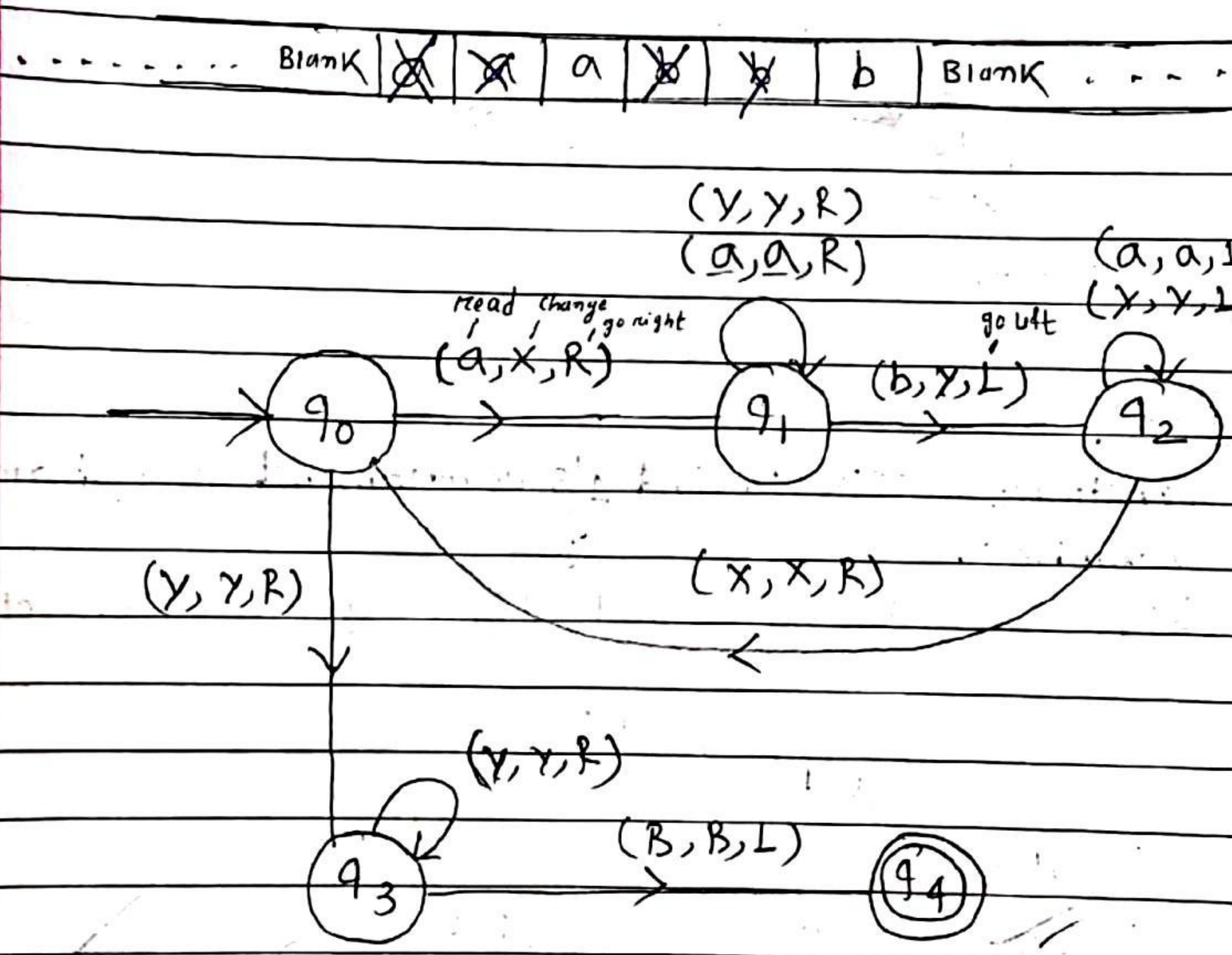
{ FA + 1 stack → PDA.
{ FA + 2 stack → TM / FA + tape → TM.

Turing Machine

→ TM is not capable of accepting 'ϵ'.
→ TM is capable of Reading, changing (writing), moving Right and moving left.
→ When the string is not in language then halt in non-final state.

Ex-1 Construct a TM that accept language $\{a^n b^n / n \geq 1\}$.

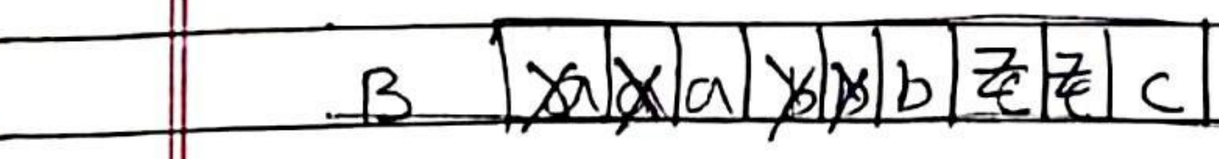
→



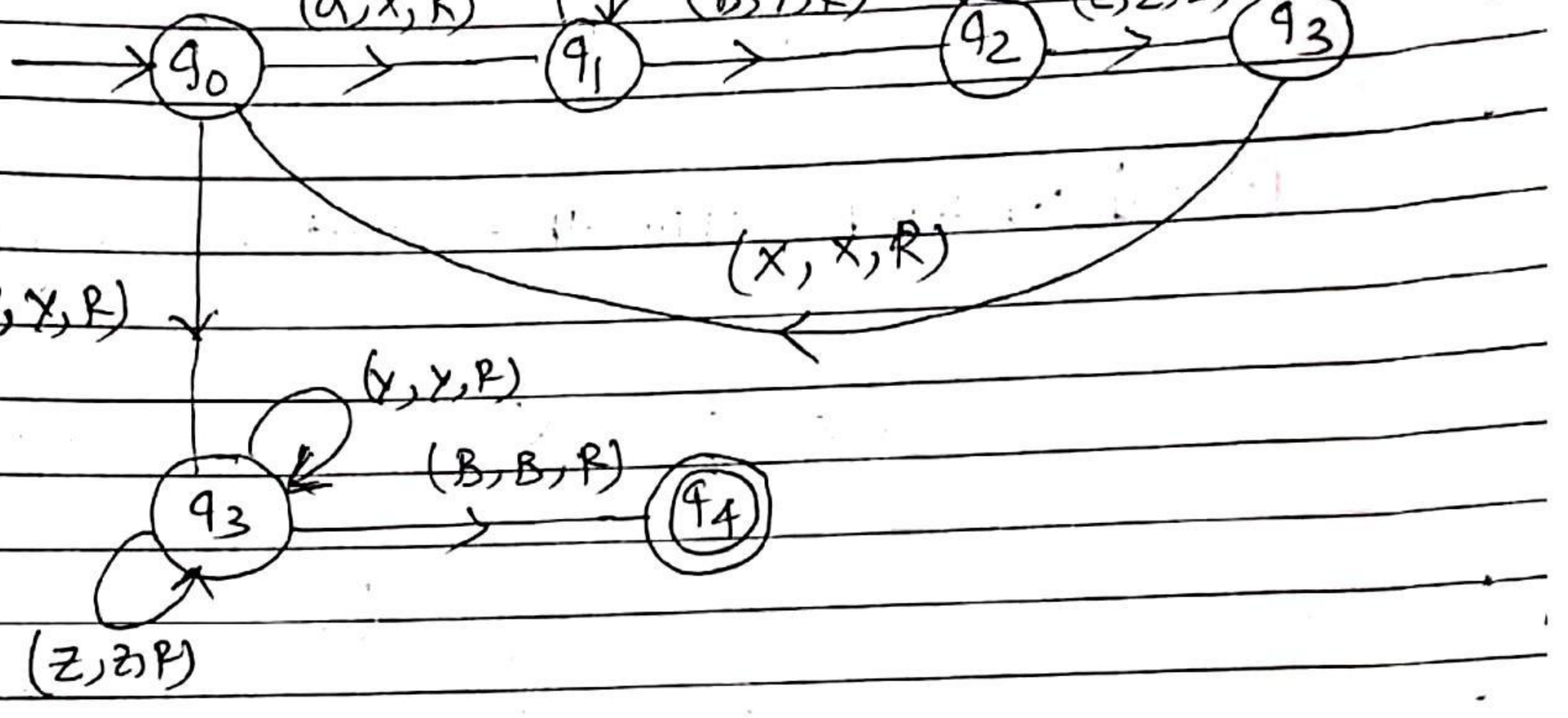
Ex-2 Construct a TM that accept language $\{a^n b^n c^n / n \geq 1\}$

