* Turing Machine :

=> A Turing Machine can be defined as a set of 7 Tuples

$T = \{Q, \Sigma, \Gamma, \delta, q_0, b, F\}$

Q -> Set of states
Σ -> Input Symboles
Γ -> Tape Symboles
δ -> Transition Function
$Q \times \Sigma \rightarrow \Gamma$
q_0 -> Initial state
b -> Black Symboles
F -> Set of Final states

Three Operations :

(i) Reading a Symbol being scanned
(ii) Modifying a Symbol being scanned
(iii) Shifting Tap Head either Right or Left.

* Universal Turing Machine :

The Language

$A_{TM} = \{\langle M, w \rangle \mid M$ is a Turing Machine and M accepts $w\}$

is Turing Recognizable.

Input :
M = the description of some TM
w = an input string for M

Action :
- Simulate M
- Behave Just like M would

Here, M accepts w - Halt & Accept
M Rejects w - Halt & Reject
M Loops on w - Not Halt

In Universal Turing Machine Input are passed and according to input it will behave.

Ex. General Purpose Computer

If we want to Run any Program in the computer then we will provide some program in computer.

This Program is become Input For the Universal Turing Machine

And Computer will be simpulate or Run the program. So, this is become Action for the Universal Turing Machine.

* Recursive Enumerable Languages:

=> Recursive Enumerable Language is a Type-0 Language which is generated by Type-0 Grammer.

This Language can be accepted or recognized by Turing Machine.

It means this Language string may or may not enter into Rejecting or Accepting states.

This Languages are also called as Turing recognizable Languages.

* Recursive Language:

=> Recursive Language is a Subset of Recursive Enumerable.

This Language can be decided by Turing machine which means it will enter into final state for the strings.

This Language are also called as Turing Decidable Languages.

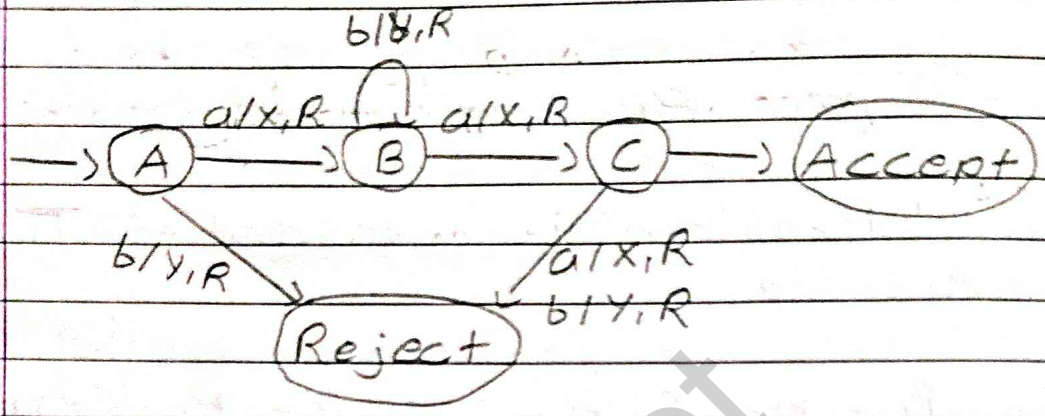
- Union : If Both L1 and L2 are Recursive then their Union will also be Recursive.
- Concatenation : If L1 and L2 are Recursive then their concatenation will also be ~~Conte~~ Recursive.
- Kleene Closure : If L1 is recursive, its kleene closure L1* will also be recursive.

K Define Decidable and Undecidable Problem.

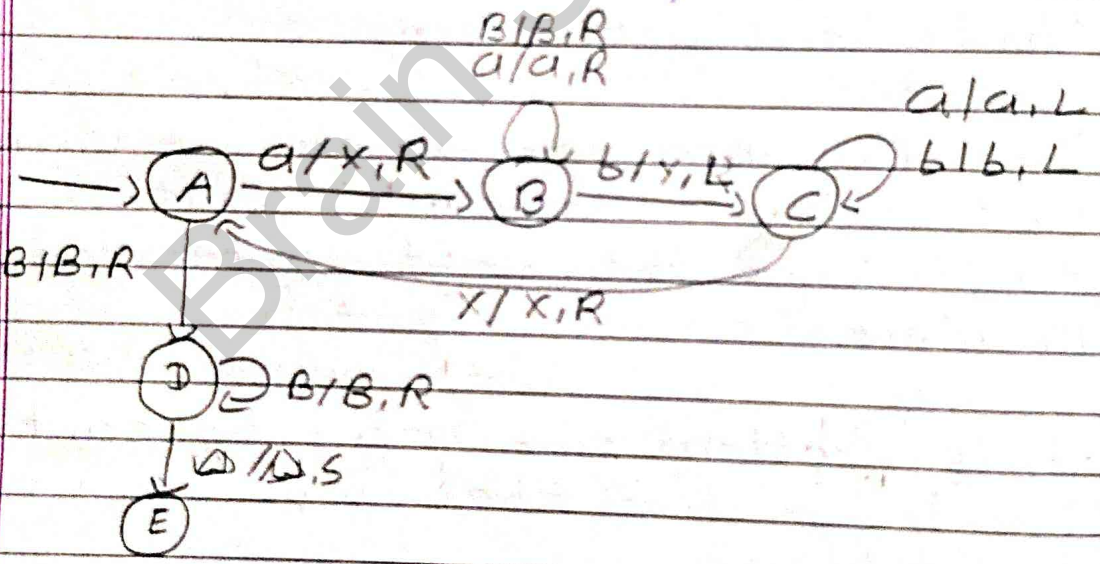
:> Decidable Problem :

