27/10/17

MODULE - 4

Where Do context free languages Fit in Big Picture.

→ Regular languages are the proper subset of context free languages. If L is regular then there is a DFSM $M = (K \Sigma \delta s A)$ which ~~rego~~ recognizes L

→ similarly there exists a PDA $M' = (K \Sigma \Gamma \Delta s A)$ which recognizes the Regular language L.

To simulate the operations of DFSM, PDA must include the following transition Rules.

$\boxed{\delta(q_i, a, q_j)}$, it can be rewritten as

$\boxed{\Delta : (q_i, a, \varepsilon)(q_j, \varepsilon)}$

Closure Properties of context free language.

1. context free languages are closed under union, concation, Kleen star, Reversal.

closed under	Not closed under
union	complementation.
concatination	Intersection.
kleen star	difference.
Reversal	
Letter substitution	

4. context free languages are closed under Union.

Proof

L_1 and L_2 are the two CFL

G_1 and G_2 are the corresponding CFG. (Grammar)

thus

$L_1 = L(G_1)$

$L_2 = L(G_2)$

Let $G_1 = (V_1, \Sigma_1, R_1, S_1)$ &

$G_2 = (V_2, \Sigma_2, R_2, S_2)$

Lets build a new grammar G_1 and

$$L(G) = L = L_1 \cup L_2$$

$G = (\{V_1 \cup V_2 \cup S\}, \{\Sigma_1 \cup \Sigma_2\}, \{R_1 \cup R_2 \cup S \to S_1 \cup S \to S_2\}, \{S\})$

Ex G_1

$G_1 : S_1 \to aS_1 | \varepsilon$

$G_2 : S_2 \to abS_2 | \varepsilon$

$G = (\{S_1, a, 1, S_2, a, b, S\}, \{0, 1, a, b\}$

$\{R_1 \cup R_2 \cup S \to S_1 \cup S \to S_2\}$ $\{S\})$.

2. Closed under concatenation

Proof

L_1 and L_2 are the two CFL

G_1 and G_2 are the corresponding CFG :

Thus $L_1 = L(G_1)$ and $L_2 = L(G_2)$.

Let $G_1 = (V_1, \Sigma_1, R_1, S_1)$ and

$G_2 = (V_2, \Sigma_2, R_2, S_2)$

Lets build a new Grammar G where

$$L(G) = L_1 . L_2$$

$G = (\{V_1 \cup V_2 \cup S\}, \{\Sigma_1 \cup \Sigma_2\}, \{R_1 \cup R_2 \cup S \to S_1 . S_2\}$

$\{S\})$

3. closed under Kleen star operation.

Let L_1 be the CFL.
Let G_1 be the corresponding CFG
and $L_1 = L(G_1)$

$G_1 = (V_1, \Sigma_1, R_1, S_1)$

Lets build a new Grammar G where

$L(G) = L = (L_1)^*$

$G = (\{V_1 \cup S\}, \{\Sigma_1\}, \{R_1 \cup S \to SS_1 | \varepsilon\}, \{S\})$

4. Not closed under complementation intersection

This property is proved by example.

Let L_1 and L_2 be the CFL's

$L_1 = \{a^n b^n c^m | n, m \geq 0\}$

$L_2 = \{a^m b^n c^n | n, m \geq 0\}$

Let us find $L_1 \cap L_2$

$S \to aS | bS | A$
$A \to cA$

L_1: $S \to asb | A | \varepsilon$
$A \to CA | \varepsilon$

L_2: $S \to aAS | A | \varepsilon$
$A \to cAb | \varepsilon$.

According Pumping Theorem of CFL, It is proved that the Language $a^n b^n c^n | n \geq 0$ is not context free

Context free languages not closed under complementation

It is proved by the method of contradiction.

1. Let L_1 and L_2 be the CFL also assume $\bar{L_1}$ and $\bar{L_2}$ are also context free
2. Find the union of $\bar{L_1}$ and $\bar{L_2}$, and it shows that $\bar{L_1} \cup \bar{L_2}$ is a context free
3. Find the complementation of $\bar{L_1} \cup \bar{L_2}$

$$\overline{(\bar{L_1} \cup \bar{L_2})} = L_1 \cap L_2 \text{ as per Demorgans Law}$$

$\therefore L_1 \cap L_2$ is not closed under CFL.

6. Not closed under difference

→ given any language L on Σ, we know that Σ^* is context free.

$\Sigma = \{a, b\}$

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cdots \Sigma^n$$

$(a \cup b)^*$ = $S \to aS$
$S \to bS$
$S \to \varepsilon$.

Let us find $\Sigma^* - L = \neg L$ $\{\bar{L}\}$

$\bar{L}$ → The languages which are not there in L.

$\bar{L}$: Therefore CFL closed under difference are not context free.

Module - 4

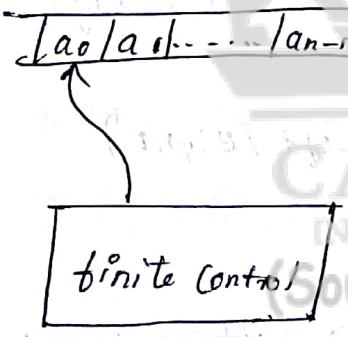
Turing Machine Model.

→ It is an ideal theoritical model for computers of today
→ It is a most general model recognizes all sort of formal languages
→ It accepts type 0 language (Recursively Enumerable language)
→ It determines the undecidability of certain languages
→ used for computing functions
→ measures the space and time complexity of the problems

Church - Turing Thesis

Statement " Any algorithmic procedure that can be carried out by computers can be carried out turing machine also".

Turing Machine Model (TM model)



1. Input tape :- divided into many cells, each cell can hold one symbol

2. Read head :- Read head is not only able to read, it can also write
Read/write head can move in both the direction left-to-right or vice versa.

