unit - 5

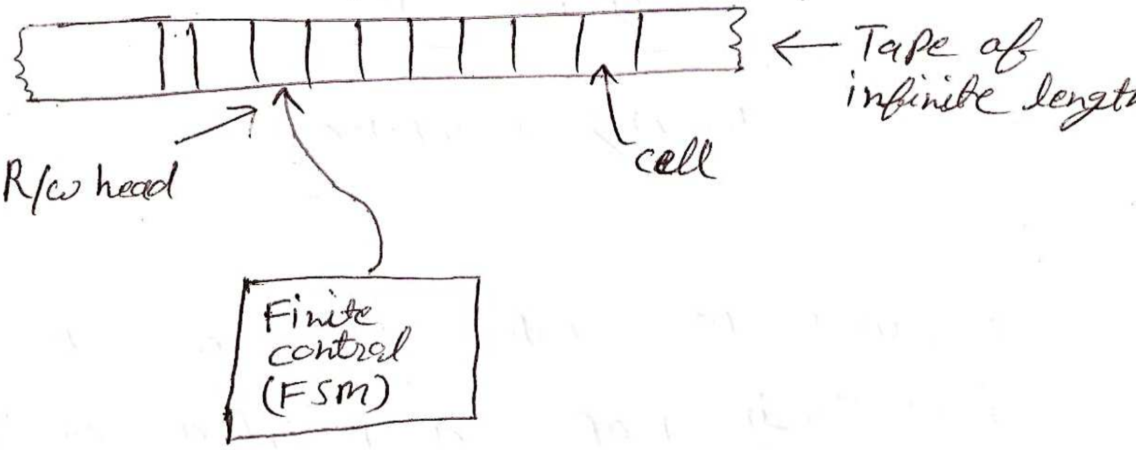
Turing machine

Arround 1936, when there was no computers, Alen Turing proposed a model of an abstract machine called the turing machine. which could Perform any computational Process carried out by the Present day's of computer. The machine was named after turing, and it is called the turingmachine. The Turing machine is the machine format of unrestricted language. i.e. all types of languages are accepted by the turing machine.

Based on the turing machine, a new theory called the "theory of undecidable Problems" is developed These types of Problums can not be solved by any computer.

Turing machine model:

The Turing machine can be thought of as finite control connected to a (Read/write) R/W head. it has one tape which is divided into a number of cells. The block diagram of the basic model for the Turing m/c is given below.



Each cell can store only one symbol. The input and output of Finite state machine are effected by the R/w head which can examine one cell at a time.

in one move, the m/c examines the Present ~~state~~ symbol under the R/w head on the tape and the Present state of an automaton to determine

(i) a new symbol to be written on the tape in the cell.
(ii) either the head moves one cell left (L) or one cell right (R).
(iii) the Next state of Automaton
(IV) whether to halt or not.

mathematically a turing m/c is a 7 tuple m/c

$$m = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

where Q is a finite set of states.
Σ is a finite set of ~~tape~~ input symbols. $B \notin \Sigma$
Γ is a finite set of tape symbols. $\{\Sigma \subset \Gamma\}$
δ is a transition function, mapping $Q \times \Gamma$ to $Q \times \Gamma \times (L/R)$
q_0 is initial state $\{q_0 \in Q\}$
B is a blank symbol of tape.
F is the final set of final states. $\{F \subseteq Q\}$

Q Design a turing m/c for $L = a^n b^n : n \geq 1$

ex.