→ ~~state transaction diagram~~ –
→ If any string not in the ~~to~~ given language, then ~~it~~ it will halt in non final state.

state transaction diagram ~~of~~ –



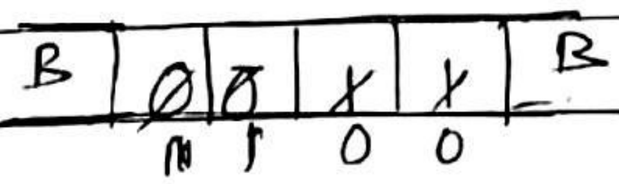
→ If the string belongs to language then it not halt.

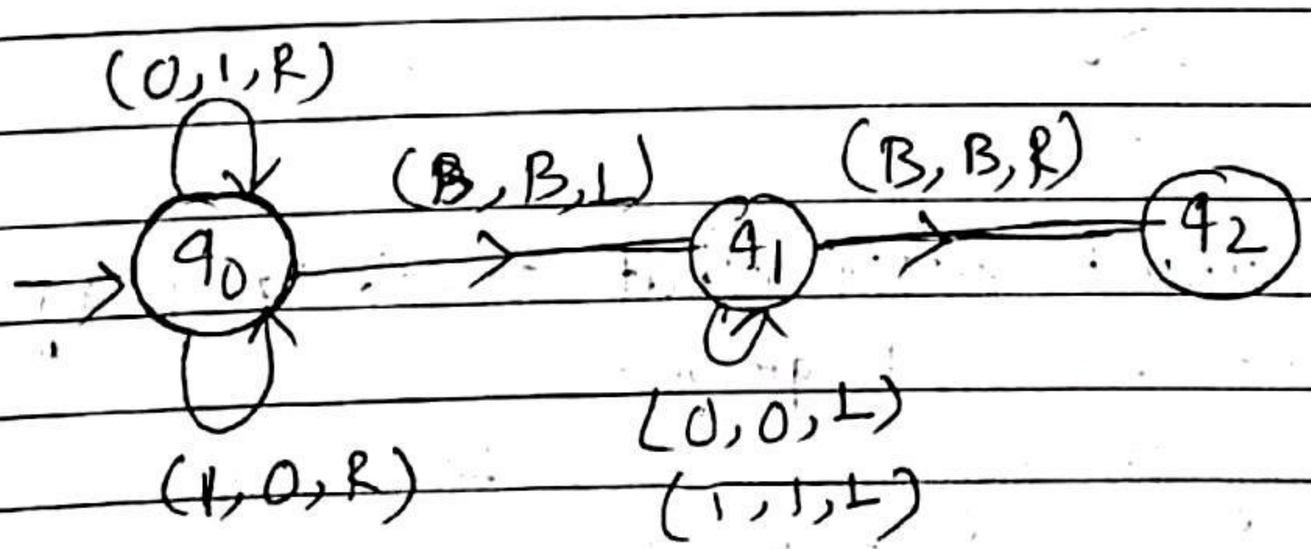


Ex-3

Construct a TM to find 1's complement of a binary number.

→ TM also has a capacity to act as Transducer.





Ex-4

Construct a TM to find 2's complement of a binary number.