3. Finite control houses the logic for recognizing the language for computing functions also to measure the time and space complexity of the problem.

Mathematical definition of Turing machine

A Turing machine is formally defined as seven tuple information.

$M = (Q, \Sigma, \Gamma, \delta, q_0, b, F)$

$Q \rightarrow$ It is a finite non empty set of states

$\Sigma \rightarrow$ set of input symbols

$\Sigma \subseteq \Gamma$

$\Gamma \rightarrow$ It is a non-empty set of tape symbols

$q_0 \rightarrow$ start state / Initial state.

$q_0 \in Q$.

$b \rightarrow$ blank space

$b \in \Gamma$

$F \rightarrow$ set of final state

$F \subseteq Q$

$\delta \rightarrow$ transition function $(q_i \times \Gamma) = (q_j \times \Gamma \times \{L, R\}$

$(q_0 X) = (q_1, X, D)$

$D = L/R$ (left/right)

Representation of turing machine

Turing machine can be represented in three ways

1. Transition table
2. Transition diagram
3. Instantaneous descriptions.

1. Design a Turing machine to recognize all even no of 1's on $\Sigma = \{1\}$

Sol. q_0 is the initial state, and the turing machine is enter the state q_1 on scanning '1' and writes 'b'

→ If M is in state q_1 it reads 1 and enters in state q_0 and writes 1 and moves to right

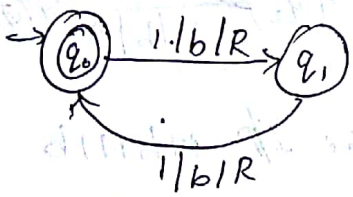
→ q_0 is the accepting state

TM $M = (\{q_0, q_1\}, \{1\}, \{1, b\}, \delta, \{q_0\}, b, \{q_0\})$

Transition table

	q
→ q_0	$q_1\ bR$
q_1	$q_0.bR$

Transition Diagram



write Instantenious description for the string "111"

$(q_0, 111b, R)$

$\vdash (q_1, b11b, R)$

$\vdash (q_0, bb1b, R)$

$\vdash (q_1, bbb1b, R)$

$\vdash (q_0, bbbbb, R)$

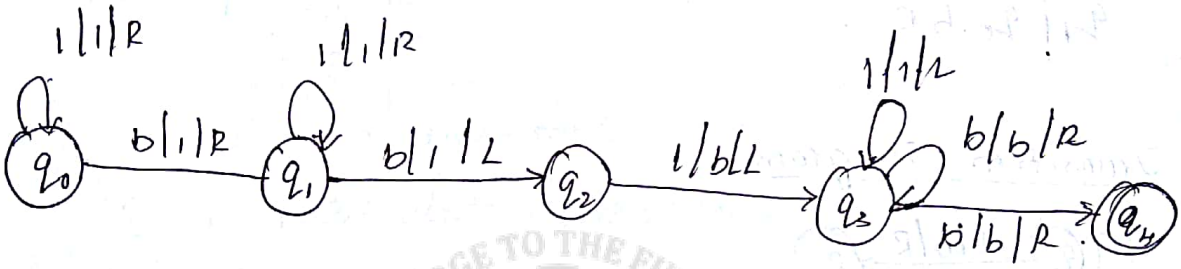
2) Design a turing machine over $\Sigma = \{1, b\}$ which can compute a concatenation function on inputs 1's.

Sol The delimiter b is found and replaced by 1

* Right most '1' is found and replaced by blank
* Read/write head should be return to the starting position

δ	1	b
$\rightarrow q_0$	$q_0, 1, R$	$q_1, 1, R$
q_1	$q_1, 1, R$	q_2, bL
q_2	$q_3\ bL$	—
$*q_3$	$q_3\ 1L$	$q_4\ bR$.

	1	b
$*q_4$	—	—



Write instantaneous description for the I/P 11b111b

$(q_0 11b111b)$
$\vdash (1 q_0 1 1b111b)$
$\vdash (11 q_0 1b111b$
$\vdash (111 q_0 b111b)$
$\vdash (1111 q_1 1 11b)$
$\vdash (11111 q_1 1b)$
$\vdash (111111 q_1 b)$
$\vdash (1111111 q_1 b)$
$\vdash (111111 q_2 1 b)$
$\vdash (11111 q_3 1 bb)$
$\vdash (1111 q_3 11 bb)$
$\vdash (111 q_3 111 bb)$
$\vdash (11 q_3 1111 bb)$
$\vdash (1 q_3 11111 bb)$
$\vdash (q_3 111111 bb)$
$\vdash (b q_4 11111 bb)$

