Module-5

Part-B

1) Describe short notes on Context Sensitive language and linear bounded automata.

Sol: **Context Sensitive language:-**

A language L is considered Context-sensitive if it can be generated by a Context-Sensitive grammar. A Context-Sensitive grammar (CSG) is a formal grammar where the production rules are allowed to have Context-dependent conditions. Formally, a Context-Sensitive grammar is defined as $G = (N, \Sigma, P, S)$ where:

N : Set of non-terminal Symbols.

Σ : Set of terminal Symbols.

P : Set of production rules in the form $\alpha \rightarrow \beta$

S : Starting Symbol.

Linear Bounded Automata:-

The LBA operates on an infinite tape, similar to a Turing machine, but the tape is bounded by the input size. Formally, an LBA is defined as a tuple

$M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

The Transition function $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L \times R\}$

where L represents a move to the left

R represents a move to the right.

2) Classify briefly about Chomsky hirach of languages

Sol: Type 0 Grammar:-

- They generate recursively enumerable languages
- They have no restrictions
- They are recognised by Turing machines
- The productions can be in the form of $\alpha \to \beta$, where α is a string of terminal and non-terminal with atleast one non-terminal and α cannot be null
- β is a string of terminals and non-terminals

Ex: $S \to ACaB$
$Bc \to acB$
$CB \to DB$
$aD \to Db$

Type 1 Grammar:-

They generate Context Sensitive languages

The productions must be in the form,

$\alpha A \beta \to \alpha \gamma \beta$, where $A \in N$ (non terminal)
$\alpha, \beta, \gamma \in (T \cup N)^*$ (String of Terminals and non-terminals)

The strings α, β may be empty, but γ cannot be empty.

These languages are recognized by the linear bounded automata

Ex: $AB \to AbBc$
$A \to bcA$
$B \to b$

Type 2 Grammar:-

They generate Context free languages

The production must be in the form: $A \to \gamma$ where $A \in N$ (Non Terminal) and $\gamma \in (T \cup N)^*$ (String of Terminals and non-terminals)

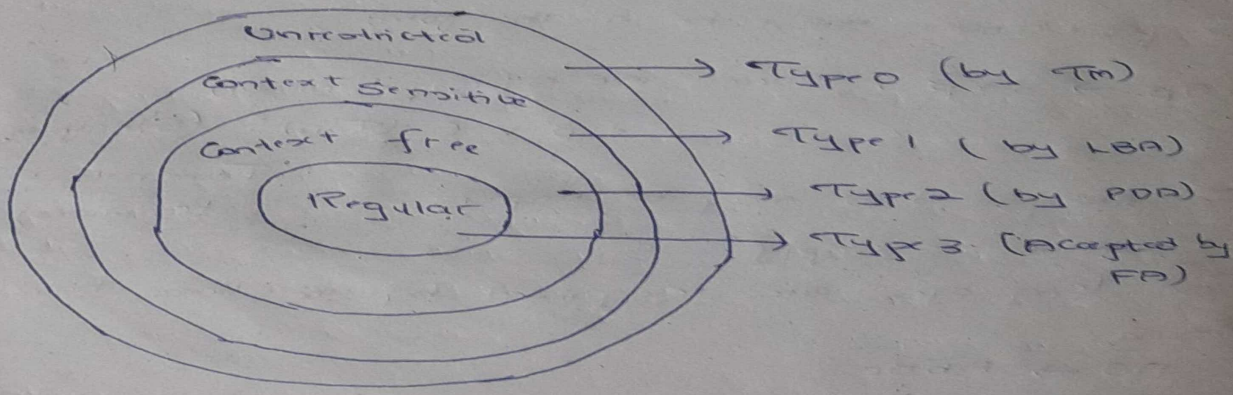
- These languages are recognized by a non-deterministic pushdown Automata.