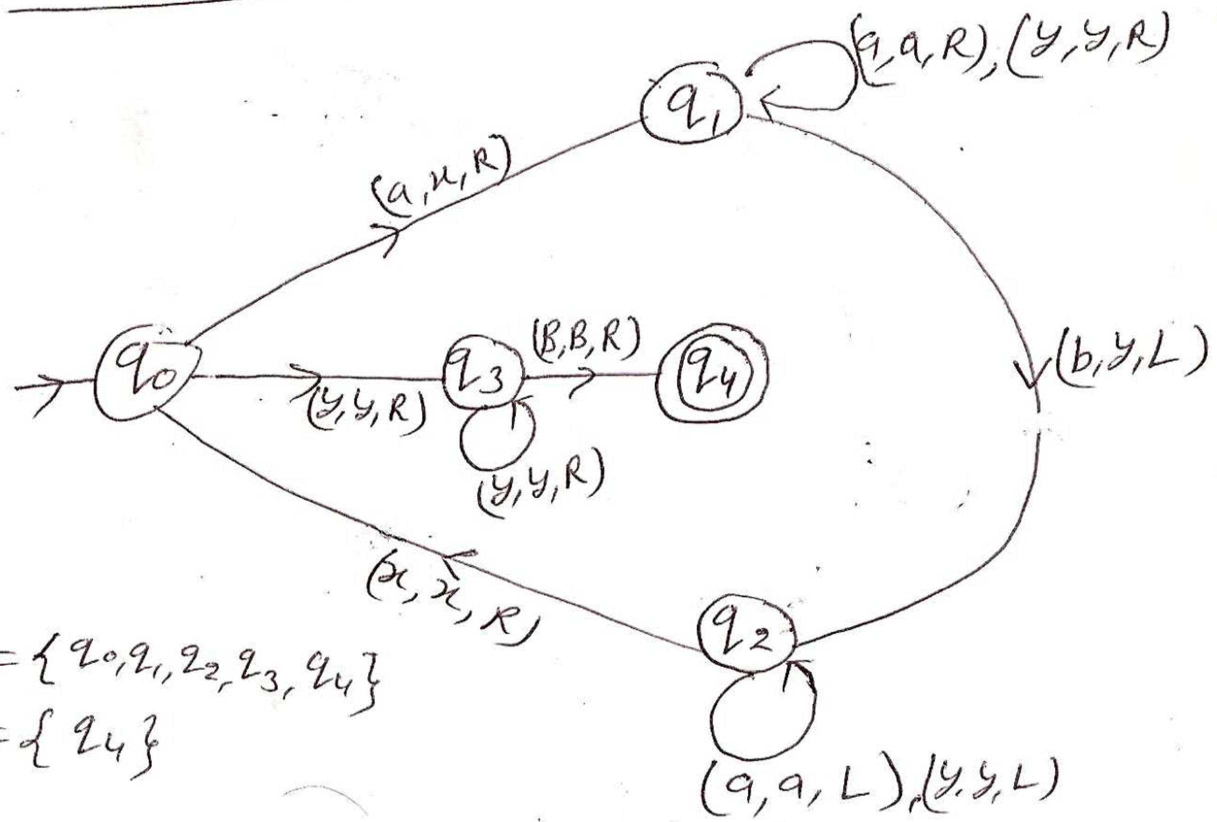
a	a	a	b	b	b	B	B	---

$M(Q, \{a,b\}, \{a,b,x,y,B\}, \delta, q_0, F)$

δ :

	a	b	x	y	B
q_0	(q_1, x, R)			(q_3, y, R)	
q_1	(q_1, a, R)	(q_2, y, L)		(q_1, y, R)	
q_2	(q_2, a, L)		(q_0, x, R)	(q_2, y, L)	
q_3				(q_3, y, R)	(q_4, B, R)
q_4					

Transition diagram:



$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$F = \{q_4\}$

Q Design a Turing m/c for $L = a^n b^n c^n : n \geq 1$

$m = (Q, \{a, b, c\}, \{a, b, c, x, y, z, B\}, \delta, q_0, F)$

ex

a	a	a	b	b	b	c	c	c	B	B	---

δ:	a	b	c	x	y	z	B
q_0	(q_1, x, R)				(q_4, y, R)		
q_1	(q_1, a, R)	(q_2, y, R)			(q_1, y, R)		
q_2		(q_2, b, R)	(q_3, z, L)			(q_2, z, R)	
q_3	(q_3, a, L)	(q_3, b, L)		(q_0, x, R)	(q_3, y, L)	(q_3, z, L)	
q_4					(q_4, y, R)	(q_4, z, R)	(q_5, B, L)
q_5							

Transition Diagram:

Instantanous Description (ID) for a string "aabbcc"

step
(1) Read always Right symbol of state.
(2) move state name and position on the basis of transition f^n's head move
(3) Rename the i/p symbol if Required.