A Problem is said to be Decidable if we can always

Ex. Draw a Turing Machine For
$L = \{ ab^*a \}$

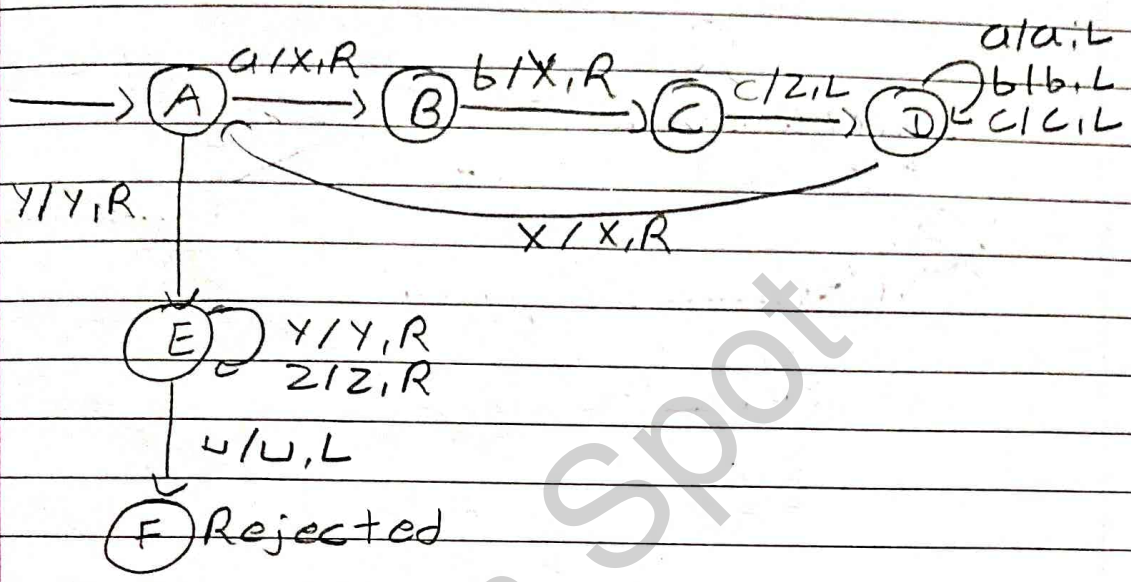


Ex. Draw a Turing Machine For
$L = \{ a^n b^n, \forall n \geq 1 \}$