→

0 1 1 1 0 0 0 — 0 0 1 1 0 0

1's - 1 0 0 0 1 1 1 — 1 1 0 0 1 1

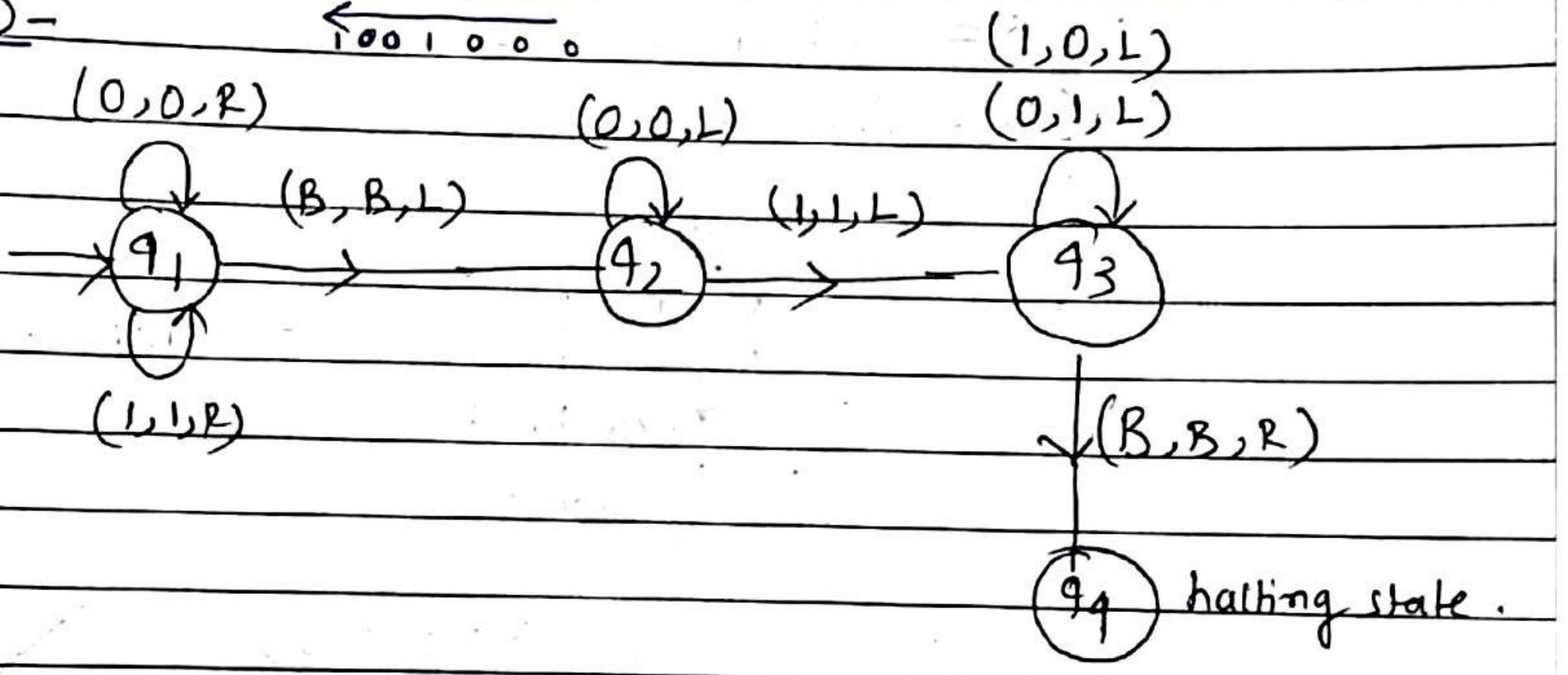
1 — 1

2's 1 0 0 1 0 0 0 ← 1 1 0 0 1 1

B B B 0 1 1 1 1 0 0 0 B B B

1 0 0 1 0 0 0

STD-



Ex-5 TM as adder-

Add two numbers using TM.

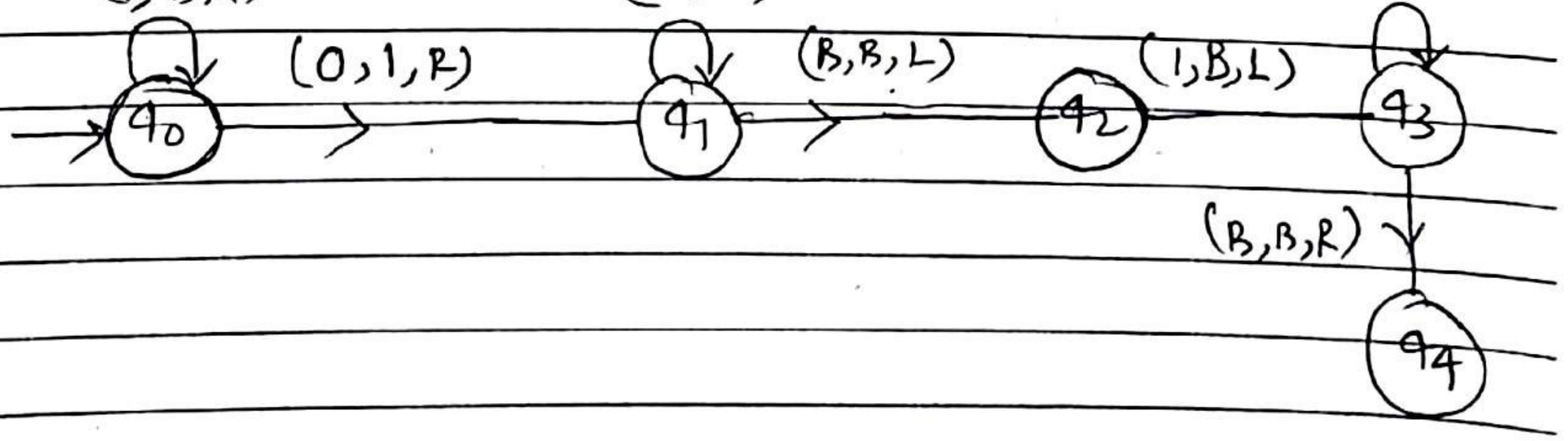
→ $3 = (111)$, $|x| = 3$

$4 = (1111)$, $|x| = 4$

$(x) + (y)$

B B B | | | 0 | | | | B B B

(B,B,L)



atlantis
Date
Page

The Standard turing machine -

$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q = Set of states.

Σ = Input alphabet. $\Sigma \subseteq M$ $\Sigma = \{a, b\}$

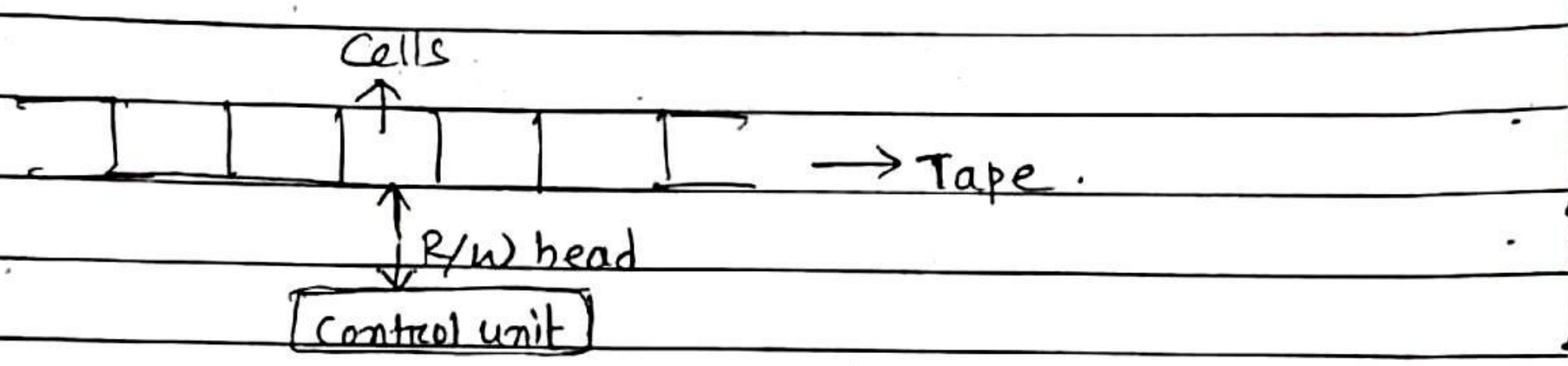
Γ = Tape alphabet. $\Gamma = \{a, b, x, y, B\}$

δ = Transition function.

$B \in \Gamma$ = is used to represent ~~esent~~ Blank.

$q_0 \in Q$ = initial state.

$F \subseteq Q$ = ~~Fin~~ set of final state.



$\delta : (Q \times \Gamma) \longrightarrow (Q \times \Gamma \times \{L, R\})$

↑ left/Right.

