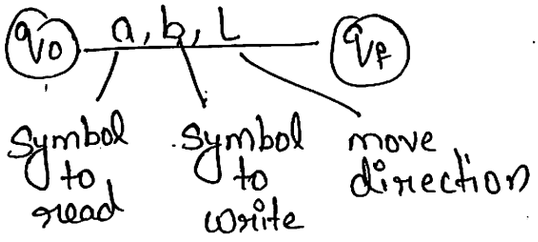
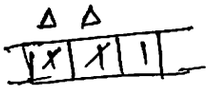
Rules of operation

1) Read the current symbol
   ↓
   update (write) the same cell
   ↓
   move exactly one cell (left or right)

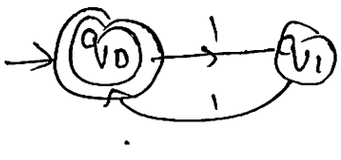


→ Construct T.M for string having even no. of 1s over $\Sigma = \{1\}$

$L = \{11, 1111, ---\}$

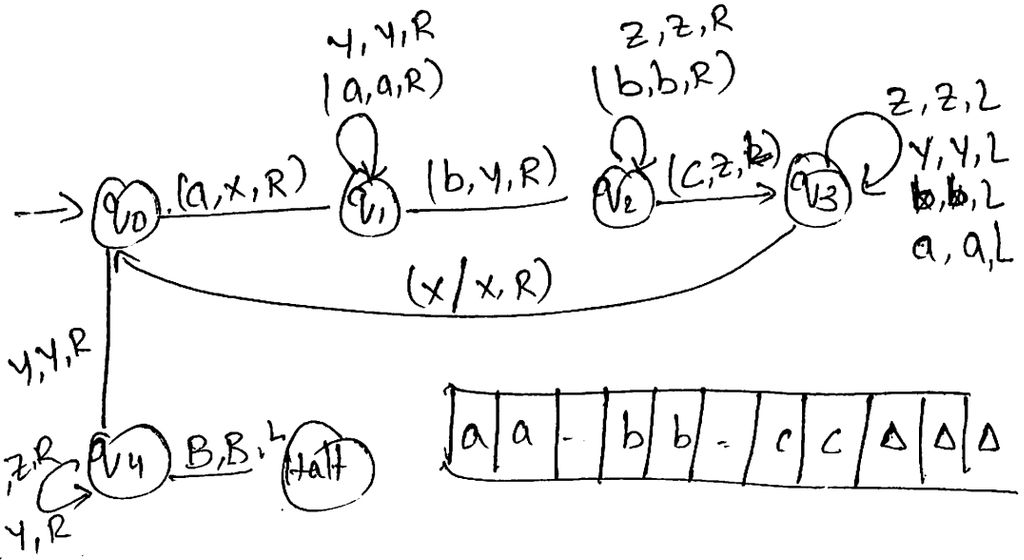


F.A



→ Construct T.M for string having $a^n b^n c^n$, $n \geq 1$

$L = \{abc, aabbcc, aaabbbccc, ---\}$



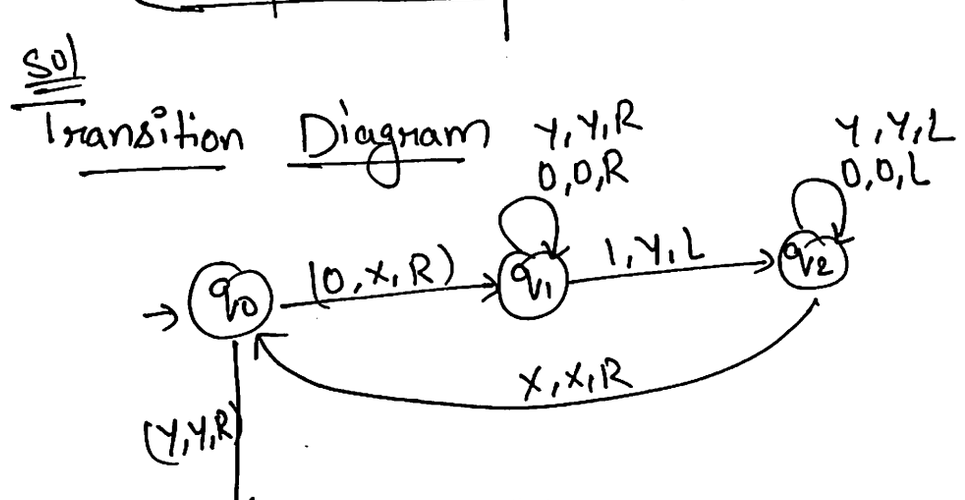
→ Construct a turing machine for $L=\{0^n1^n \mid n\geq 1\}$

Transition function is given as:

States	0	1	X	Y	B
q_0	(q_1, X, R)			(q_3, Y, 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_3, Y, R)	Halt

Sol

Transition Diagram



Consider 0011

B	0	0	1	1	B

$\delta(q_0, 0) = (q_1, X, R)$

B	X	0	1	1	B

$\delta(q_1, 0) = (q_1, 0, R)$

B	X	0	1	1	B

$\delta(q_1, 1) = (q_2, Y, L)$

B	X	0	Y	1	B

$\delta(q_2, 0) = (q_2, 0, L)$

B	X	0	Y	1	B

$\delta(q_2, X) = (q_2, X, R)$

B	X	0	Y	1	B

$\delta(q_0, 0) = (q_1, X, R)$

B	X	X	Y	1	B

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

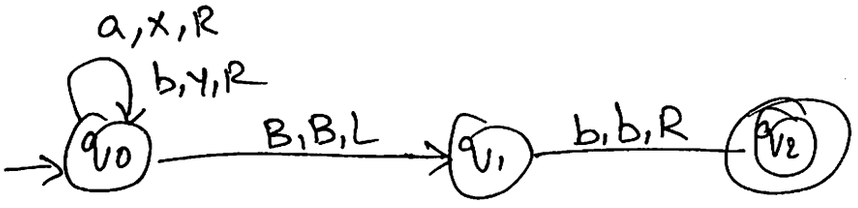
B	X	X	Y	1	B

$\delta(q_1, 1) = (q_2, Y, L)$

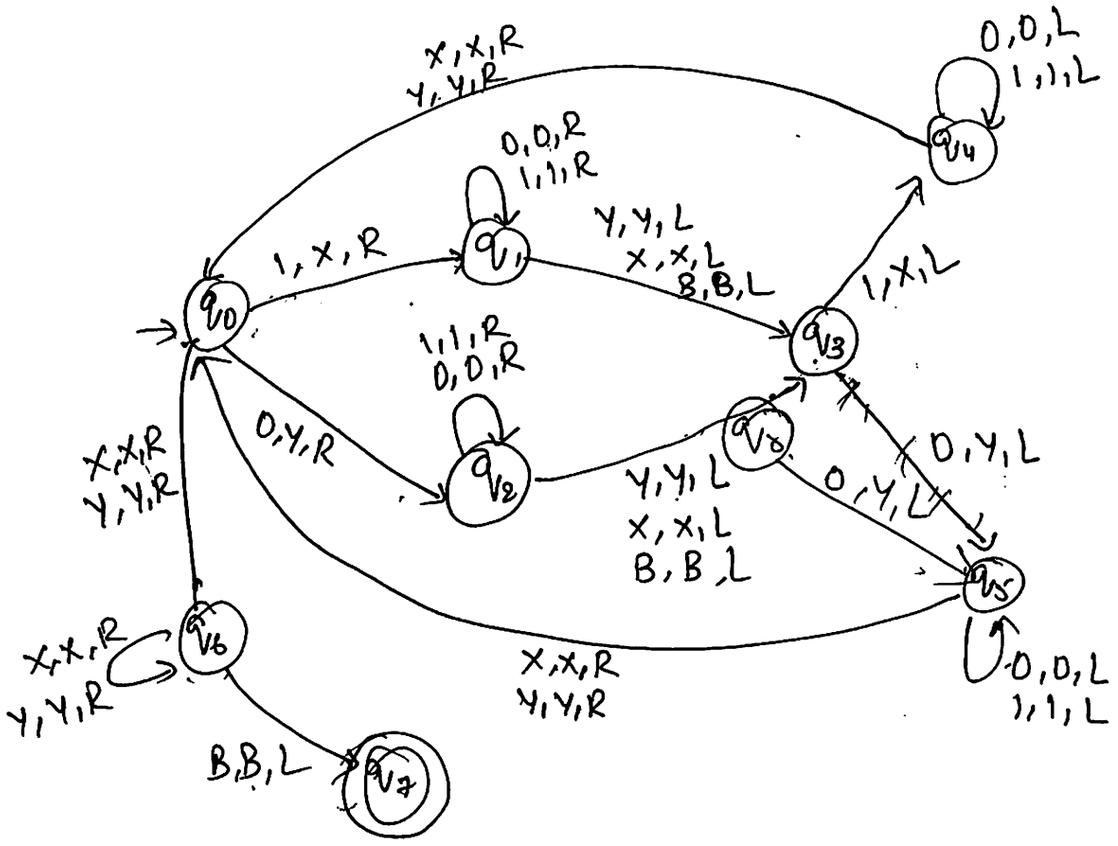
B	X	X	Y	Y	B

Continue the Process until it halts

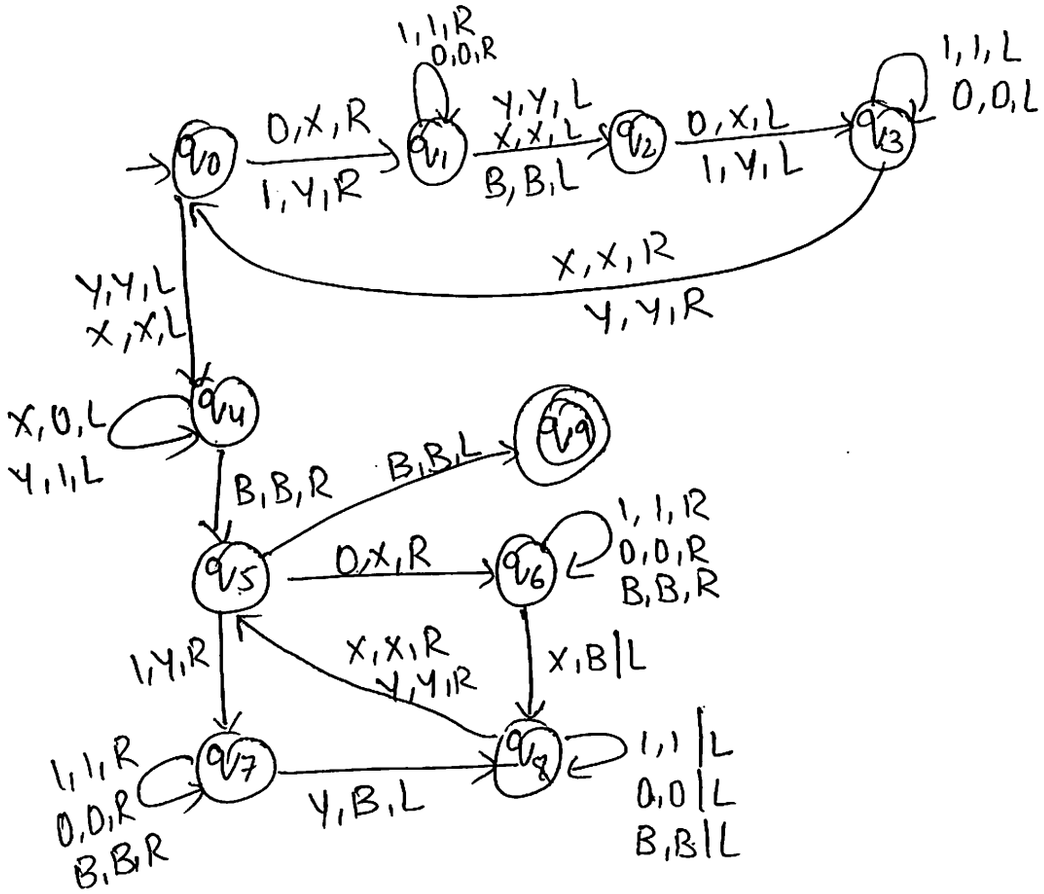
→ $\Sigma = \{a, b\}$ & String ends with 'b'