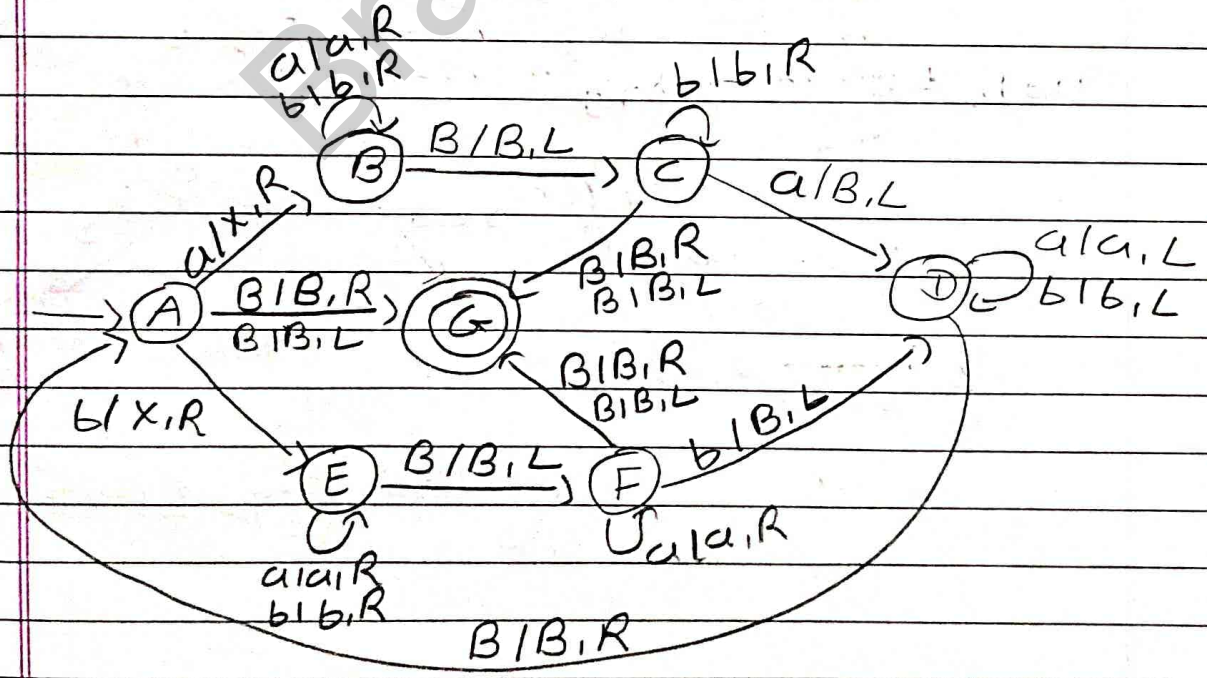


B = Black state

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

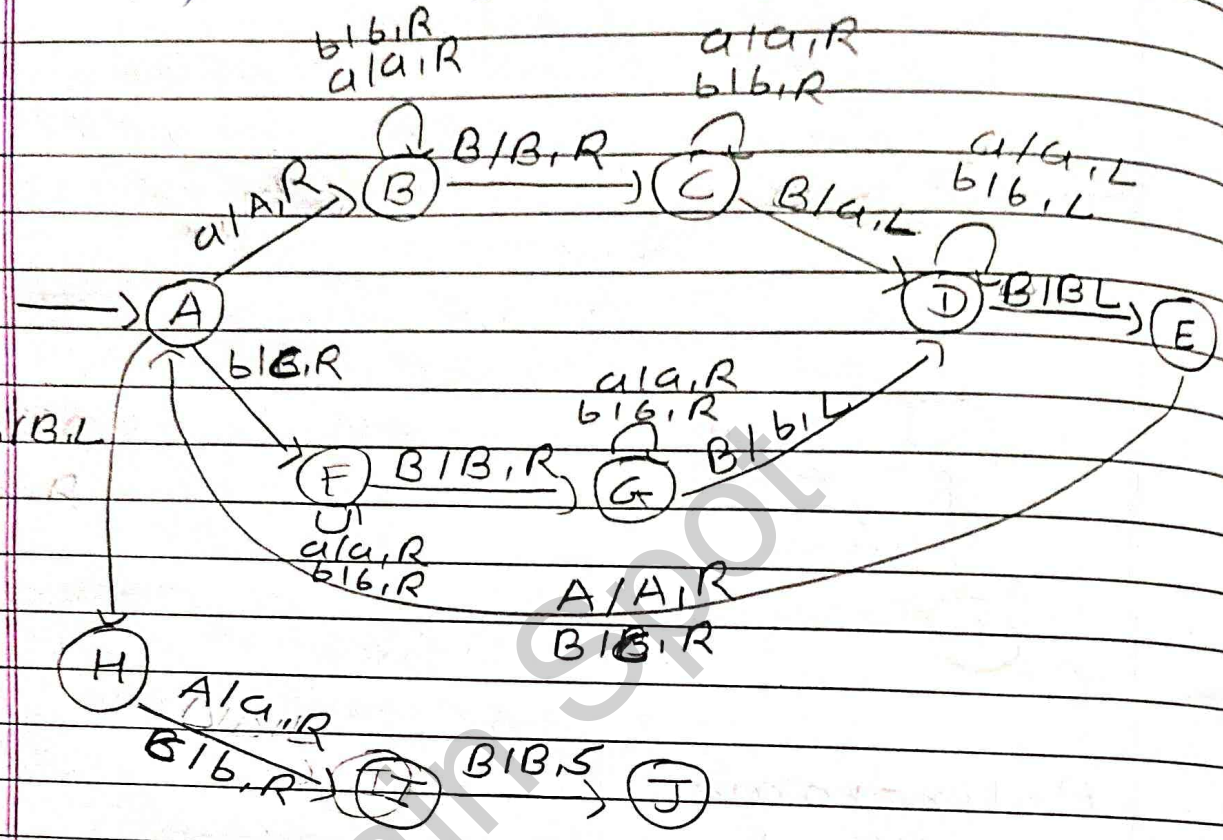


Ex. Palindrome



B = Black state

Ex. Copy String Turing Machine.



Ex Delete String: