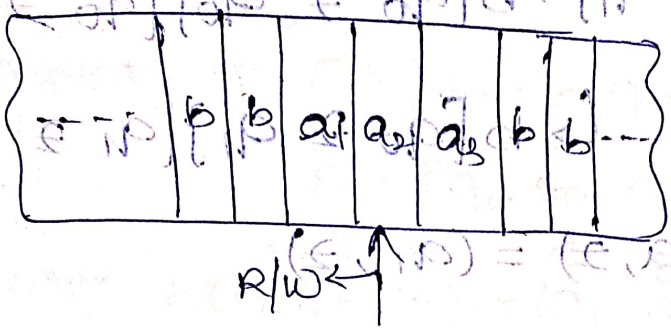
Turing Machines.

CSG - Context sensitive Grammar

↳ LBA - linear bounded Automata

Turing Machine - R.E or unrestricted Grammar.

Recursively Enumerable language



The turing machine consists of a finite state control connected to a read/write head which is pointed to any one of the tape symbol at a time.

→ It has one tape which is divided into infinite no. of states.

→ Each cell can store only one symbol.

→ In one move or transition, turing machine examines the present symbol under read write head on the tape.

and the present state of the turing machine to determine a) new symbol to be written on the tape in the set.

b) Movement of the read write head. (Read or write).

c) The next state of the turing machine.

→ A turing machiene M is a 7 tupples

$(Q, \Sigma, \Gamma, \delta, q_0, b, F)$

where Q = finite non empty set of states

Σ = Finite non empty set of input symbols.

Γ = finite non empty set of tape symbols.

q_0 - Initial state

F = subset of Q, $F \subseteq Q$.

B| b = A special symbol call blank symbol. $b \in \Gamma$

δ = Transition function which maps

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

→ δ may not be defined for some elements of $Q \times \Gamma$.

→ The acceptability of a string is decided by the reachability from the initial state to some final state when turing machiene is halted.

→ Representation of a Turing Machiene

ID- Instantaneous description

Transition table

Transition diagram.

Instantaneous Description:-

If the input string to be processed is $x_1, x_2, \ldots x_n$ and present symbol Under read write head is x_i so the ID before processing x_i for the move $\delta(q, x_i) = (P, y, L)$ is

$x_1 x_2 \ldots q_0 x_i \ldots x_n$

after processing the move

$x_1 x_2 \ldots P x_{i-1} y x_{i+1} \ldots x_n$

→ If the move is

$\delta(q, x_i) = (P, y, R)$ then the resultant ID is

$x_1 x_2 \ldots x_i y P x_{i+1} \ldots x_n$

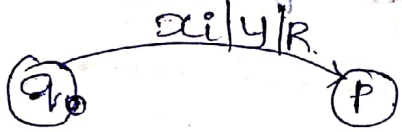
Transition table:-

If we give the definition of δ in the form of a table then that table is called as transition table.

Transition Diagram:-

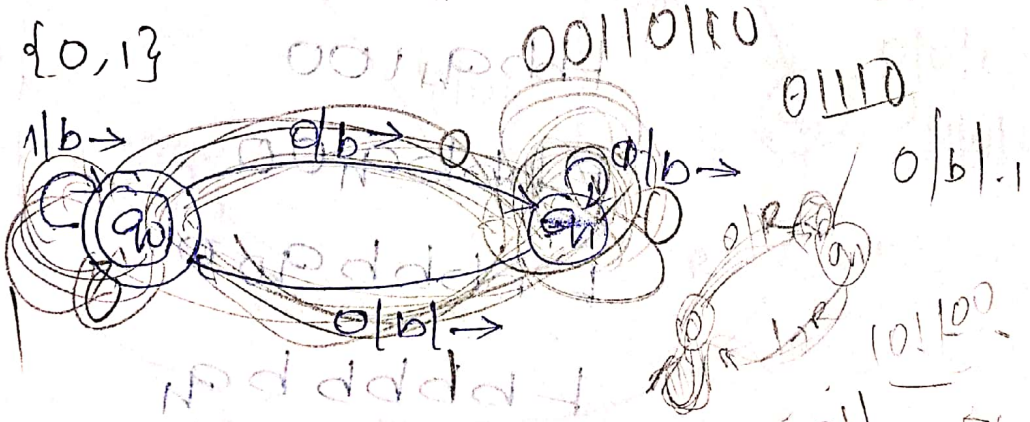
We can represent transitions in the form of a diagram then it is called transition diagram.

→ If the move is $\delta(q_1, x_i) = (P, y, R)$ then



Design a turing machiene to recognise all strings of 0's & 1's such that strings should consists of even no of 0's and 1's over the alphabet of

$\Sigma = \{0, 1\}$



δ	0	1
→ q_0	(q_1, b, R)	(q_0, b, R)
	(q_0, b, R)	(q_1, b, R)

$M = (\{q_0, q_1\}, \{0,1\}, \{0,1,b\}, \delta, q_0, b, \{q_0\})$

$q_0\ 00100 \vdash b q_1 0100$

$\vdash bb q_0 100$

$\vdash bbb q_0 00$

$\vdash bbbb q_1 0$

$\vdash bbbb b q_0$

When the turing is machiene is halted the state of the turing machiene is q_0

→ As q_0 is final state so the given string is accepted.

Consider 01100.

$q_0\ 01100 \vdash b q_1 1100$

$\vdash bb q_1 100$

$\vdash bbb q_1 00$

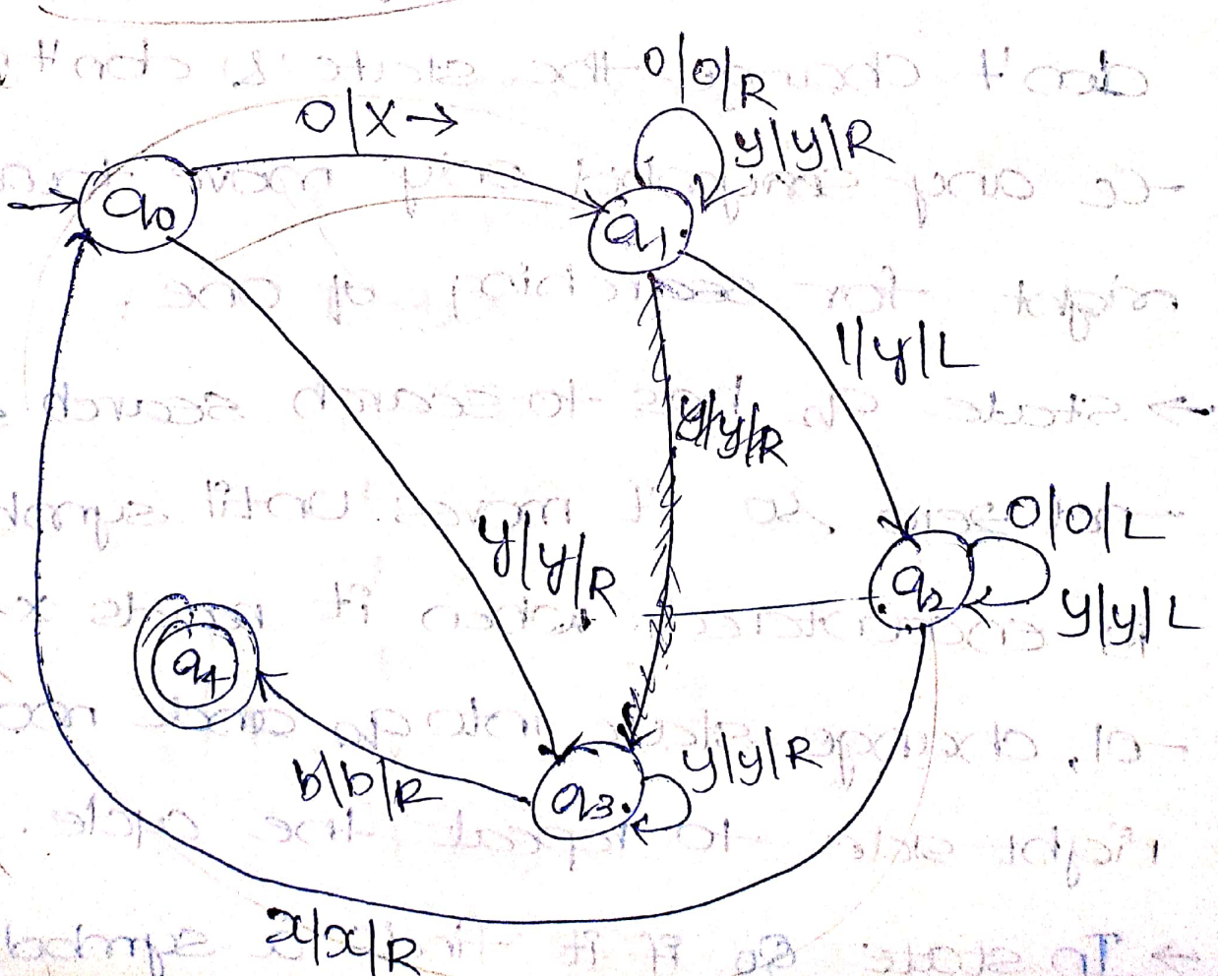
$\vdash bbbb q_0 0$

$\vdash bbbbb q_1$

When the turing machine is halted the state of turing machine is q_1 so, the string is rejected.

→ Design a turing machiene for the language $L = \{0^n 1^n \mid n \geq 1\}$

δ	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)	(q_4, b, R)
(q_4)	—	—	—	—	—



→ When the turing machine is in initial state i.e in q_0 when it reads symbol 0 it replaces with x and change state into q_1, moves right side for searching of first "1".

→ In q_1 state on reading symbol '1' replace "1" with 'y' and change state into q_2 and move left to search for second zero.

→ If the symbols 0, y are encountered don't change the state & don't replace any symbol only move towards right for searching of one.

→ State q_2 has to search search second zero so it moves until symbol x is encountered. when it reads x symbol, change state into q_0 and move right side to repeat the cycle.