Write instantaneous description for the inputs

1	1	1	b	1	1	1	b	b

$\vdash (q_0 \underline{1}\,11b111bb)$

$\vdash (1\,q_0\,\underline{1}\,11b111bb)$

$\vdash (11\,q_0\,\underline{1}\,1b111bb)$

$\vdash (111\,q_0\,\underline{b}\,111bb)$

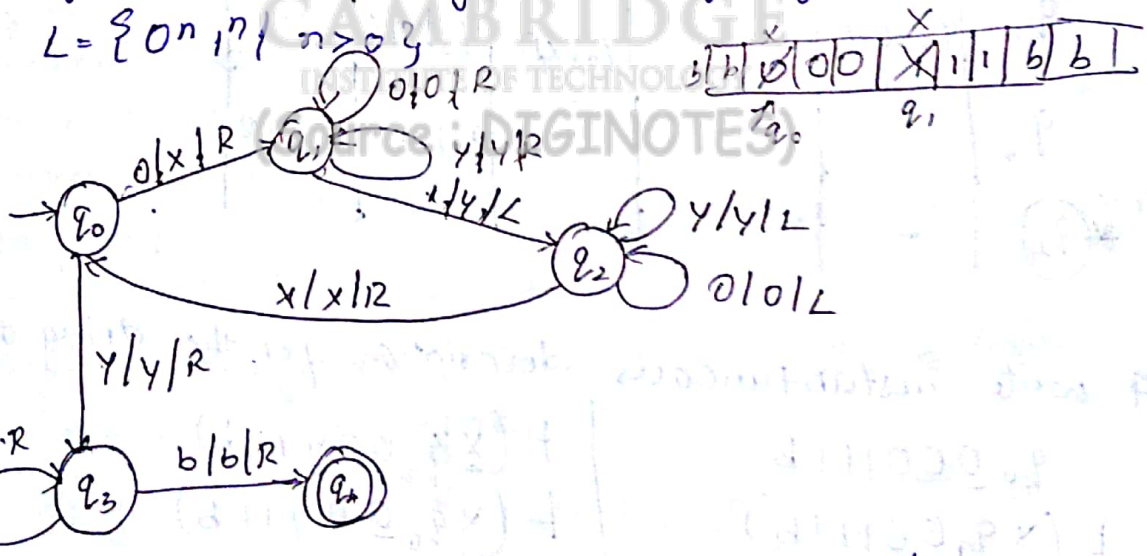
$\vdash (1111\,q_4\,\underline{1}\,11bb)$

$\vdash (11111\,q_1\,\underline{1}\,1b)$

31/10/17

3. design a TM to recognize a Language L =

$L = \{0^n 1^n \mid n \geq 0\}$



write instantaneous description for the string 0011

$(q_0 \underline{0}011\,b)$

$\vdash (xq_1 \underline{0}11b)$

$\vdash (x\,0\,q_1\,\underline{1}\,1\,b)$
$\vdash (x\,\underline{0}\,q_2\,y\,1\,b)$
$\vdash (x\,\underline{x}\,q_2\,0\,y\,1\,b)$
$\vdash (x\,q_0\,\underline{0}\,y\,1\,b)$
$\vdash (x\,x\,q_1\,\underline{y}\,1\,b)$
$\vdash (x\,x\,y\,q_1\,\underline{1}\,b)$
$\vdash (x\,x\,\underline{y}\,q_2\,y\,b)$
$\vdash (x\,\underline{x}\,q_2\,y\,y\,b)$
$\vdash (x\,x\,q_0\,\underline{y}\,y\,b)$

$\vdash (x\,x\,y\,q_3\,y\,b)$
$\vdash (x\,x\,y\,y\,q_3\,\underline{b})$
$\vdash (x\,x\,y\,y\,b\,q_4\,\underline{b})$

String is accepted.

	0	1	x	y	b
$\rightarrow q_0$	q_1,x,R	—	—	q_3,y,R	—
q_1	$q_1,0,R$	q_2,y,L	—	q_1,y,R	—
q_2	$q_2,0,L$	—	q_0,x,R	q_2,y,L	—
q_3	—	—	—	q_3,y,R	q_4,b,R
*(q_4)	—	—	—	—	—

4 write instantaneous description for the string 000111

$q_0\,\underline{0}\,0\,0\,1\,1\,1\,b$
$\vdash (x\,q_1\,\underline{0}\,0\,1\,1\,1\,b)$
$\vdash (x\,0\,q_1\,\underline{0}\,1\,1\,1\,b)$
$\vdash (x\,0\,0\,q_1\,\underline{1}\,1\,1\,b)$
$\vdash (x\,0\,\underline{0}\,q_2\,y\,1\,1\,b)$
$\vdash (x\,\underline{0}\,q_2\,0\,y\,1\,1\,b)$

$\vdash (\underline{x}\,q_2\,0\,0\,y\,1\,1\,b)$
$\vdash (x\,q_0\,\underline{0}\,0\,y\,1\,1\,b)$
$\vdash (x\,x\,q_1\,\underline{0}\,y\,1\,1\,b)$
$\vdash (x\,x\,0\,q_1\,\underline{y}\,1\,1\,b)$
$\vdash (x\,x\,0\,y\,q_1\,\underline{1}\,1\,b)$
$\vdash (x\,x\,0\,y\,q_2\,\underline{y}\,1\,b)$

$\vdash (xx\underline{0}q_2yy1b)$
$\vdash (x\underline{x}q_20yy1b)$
$\vdash (x\underline{x}q_00yy1b)$
$\vdash (xxxq_1\underline{y}y1b)$
$\vdash (xxxyq_1\underline{y}1b)$
$\vdash (xxxyyq_1\underline{1}b)$
$\vdash (xxxyy\underline{y}q_2yb)$
$\vdash (xxx\underline{y}yq_2yyb)$
$\vdash (xx\underline{x}q_2yyyb)$
$\vdash (xxxq_0yyyb)$

$\vdash (xxxyq_3yyb)$
$\vdash (xxxyyq_3yb)$
$\vdash (xxxyyyq_3b)$
$\vdash (xxxyyyyq_4b)$

