UNIT V : TURING MACHINES

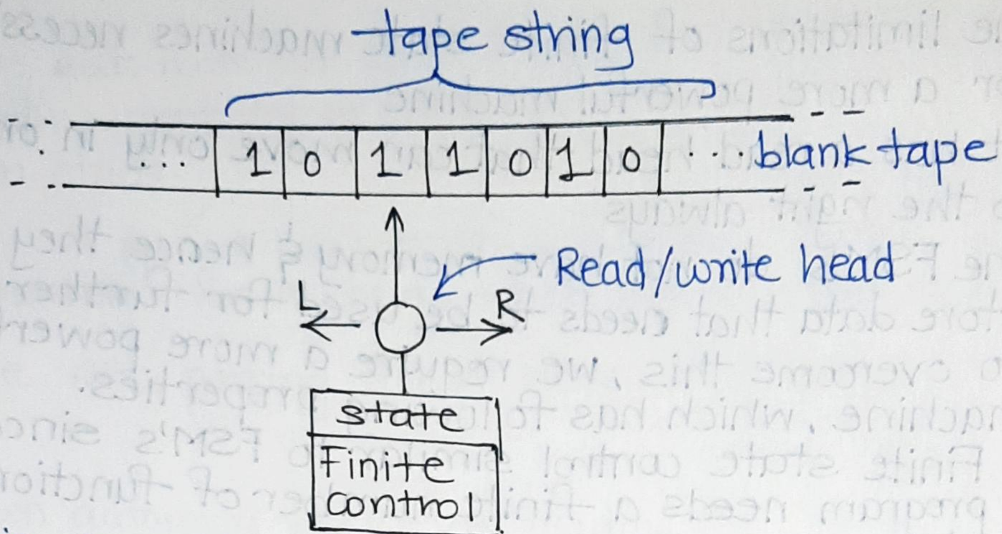
- The limitations of finite state machines necessitates for a more powerful machine
- It has a read head that can move only in one direction to the right always
- The FSM's do not have memory & hence they cannot store data that needs to be used for further computation
- To overcome this, we require a more powerful machine, which has following properties.

1. Finite state control similar to FSM's since many program needs a finite number of functions or states
2. External memory capable of remembering arbitrarly long sequence of i/p's to achieve this, the machine should have unbounded one dimentional power from which it can choose any required part by moving to the left right or by staying in one position (i.e no movement)
   - the entire unbounded tape acts as an infinite memory.
3. Ability to consume its own o/p as i/p during execution at a later time.
4. Ability to store (write onto tape) - the machine head must be a read/write head.

- All the aforementioned characteristics are satisfied by the turing Machine (TM), which was proposed by Alan Turing, a decade prior to the 1st shared program computer.
- This concept of TM lead to the concept of algorithm & finite procedures since the basis of TM is to 1st divide the process into primitive operations (functions/procedures/states) such that they cannot be further divisible & then execute operations in sequence.

[FSM - Finite state machine TM - Turing Machine]

ELEMENTS OF TURING MACHINE



1) Head is a part which can read/write at a time & move either to the left or right or remain at same position
2) An infinite tape extending on either sides of the head marked off into squares on which the symbols from an alphabet set can be written.
3) A finite set of symbols called external alphabets set I, which consist of small letters, punctuation mark & Blank character.
4) A finite set of states denoted by 's' the machine resides in one of these states.

FORMAL DEFINITION:

$M = \{Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Q - Finite set of states , Σ - finite set of input symbols.

Γ - A finite set of symbols in tape. $\cup B$, B - blank symbol

q_0 - Initial state , F - final state. set

δ - functional matrix.

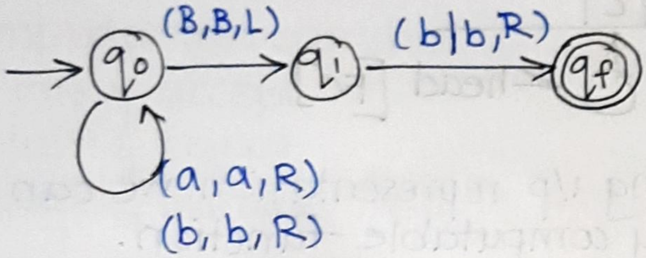
$\delta : (Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{L, R, M\})$.

$L \rightarrow$ Left move, R - right move, M - no, move

TYPES OF TM

1.) Composite turing machine
2.) Iterative turing machine
3.) Multitape turing machine
4.) Multihead turing machine
5.) Non deterministic TM
6.) Multi dimentional TM
7.) Universal Turing Machine.

example:
construct a TM that accepts language over {a, b} that contain string ending with 'b'



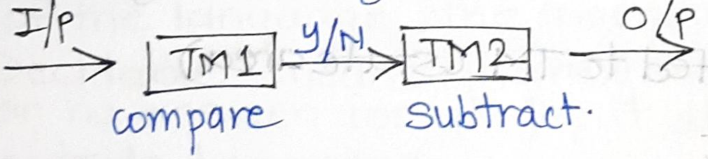
LANGUAGE ACCEPTED BY TM.