Ex. $S \to Xa$ $\quad X \to abc$
$X \to a$ $\quad X \to \varepsilon$
$X \to aX$

Type 3 Grammar :-

- They generate regular languages.
- They must have a single non-terminal on left hand side and right hand side consisting of a single terminal or a single terminal followed by a non-terminal.
- The productions must be in the form $X \to a$ (or) $X \to aY$ where, $X, Y \in N$ (non-terminal) and $a \in T$ (terminal)

Ex $X \to a \mid aY$; $Y \to b$.



3) Describe a Turing machine with a neat diagram, explain the working of a Turing machine.

Sol. • A Turing machine is an accepting device which accepts the languages (recursively enumerable set) generated by type 0 Grammars.

It was invented by Alan Turing in 1936.

A Turing machine is a mathematical model which consists of an infinite length tape divided into cells on which input is given.

It consists of a head which reads the input tape.

State register stores the state of the Turing machine.

It is represented by 7-tuple notation,

$$(Q, X, \Sigma, \delta, q_0, B, f)$$

Q: Finite Set of States

X: Tape Alphabet

Σ: input alphabet

δ: Transition function

q_0: initial State

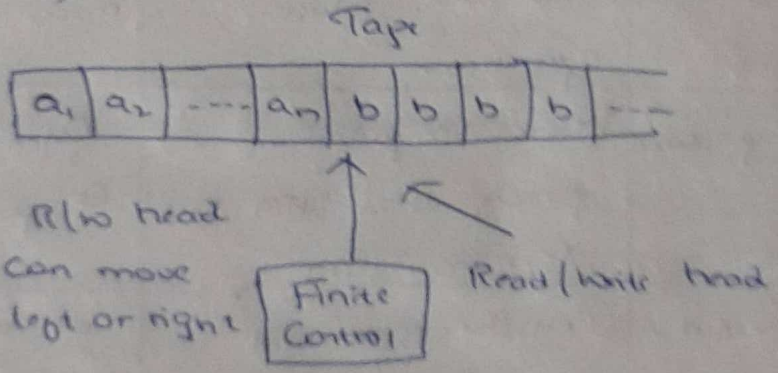
B: blank Symbol

f: Set of final states

Time and Space Complexity of a TM:-

$T(n) = O(n \log n)$

$S(n) = O(n)$



Working:-

Initially, the TM is in a Specific start state, the input string is placed on the tape and the tape head is positioned at the leftmost cell of the input.

The TM reads the Symbol under the tape head and looks up the current state.

Based on the current state, and the Symbol read, the finite Control unit provides the next state, the Symbol to

writing on the tape and the direction in which tape head should move.

4) The process continues until the TM enters a halting state. If the machine reaches a halting state, it stops and computation is complete.

4) Compare Turing machine with other automata.

A)

Finite Automata	(PDA)	Turning machine (TM)
Recognizes Regular languages	Recognizes Context-free languages	Recognizes Recursively Enumerable languages
No memory (limited to current state)	Uses a stack for memory	Infinite tape as memory.
Regular languages	Context-free languages	Recursively Enumerable languages.
Reads input symbols and transitions between states	Reads input symbols and use stack operations.	Reads, writes and moves on the tape.
Limited computational power	More computational power than FA but less than TM.	General purpose computing device.
Can be represented using state diagrams	Can be represented using state diagrams with additional stack information.	Can be represented using state & transition tables with tape information.

5) Construct a Transition diagram for TM to accept the language $L = \{w + w^R \mid w \in (a+b)^*\}$

Sol:

6) Express short notes on Recursive and Recursively Enumerable languages.

Recursive languages and Recursively Enumerable languages are two classes of formal languages in the ToC. These classes are defined based on the computational properties of TM.

1) Recursive languages:-

A language L is said to be Recursive if there exists a TM that can decide whether a given input string belongs to L or not.

In other words, a TM can halt and provide a definite answer of "yes" or "no" for any input string in

properties :-

- Decidability :- Recursive languages are decidable, that is there is an algorithmic procedure to determine membership in the language.
- Halting :- TM always halts on every input, providing a definite answer.
- Acceptance :- All strings in the language are accepted and all strings not in the language are rejected.

