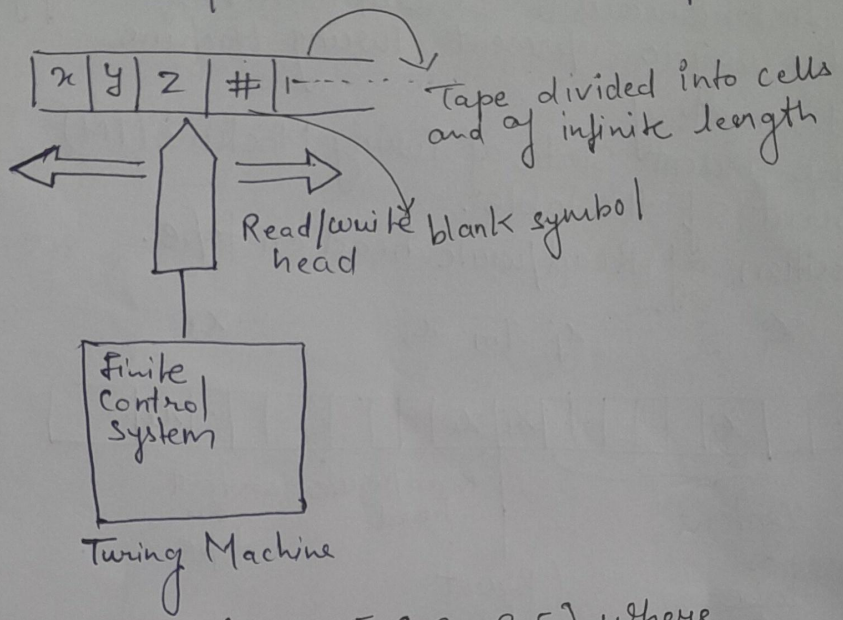
Turing Machines

A Turing Machine is a very simple machine, but logically speaking, has all the power of any digital computer. The Turing Machine (TM) is a abstract model created by Alan Turing (1912-1954), who was Mathematician by trade and the inspiration behind the computers.



Definition

A TM has a 7 tuple $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, where

1. Q is a finite, non-empty set of states
2. Σ is a non-empty set of input symbols and is a subset of Γ and $B \notin \Sigma$
3. Γ is a finite non-empty set of Tape symbols
4. $B \in \Gamma$ is the blank.
5. δ is the transition function, which maps
   $$Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$$
6. $q_0 \in Q$ is the initial state
7. $F \subseteq Q$ is the set of final states.

Representation of Turing Machine

Generally we have three technique to represent the TM :-

1. Representation by Instantaneous description
2. Representation by Transition Table
3. Representation by Transition Diagram

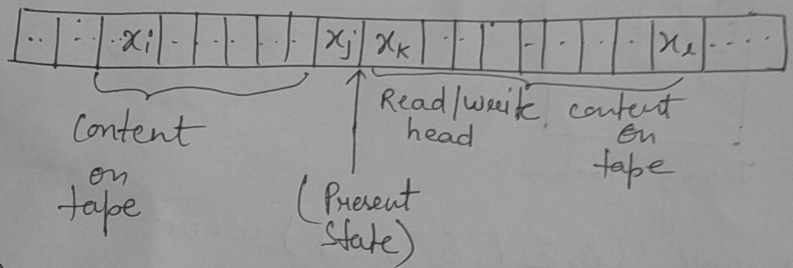
- Representation by ID

An Instantaneous Description (ID) or configration is one of the way to represent Turing Machine

It consists of

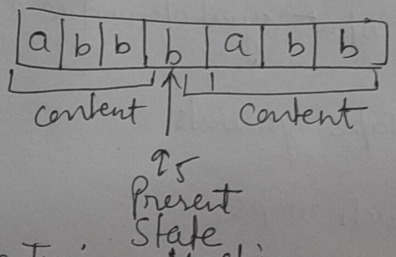
1. The present state of Turing Machine (TM)
2. Content of the table.
3. Position of Read/write head on tape.

