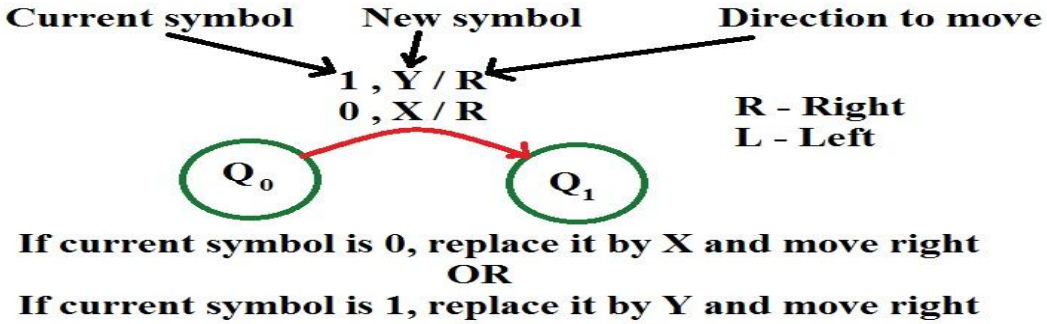
Turing machine:
A Turing machine is a finite automaton that can read, write, and erase symbols on an infinitely long tape. The tape is divided into squares, and each square contains a symbol. The Turing machine can only read one symbol at a time, and it uses a set of rules (the transition function) to determine its next action based on the current state and the symbol it is reading.
The Turing machine model uses an infinite tape as its unlimited memory. It
has a tape head that can read and write symbols and move around on the tape.

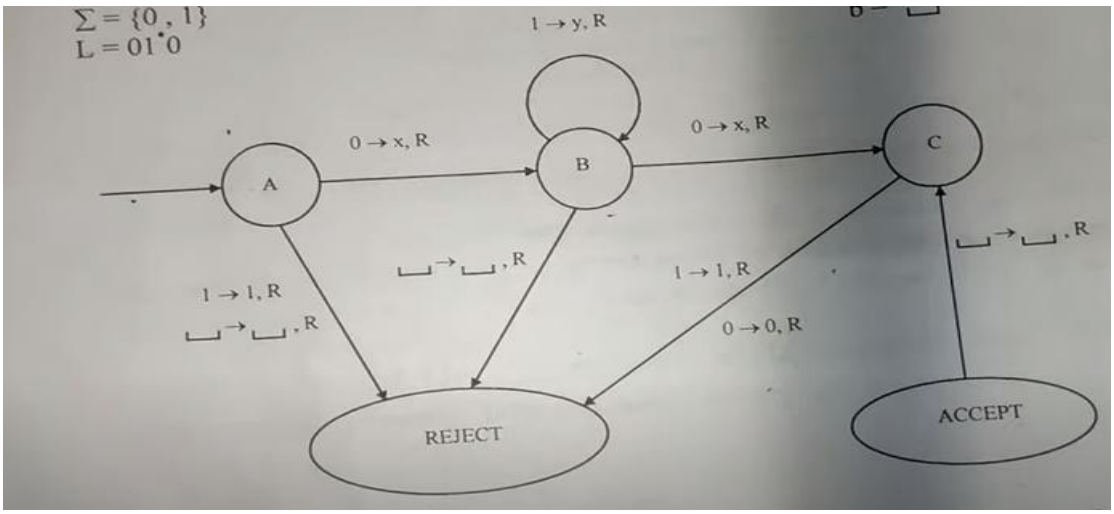
Formal Definition of a Turing machine
A ***Turing machine*** is a 7-tuple, (Q, Σ, Γ, δ, q0, qaccept, qreject), where
Q, Σ, Γ are all finite sets and
1. Q is the set of states,
2. Σ is the input alphabet not containing the ***blank symbol*** ,
3. Γ is the tape alphabet, where ∈ Γ and Σ ⊆ Γ,
4. δ : Q × Γ−→Q × Γ × {L, R} is the transition function,
5. q0 ∈ Q is the start state,
6. qaccept ∈ Q is the accept state, and
7. qreject ∈ Q is the reject state,

Transition Format:



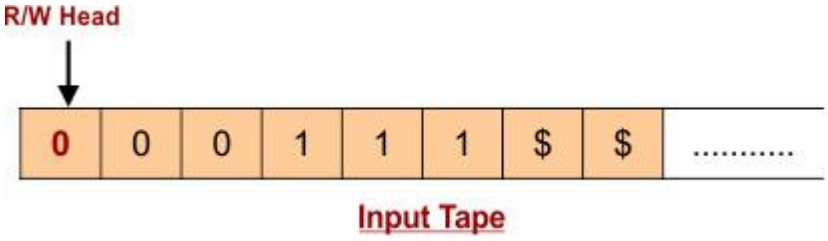
Designing Turing Machine:
1. Design a turing machine which recognizes the language L=01*0
Input Symbol:

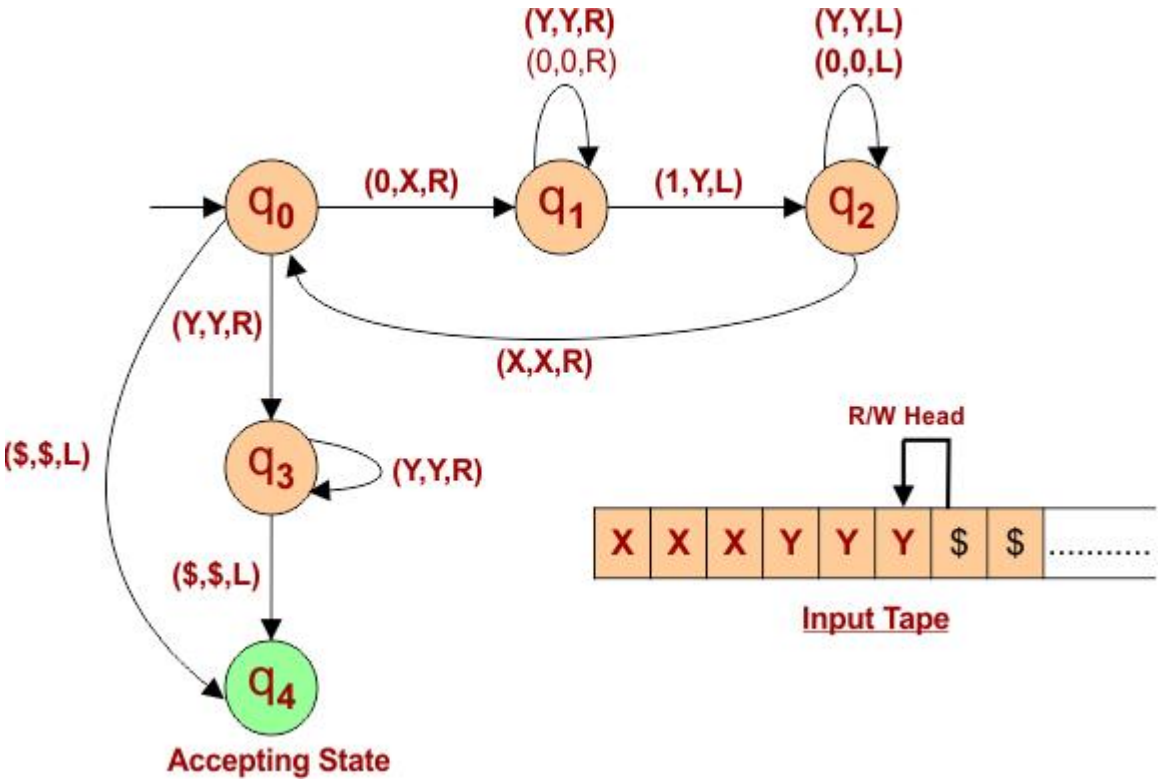
	0	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**B**



