CSC-257
Theory Of Computation
(BSc CSIT, TU)

Ganesh Khatri
kh6ganesh@gmail.com

Language of Turing Machine

- If $T = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ is a Turing machine and $w \in \Sigma^*$, then language accepted by T is defined by, $L(T) = \{w \mid w \in \Sigma^* \text{ and } q_0 w \vdash^* \alpha p \beta\}$ for some $p \in F$ and any tape string α and β.
- The set of languages that can be accepted using TM are called recursively enumerable languages or RE languages.

Language of Turing Machine

- The Turing Machine is designed to perform at least the following three roles :

1. **As a language recognizer :** TM can be used for accepting a language like Finite Automaton and Pushdown Automata.
2. **As a Computer of function :** A TM represents a particular function. Initial input is treated as representing an argument of the function and final string on tape when the TM enters the halt state; treated as representative of the value obtained by an application of the function to the argument represented by the initial string
3. **As an enumerator of string of a language :** It outputs the strings of a language, one at a time in some systematic order that is as a list.

Turing Machine for Computing a Function

- A Turing Machine can be used to compute functions.
- For such TM, we adopt the following policy to input any string to the TM which is an input of the computation function :
  - The string w is presented into the form BwB, where B is a blank symbol, and placed onto the tape; the head of TM is positioned at a blank symbol which immediately follows the string w
  - We can show by underlining that symbol to the current position of machine head in the tape as (q, BwB) or, we can represent it by ID of TM as BwqB
  - The TM is said to halt on input w if we can reach to halting state after performing some operation. i.e. If TM = (Q, Σ, Γ, δ, q0, B, F) is a Turing machine.

Turing Machine for Computing a Function : Formal Definition

- A function $f(x) = y$ is said to be computable by a TM and defined as; $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$ If $(q_0, BxB) \vdash^* (q_a, ByB)$; where $x \in \Sigma^*$, and $y \in \Sigma^*$
- It means that if we give input x to the Turing machine, it gives output as a string if it computes the function $f(x) = y$.

Turing Machine for Computing a Function : Formal Definition

- **Example :** Design a TM which computes the function f(x) = x+1 for each x belonging to set of natural numbers.
- Given the function, f(x) = x+1. Here, we represent the input x on the tape by a number of 1's on the tape
- i.e. for x=1, input tape will have B1B, for x=2 input tape will have B11B and so on.
- Similarly, output can be seen by the number of 1's on the tape when machine halts

	B	**1**
q0	(q0, 1, S)	(qf, 1, R)
q1		

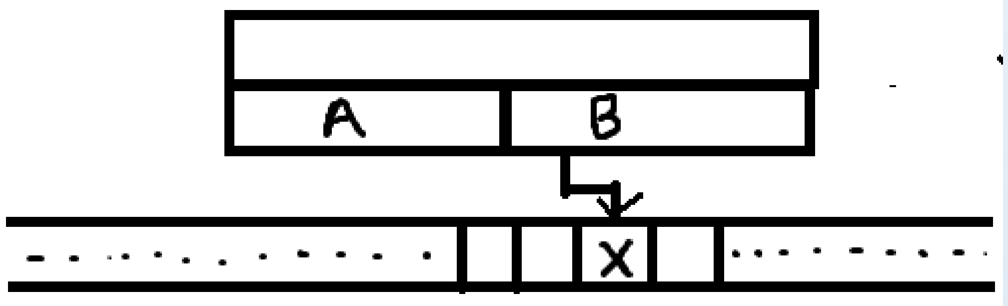
- So, let the input be x = 4. So, input tape at initial step consists of B1111B.
- (q0, B1111B) $\vdash^*$ (q0, B11111) $\vdash$ (q_f, B11111B).Which means output is 5 accepted

Turing Machines and Halting

- There is another notion of acceptance that is commonly used for Turing machines : acceptance by Halting.
- We say a TM halts if it enters a state q, scanning a tape symbol X and there is no move in this situation; i.e $\delta(q, X)$ is undefined.
- The Turing machine described above was not designed to accept any language rather we viewed it as computing an arithmetic function.
- We always assume that a TM halts if it accepts. i.e. without changing language accepted, we can make $\delta(q, X)$ undefined whenever q is an accepting state

Turing Machine with Storages in the state

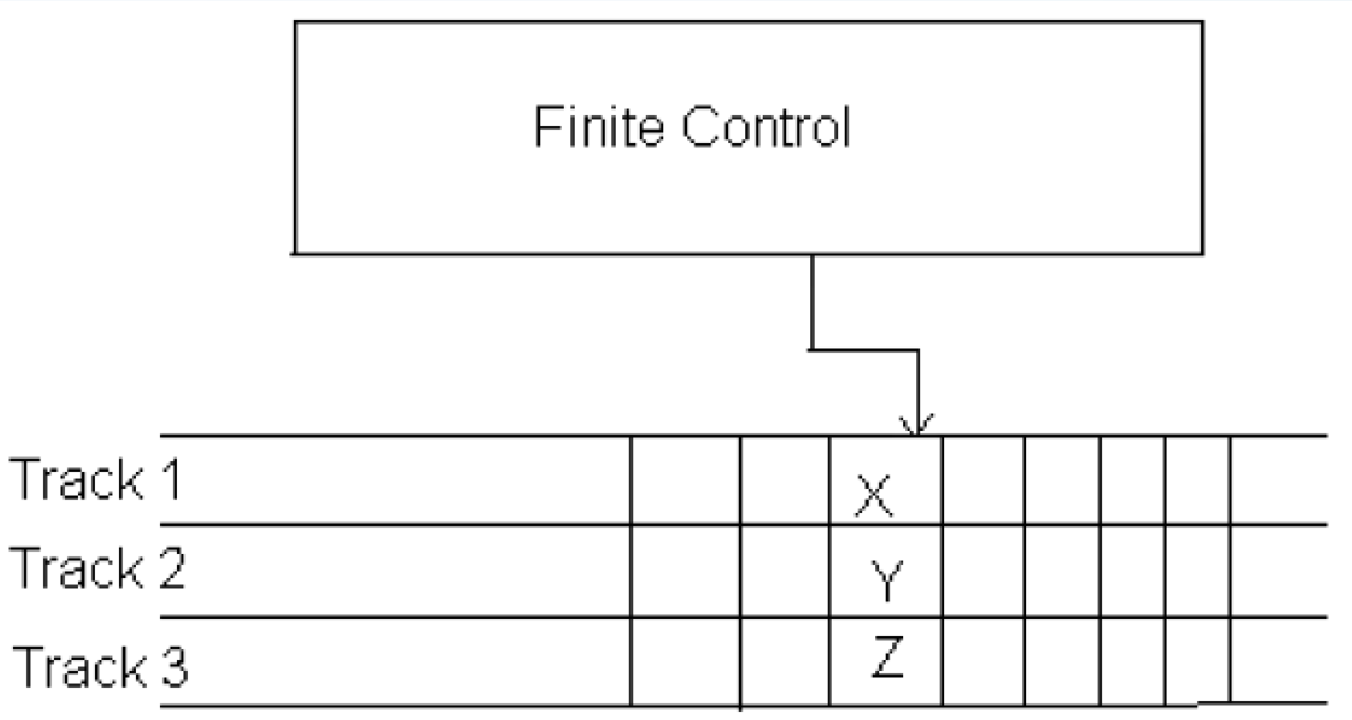
- In Turing Machine, generally, any state represents the position in the computation.
- But the state can also be used to hold a finite amount of data.
- We can use the finite control not only to represent a position in the computation / program of the Turing Machine, but to hold a finite amount of data.
- In this case, a state is considered as a tuple; (state, data).



- δ is defined by : δ([q, A], X) = ([q1, X], Y, R); means that q is the state and data portion associated with q is A. The symbol scanned on the tape is copied into the second component of the state and moves right entering state q1 and replacing tape symbol by Y.

Turing Machine with Multiple Tracks

- The tape of TM can be considered as having multiple tracks.
- Each track can hold one symbol, and the tape alphabet of the TM consists of tuples, with one component for each track.
- Following figure illustrates the TM with multiple tracks :