ID $x_i \cdots\cdots x_j\ q_m\ x_k \cdots\cdots x_s$



eg →

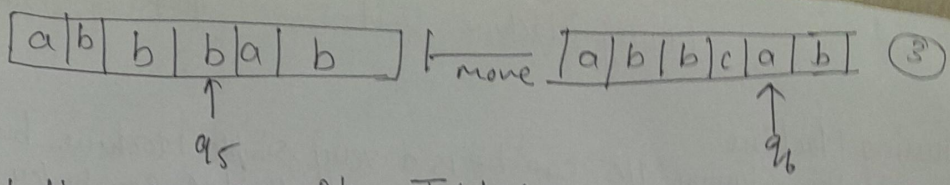
a b b q_5 b a b b



Move in Turing Machine

A move in TM can be represented as a pair of instantansous Description (ID)

$\underbrace{a\,b\,b\,q_5\,b\,a\,b}_{ID_1} \vdash_{move} \underbrace{a\,b\,b\,c\,q_6\,a\,b}_{ID_2}$



(3)

Representation by Transition Table:

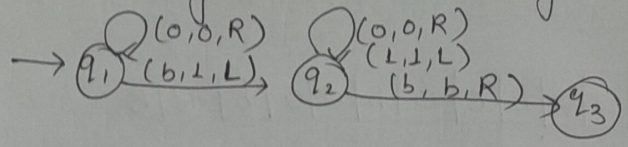
If $\delta(q,a) = (r,\alpha,\beta)$, we write $\alpha\beta r$ under a-column and q-row, so if we get $\alpha\beta r$ in the table, it means that α is written in the current cell, β gives the movement of the head (L or R) and r denotes the new state into which the turing M/c enters.

Present State	Type ~~sym~~ symbols		
	b	0	1
$\rightarrow q_1$	$1Lq_2$	$0Rq_1$	—
q_2	bRq_3	$0Lq_2$	$1Lq_2$
q_3	—	bRq_4	bRq_5
q_4	$0Rq_5$	$0Rq_4$	$1Rq_4$
$*q_5$	$0Lq_2$	—	$1Rq_4$

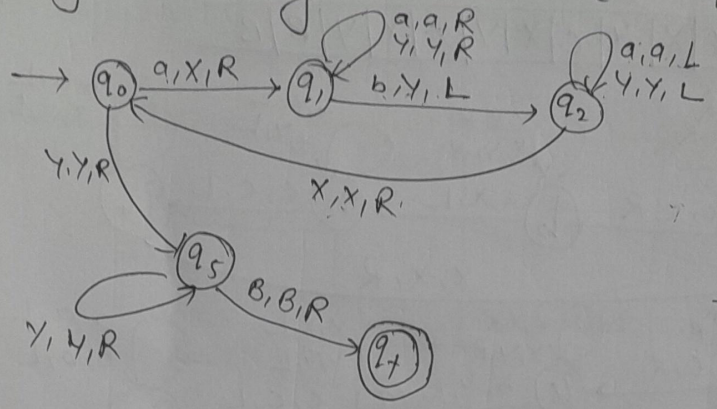
Draw the computation sequence of input string 00.

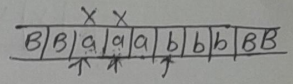
Ans. $q_100 \vdash 0q_10 \vdash 00q_1b \vdash 0q_20 \vdash q_200 \vdash q_2b001 \vdash bq_3001 \vdash bbq_401 \vdash bb0q_41 \vdash bb01q_4b \vdash bb010q_5b \vdash bb01q_200 \vdash bb0q_2100 \vdash bbq_20100 \vdash bq_2b0100 \vdash bbq_30100 \vdash bbbq_4100 \vdash bbb1q_400 \vdash bbb10q_401 \vdash bbb100q_4b \vdash bbb1000q_5b \vdash bbb100q_200 \vdash bbb10q_20000 \vdash bbb1q_20000 \vdash bbbq_210000 \vdash bbq_2b10000 \vdash bbbq_310000 \vdash bbbbq_50000.$