→ $L = \{ ww^R \mid w \in \{0,1\} \}$



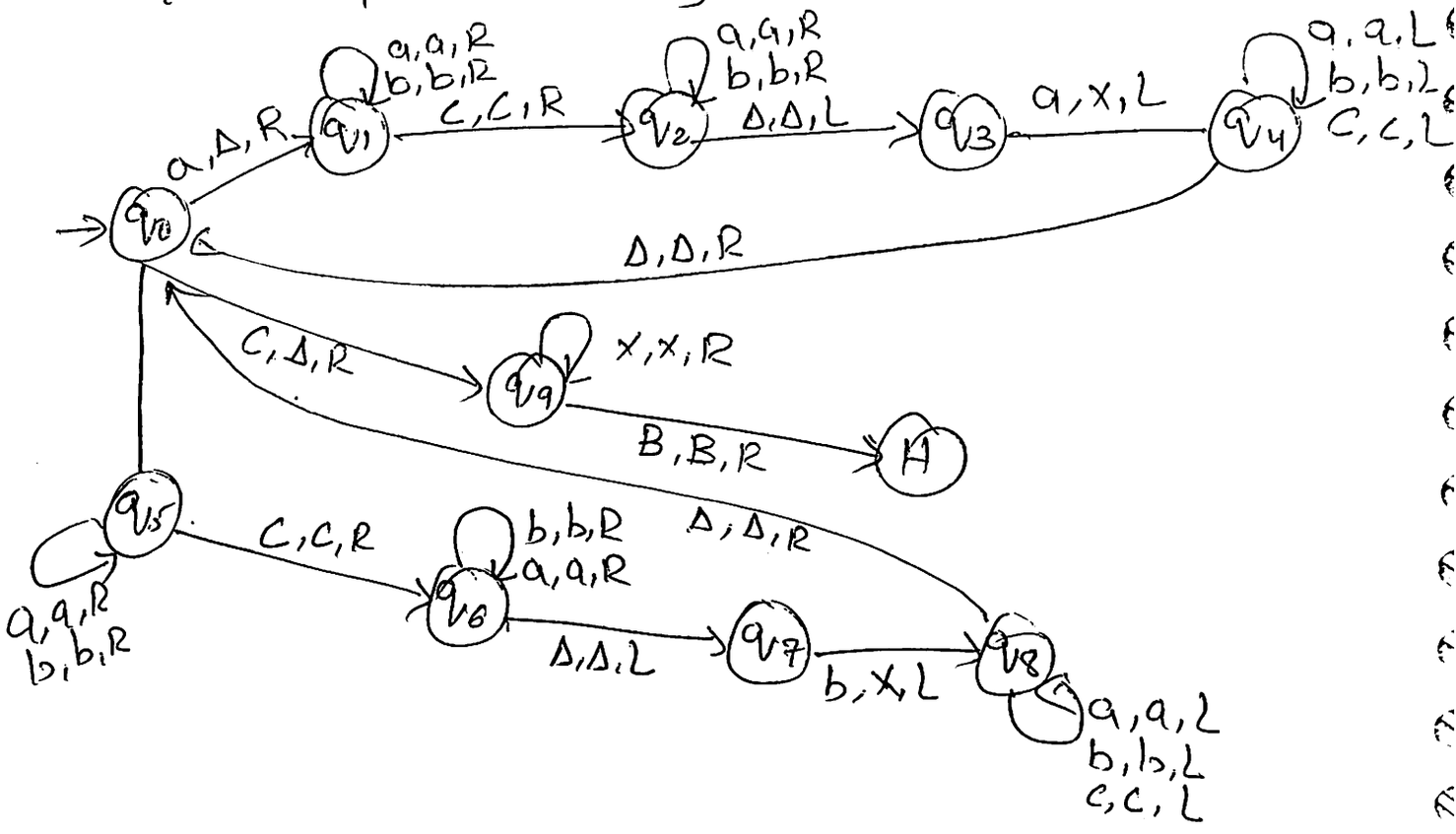
→ T.M for $L = \{ww \mid w \in \{0,1\}\}$



i/p - aba.aba | 010.010

010 010	⇒ XYX.XYX
X10010	⇒ XY0.XYX
X1001X	= X10.XYX
XY001X	BQ10.XYX
XY00YX	X10.BYX
XY0YYX	XY0 BBX
XYXYYX	XYX BBB

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



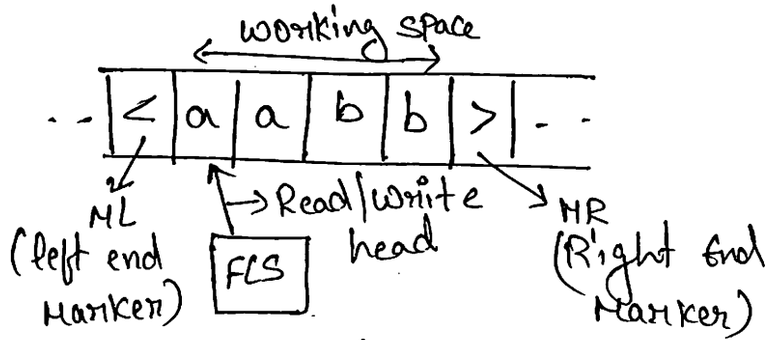
Church's thesis (or) church turning thesis:

Linear Bounded Automata [LBA]

A turing machine that uses only the tape space occupied by the i/p string is called as LBA. A LBA is described by 8 tuple notation

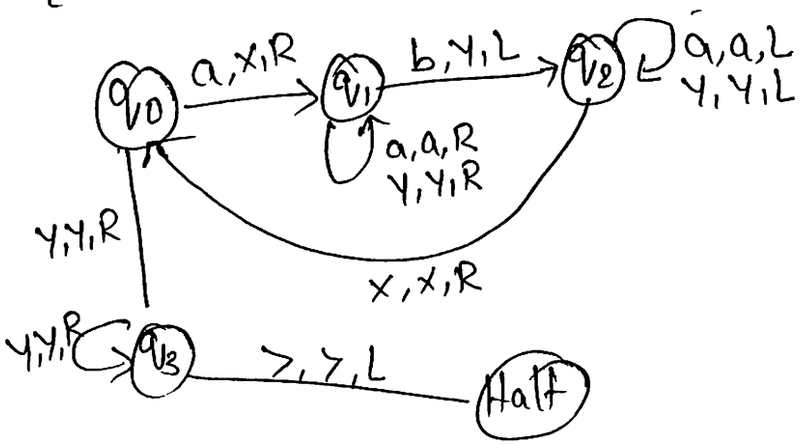
i.e $M = (Q, \Sigma, q_0, \delta, F, \Gamma, \angle, \rangle)$

$\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$



i) $L = \{a^n b^n \mid n > 0\}$

$L = \{ab, aabb, aaabbb - - -\}$



a a b b
X a Y b
X X Y b
X X Y Y

It is similar to multi stack turing machine but the only the difference between this is in place of each stack, there is a counter, which contains non -ve integers each move of counter machine depends on its state, current i/p symbol.

In one move of counter machine, CM can

a) Change state
b) add or subtract one from any one of its counters.

'-' ve counters are not allowed at all

→ Counter Machine is similar to restricted multi stack Machine which following restrictions.

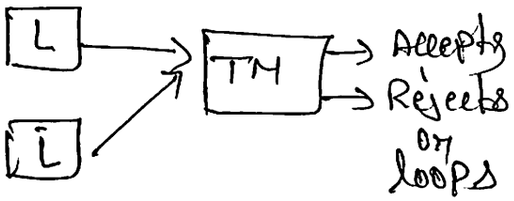
i) There are only two stack symbols Z_0, X
ii) Z_0 is bottom of stack marker. It is initially on each stack.
iii) Replace Z_0 only by string of the form $X^i Z_0$; $i \geq 0$
iv) Replace X only by X^i, $i \geq 0$ i.e Z_0, appears on the bottom of each stack & all other stack symbols are X
v) Every language accepted by CM is recursively enumerable.

Recursively enumerable language:

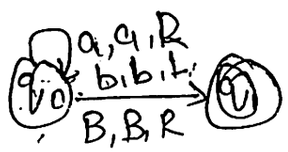
A language L is recursively enumerable then there is a turing machine that accepts every word in L & either rejects or loops for every string in Lang $\bar{L}$ (i.e complement of L)

Accept (M) = L

Rejects (M) or loops (M) = $\bar{L}$



eg:-



L = aa^*

The language accepted by this TM is aa^*

1) For the i/p aaa, it accepts
2) For the i/p aba, it loops

Recursive languages:

let L be a language L is said to be recursive language if there exists turing machine 'T' that accepts every string or word in L' & rejects every word or string in L'.