2. Design a Turing Machine which recognizes the language L = 0^n1^n where n>0.

consider a input-tap which hold L="000111" as given below

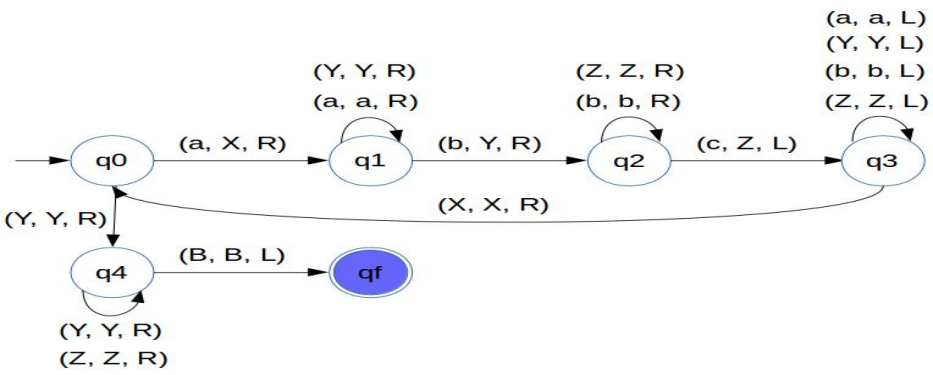


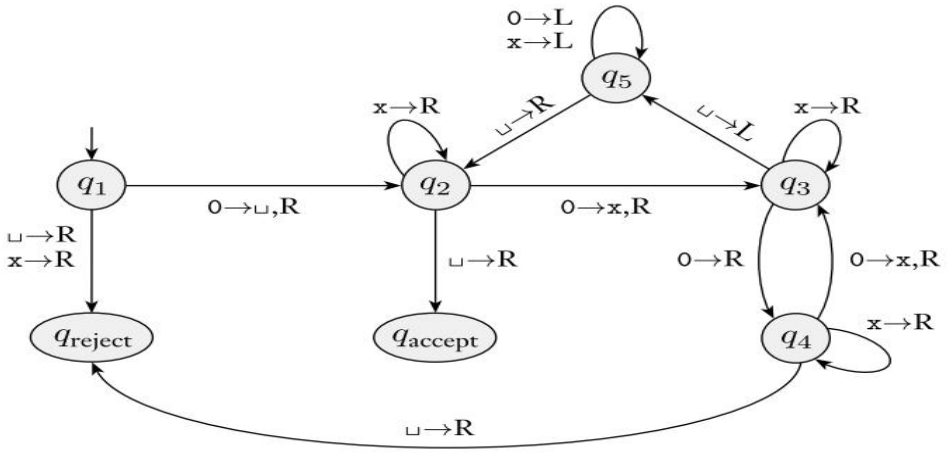


3. Design Turing Machine for the language L= $a^nb^nc^n$ | n ≥ 1

Input Tape:

B	a	a	a	b	b	b	c	c	c	B





Give The sequence of configurations the machine enters when started with the following strings
The input is 0000

So,The starting con-figuration is $q_1$0000

The sequence of configurations the machine enters appears as follows

For Practice See: https://www.youtube.com/watch?v=j2hlSTh9-BY&t=592s

Describing TMs

Formal definition: set of states, input alphabet, tape alphabet, transition function, state state, accept state, reject state.

Implementation-level definition: English prose to describe Turing machine head movements relative to contents of tape.

High-level description: Description of algorithm, without implementation details of machine. As part of this description, can "call" and run another TM as a subroutine.

Implementation-level descriptions of a Turing machine

- **Give an Implementation-level descriptions of a Turing machine that decides the following language over the alphabet {0, 1}.**

{w|w contains an equal number of 0s and 1s }

M = "On input string w −

- Scan tape and mark the first 0 which has not been marked. If no unmarked 0 is found, go to stage 4. Otherwise, move the head back to the front of the side of the tape.
- Scan tape and mark first 1 which has not been marked. If no unmarked then 1 is found, reject.
- Move the head back to front of the tape and go to stage 1.
- Move the head back to the front of the tape. Scan the tape to see if any unmarked 1s still remain. If none are found, accept; otherwise, reject."

- **Give an implementation-level description of a Turing machine that decides the language B = { 0 n1 n2 n | n ≥ 0 }.**

M = "On input string w:
1. Scan the input from left to right to make sure that it is a member of 0 ∗ 1 ∗ 2 ∗ , and reject if it isn't.
2. Return tape head to left-hand end of tape.
3. Repeat the following until no more 0s left on tape.
4. Replace the leftmost 0 with x.
5. Scan right until a 1 occurs. If there are no 1s, reject.
6. Replace the leftmost 1 with x.
7. Scan right until a 2 occurs. If there are no 2s, reject.
8. Replace the leftmost 2 with x.
9. Return tape head to left-hand end of tape, and go to stage 3.
10. If the tape contains any 1s or 2s, reject. Otherwise, accept."