$[q_0 aabbccB] \vdash [X q_1 abbccB] \vdash [Xa q_1 bbccB]$

$\vdash [Xa Y q_2 bccB] \vdash [XaYb q_2 ccB] \vdash [XaY q_3 bZcB]$

$\vdash [Xa q_3 YbZcB] \vdash [X q_3 aYbZcB] \vdash [q_3 XaYbZcB]$

$\vdash [X q_0 aYbZcB] \vdash [XX q_1 YbZcB] \vdash [XXY q_1 bZcB]$

$\vdash [XXYY q_2 ZcB] \vdash [XXYYZ q_2 cB] \vdash [XXYY q_3 ZZB]$

$\vdash [XXY q_3 YZZB] \vdash [XX q_3 YYZZB] \vdash [X q_3 XYYZZB]$

$\vdash [XX q_4 YYZZB] \vdash [XXY q_4 YZZB] \vdash [XXYY q_4 ZZB]$

$\vdash [XXYYZ q_4 ZB] \vdash [XXYYZZ q_4 B] \vdash [XXYYZZ q_5 B]$

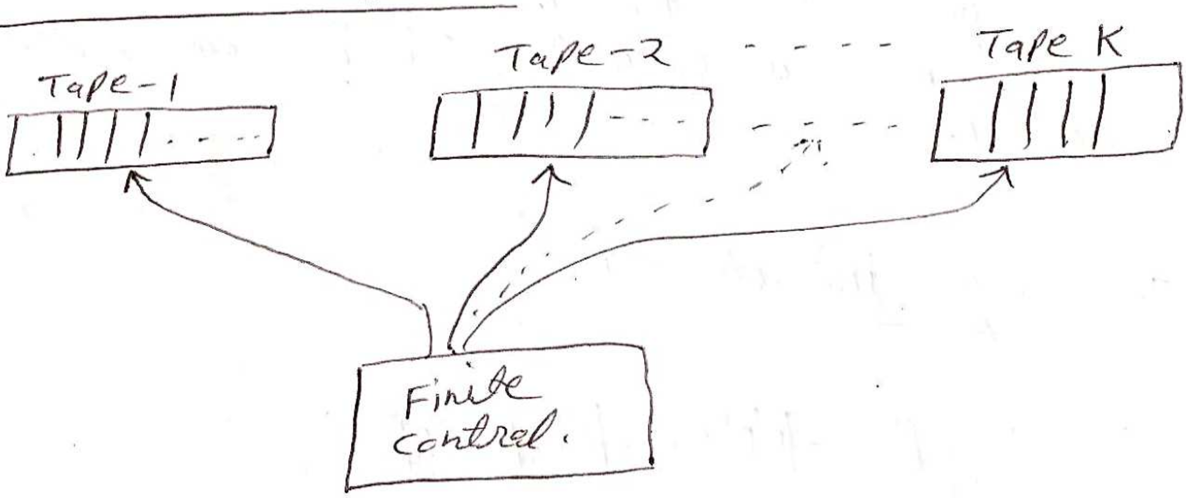
Language Acceptability by turing m/c :

Let us consider the turing m/c $M = (Q, \Sigma, \Gamma, \delta, q_0, b, F)$. A string w in Σ^* is said to be accepted by M if $q_0 w \vdash^* \alpha_1 q_f \alpha_2$ for some $q_f \in F$ and $\alpha_1, \alpha_2 \in \Gamma^*$.

M does not accept w if the machine M either halts in nonaccepting state or does not halt.

Variants of turing m/c :

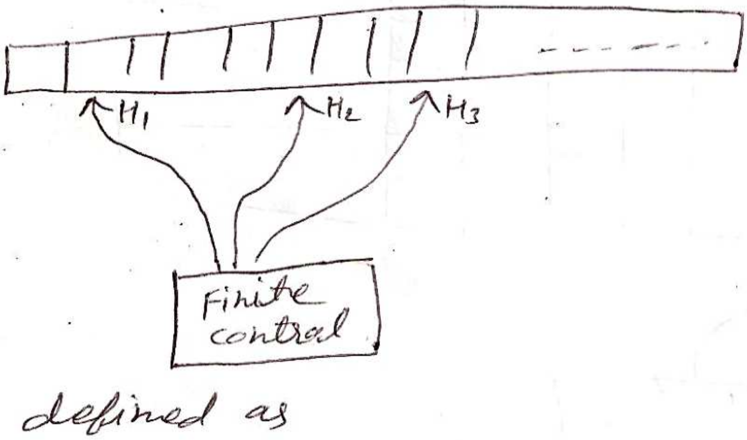
(I) multitape turing m/c



here δ is defined as.

$$Q \times (\Gamma_1, \Gamma_2, \Gamma_3, \cdots \Gamma_k) \longrightarrow (Q \times (\Gamma_1, \Gamma_2, \cdots \Gamma_k) \times ((L/R), (L/R), \cdots$$

(II) multihead turing m/c :



here δ is defined as

$$Q \times (\Gamma_1, \Gamma_2, \Gamma_3) \longrightarrow (Q \times (\Gamma_1' \times (L/R)) \times (\Gamma_2' \times (L/R)) \times (\Gamma_3' \times (L/R)$$

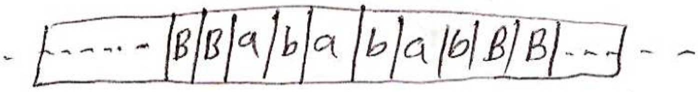
read by head H_1

write by head H_1

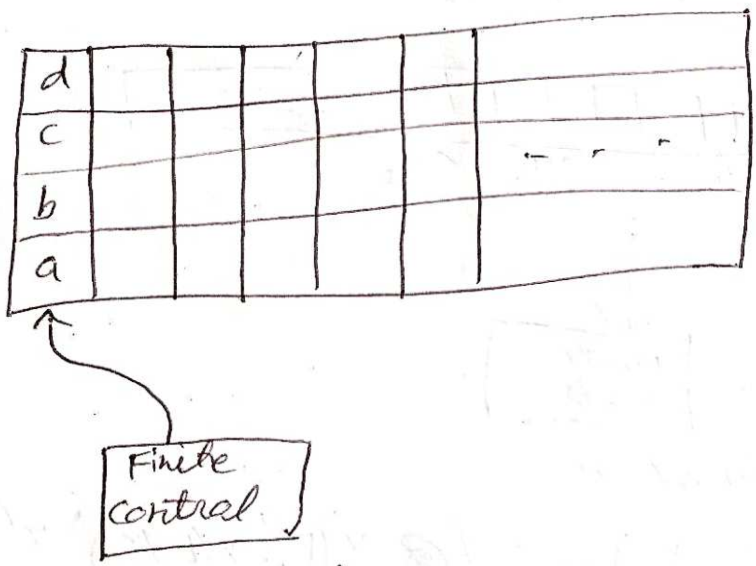
A situation may arise when more than one heads are scanning a particular cell at a time but symbol written by one head is different from the symbol written by other head.

in this situation, Priority among the heads is defined.

(3) Two-way infinite tape:



(4) K dimensional / multitrack turing m/c:



$$\delta(q_0, (a,b,c,d)) = (q_1, (x,y,c,d), R)$$

Non Deterministic turing m/c :

For a combination of single present state and a single input symbol there may be more then one move then m/c is called Non deterministic turing m/c

$$Q \times \Sigma \rightarrow \text{Power set of } (Q \times \Gamma \times (L/R))$$

Universal turing m/c :

The turing m/c can Perform any computational Process carried out by the Present day's computer. the difference between a turing m/c and a real computer is that the turing m/c designed to execute only one Program but real computers are reprogrammable.

A turing m/c is called universel turing m/c if it simulate the behaviour of a digital computer. So a universal turing m/c can simulate all the turing m/c's designed for each separate task.

CHURCH's Thesis:

Alonzo church, an Americal mathematician Proposed that any machine that can Perform a certain list of operations will be able to Perform all Possible algorithms. And Alen turing. Proposed that machine called the turing machine. means every computer algorithm can be implemented as a turing machine.

For writing a Program, we need to construct an algorithm first. 'No computational Procedure is considered as an algorithm unless it is represented by turing m/c. This is known as the church thesis. This thesis can not be Proved; it is generally accepted truth. for this reason it is called a thesis not a theorem.

Recursive and Recursively enumerable:

A formal language is recursive if there exists a turing m/c which halts for every given input and always either accepts or Rejects candidate strings. This is also called decidable language.

A recursively enumerable language require that some turing m/c halts and accepts when Presented with a string in the language. it may either halt and reject, or loop forever when Presented with a string not in the language.

* every Recursive language is Recursively enumerable.

Properties of Recursive & Recursively Enumerable Language

(1) The union of two recursive language is recursive

(2) The union of two recursively enumerable language is Recursively enumerable.

(3) The intersection of two recursive language is recursive.

(4) The intersection of two recursively enumerable languages is recursively enumerable.

(5) the complement of a recursive language is Recursive.

* (6) if a language and its complement both are Recursively enumerable, then the language and its complement both are recursive.

(7) the concatenation of two recursive language is recursive

(8) the concatenation of two recursive enumerable language is recursively enumerable.

* (9) The intersection of a recursive & a recursively enumerable language is recursively enumerable.

(10) The Kleene closure operation on a Recursive language is Recursive language.

(11) The Kleene closer operation on a Recursively enumerable language is recursively enumerable.

Halting Problem Of Turing machine:

There are certain set for which turing machine can be design but the Problum accure that after the exhausting the input string, the head of turing machine will not halt. it will continiously either toggle between two position or moving continiously is in one direction.