22/10/17

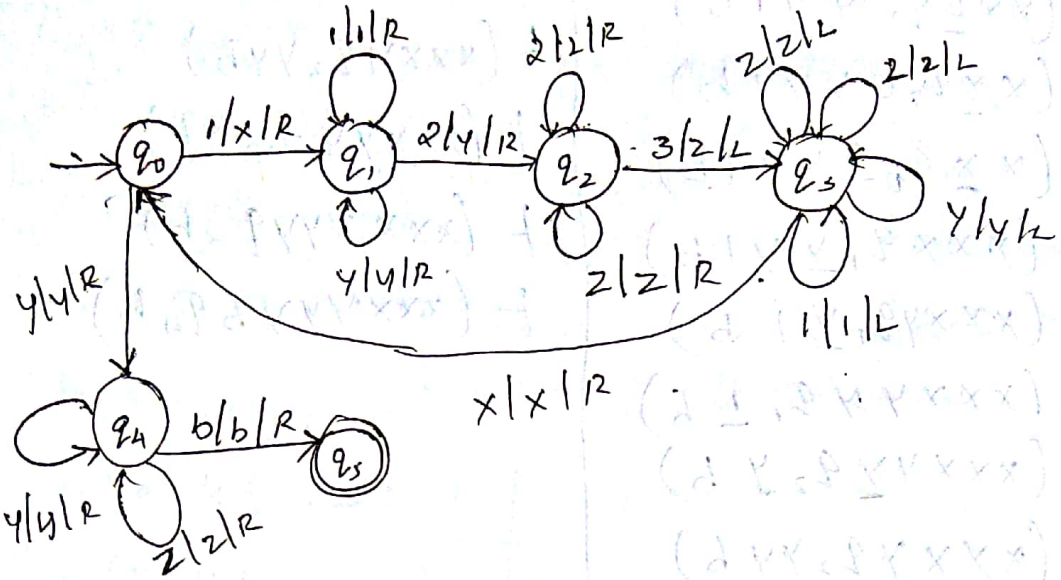
5. Design a TM.

$L = \{ 1^n 2^n 3^n \mid n>0 \}$

b	b	1	1	1	2	2	2	3	3	3	b	b

→ first it reads 1 and replaces it with 'x' and moves Right.
→ once it Reads 2 it replaces it with 'y' and moves right
→ once it Reads '3' it replaces it with 'z' and changes direction to Left.
→ while moving left it has to pass ~~b~~ from 2 → y → 1 and reach x.
→ after reaching x, it changes its direction to Right by replacing 1 with x and process continues.

x	1	1	y	2	2	z	3	3



	1	2	3	x	y	z	b
$\rightarrow q_0$	q_1, x_1, R				q_4, y, R		
q_1	$q_1, 1, R$	$q_2 \mid y \mid R$			$q_1 \mid y \mid R$		
q_2		$q_2, 2, R$	q_3, z, L			q_2, z, R	
q_3	$q_3, 1, L$	$q_2, 2, L$		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, R
(q_5)	—	—	—	—	—	—	—

Techniques for Turing machine construction

Subroutines

It can be implemented using turing machines, one of the popular example is multiplication operation.

→ To carry out $m * n$ lets use unary representation to represent m and n in the i/p tape.

→ i/p : $0^m 1 0^n 1$

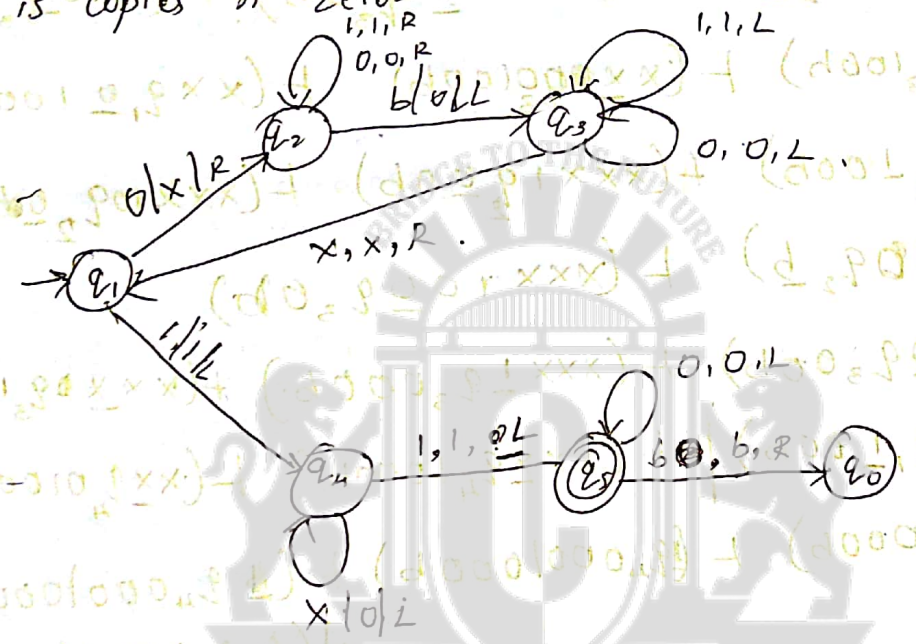
o/p 0^{mn}

→ m and n are represented in unary zero, m is to be copied m no of times.

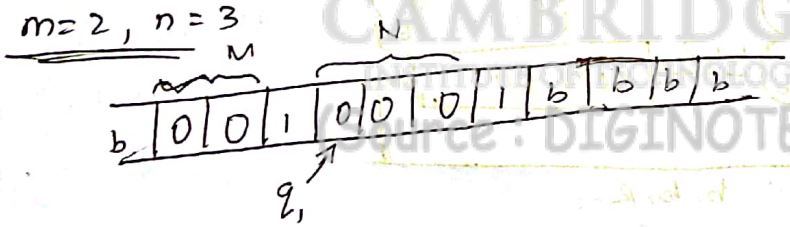
→ Then the i/p as well as the delimeter will be replaced by blank [The pattern $0^m 1 0^n$ will be replaced by blank]