2) Recursively Enumerable Languages :-

- A language L is called Recursively enumerable if there exists a TM that can accept all strings in L, but it may either reject or loop indefinitely on strings not in L.
- In other words, a TM can accept strings in the language, but there is no guarantee it will halt and rejects strings outside the language.

properties :-

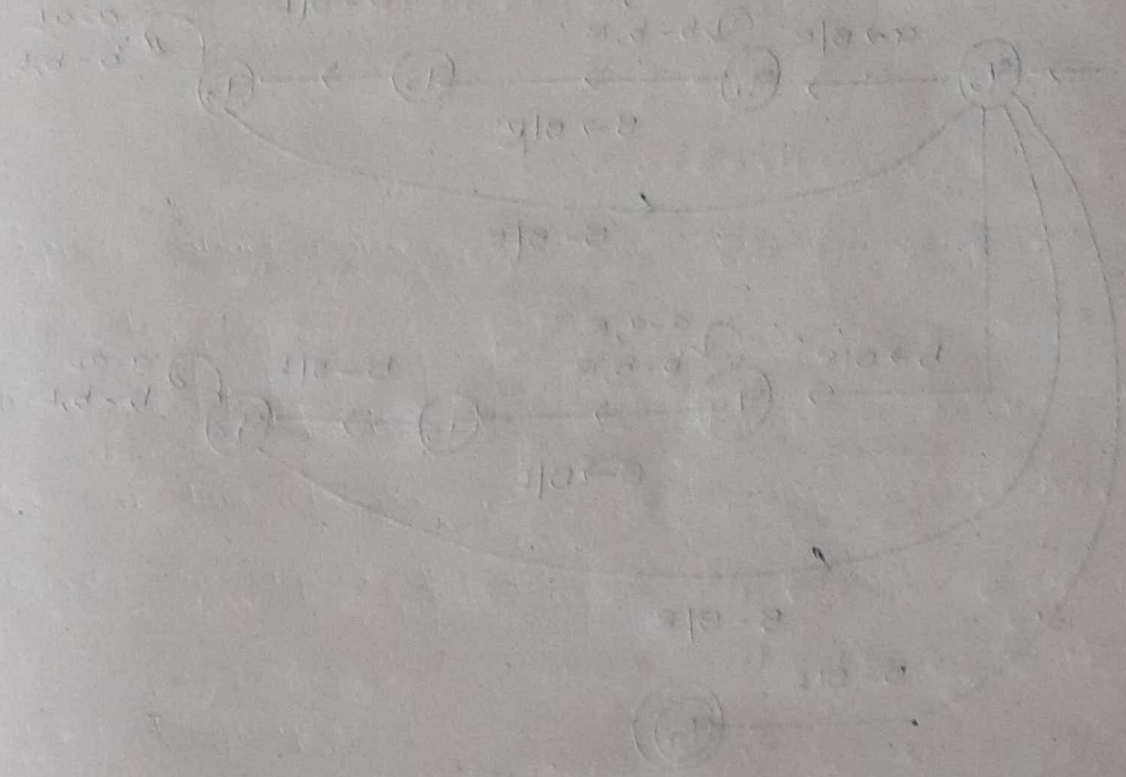
- Semi-decidability :- These are semi-decidable, meaning there is a TM that can accept strings in the language, but there may not be an algorithmic procedure to reject strings not in the language.
- Halting :- The TM may either halt and accept a string in the language or loop or reject for strings not in language.
- Acceptance : (Same above)

7) Describe the properties of recursive and recursively enumerable languages.

Sol: Refer (part-B 6th Question)

8) Design a Turing machine to accept Strings formed with 0 and 1 having sub string 000.