Representation by Transition Diagram

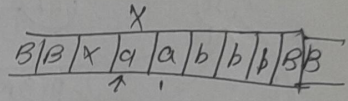


Design a Turing Machine $L = \{a^n b^n \mid n \geq 1\}$

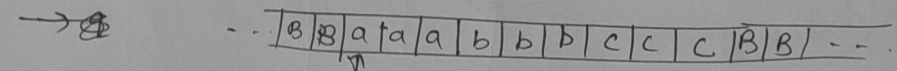




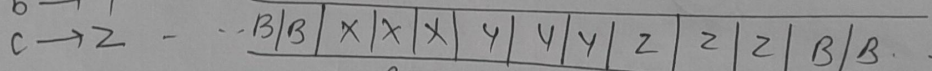
a → X
b → Y

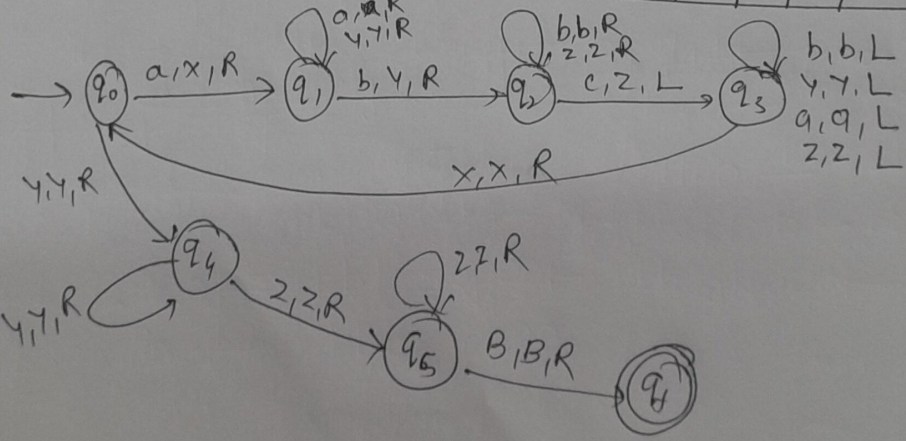


Design a Turing Machine $L = \{a^n b^n c^n \mid n \geq 1\}$



a → X
b → Y
c → Z

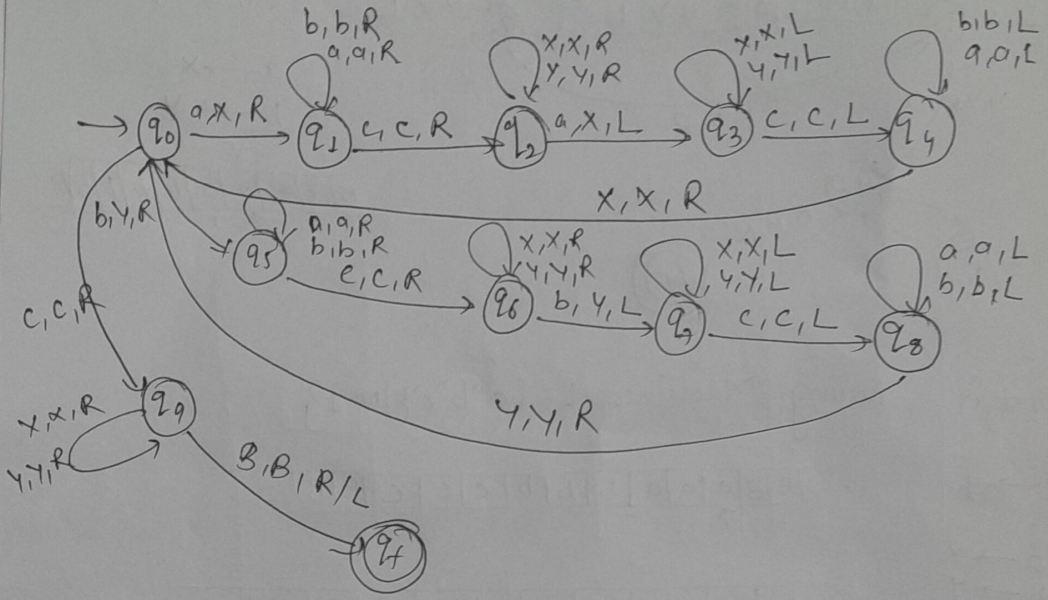




Design a Turing Machine $L = \{wcw \mid w \in (a,b)^*\}$ (5)