Accept (M) = L

Reject (M) = L'

Church's Hypothesis:

The assumption that the intuitive notion of computable functions can be identified with partial recursive functions

→ However this hypothesis cannot be proved

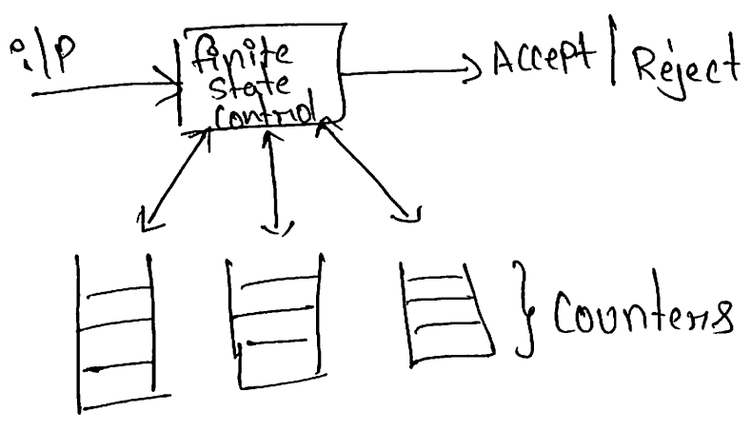
→ Computability of recursive functions is based on the following assumption

i) Each elementary function is computable.

ii) let f be computable function & A be another function which can be obtained by applying an elementary operations to f, then A becomes computable function.

iii) Any function becomes computable if it is obtained by rule i & ii

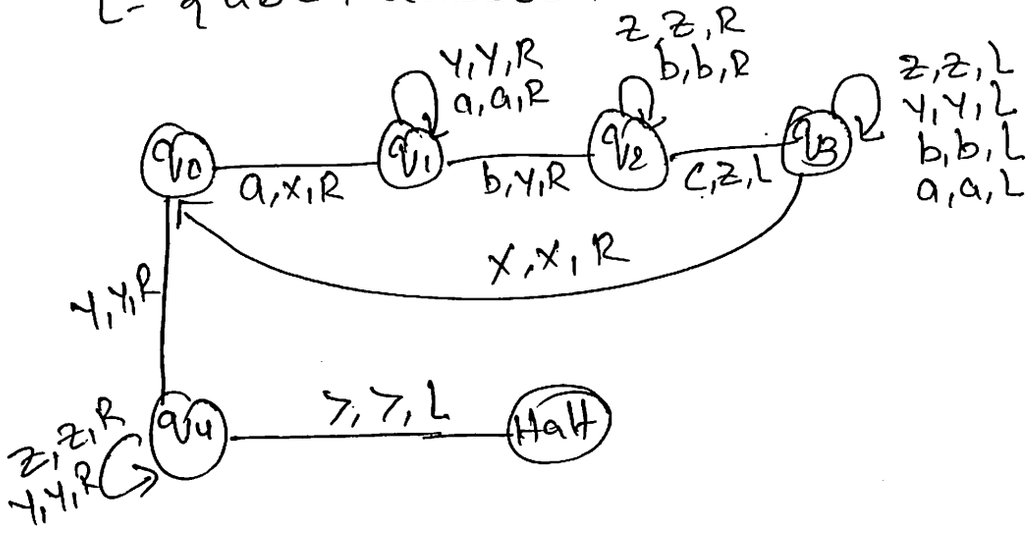
Counter Machine:



Counter Machine

a) $L = \{a^n b^n c^n \mid n > 0\}$

$L = \{abc, aabbcc, aaabbbccc \ldots\}$



Content Sensitive languages

Content sensitive grammar [CSG]

$G = (V, T, P, S)$

The production rules are in the following form

$\alpha \to \beta$ where α, β are set of terminals & non terminals. $|\beta| \geq |\alpha|$

→ $\in$ Productions are not allowed in CSG

① Find out the lang for the foll grammar

$S \to aSBC \mid aBC$

$CB \to BC$

$aB \to ab$

$bB \to bb$

$bC \to bc$

$cC \to cc$

$S \rightarrow aBC$ [$aB \rightarrow ab$]

abC [$bC \rightarrow bc$]

abc

$S \rightarrow aSBC$ [$S \rightarrow aBC$]