Sol:



) Construct a Transition diagram for TM to accept e language $L = \{ww^R \mid w \in (a+b)^*\}$

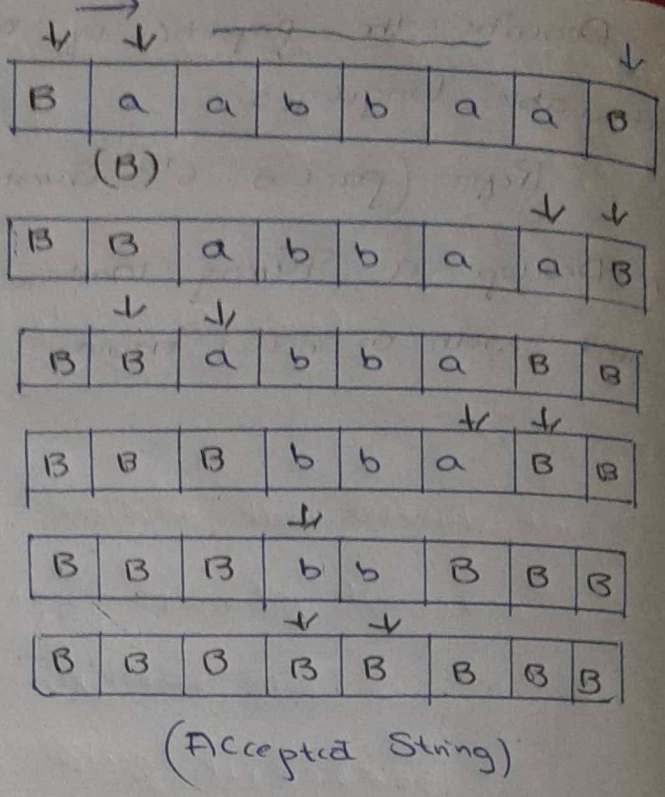
: Consider- w = aab L = {abba, aabbaa, ...}

The String will be aabbaa

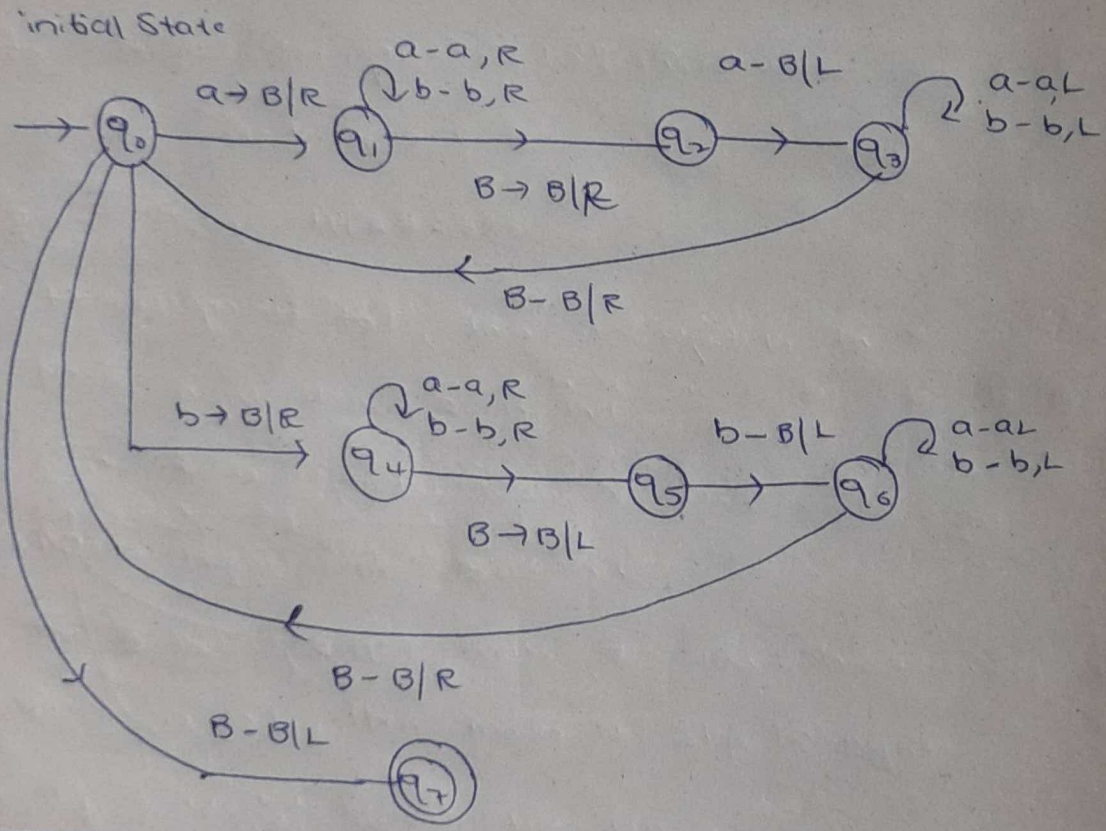
B - Blank

R - Right

L - Left



(Accepted String)

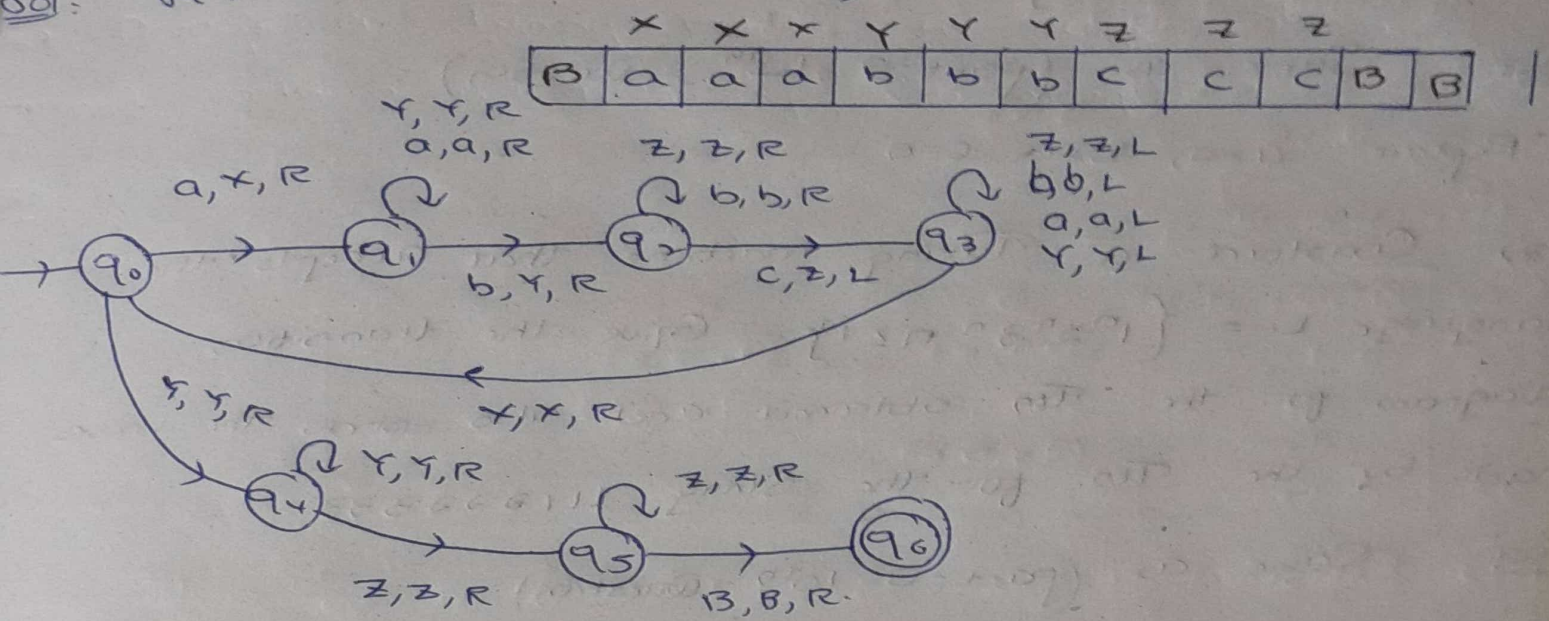


	b	a	B	R	
q_0	q_4, B, R	q_1, B, R	q_7, B, L		
q_1	q_1, bR	q_1, a, R	q_2, B, R		
q_2	q_3, B, L				
q_3	q_3, b, L	q_3, a, L	q_0, B, R		
q_4	q_4, b, R	q_4, a, R	q_5, B, L		
q_5	q_6, B, L				
q_6	q_6, b, L	q_6, a, L	q_0, B, R		
q_7					

10) Design a Transition table for TM L =

$\{a^n b^n c^n \mid n \geq 1\}$

Sol: Let us take a String, aaabbbccc



Transition Table :-

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

$\delta(q_0, Y) \to (q_4, Y, R)$

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

$\delta(q_1, b) \to (q_2, b, R)$

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

$\delta(q_2, b) \to (q_2, b, R)$

$\delta(q_2, c) \to (q_3, Z, L)$

$\delta(q_2, Z) \to (q_2, Z, R)$

$\delta(q_3, a) \to (q_3, a, L)$

$\delta(q_3, b) \to (q_3, b, L)$

$\delta(q_3, Z) \to (q_3, Z, L)$

$\delta(q_3, Y) \to (q_3, Y, L)$

$\delta(q_3, X) \to (q_0, X, R)$

$\delta(q_4, Y) \to (q_4, Y, R)$

$\delta(q_4, Z) \to (q_5, Z, R)$

$\delta(q_5, Z) \to (q_5, Z, R)$

$\delta(q_5, B) \to (q_6, B, R)$

Present State	a	b	X	Y	☐ B
→ q_0	(q_1, X, R)			(q_4, Y, R)	
q_1	(q_1, a, R)	(q_2, b, R)		(q_1, Y, R)	
q_2		(q_2, b, R)			
q_3					
q_4					(q_6, B, R)
q_5					
q_6					

11) Construct a Transition table for Turing machine to accept the following language.

$L = \{0^n 1^n 0^n \mid n \geq 1\}$

Sol: Same as (part-B 10th Question)

Replace a=0, b=1, c=0

12) Construct a Turing machine that accepts the language $L = \{1^n 2^n 3^n \mid n \geq 1\}$. Give the transition diagram for the TM obtained and also show the moves made by the TM for the string 111222333.

Sol: Same as (part-B 10th Question)

Replace a=1, b=2, c=3

Given String 111222333

- Head point to 1, then it update to x
- Move right until 2 is hit
- The 2 is updated with Y, and move right
- Move right until 3 is hit
- The 3 is updated with Z, and head move left until x is hit, then moves a step to right
- If it is 1, repeat above process
- If we hit Y, head move to right until 'b' is hit
- Halt.

13) Enumerate Linear Bounded Automata and explain its model?

Sol: Refer (part-B 1st Question)

4) Demonstrate the power and limitations of Turning machine.

Sol: Power of TM :-

. The TM has great Computational Capabilities, so it can be used as a general mathematical model for modern computers also.

. Turing machine can model even recursively enumerable languages.

Thus, the advantage of TM is that it can model all the Computable functions as well as the languages for which the algorithm is possible.

Limitations of TM :-

They donot model the Strengths of a particular Arrangement well

When TM are used as basis of bounding running time a else lower-bound may be proven, because of no memory indexing.

The Solutions are may not optimal because of not implementing an indexed memory.

Another limitation of TM is that they do not model Concurrency well.

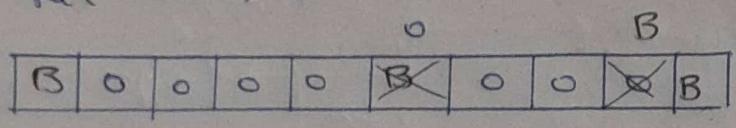
5) Construct Transition diagram for TM

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

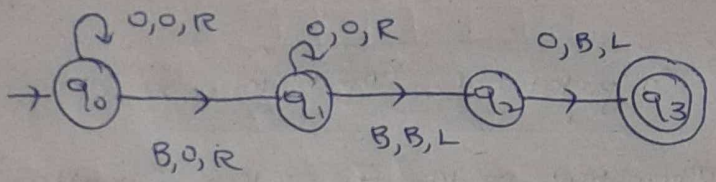
Sol: Refer (part-B 10th Question)

6) Construct a Transition diagram for TM to implement addition of two unary numbers $(x + y)$.

Sol: Let a=4, b=3



Steps :-



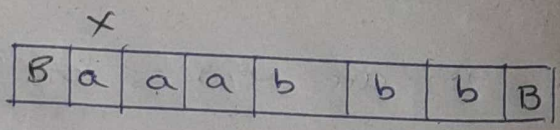
Transition diagram :-

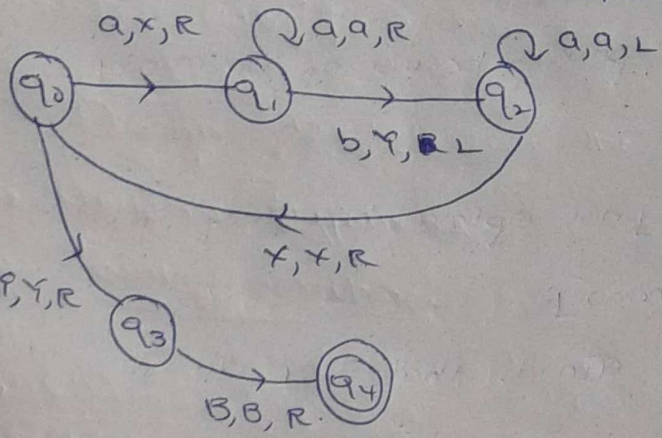
Present State	0	B
→ q0	(q0,0,R)	(q1,0,R)
q1	(q1,0,R)	(q2, B,L)
q2	(q3,B,L)	
q3	—	—

17) Construct a linear Bounded automata for a language where L= {$a^n b^n$ | n≥1}

Sol: Consider, L= { ab, aabb, aaabbb --- }

String = aaabbb

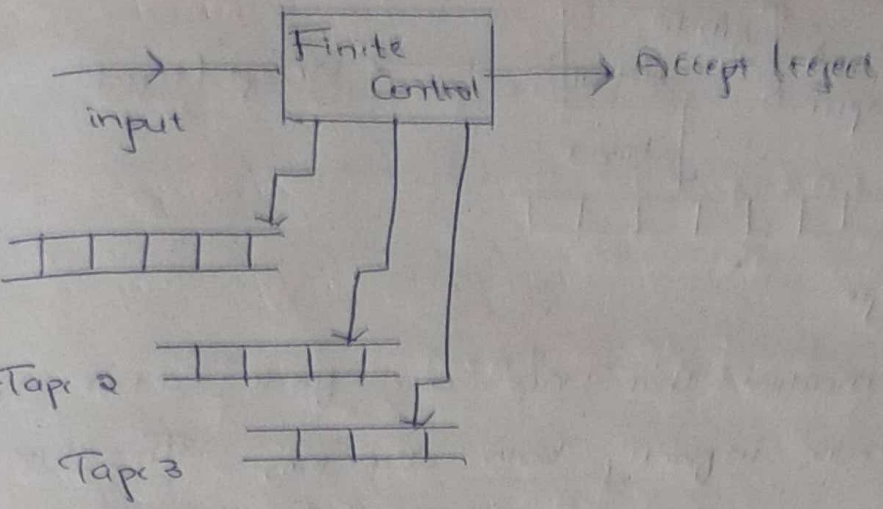




18) Classify the types of Turing machines.

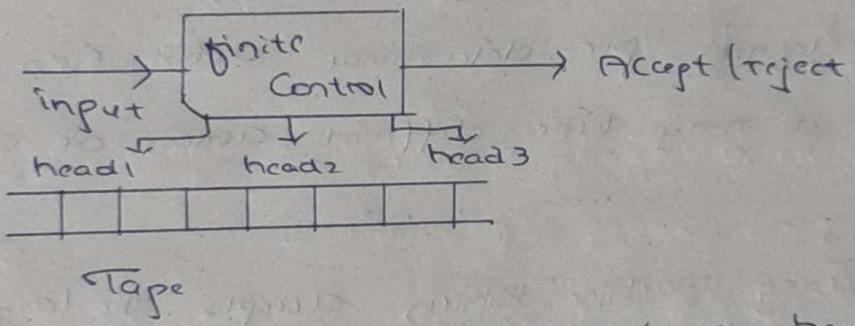
Sol: Types of Turing machines :-

1) Turing machine with multiple tapes
2) Turing machine with one tape multiple heads
3) Turing machine with two dimensional tape
4) Turing machine with infinite tape
5) Non-deterministic turing machine.

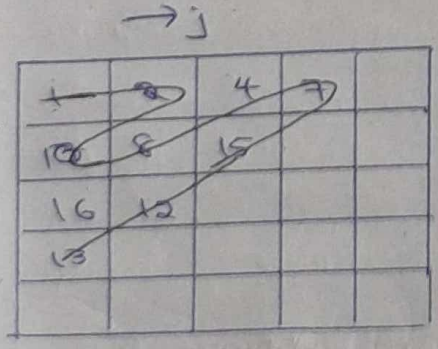


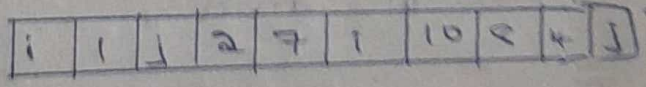
- It has a finite control, more than one tapes having its on read and write head.

The language accepted by the n-tape turing machine n be accepted by 1-tape turing machine.



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





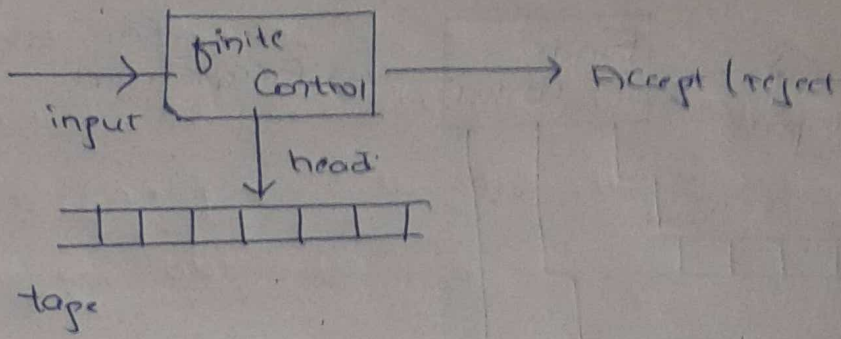
1D tape

It is divided into small squares formed due to corresponding rows and columns.

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

The head of 2-D tape moves one square upward 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.

5) Similar to NFA, for any state & any input symbol. It can take any action from a set rather than a definite pre-determinal action even in some situations it may take different actions at different times.

Eg: Turing machine which accepts the language L = {ww / w∈ (a+b)*} is non-deterministic TM.

19) Describe briefly about the following.

a) Church's Hypothesis

b) Counter machine.

Sol.

a) Church's Hypothesis :-

- Church's Hypothesis, also known as Church's Thesis. It was formulated by the mathematician and logician Alonzo Church in 1930s. The hypothesis stats that any function that is computable by an effective algorithm can be expressed or represented using the lambda calculus formalism.

Lambda Calculus is a formal system developed by Church to study the Concept of Computability and functions. It involves variables, abstraction and applications of functions.

This hypothesis has played a Crucial role in the development of theoretical Computer Science and the understanding of Computability.

b) Counter Machine :-

A Counter machine is a Computational model that operates on an infinite Sequence of natural numbers stored in Counters. The machine consists of a finite Control unit, which has a set of states and an infinite number of Counters.

At a given time, the machine is in one of its states, and can perform actions based on the Current state and the value in Counters. These actions can include incrementing or decrementing the value in a Counter, changing the state or halting the Computation.

Counter machines can be Classified based on their capabilities, such as number of Counters they have or whether they allow Conditional branching.

c) Construct Transition diagram for TM that accepts the language $L = \{0^n 1^n \mid n \geq 1\}$. Give the transition diagram for the TM obtained and also show the moves made by the TM for the String 000111

Ans: Refer (Part-B 17th Question) + (Transition table)