B	B	a	b	a	a	b	c	a	b	a	a	b	B	B
		↑												

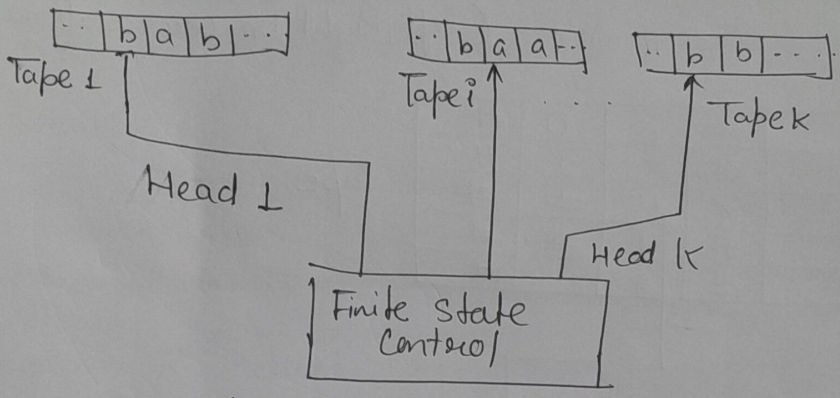
B	B	X	Y	X	X	Y	c	X	Y	X	X	Y	B	B



Variants of Turing Machine

• Multitape Turing Machine

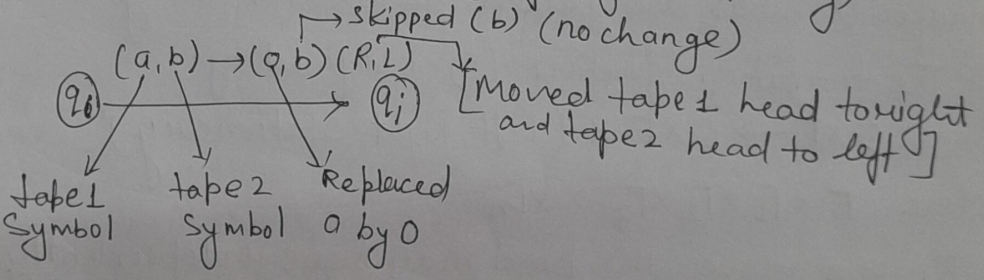
A Turing Machine with several input tapes is said to be a multitape TM. In a multitape TM each tape is controlled by its own independent R/W (Read/write) head. A multitape TM with k-tapes is shown in fig.



A multitape TM Model

Typically an n-tape TM can be defined as $M = \{Q, \Sigma, \delta, q_0, B, F\}$ where all has some meaning except transition function (δ)

Here $\delta: Q \times \Gamma^i \longrightarrow Q \times \Gamma^i \times (\text{Left, Right, Stationary})^i$

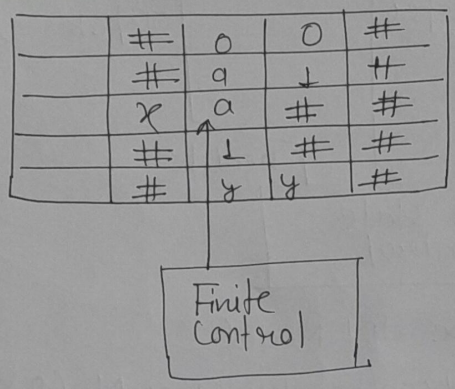


Multidimensional (k-dimensional) Turing Machine ⑦

* A turing Machine is said to be multidimensional if input tape extends infinitely in more than one dimension.
* The transition function (δ) is defined by

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times (L, R, U, D, S)$$

(Left, Right, Up, Down, stationary)



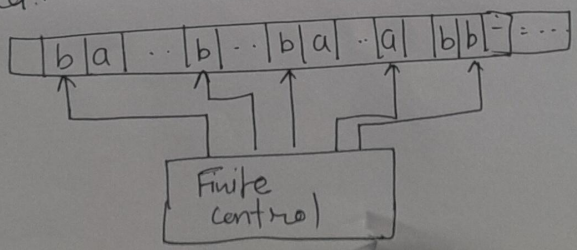
Multi-head Turing Machine

* A turing machine with single tape but several independent read/write heads are called multi-head turing Machine.
* The transition function (δ) is defined by:

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times (\text{Left, Right, stationary})^n$$

where $\Gamma = \Gamma_1 \times \Gamma_2 \times \Gamma_3 \cdots \times \Gamma_n$

* In one move, the head may move independently & their movement depends upon the state & symbol scanned by each head.



Church's Thesis

(Alonzo church) → Lambda calculus

(Alan Turing) → Turing Machine

* It says that any real-world computation can be translated into an equivalent computation involving a Turing Machine.
TM and Lambda Calculus are equivalent in power.
Algorithmically computable means computable by a turing Machine.

* The church - Turing thesis encompasses more kinds of computations than those originally envisioned such as those involving cellular automata, combinations, register Machines etc. It also applies to other kind of computations found in theoretical computer science such as quantum computing & probabilistic computing.