Type 0 grammar (Unrestricted grammar) generated language accepted by TM

Composite TM:

when o/p of TM1 is used as input of TM2, then it is composite
Each TM has a seperate Finite control

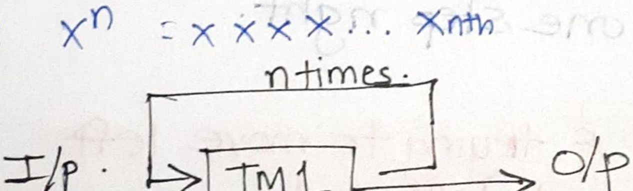
eg: $f(x,y) \begin{cases} x-y & \text{if } x \neq y \\ 0 & \text{else.} \end{cases}$



Iterative TM:

When problem requires same logic to be repeated a finite number of times then it is Iterative TM

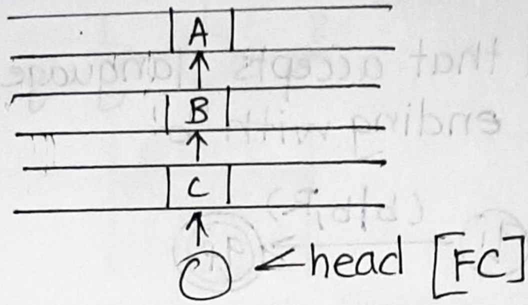
eg: $x^n = x\,x\,x\,x \ldots x_{nth}$



Multiple / Multitape TM

Model of modern computer where there are many i/p tape & one read /write head

There is one central FC (Finite control) which is creating multiple instances of problem basedon i/p



Universal TM :

Turing stated that by adjusting i/p representation we can cause TM to compare any computable function.

Theabove capability results in a universal TM

UTM has property that for every TM "T", there is string of symbol "d_T" such that if no 'x' is written in unary notation on blank tape followed by string d_T & UTM has started on state q_0 on left most symbol of d_T then f(x) will appear on tape & Machine stops

In UTM following information is stored. on tape.

a) Description of TM in terms of finite automata (operation area)

b) Initial configuration of TM i.e current state & symbol scanned (state area)

c) Processing data to be fed to TM (state area)

- OPERATION ON THE TAPE

- Read / scan symbol below the tape Head
- Update / write a symbol below the tape Head
- Move the tape head one step left
- Move the tape head one step right.

If at left end of tape & trying to move left then do not move stay at left end

Symbol to read ← a, b, R → Direction to move

↓

Symbol to write

- Rules of operation.
  - Control is with a sort of FSM
  - Initial state
  - Final states : (There are 2 final state)
    - • Accept state • Reject state
- Computation can be
  1) Halt & accept
  2) Halt & reject
  3) Loop (machine fails to halt)

TURING UNRECOGNIZABLE LANGUAGE

A Turing unrecognizable language (or non-recursively enumerable language) is a language for which there is no turing machine that can decide whether a given string belongs to the language. In other words, while a turing machine may halt and accept strings that are in the language, it may never halt and loop forever for strings not in language.

Recognizable : means a turing machine can accept string in the language, but for string outside of the language, the machine may not halt.

Decidable : means a turing machine halts with a yes or no answer for all input strings (both inside & outside language.

- A unrecognizable language cannot be recognized by any Turing machine.

REDUCIBILITY :

In computational theory is a way of relating two problems in terms of their difficulty. If problem A can be reduced to problem B, then a solution to problem B can be used to solve problem A

There are many types of reductions.

Many-one reduction (mapping reduction):

problem A is reducible to problem B if there exist a computable function that transforms any instance of A into an instance of B preserving the answer.

Turing reduction :

Problem A is reducible to problem B if there exist a Turing machine that ~~uses~~ solves A using an black box machine which solves B in single step

If A is reducible by B it means that if we solve B we can also solve A though reverse is not necessarily true.

RECURSION THEOREM

The recursion theorem (Fixed point theorem) is a result in computational theory that states that for any computable function, there exists an input such that the function produces that input as output. This ~~forms~~ is a form of self reference in computation.

- For any TM M there exist input ω such that $M(\omega) = \omega$

LINEAR BOUNDED AUTOMATA

It is a restricted form of TM

LBA is a TM that uses tape space that is linearly bounded by the size of the input. If the input is of length n then the tape has O(n) cells of tape

i.e tape size is ~~linearly~~ linearly bounded which makes it more powerful than FA but less powerful than TM