Copy Subroutine

It is copies n zeros 1 time



m=2, n=3



N-2+1=5

$(q_1 0\,0\,0\,1\,b)$

$\vdash (x\,q_2\,0\,0\,1\,b)$

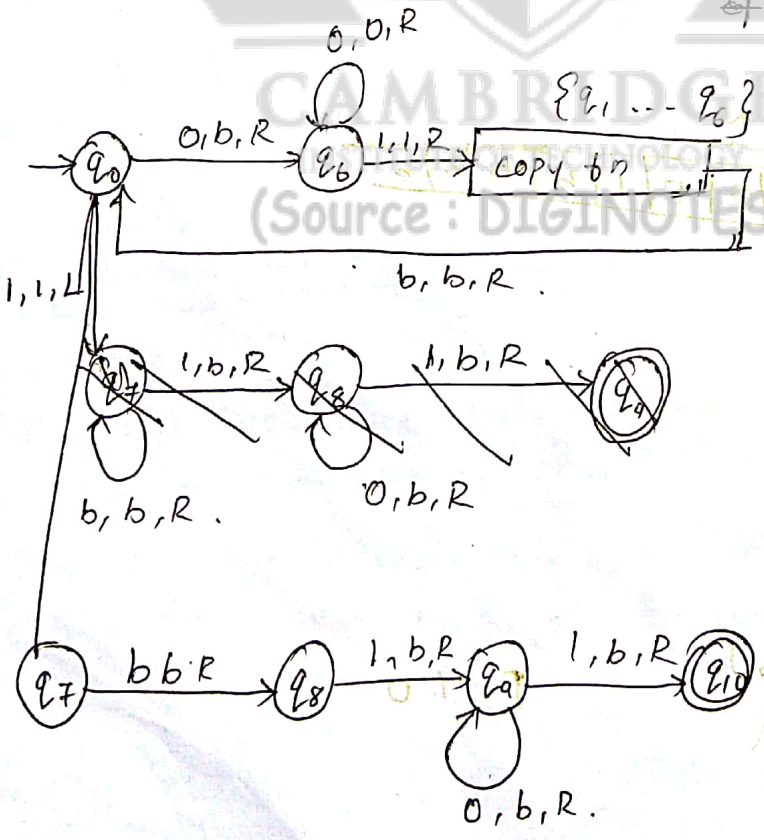
$\vdash (x\,x\,q_2\,0\,1\,b)$

$\vdash (x\,x\,x\,q_2\,1\,b)$

$\vdash$ (

P.T.O

$(q_1 \overbrace{0000}^{n} 1b) \vdash (xq_2 000 1b) \vdash (x0q_2 01b) \vdash (x00q_2 1b)$
~~$\vdash (x00 1 q_3 0b)$~~ $\vdash$
$\vdash (x001q_1 b) \vdash (x001q_3 0bb) \vdash (x00q_3 10bb)$
$\vdash (x0q_3 010bb) \vdash (xq_3 0010bb) \vdash (xq_1 0010bb)$
$\vdash (x0q_2 010bb) \vdash (x00q_2 10bb) \vdash (x00 1 q_2 0b)$
$\vdash (x00 1 0 q_2 b) \vdash (x00 1 q_3 0b) \vdash (x00 1 q_3 0bb)$
$\vdash (x00q_3 100bb) \vdash (x00q_3 100bb) \vdash (x0q_1 0100bb)$
$\vdash (x00q_2 100bb) \vdash (x00 1 q_2 00b) \vdash (x00 1 0 q_2 0b)$
$\vdash (x00 1 00 q_2 b) \vdash (x00 1 0 0 q_3 0b)$
$\vdash (x00 1 0 q_3 00b) \vdash (x00 1 q_3 000b) \vdash (x00q_3 1000b)$
$\vdash (x0q_1 0 1000b) \vdash (x00q_4 1000b) \vdash (x0q_4 01000b)$
$\vdash (xq_4 001000b) \vdash (q_4 0001000b) \vdash (bq_4 0001000b)$
$\vdash (bq_5 1000 1000b)$



Instantaneous description

001 000 | bbb
m n.

$\vdash (q_0 \underline{0}0010001bbb)$

$\vdash (bq_0\underline{0}10001bbb)$

$\vdash (bbq_0\underline{0}0q_0\underline{1}0001bbb)$

$\vdash (bb1q_1\underline{0}001bbb)$

$q_1 - q_5$ is executed previously

$\vdash (bb0q_5\underline{1}0001000b)$

$\vdash (bbq_5\underline{0}1000100b)$

$\vdash (bbq_0\underline{0}1000100b)$

$\vdash (bbq_61\underline{0}001000b)$

$\vdash (bbq_1\underline{0}001000\ldots)$

copy function.

$\vdash (bb\underline{1}q_50001000000)$

$\vdash (bb\underline{b}q_51000100000)$

$\vdash (bbbq_0\underline{1}0001000000)$

$\vdash (bb\underline{b}q_7\underline{1}0001000000)$

$\vdash (bbbq_8\underline{1}000|000000)$

$\vdash (bbbq_9\underline{0}00|000000)$

$\vdash (bbbbq_9\underline{0}01.000000)$

$\vdash (bbbbbq_9\underline{0}|000000)$

$\vdash (bbbbbbq_91000000) \rightarrow \vdash (bbbbbbbq_{10}000000)$

2. Turing machine with stationary head

Let us include the option that the Read/write head can continue to rest in the same cell for some i/p, its transition function is given as follows.

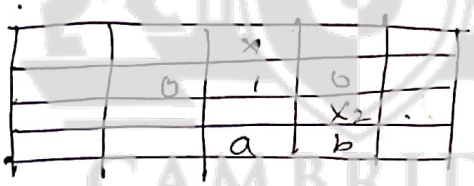
$$(q_i, a) \rightarrow (q_j, Y, \{L, R, S\})$$

S → stay option.

3. Multiple track turing machine

In multiple track turing machine a single tape is divided into several tracks. for example K – number of tracks. The tape alphabet is required to consist of K-number of tape symbols.

→ The major difference b/w the standard turing machine and the multitrack TM is that the tape symbols which is Γ and Γ^K



$$(q_0, b) \rightarrow (q_1, X, Y, L, R)$$

(4) Storage in the state :-

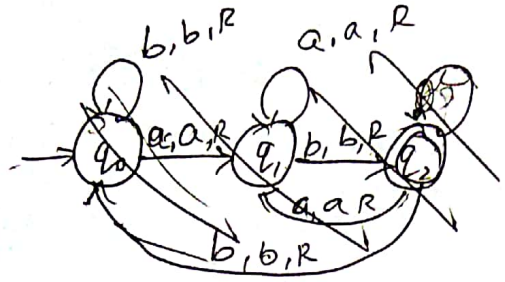
In this technique T.M's can be constructed to store or remember a symbol along with the state as well so the state becomes a pair of information (q, a) where $q \rightarrow$ current state of TM

$a \rightarrow$ i/p symbol read previously.

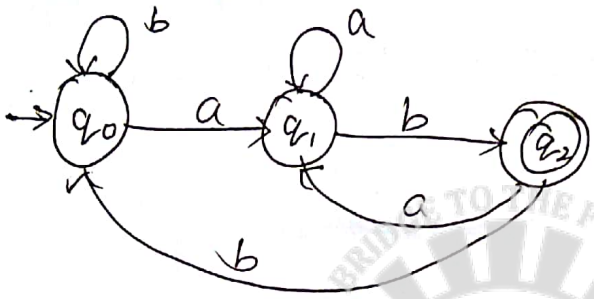
$$((q, a), b) = q_2, X, R.$$

Previously read — Currently reading

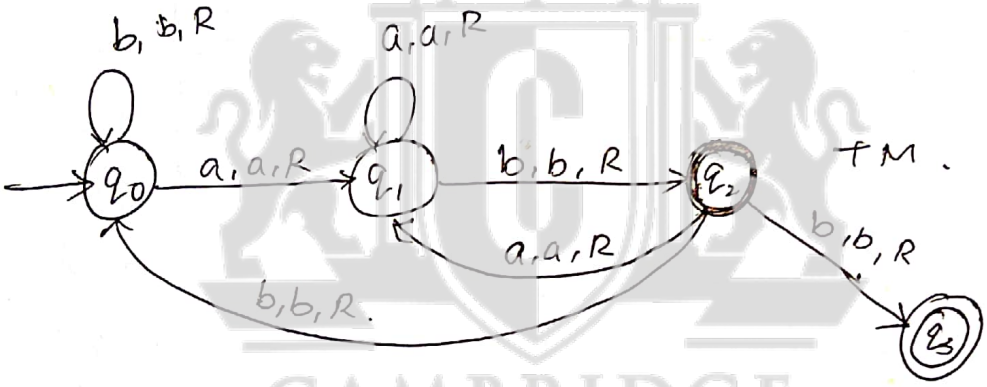
Q] Design TM to accept the language that contains strings of a's and b's that end with ab.



DFSM.



DFSM



TM.

For Regular Languages, we can desing T.M like this way.

$0^n 1^r \mid n > 0$

7/11/17

Pumping lemma for CFL

→ Proof by contradiction

if L were context free, then it would posses certain properties, but it does not posse those properties, it is not context free.

→ To define the properties of context free language we exploit the following facts.

1. context free grammar.
2. The structure of parse tree.
3. Height of the tree [length of the longest path from root to leaf]
4. branching factor: [b] it is nothing but longest no of children of any node in the tree.
   Or it is the length of longest right hand of any Rule.
5. yield of tree :- The string from left to right

Let G be context free grammar,
Let $|N| = |V - \Sigma|$ (no of non terminals)
Let b be the branching factor.

The longest string that corresponds to the parse tree T has always a length greater than or equal to b^n

This length could be acheived from the Grammar which posses atleast one recursive variable.

$S \Rightarrow uXz \Rightarrow uvxyz$

$\Rightarrow uvvxyyz$

based on the observations made on the parse tree, if L is a context free language, then there exists a

$$\exists k \geq 1 \left(\forall \text{string } w \in L, \text{ where } |w| \geq k \left(\exists u,v,x,y,z \left(w = uvxyz,\ vy \neq \varepsilon \right.\right.\right.$$

$$\left.\left.\left. |vxy| \leq k \text{ and } \forall q \geq 0 \left(uv^q xy^q z \text{ is in } L\right)\right)\right)$$

Problems

① Show that language $L = \{a^n b^n c^n : n > 0\}$ is not context free

Sol:

Consider the language is regular.

Let $n = 3$.

$\therefore w = \underset{u}{a}\,\underset{v}{a}|\underset{x}{a}\,\underset{y}{b}|\underset{z}{b}\,b|c\,c|c$.

Case i Let $i = 2$. $uv^i xy^i z$.

uv^2xy^2z.

$= aa|abab|bb|cccc|c \notin L$

$\therefore$ it is not context free.

Case ii

$\varepsilon\ a\ a|a\ b|b|b\ c\ c\ c$

$v\ \ x\ y\ z$.

$q = 2$.

uv^2xy^2z

$aaa\,a\,abbb\,b\,ccc \notin L$.

Case 3

$a|aa|bb|bc|cc \in L$

$u \quad v \quad x \quad y \quad z$

$i = 2$

uv^2xy^2z

$= aaaaabbbcbccc \notin L$

$\therefore a^n b^n c^n$ is not context free.

10/1/17

Recursive	Recursive Enumerable
Decidability	Undecidability
A_{CFG}	A_{TM}
A_{DFA}	Halting problem PCP

Decidability

→ A Language is decidable if it is Recursive.

→ A Language is Undecidable if it is Recursively enumerable.

A_{DFA} is decidable

For this theorem

$\exists_{TM}$ A which simulates the operation of DFA.

if DFA accepts TM also accepts, IF DFA rejects DFA also rejects.

A_{CFG} is decidable

$\exists_{TM}$ A which simulates the operations of parser (PDA)

→ To prove this every context free Grammar (CFG) is converted into its CNF (comsky normal form)

→ Every string w from CNF can be derived in $(2k-1)$ steps, where k is the length of the right hand side of the production rule.