Decidability And Undecidability

Recursively Enumerable Language

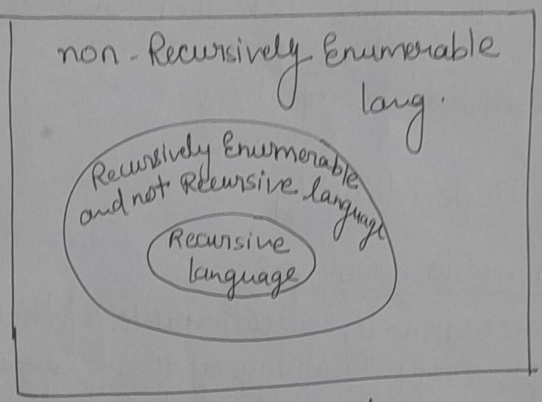
→ A language is recursively enumerable if there exists a Turing Machine accepts every string of the language and a language is recursive if there is a TM that accepts every string of the language and does not accept strings that are not in the language.
So, every recursive language is also recursively enumerable.
Hence, Statement of theorem is true.

→ These are turing recognizable also known as partially decidable.

* Recursive language

→ A language L is said to be recursive if there exists a Turing Machine which will accept all the string in 'L' & reject all the strings not in 'L'.
→ The turing Machine will halt everytime & give an answer (accepted or rejected) for each & every string input.
These are turing decidable.

Note An undecidable language may sometimes be partially decidable but not decidable. If a lang is not even partially decidable, then there exist no TM for that language.



~~For~~ From these two language following observation are made:
① Every recursive language is recursively enumerable lang
② Not all recursivelly language ~~to~~ are recursive that means every TM which accepts the enumerable language must have some wards for which its loops forever.

Halting Problem

Given a program / algorithm will even halt or not?

* Halting means that the program on certain input will accept & halt / reject & halt & would never go into an infinite loop.

→ Basically, halt means to terminate. In terms of TM, will it terminate when run on some machine with some particular given input string.

→ No, we cannot design a generalize algorithm which can appropriately say that given program will ever halt or not.

→ This is an undecidable problem because we cannot have an algorithm which will tell us whether a given program will ~~not~~ halt or not in a generalized way.

The best way is to run the program & see whether it halts or not.

Post Correspondence problem

The post's correspondence problem consists of two lists of string that are of equal length over the input. The two lists are $A = w_1, w_2, w_3 \cdots w_n$ and $B = x_1, x_2, x_3 \cdots x_n$ then there exists a non empty set of integers $i_1, i_2, i_3 \cdots$ in such that $w_1, w_2, w_3 \cdots w_n = x_1, x_2, x_3 \cdots x_n$"

eg → Consider the correspondence system as given below. $A = (b, bab^3, ba)$ and $B = (b^3, ba, a)$. The input set is $\Sigma = \{0, 1\}$.

Sol A solution is 2, 1, 1, 3. That means $w_2 w_1 w_1 w_2 = x_2 x_1 x_1 x_3$

The constructed string from both lists is $bab^3 b^3 a$.

w_2	w_1	w_1	w_3
bab^3	b	b	ba
x_2	x_1	x_1	x_3
ba	b^3	b^3	a

eg→

B	A	CA	ABC
CA	AB	A	C
①	②	③	④

Sol

A	B	CA	A	ABC
AB	CA	A	AB	C

→ ABCAAABC
→ ABCAAABC
is the solution of instance of PCP.

Q) Let $\Sigma = \{0,1\}$. Let X & y be list of three string each defined as follows.

	list X	list Y
i	w_i	x_i
1	1	111
2	10111	10
3	10	0

$\frac{1}{111} \rightarrow w_1 \rightarrow x_1$

$\frac{10111}{10} \rightarrow w_2 \rightarrow x_2$

$\frac{10}{0} \rightarrow w_3 \rightarrow x_3$

$\frac{10111}{10} \; \frac{1}{111} \; \frac{1}{111} \; \frac{10}{0} \longrightarrow 10111111110$

$\frac{1}{111} \; \frac{1}{111} \; \frac{1}{111} \; \frac{10}{0} \longrightarrow 10111111110$

$\therefore \boxed{w_2 w_1 w_1 w_3 = x_2 x_1 x_1 x_3 = 10111111110}$

Q)

	list X	list Y
i	w_i	x_i
1	10	101
2	011	11
3	101	011

$\frac{10}{101}$

$\frac{011}{11}$

$\frac{101}{011}$

Possibility ① $\frac{10}{101}\ \frac{10}{101} = 1010\ 10101$ ✗ (13)

Possibility ② $\frac{10}{101}\ \frac{011}{11} = \frac{10011}{10111}$ ✗

Possibility ③ $\frac{10}{101}\ \frac{101}{011}\ \frac{101}{011}\ \frac{101}{011}\ \ldots$

$= \frac{10101101101}{10101101011}$ .... ✗

This will continue & goes to infinity. And does not come to an end / halt. So this particular PCP cannot ~~be~~ be solved.

