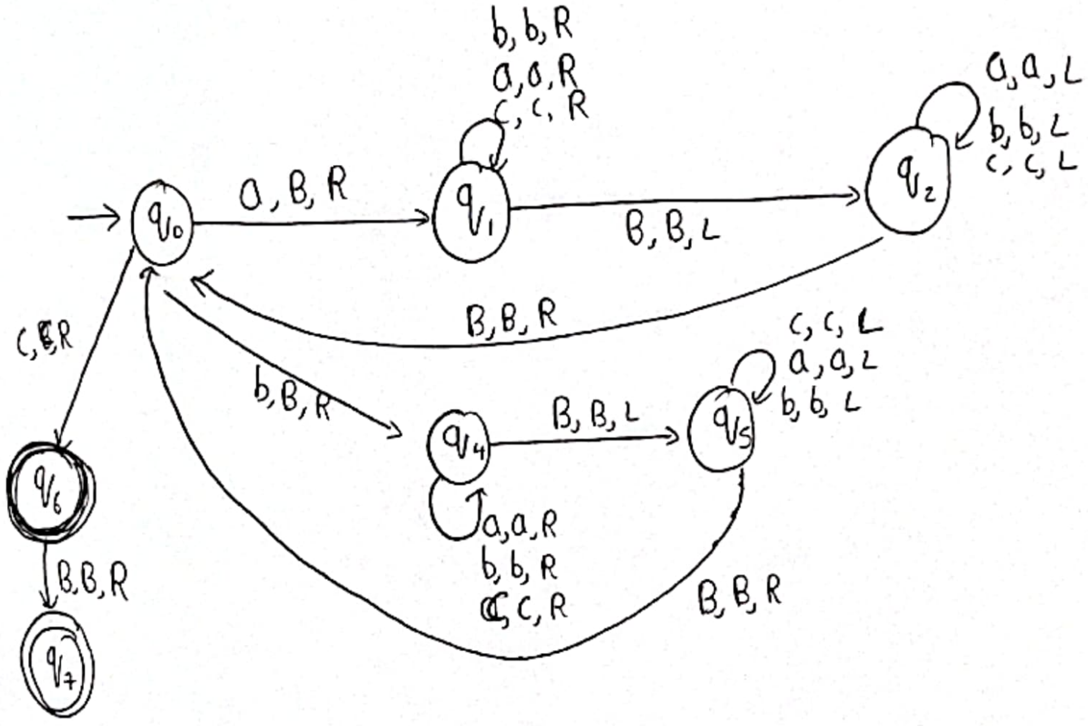
…	B	a	a	b	b	c	c	B	…



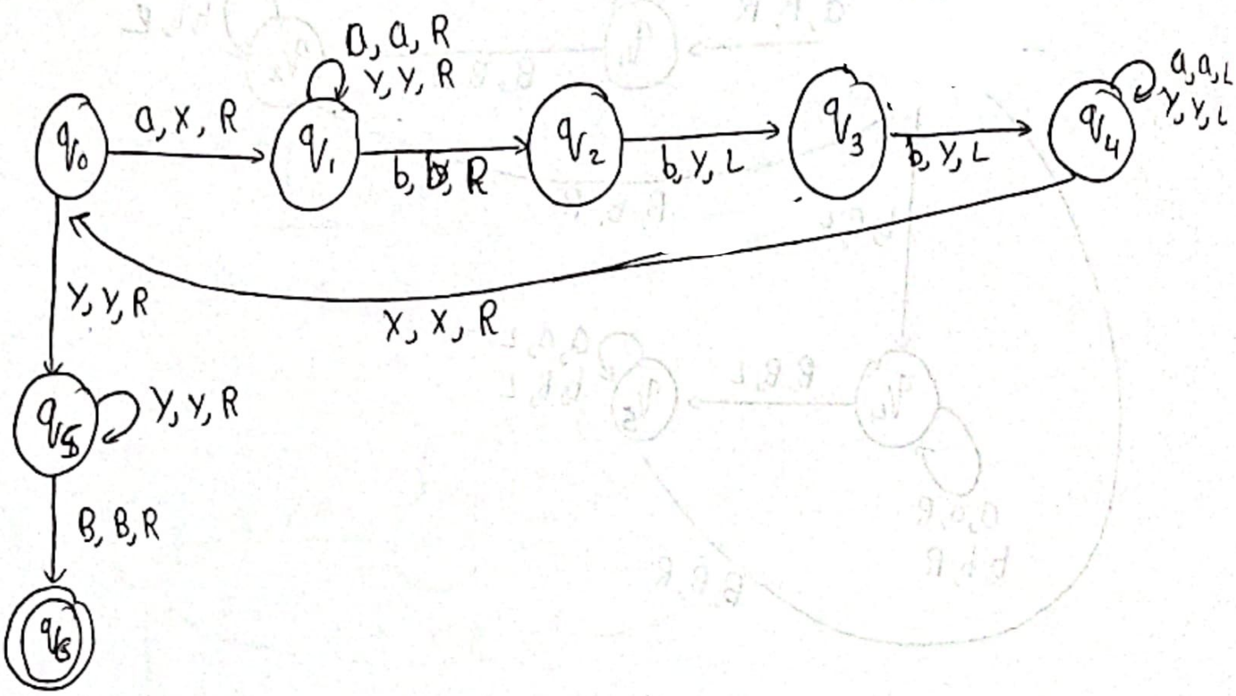
(iii) $L = \{ww^R \mid w \in (a, b)^*\}$