$\rightarrow aaBCBC$ [$aB \rightarrow ab$]

$\rightarrow aabCBC$ [$CB \rightarrow BC$]

$\rightarrow aabBCC$ [$bB \rightarrow bb$]

$\rightarrow aabbCC$ [$bC \rightarrow bc$]

$aabbcC$ [$cC \rightarrow cc$]

$aabbcc$

$\therefore L = \{abc, aabbcc, \ldots\}$

(2) Find the lang for the foll grammar

$S \rightarrow ABC \mid ABCS$

$AB \rightarrow BA$

$BC \rightarrow CB$

$BA \rightarrow AB$

$CB \rightarrow BC$

$CA \rightarrow AC$

$A \rightarrow a$

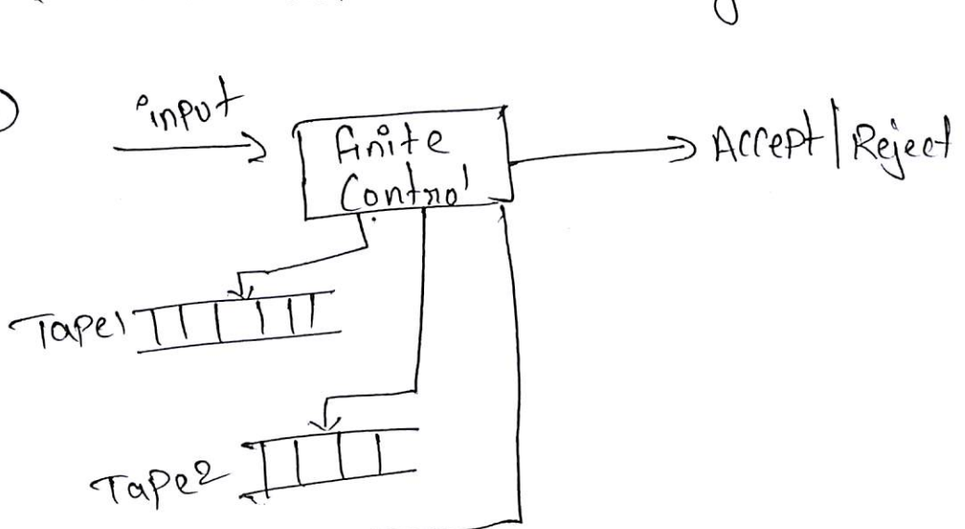
$B \rightarrow b$

$C \rightarrow c$.

$\therefore L = \{abc, bac, acb, \ldots\}$

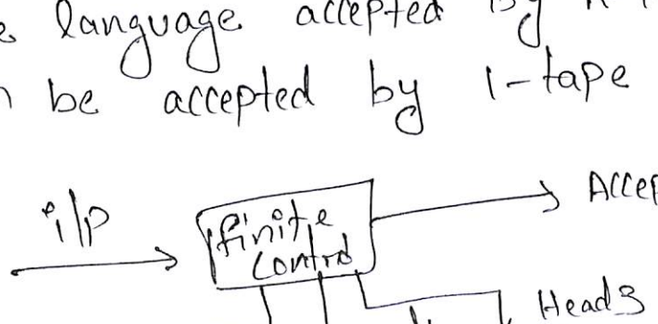
Types of turing machine:

1) Turing machine with multiple tapes
2) Turing machine with one tape mutiple Heads
3) Turing machine with Two Dimensional tape
4) Turing machine with infinite tape.
5) Non - deterministic turing machine

① 

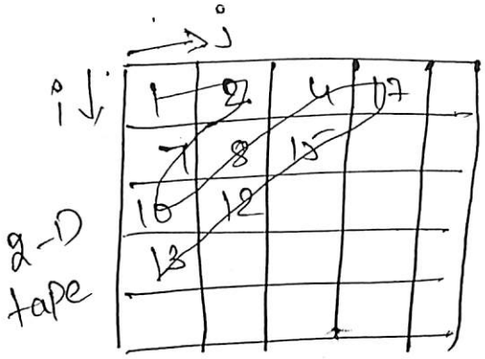
It has finite control, more than one tapes having its own read write head.