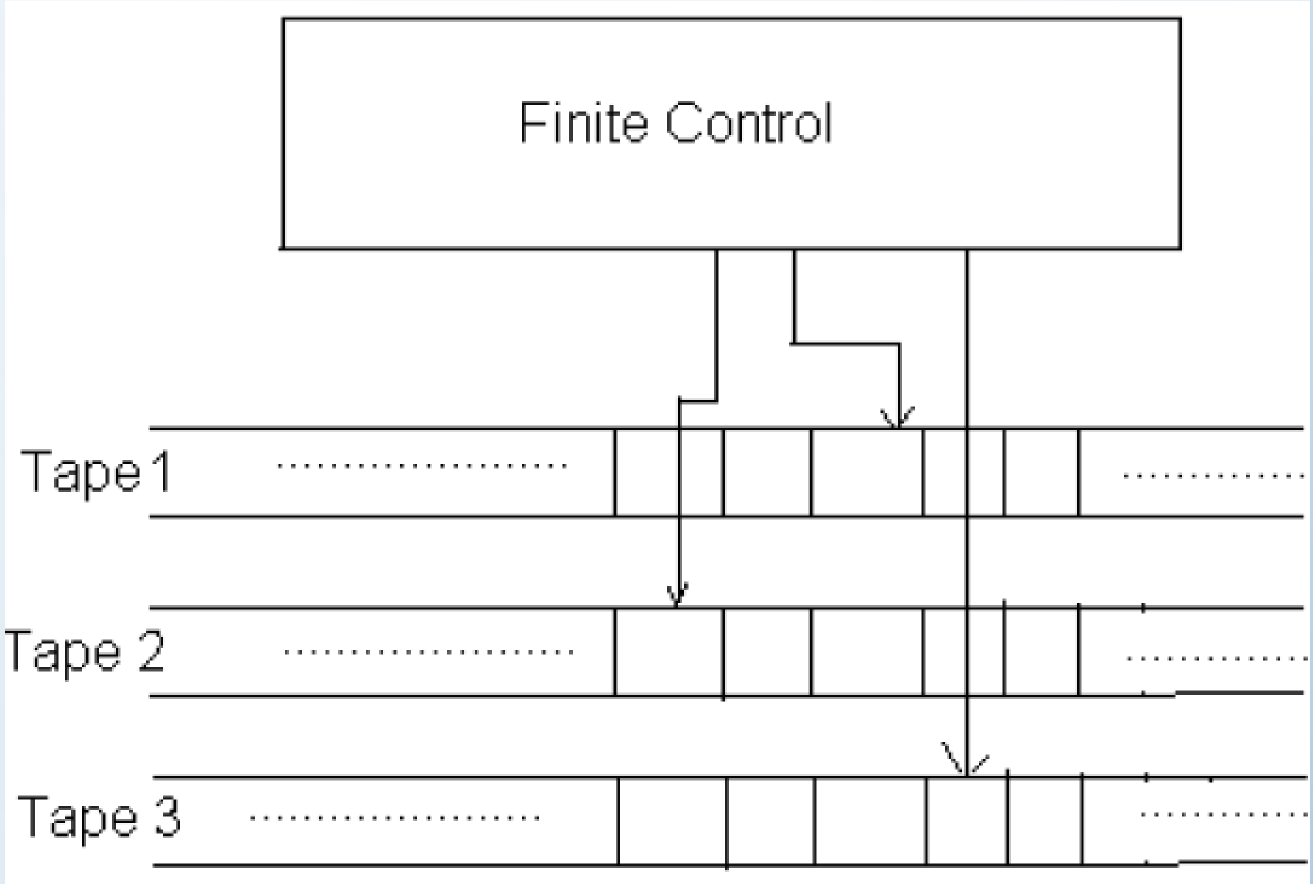


Turing Machine with Multiple Tracks

- The tape alphabet Γ is a set consisting of tuples like : Γ = { (X, Y, Z), ……… }
- The tape head moves up and down scanning symbols in the tape at one position.
- Like the technique of storage in the state, using multiple tracks does not extend what the TM can do.
- It is simply a way to view tape symbols and to imagine that they have a useful structure

Multi Tape Turing Machine

- Modern computing devices are based on the foundation of TM computation models.
- To simulate the real computers, a TM can be viewed as multi-tape machine in which there is more than one tape.
- However, adding extra tape adds no power to the computational model, only the ability to accept the language is concerned.



Multi Tape Turing Machine

- A multi-tape TM consists of finite control and finite number of tapes.
- Each tape is divided into cells and each cell can hold any symbol of finite tape alphabets.
- The set of tape symbols include a blank and the input symbols.
- Initially,
  - The Input (finite sequence of input symbols) w is placed on the first tape.
  - All other cells of the tapes hold blanks.
  - TM is in initial state q0.
  - The head of the first tape is at the left end of the input.
  - All other tape heads are at some arbitrary cell. Since all other tapes except first tape consists completely blank

Multi Tape Turing Machine

- A move of multi- tape TM depends on the state and the symbol scanned by each of the tape head. In one move, the multi-tape TM does the following :
  - The control enters in a new state, which may be same previous state.
  - On each step, a new symbol is written on the cell scanned, these symbols may be same as the symbols previously there.
  - Each of the tape head make a move either left or right or remains stationary. Different head may move different direction independently i.e. if head of first tape moves leftward; at same time other head can move another direction or remains stationary