(iv) $L = \{wcw^R \mid w \in (a, b)^*\}$



(v) $L = \{a^n b^{2n} \mid n \geq 1\}$



Pumping Lemma :-

For any infinite language 'L', there exists a positive integer 'n' such that any string 'w' belonging to the language has a length greater than (or) equal to n i.e $|w| \geq n$ and 'w' can be written as xy^iz such that it satisfies the following conditions.

(i) for all $i \geq 0$, $xy^iz \in L$

(ii) $|xy| \leq n$

(iii) $|y| \geq 1$

In Simple terms, it means that if a string 'y' is pumped i.e if 'y' is inserted any number of times, the resultant String still remains in 'L'.

The Pumping Lemma is used to prove irregularity of a language. If there exists a string in any language which does not satisfy pumping lemma, then the language is not regular. But if all strings of a language satisfy pumping lemma, then the Language may (or) may ~~be~~ not be regular.

Let us prove $L = \{0^n 1^n \mid n \geq 0\}$ is irregular.

To prove this first assume 'L' is regular, then it Satisfies all the conditions of Pumping lemma.

Let, $w \in L$ and $|w| \geq n$, So by pumping lemma.

$w = xyz$ such that conditions ①, ⓘⓘ, ⓘⓘⓘ holds.

Now, we have to show that ⓘ, ⓘⓘ ⓘⓘⓘ do not hold:

If ⓘ and ⓘⓘ hold, then

$$w = 0^n 1^n = xyz \quad |xy| \leq n \quad |y| \geq 1$$

So, $x = 0^a$, $y = 0^b$, $z = 0^c 1^n$,

where $b \geq 1$, $c \geq 0$, $n = a+b+c$, $a+b \leq n$

But, then ③ fails because

for $i = 0$

$$w = xy = 0^a 0^c 1^n = 0^{a+c} 1^n \notin L$$

This implies $a+c = n$ which is not true.

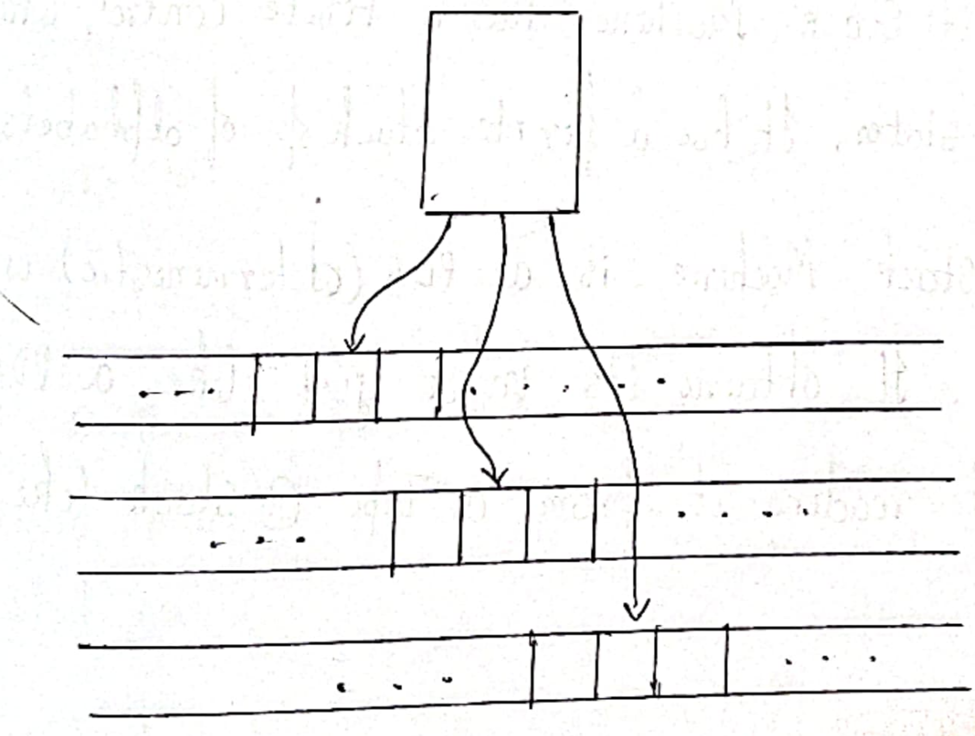
$\therefore$ The langeuage 'L' is Irregular.