Summary -

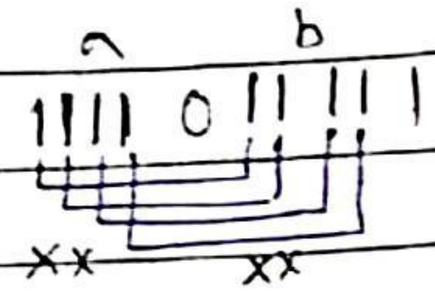
→ The tape is unbounded, so any of Right and left moves possible.

→ It is deterministic, i.e atmost one move for each configuration.

→ No special i/p or o/p file.

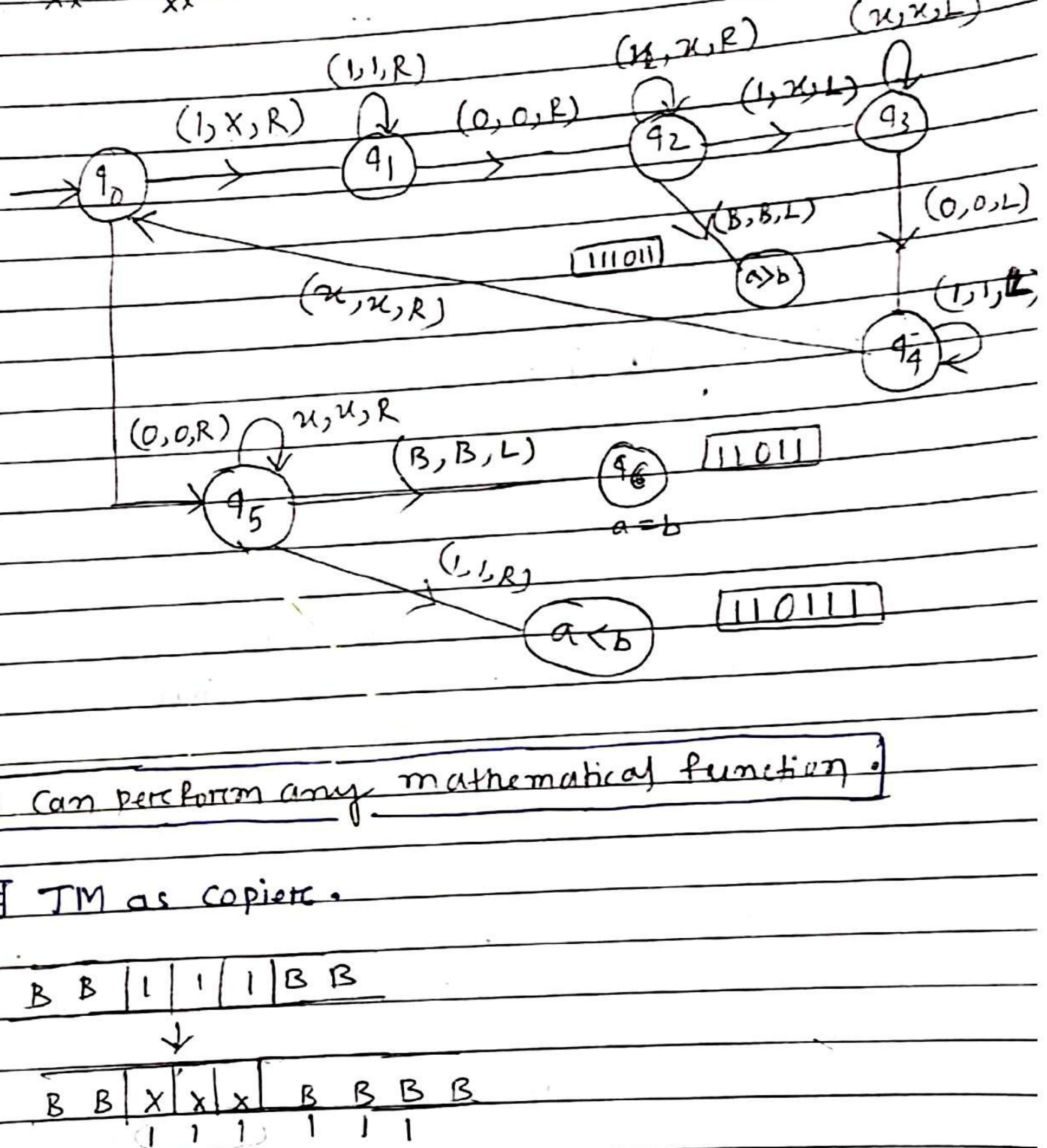
a=b a<b a>b

Ex-6 TM as Comparator.



a = b

B	*	*	0	*	*	B

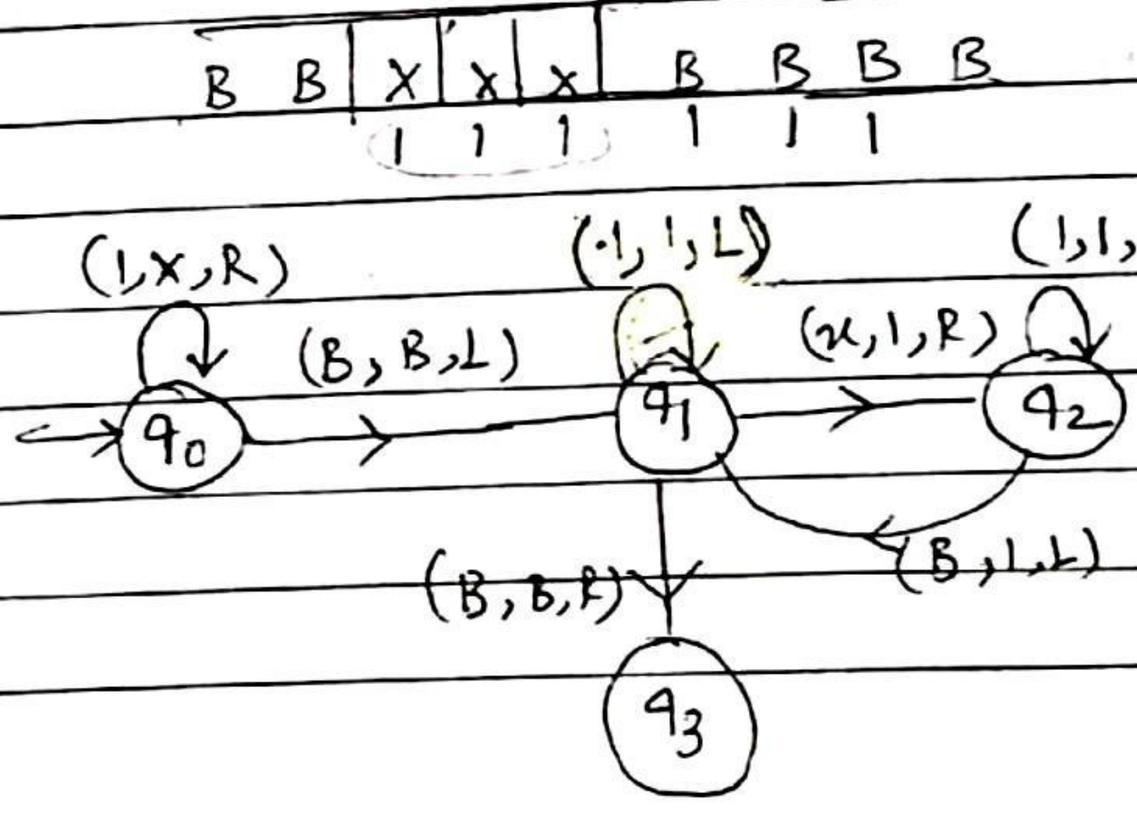


** TM can perform any mathematical function.

Ex-7 TM as copier.

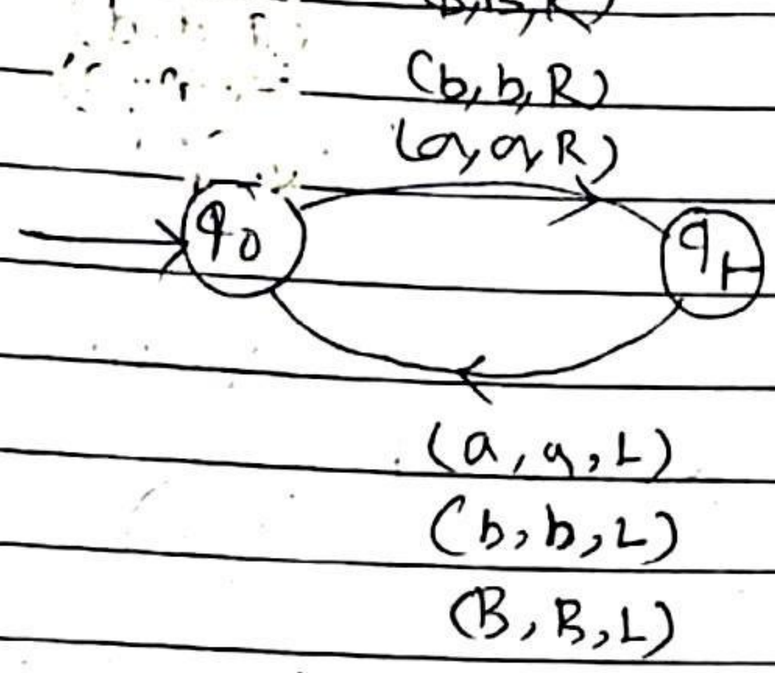
→ B B | 1 | 1 | 1 | B B

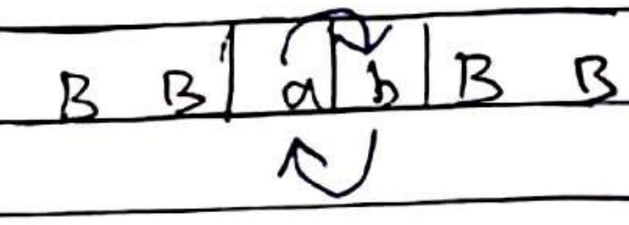
B	B	X	x	x	B	B	B	B
		1	1	1	1	1	1	



→ Using for copying.

[Ex-8] Non halting TM:





→ all the turing machine might not halt.

not for gate

• Turing Thesis:-

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

Some arguments why turing thesis is accepted as definition of mechanical computation or computer-

a) Anything that can be done by existing digital computer can also be done by 'TM'.

b) No one has yet been able to suggest a problem, solvable by what we intuitively consider an algorithm, for which a turing machine program cannot be written.

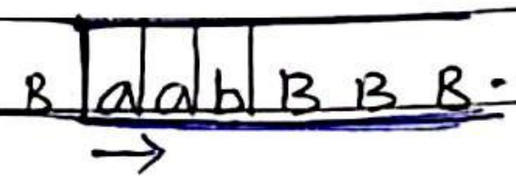
c) Alternative models have been proposed, but non of them are more powerful than the turing machine model.

→ Standard TM as powerful as modified TM.
To accept number of language.

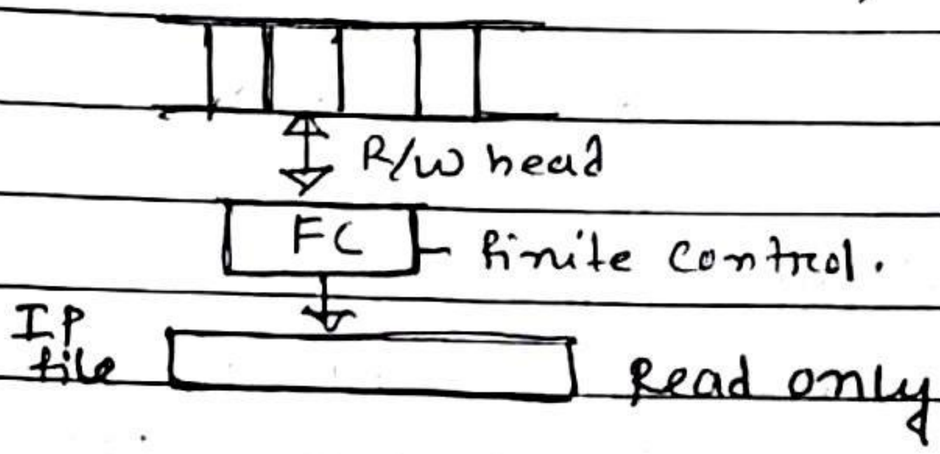
- Modification of standard Turing machine-

1) TM with stay option. $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times (L, R, S)$.

2) TM with semi infinite tape.

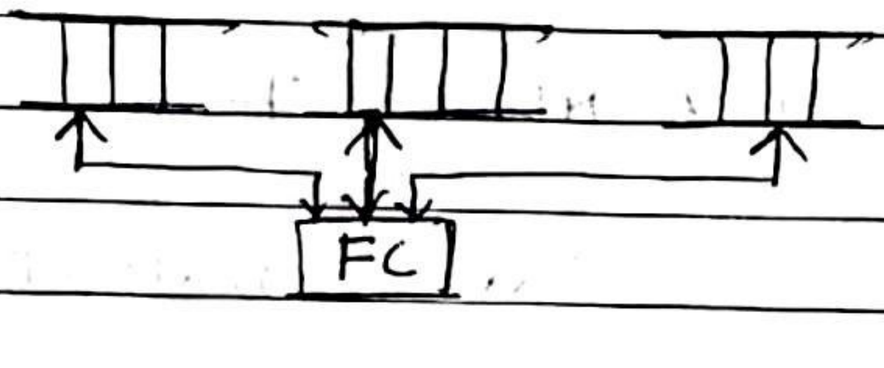


3) Offline TM.



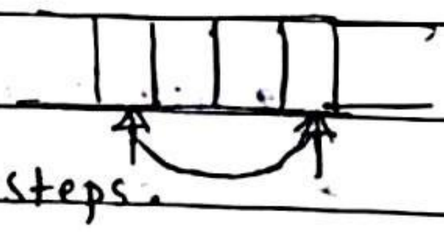
4) Multitape TM.

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



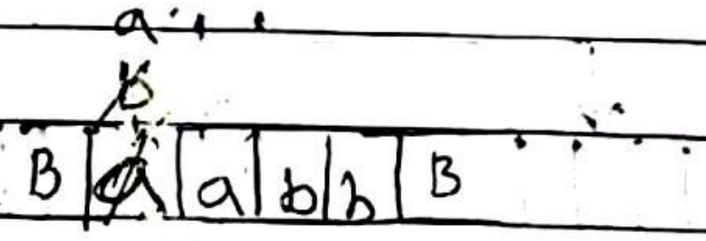
5) Jumping TM.

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



6) Non erasing TM.

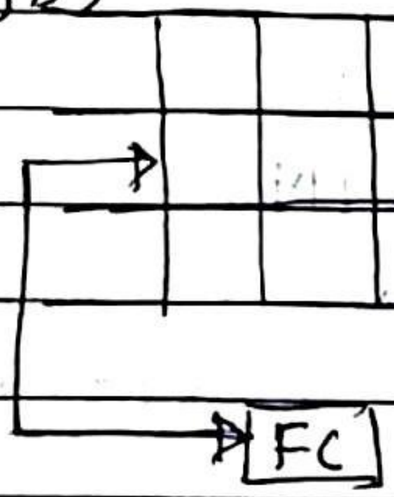
7) Always writing TM.



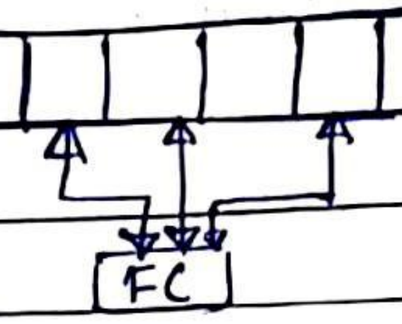
8) Multidimensional TM.

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

U → up.
D → down



9) Multihead TM.



10) Automata with a queue.

* 11) TM with only 3 states.

* 12) TMultitape TM with stay ~~operation~~ option and at most 2 states.

** (NTM ≅ DTM) 13) Non-deterministic TM

$\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$

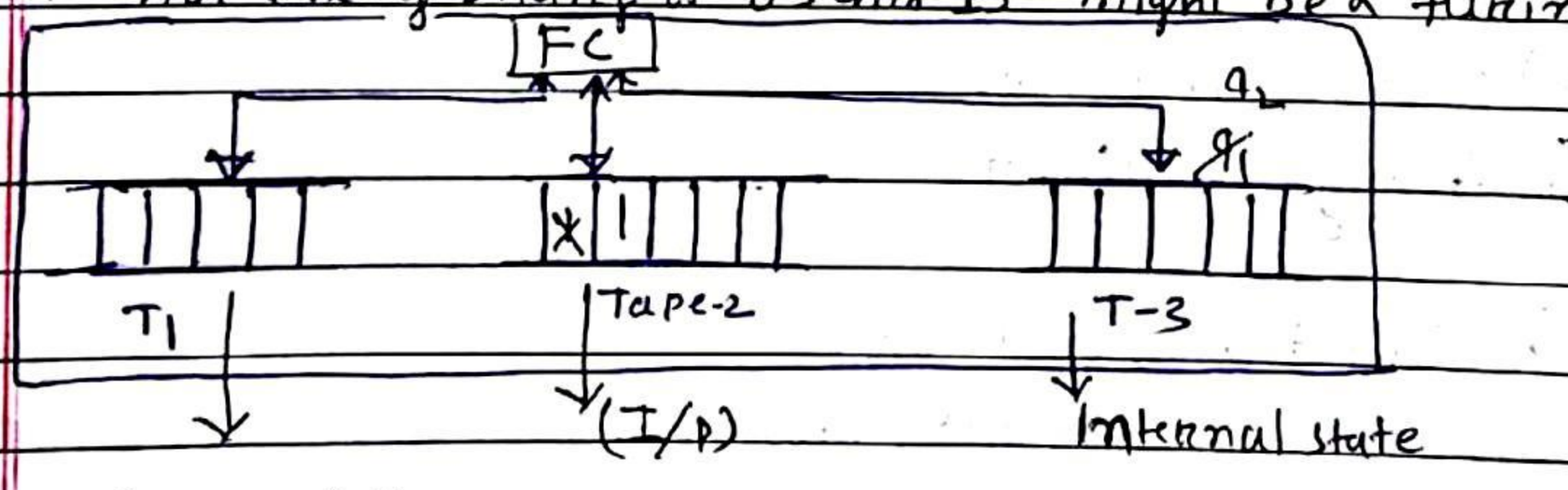
14) A NPDA with two independent stacks.

15) $\delta : Q \times (\Sigma \cup (\epsilon)) \times \Gamma \times \Gamma \rightarrow 2^{Q \times \Gamma^* \times \Gamma^*}$

• Universal TM and encoding of TM :-

* → Every TM can be represented as a string of 0's and 1's.

* → not every string of 0's and 1's might be a turing machine.



Representation of TM

$(q_1, 1) \rightarrow (q_2, X)$

$(q_2, 1) \rightarrow \ldots$

Tape-3 : To represent in which state we are in at present.

Tape-2 : To represent what ~~in~~ are the input looking at and what has been done to the input.

Tape-1 : To represent TM.

Original turing machine has,

$Q = (q_1, q_2, q_3, q_4 \ldots)$

$\Gamma = (a_1\ a_2\ a_3\ a_4 \ldots)$

encoding -

q_1 - 1 a_1 - 1

q_2 - 11 a_2 - 11

q_3 - 111 a_3 - 111

q_4 - 1111 a_4 - 1111

:

'0' - separator.

transation function -

$\delta(q_1, a_1) = (q_2, a_2, R)$

1 1 1

010 10 110 110 110

q_1 a_1 q_2 a_2 R

→ Turing machine can be used as a computer.

- TM + Tape (finite) = FA (Finite Automata)
- NTM + Tape(stack) = PDA (Pushdown Automata)
- TM + Tape (use part of i/p present) = LBA (Linear bounded Automata)

[a a bb]

**

FA < PDA < LBA < TM

D ≅ N D < N D ≅ N

- Some standard language that accepted by ~~PT~~ LBA :

D L = $\{a^n b^n c^n : n \geq 1\}$

2) L = $\{a^{n!} : n \geq 0\}$

3) L = $\{a^n : n = m^2\}, m \geq 1\}$

4) L = $\{a^n : n$ is a prime$\}$

5) L = $\{a^n : n$ is not a prime$\}$

6) L = $\{ww : w \in \{a,b\}^+\}$

7) L = $\{w^n : w \in \{a,b\}^+, n \geq 1\}$

8) L = $\{wwwR : w \in \{a,b\}^+\}$.

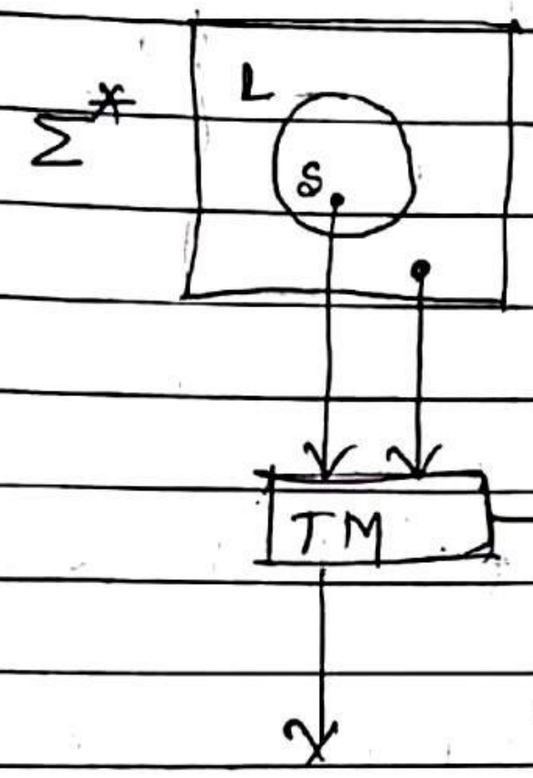
a a b a a b

→ A A B A A B ←

** Recursively enumerable language =

→ Any language accepted by TM is called Recursively enumerable language.

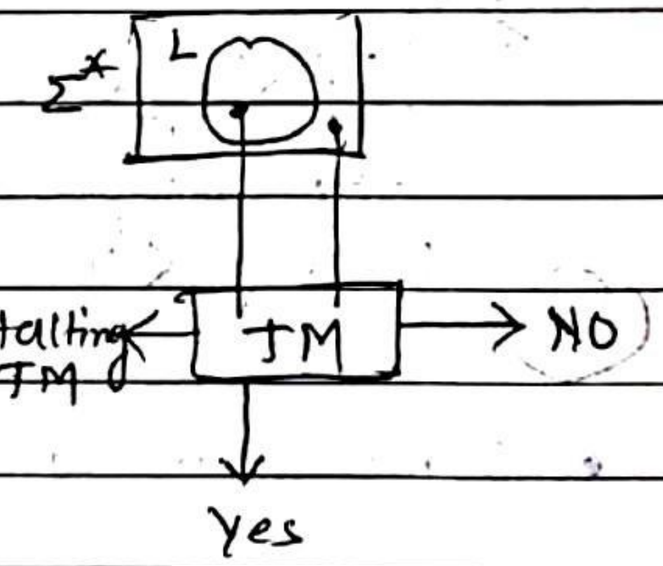
→

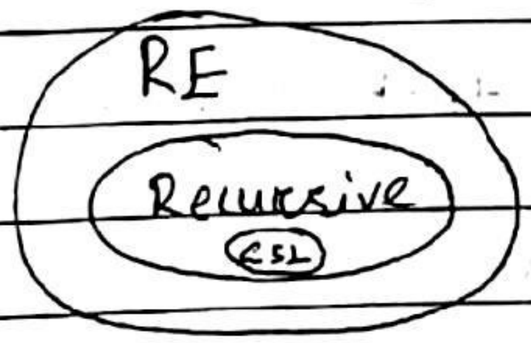


→ If string belongs to the language then it accepted by TM.

→ If string not belongs to the language then it halt or going to fall in infinite loop.

→ TM which always halt then that language is called Recursive language.





→ The Any language accepted by LBA (Linear bounded automata) is called CSL (Context sensitive language).

Unrestricted Grammar –

→ A grammar is called unrestricted if all the productions are of the form $u \rightarrow v$

Where, u is in $(V \cup T)^+$ $\quad (u \neq \epsilon)$

v is in $(V \cup T)^*$

What language does the following unrestricted grammar unrestricted grammar derive –

$S \rightarrow S_1 B$
$S_1 \rightarrow a S_1 b$
$bB \rightarrow bbbB$
$aS_1 b \rightarrow aa$
$B \rightarrow \lambda$

$L = \{a^{n+1} b^{n+K}, n \geq 1, K = -1, 1, 3, 5, \cdots\}$
$= \{aa, aabb, \cdots\}$

✓ $S \Rightarrow S_1 B$
$\Rightarrow a S_1 b B \quad [S_1 \Rightarrow a S_1 b]$
$\Rightarrow aaB \quad [aS_1 b \Rightarrow aa]$
$\Rightarrow aa \quad [B \Rightarrow \lambda]$

✓ $S \Rightarrow S_1 B$
$\Rightarrow a S_1 bB \quad [S_1 \rightarrow a S_1 b]$
$\Rightarrow a S_1 bbbB \quad [bB \rightarrow bbbB]$
$\Rightarrow aaabbB \quad [aS_1 b \Rightarrow aa]$
$\Rightarrow aaabb$

atlantis
Date
Page

• Context sensitive grammars =

⟹ A grammar is said to be context sensitive if all the productions are of form

$$\alpha \rightarrow \gamma$$

where, $\alpha, \gamma \in (V \cup T)^+$ and

$$|\alpha| \leq |\gamma|$$

• What is language generated by the following CSL-

$S \rightarrow abc / aAbc$	$S \Rightarrow aAbc$
$Ab \rightarrow bA$	$\Rightarrow abAC$
$AC \rightarrow Bbcc$	$\Rightarrow abBbcc$
$bB \rightarrow Bb$	$\Rightarrow aBbbcc$
$aB \rightarrow aa / aaA.$	$\Rightarrow aabbcc$
	$a^2 b^2 c^2$

$$L = \{a^n b^n c^n, n \geq 1\}.$$

not important for gate

• $a^n b^m c^n d^m$ $/ n, m \geq 1$, ~~of~~ for this language construct csg.