→ Since it is definite to derive w which belongs to CFG, then designing a TM simulating is also possible

There exists Languages over Σ which are not recursively Enumerable

→ We know that Σ is finite set.

→ we say that Σ^* is also finite based on the property ie one to one correspondence b/w Σ^* and N

→ $\mathcal{L}$ is the set of Languages over Σ^*

→ There is a TM M which is defined as a 7 touple information.

→ since every member of touple is finite,

I: the set of Turing machine is also finite.

→ assume that $\mathcal{L}$ is finite (countable)

so each member of $\mathcal{L}$ is defined as a binary sequence. $L_i = x_{i1}\, x_{i2}\, x_{i3}\, x_{i4} \ldots$

$$x_{ij} = \begin{cases} 1 & \text{if } w \in L_i \\ 0 & \text{otherwise} \end{cases}$$

Let us find $L_k = y_{k1}\, y_{k2}\, y_{k3} \ldots$

$$\boxed{y_{kj} = 1 - x_{ij}}$$

$$\therefore \boxed{L_i \neq L_k}$$

→ It means if $w \in L_i$, iff $w \notin L_k$

after finding L_k, it is proved that L_k is infinite.

if L_k is infinite obviously L_i is also infinite.

Conclusion : I : The set of turing machines is finite

This shows that there are some Languages exits which cannot be recognized by TM.

① Post Correspondance problem [PCP]

→ PCP is an undecidable problem.
→ This problem was founded by Emiley post
→ There are 2 list given, x and y.

$x = (b, bab^3, ba)$ $\quad y = (b^3, ba, a)$

Let us find a string from x that should be equivalent to the string of y.

① for example $\quad bab^3 \; b \; b \; ba \rightarrow x$

$= bab^3 \; b^3 \; a \rightarrow y$

It is same string.

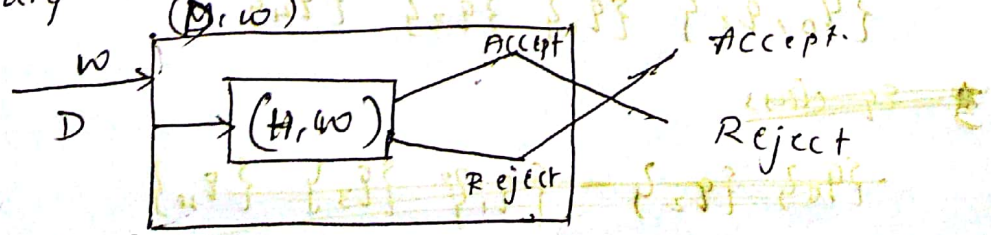
② $x = ((10)^2, 1, 0)$ $\quad y = (0, 1, 0)$

$x \rightarrow 1010$
$y \rightarrow 0101$ $\neq$

The problem over the set of alphabets is supposed to belong to a class of yes or no.

② A_{TM} is undecidable :-

We've a machine M. which reads input 'w' and say whether it is accepted. , we know that A_{TM} recognizes Recursively enumerable languages. In contrary to that we assume A_{TM} is decidable.



Assignment -3

1. Multitape Turing Machine

A multitape Turing machine is like an ordinary TM with several tapes. Each tape has its own head for reading and writing. Initially the i/p appears on tape1 and other start out blank.

No matter how many tapes can be simulated by a single-tape machine using only quadratically more computation time

The multitape machines cannot calculate any more functions than a single-tape machines and none of the robust complexity classes are affected by a change blw single-tape and multitape machines.

A k-tape TM can be described as 6-tuple

$M = (Q, \Gamma, q, b, F, \delta)$ where

Q is a finite set of states.

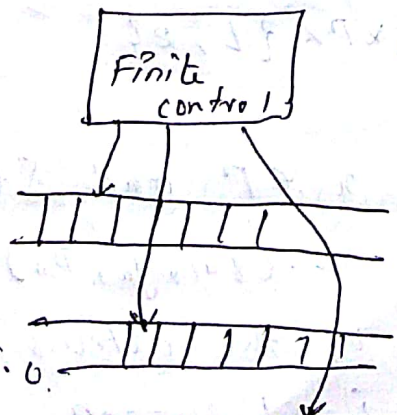
Γ is a finite set of tape alphabet.

$q_0 \in Q$ is the initial state.

$b \in \Gamma$ is a blank symbol

$F \subseteq Q$ is the set of final states.

$\delta : Q \times \Gamma^k \to Q \times (\Gamma \times \{L, R, S\})^k$ is a transition function.



In a typical move.

i) M enters a new state

ii) one each tape, a new symbol is written in the cell under the head

iii) each tape head moves to the left or right or remain stationary. The heads move independently

The initial ID has the initial state q_0, the i/p string w in the first tape, empty string of b's in the remaining $k-1$ tapes.

An accepting ID has a final state.

Q. Non-deterministic Turing Machine

A non-deterministic turing machine is a 7 tuple information $(Q, \Sigma, \Gamma, \delta, q_0, b, F)$ where

Q – Final set of states

$\Gamma \rightarrow$ finite non-empty set of tape symbols.

$b \rightarrow \in \Gamma \rightarrow$ blank symbol.