→ Multitape Turing Machine:-

It is a Turing machine that consists of a Finite Control and a finite number of tapes.

Initially:-

(i) The input String is placed on the first tape
(ii) All other cells of all the tapes are Blank.
(iii) The finite control is in Initial State
(iv) The head of first tape is at the left end of the first tape
(v) The head of all other tapes are in some arbitrary position. Since, they are filled with blank, it does not matter, they all look the same.



In One Move, The Multitape TM does the following:-

(i) The control enters a new state which may be same as the previous state

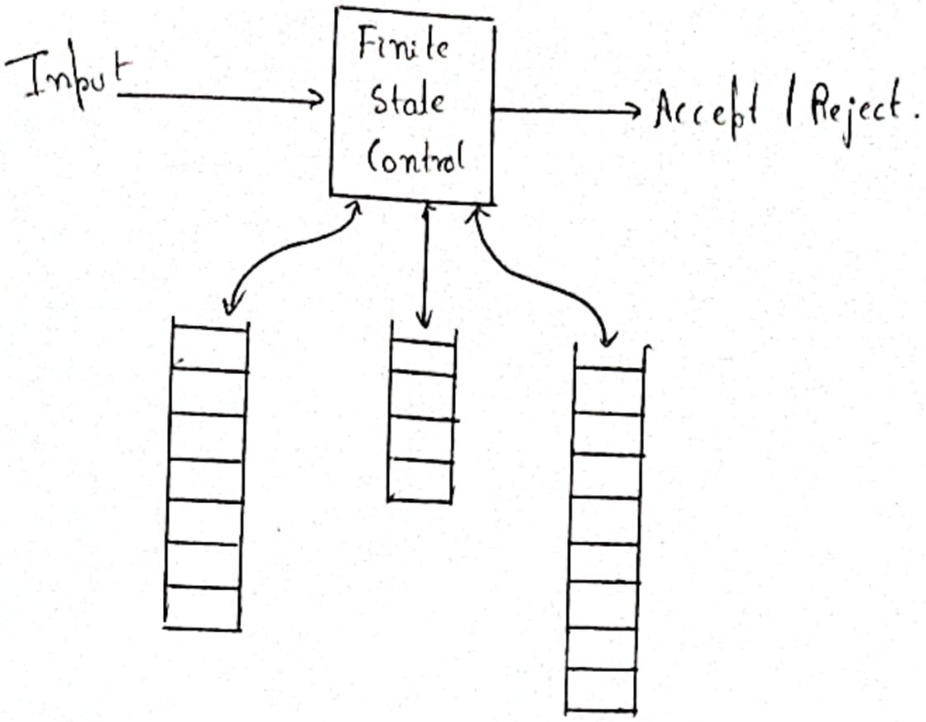
(ii) On each tape, a new tape symbol is written on the cell scanned which may be same as the symbol previously there.

(iii) Each of the tape head makes a move which may be either right, left (or) Stationery. The tape heads move independently so, different tape heads can move in different directions.

⇒ Multi Stack Machines:-

* A Multi Stack Machine has a Finite control, which is in one of its states, It has a finite stack of of alphabets.

* A K-stack Machine is a PDA (deterministic) with K-stacks. It obtains its input just like a PDA does rather that reading it from a Tape (or) stack like TM.



The move of a Multistack Machine is based on:-

(i) The state of finite control.

(ii) The input symbol read, which is chosen from an input alphabet. The Multistack Machine can make a move using ϵ input, but to make the machine deterministic it is avoided.

(iii) The top stack symbol of each Stack.

In one move, the Machine can

a) Change to a new state

b) Replace the Top symbol of each stack with a string of zero or more stack symbols. The new symbol written is generally different on each stack.

$$(q, a, X_1, X_2, \ldots X_K) = (p, \alpha_1, \alpha_2, \ldots \alpha_K)$$