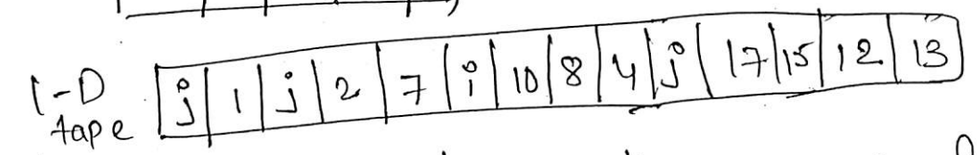
The language accepted by n-tape turing machine can be accepted by 1-tape turing machine

② 

There are n-heads but in state only one head can move. This type of turing machine are powerful as one tape turing machine

(3)



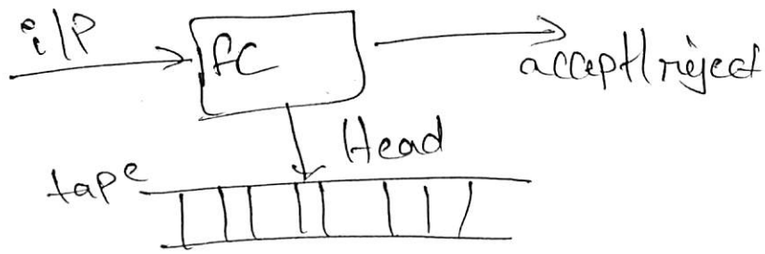


It is divided into small squares formed due to corresponding rows & collems

→ Turing machine with 1-D tape is equally powerful to that of 2-D tape

→ The head of 2-D tape moves one square updown left right

(4)



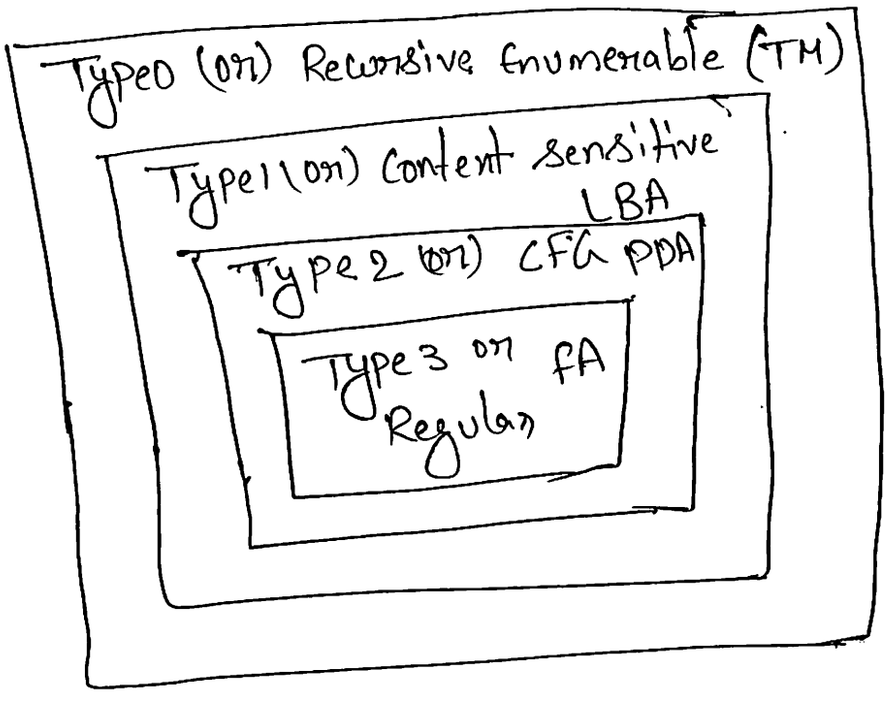
Turing machine that have one finite control & one tape which extends infinetly both directions. This type of turing machine is powerful as one-tape TM whose tape has left end.

(6) Similar to NFA, for any state & any i/p symbol. It can take any action from a set rather than a definite Pre-determined action even in some situations it may take different actions at different times.

Eg: Turing machine which accepts language L

$L = \{ww \mid w \in (a+b)^*\}$ is non-deterministic TM

Chomsky Hierarchy of languages:



One way of exhibiting relationship between languages in chomsky Hierarchy. Initial classification into 4 categories of language

→ Type0
→ Type1
→ Type2
→ Type3.