→ $S \rightarrow aACD / aBcD$

$A \rightarrow aAc / aBc$

$Bc \rightarrow cB$

$Bb \rightarrow bB$

$BD \rightarrow Fd$

$cE \rightarrow cE$

$bE \rightarrow Eb$

$aE \rightarrow ab$

$aaBbbccDdd$

$BD \rightarrow FDd$

$cF \rightarrow Fc$

$bF \rightarrow Fb$

$aF \rightarrow abB$

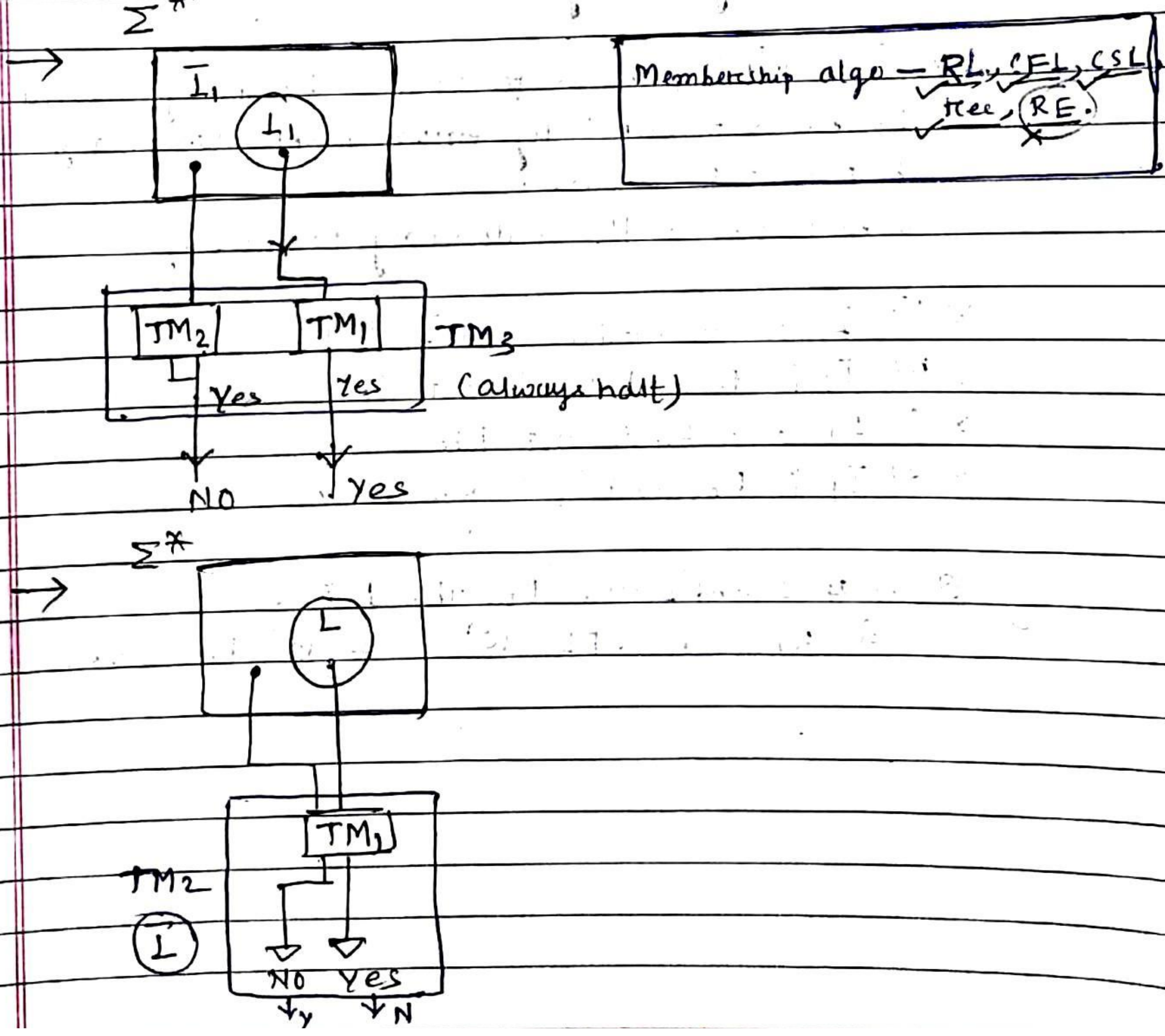
• Important theorem on recursive and RE languages

→ If a language L and its complement $\bar{L}$ are both recursively enumerable, then both languages are recursive. ✓

→ If L is recursive, then $\bar{L}$ is also recursive and consequently both are Recursively enumerable.

→ NO membership Algorithm to Recursively enumerable language.

→ membership algorithm exist to Recursive lang.



atlantis
Date
Page

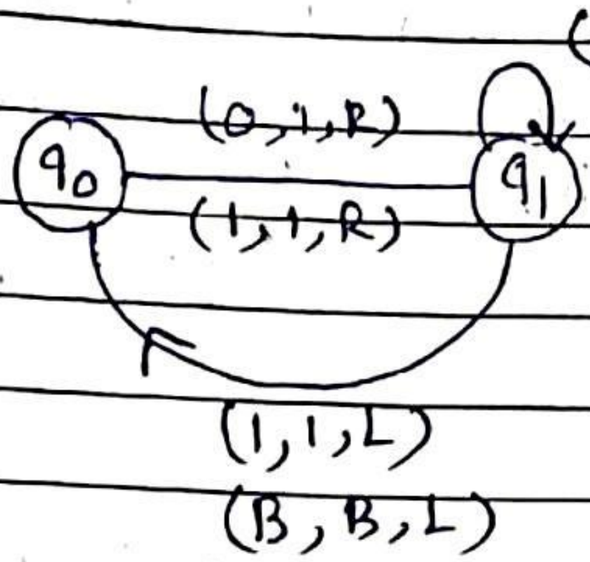
or-2003.

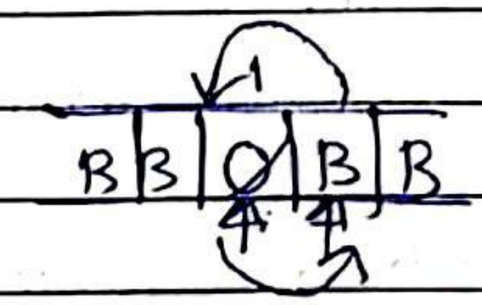
Q-1 TM Given -

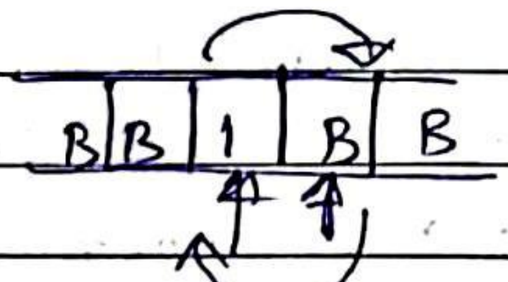
	0	1	B
q_0	$q_1, 1, R$	$q_1, 1, R$	Halt
q_1	$q_1, 1, R$	$q_0, 1, L$	q_0, B, L

whether strings halt or not.

→







for any string it is not going to halt.

Ans Q-2 Let L_1 be a recursive language, and L_2 be a recursively enumerable language but not Recursive.
Which one of the following is true?

✗ a) $\bar{L}_1$ is recursive and $\bar{L}_2$ is RE.
✓ b) L_1 is R and $\bar{L}_2$ is not RE.
✗ c) $\bar{L}_1$ and $\bar{L}_2$ is not RE
✗ d) $\bar{L}_1$ is RE and $\bar{L}_2$ is Rec.

Q-3 If L and $\bar{L}$ are 'RE' the L is,
(a) RL (b) CFL (c) CSL (d) Recursive.

• Diagram — (Relation between Machine, lang, Grammars).