→ In state q_0 if it finds a symbol 'y' i.e all zeros are already replaced by 'x' then 'n' enters into a state q_3.

→ In state q_3 machine checks is there

are any 1's or not. If any 1 is remaining string has to be rejected otherwise string has to be accepted. For implementing this q_3 moves right side until it encounters one or blank.

→ q_3 doesn't do anything if it encounters a symbol 'y'. If blank is encountered it enters into a final state q_4.

→ Consider an example 0011, it will be processed by the turing machine as follows.

$q_0 0011 \vdash X q_1 011$

$\vdash q_2 X 0 Y 1 \; X 0 q_1 11$

$\vdash X q_2 0 Y 1$

$\vdash q_2 X 0 Y 1$

$\vdash X q_0 X Y 1$

$\vdash X X q_1 Y 1$

$\vdash X X Y q_1 1$

$\vdash X X q_2 Y Y$

$\vdash X X q_2 Y Y$

$\vdash X q_2 X Y Y$

$\vdash X X q_0 Y Y$

$\vdash X X Y q_3 Y$

$\vdash X X Y Y q_3 b$

$\vdash X X Y Y b q_4 b$

Since we did'nt define any state for q_4, the machine halts, when halted the machine is in q_4 state. since q_4 is the final state. so the string is accepted.

→ Consider an example of string 00111 it will be processed by the turing machine as follows.

$q_0 00111 \vdash X q_1 0111$

$\vdash X 0 q_1 111$

$\vdash X q_2 0 Y 11$

$\vdash q_2 X 0 Y 11$

$\vdash X q_0 0 Y 11$

$\vdash X X q_1 Y 11$

$\vdash X X Y q_1 11$

$\vdash X X q_2 Y Y 1$

$\vdash X q_2 X Y Y 1$

$\vdash X X q_0 Y Y 1$

$\vdash X X Y q_3 Y 1$

$\vdash X X Y Y q_3 1$

Since we didiot define any state for q_3 on 1 the machine is halted. in q_3

→ Since q_3 is not the final state the machine rejects the string.