Universal Turing Machine M/C (UTM)

→ It simulates a turing Machine
→ It can be considered as a subset of all the turing Machines, it can match or surpass other turing Machine including itself.
→ It is like a single turing Machine that has a solution to all the problems that is computable.
→ It contains a turing Machine description as input along with an input string, runs the turing Machine on the input & returns a result.
→ It is a recognize (not a decider) for a TM as it may accept / reject / loop for the particular string.

~~Halting Problem~~

~~Given a program / algorithm will ever halt or not?~~

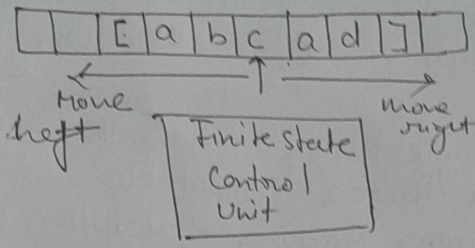
Linear Bounded Automaton (LBA)

→ A linear Bounded Automaton (LBA) is the restricted model of turing machine.
→ It is a non-deterministic, multitrack turing machine with bounded finite length of the tape.
→ In LBA, the computation is restricted to the bounded tape area. The tape head can be not move outside the available tape.

LBA consists of 8 tuples.

Q → Finite set of states
Σ → input alphabet
γ → tape alphabet
q_0 → initial state
F → set of finial states
$[$ → ~~left~~ left end marker
$]$ → Right end marker
δ → transition function.

~~Turing Machine as Computer of Integer Functions~~

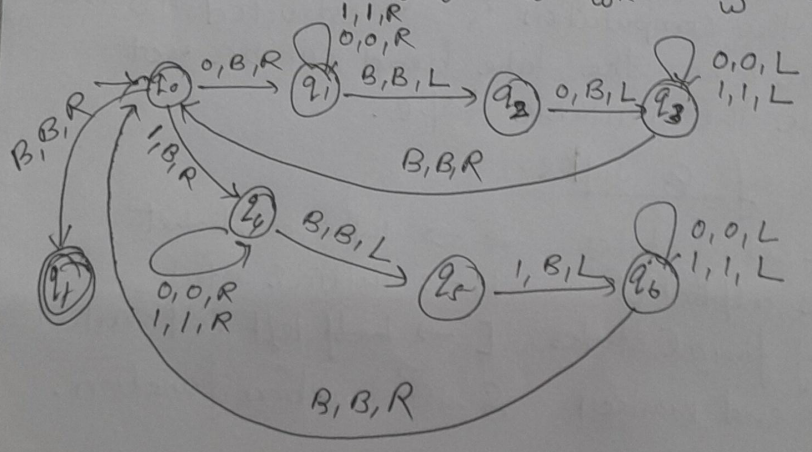


Block diagram of LBA

* Language accepted by LBA
  - (i) $\{L = a^n b^n c^n \mid n \geq 1\}$
  - (ii) $\{L = a^{L^n} \mid n \geq 0\}$
* LBA accepts CSL (Context sensitive language)
* LBA is more powerful than NPDA but less ~~Power~~ powerful than TM.

Q) Design a turing Machine for $L = ww^R$ where $w \in (01)^*$ $\{0,1 \to B\}$

Sol $L = ww^R \Rightarrow \{\underbrace{0}_{w}\underbrace{110}_{w^R}, \underbrace{100}_{w}\underbrace{001}_{w^R}, \underbrace{1010}_{w}\underbrace{0101}_{w^R}$ and ~~B, SHIFE~~



Turing Machine as computer of Integer Functions (16)

An integer function is a function which takes integer as input and returns the result an integer.
to seperate the arguments blank is used.

eg →

$f(x) = x + y$ 4+2

tape $0^x B 0^y$