This Problem of turing machine is called halting Problum.

This Problem of turing machine is known as undecidable Problems.

Undecidability:

There does not exist any turing machine which accepts the language and makes a decision by halting for every input string (may halt for some strings but not for all).

Reducibility:

We say that Problem A is reducible to Problem B if a solution to Problem B can be used to solve Problem A.

ex. A is the Problem to finding some root of $x^4-3x^2+2=0$ and B is the Problem to finding some root of $x^2-2=0$ then A is reducible to B. As x^2-2 is a factor of x^4-3x^2+2, a root of $x^2-2=0$ is also a root of $x^4-3x^2+2=0$.

Post's correspondence Problem (PCP):

The Post correspondence Problem (PCP) was Proposed by an American mathematician Emil Leon Post in 1946. this is a well- Known undecidable Problems in the theory of computer science. PCP is very useful for showing the undecidability of many other Problems by reducibility.

consider the two lists $x = (x_1, x_2, \ldots x_n)$, $y = (y_1, y_2 \ldots y_n)$ of nonempty strings over alphabet Σ. the PCP is to determine whether or not their exist $i_1, i_2, \ldots i_m$, where $1 \leq i_j \leq n$ such that

$$x_{i_1} \ldots x_{i_m} = y_{i_1} \ldots y_{i_m}$$

Note: The indices i_j's need not to be distict and m may be greather then n. Also if there exists a salution to PCP, there exist infinitely many salutions.

ex: Does the PCP with two list $x = (b, bab^3, ba)$ and $y = (b^3, ba, a)$ have a salution?

Salution: we have to determine whether or not there exist a sequence of substrings of x such that the string formed by this sequence and the string formed by the sequence corresponding substrings of y are identical the required sequence is given by $i_1 = 2, i_2 = 1, i_3 = 1, i_4 = 3$ i.e. (2, 1, 1, 3) and $m = 4$

The corresponding strings are

$$\underbrace{\boxed{bab^3}}_{x_2}\ \underbrace{\boxed{b}}_{x_1}\ \underbrace{\boxed{b}}_{x_1}\ \underbrace{\boxed{ba}}_{x_3} = \underbrace{\boxed{ba}}_{y_2}\ \underbrace{\boxed{b^3}}_{y_1}\ \underbrace{\boxed{b^3}}_{y_1}\ \underbrace{\boxed{ba}}_{y_3}$$

Thus the PCP has a solution.

a Does the PCP with two list W(11, 100, 111),
V(111, 001, 11) have a solution?

$w_1 = 11$, $w_2 = 100$, $w_3 = 111$
$v_1 = 111$ $v_2 = 001$ $v_3 = 11$

$$\underbrace{11}_{w_1}\ \underbrace{100}_{w_2}\ \underbrace{111}_{w_3} = \underbrace{111}_{v_1}\underbrace{001}_{v_2}\underbrace{11}_{v_3}$$

So (1, 2, 3) is a solution

$$\underbrace{11}_{w_1}\underbrace{111}_{w_3} = \underbrace{111}_{v_1}\underbrace{11}_{v_3}$$

So (1, 3) is a solution

$$\underbrace{111}_{w_3}\underbrace{11}_{w_1} = \underbrace{11}_{v_3}\underbrace{111}_{v_1}$$

So (3, 1) is a solution

<u>Note:</u> (I) if $\forall i$ $|w_i| > |v_i|$ or $|v_i| \geq |w_i|$
then no solution possible.
(II) if first letter is not matching in any of string than no PCP solution.

Q Does the PCP with two list $w (w_1, w_2, w_3)$ & $V (V_1, V_2, V_3)$ have a solution ? where

$w_1 = 11$, $w_2 = 101$, $w_3 = 110$

$V_1 = 1$, $V_2 = 10$, $V_3 = 01$

Sol. no solution because $|w_i| > |V_i|$, $\forall i$

modified Post corresponding Problem :- (MPCP)

it is restricted form of PCP. The restriction is that ~~it~~ it must be start with first string (w_1, V_1)

that is, There is MPCP solution exist if there exist a sequence of integer $i_1 \cdots i_m$ such that $\underline{w_1}\, w_i w_i \cdots w_k = \underline{V_1}\, V_i V_i \cdots V_k$

fix

that is MPCP solution must start with w_1 & V_1 on right side.

<u>Note</u> : ① if MPCP solution will exist then PCP must have solution.

(II) if PCP solution exist then MPCP ~~#~~ solution may or may not be possible.