→ $L = \{1^n 2^n 3^n \mid n \geq 1\}$

$q_0 11 2233 \vdash X q_1 1 2233$

$\vdash X 1 q_1 2233$

$\vdash X 1 Y q_2 233$

$q_0, 1 \quad \vdash X 1 Y 2 q_2 33$

$q_0 ,$ $\vdash X 1 Y 2 2 q_3 3$

$\vdash X 1 q_3 Y 2 2 3$

$\vdash X q_3 1 Y 2 2 3$

$\vdash X q_3 X 1 Y 2 2 3$

$\vdash X q_0 1 Y 2 2 3$

$\vdash X X q_1 Y 2 2 3$

$\vdash X X Y q_1 2 2 3$

$\vdash X X Y Y q_2 2 3$

$\vdash X X Y Y 2 q_2 3$

$\vdash X X Y Y q_2 2 q_3$

$\vdash X X Y Y q_3 2 2$

$\vdash X X Y q_3 Y 2 2$

$\vdash X X q_3 Y Y 2 2$

$\vdash X q_3 X Y Y 2 2$

$\vdash q_3 X X Y Y 2 2$

$\vdash xxq_0yyzz$

$\vdash xxyq_4yzz$.

$\vdash xxyyq_4zz$.

$\vdash xxyyq_5z$.

$\vdash xxyyzq_5b$

$\vdash xxyyzzq_6$.

q_6 is the final state.

2. Design a turing machine for the set of all pallindromes over $\Sigma = \{0,1\}$

→ If M is in state q_0 and it scans '0' then it is replaced by "x" and enters into state 'q_1' and move right all zeros and one's that it finds on the tape, remaining in the same state until it finds a symbol x, y or blank (b). Now "m" moves one step to the left and changes to state q_3.

→ It verifies that the symbol read is zero and changes 0 to "x" and go es to state q_5.

→ If M is in same q_0 and it stands a symbol 1 and it goes to state q_2 changes 1 to 'y' and moves right over all zeros & 1's; that it finds on the tape, remaining in the same state.

→ Now "m" moves one step to the left & changes to state q_4 it then verifies that symbol read is 1 &

changing it to "y" & goes to state q_5

→ In state q_5 "M" (machine) moves left over all zeros & 1's until it finds x or y. Now it changes state to q_6 moves right.

→ Once again there are 2 cases

→ If M finds 0's or 1's repeat the matching cycle we have just described.

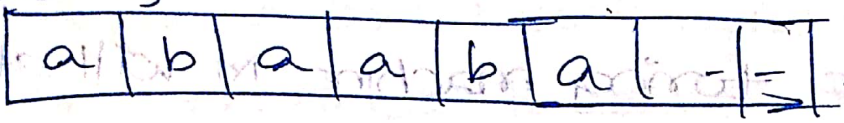
→ If M finds x or y then it has changed all 0's to x's and 1's to y's and hence the input was of an a palindrome of even length so entered into q_6 and halts.

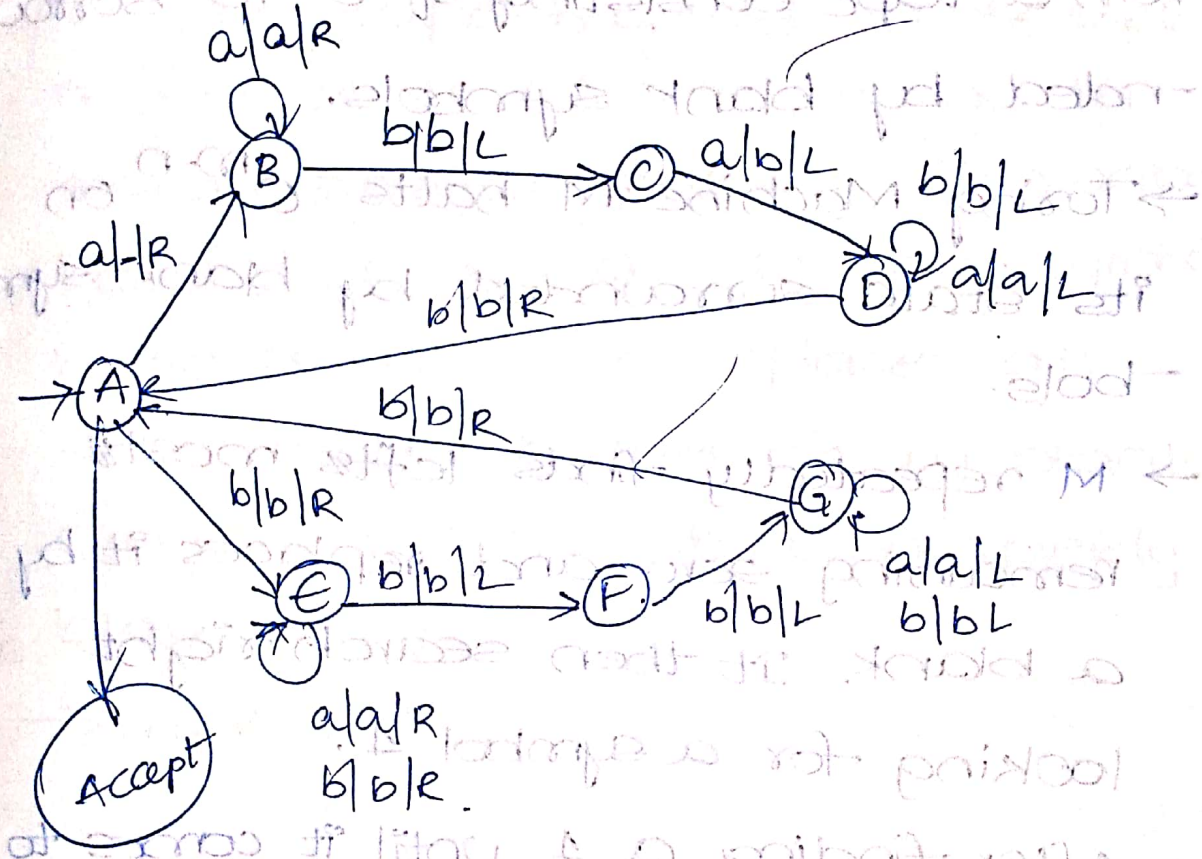
→ In case M is in state q_3 (either in q_4) and reads x or y instead of 0 (1).

→ It concludes that the i/p was a palindrome of odd length. In this case also it changes its state to q_6 and halts. On the other hand if M encounters

a symbol 1 in state q_3 or a symbol 0 in state q_4 then the i/p is not a palindrome and so M dies without accepting.

considering a b a a b a





Types of Turing Machine.

Non-Deterministic turing Machine:
(Linear bounded automata)

It is a turing machine at current state & for the tape symbol it is reading, There may be diff possible actions to be performed.

Turing Machine with 2D tapes:

Turing Machine with a 2dimensional tape is a kind of turing machine that has one finite control, one read write head & one 2dim. tape.

→ The cell in the tape is 2D (rows & colum-ns). The read write head can move left direction, right, upward or downward direction.

Turing Machine with multiple tapes:-

It is a kind of turing machine that has one finite control & more than 1 tape. Each with its own read, write head.

Turing Machine with multiple heads:

It is a kind of turing machine that has one finite control & one tape but more

than one read write head. In each state only one of the heads is allowed to read & write.

→ Turing Machine with semi infinite tape:

It is a kind of turning machine that has a tape which is bounded by one side & infinite at the other side. This machine also can perform all the functionalities which are carried out by basic turing machines.

→ Linear Bounded Automata.

A non-deterministic turing machine is also called as linear bounded automata (LBA). In this automata a tape is bounded i.e the length of the tape is determined by using a linear fn.

→ It contains all the tupples (or) symbols as that of turing machine and also contains 2 end markers ¢ (left end marker) & $ (right end marker)

→ It has no moves beyond these end markers.

⇒ It never changes these end marke
-rs symbols.

→ The language accepted by LBA is called context sensitive language

→ The Grammar describing this lang
-uage is called context sensitive Grammar.

Context Sensitive Grammar:

A Grammar is said to be CSG if all productions is of the form $\alpha \to \beta$ where $\alpha, \beta \in (V \cup T)^*$. but the length of β is atleast as much as the len
-gth of the α.

Consider the Grammar $S \to abc | aAbc$

$Ab \to bA$

$Ac \to Bbcc$

$bB \to Bb$

$aB \to aa | aaA$.

Generate the string aabbcc, aaabbbccc. for the above Grammar.