$\Sigma \rightarrow$ non-empty subset of Γ, called the set of input symbol - we assume $b \notin \Sigma$

$q_0 \rightarrow$ initial state

$F \subseteq Q \rightarrow$ set of final states

δ is partial function $Q \times \Gamma$ into the power set of $Q \times \Gamma \times \{L, R\}$

If $q \in Q$ and $x \in \Gamma$ and $\delta(q, x) = \{(q_1, y_1, D_1), (q_2, y_2, D_2) \ldots\ldots (q_n, y_n, D_n)\}$ then NTM can choose any one of the actions defined by (q_0, y_i, D_i) $i = 1, 2, 3 \ldots\ldots n$

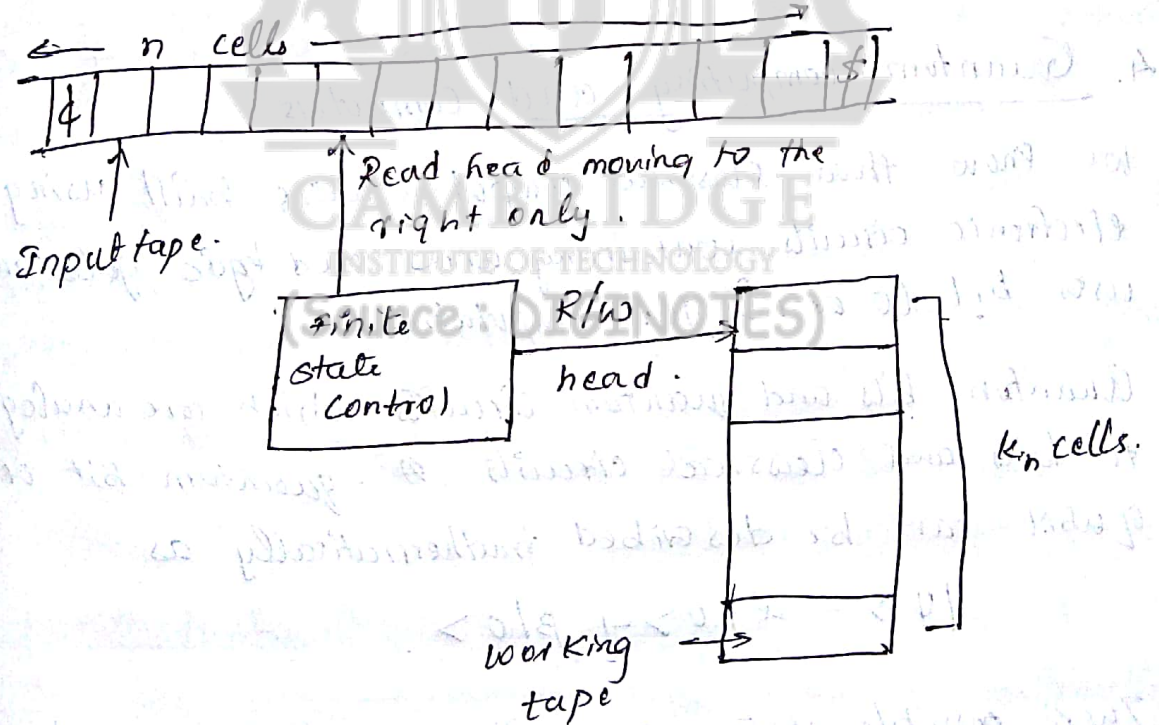
So on reading the i/p symbol, the NTM M whose current ID is $z_1, z_2 \ldots q \times z_{k+1} \ldots z_n$ can change to any one of the three ID's

3. Model of LBA

This model is important because

1. set of context-sensitive languages is accepted by the model.
2. The infinite storage is restricted in size but no in accessibility to the storage in comparision with TM. It is called the linear bounded automation

A linear bounded automation is a non deterministic TM which has a single tape whose length is not infinite but bounded by a linear function of the input length of the input string.



The model can be described formally by the following set format

$$M = (Q, \Sigma, \Gamma, \delta, q_0, b, ¢, \$, F)$$

¢ → Left end marker
$ → Right end marker

In case of LBA, an ID is denoted by (q, w, k) where $q \in Q$, $w \in \Gamma$ and k is some integer blw 1 and n. Thus transition of ID's is similar except that k changes to $k-1$ if the R/w head moves to the left and to $k+1$ if the head moves to the right.

The language accepted by LBA is defined as the set

$$\{w \in (\Sigma - \{¢, \$\})^* \mid (q_0, ¢w\$, 1) \vdash^* (q, \alpha, \beta)$$

for some $q \in F$ and for some integer i blw 1 and n $\}$

4. Quantum computing and computers

We know that classical computers are built using electronic circuits containing wires and logic gates and uses bit (0 or 1) for computation.

Quantum bits and quantum circuits which are analogous to bits and classical circuits. A quantum bit or qubit can be described mathematically as

$$|\psi\rangle = \alpha|0\rangle + \beta|0\rangle.$$

Two possible states of qubit are $|0\rangle$ and $|1\rangle$ unlike a classical bit, a qubit can be infinite numbers of states other than $|0\rangle$ and $|1\rangle$. It can be in a state $|\psi\rangle = \alpha|0\rangle + \beta|0\rangle$

where α and β are complex numbers such that

$$|\alpha|^2 + |\beta|^2 = 1$$

In classical case, we can observe it as 0 or a 1, But it is not possible to determine the quantum state on operation. When we measure a qubit, we get either the stat $|0\rangle$. with probability $|\alpha|^2$ or the state $|1\rangle$ with probability $|\beta|^2$

This is difficult to visualize using our classical thinking, but the source of the source of power of the quantum computation

Now we define the qubit gates. The classical NOT gate interchanges 0 and 1. In the case of the qubit. the NOT gate ; $\alpha|0\rangle + \beta|1\rangle$ is changed to $\alpha|1\rangle + \beta|0\rangle$.

The qubit gate corresponding to NOR is the controlled NOT or CNOT gate. it can be describes as

$$|A, B\rangle \rightarrow |B, B \oplus A\rangle$$

A ~~regul~~ quantum computer is a built from quantum circuits containing wires and elementary quantum gates, to carry out manipulation of quantum information

5. <u>church Turing Thesis</u>

since 1970's many techniques for controlling the single quantum systems have been developed but with only modest success.

church Turing thesis says that "any algorithm ~~that~~ can be performed on a Turing machine as well"

An algorithm requiring polynomial time was considered as an efficient algorithm, a strengthened version of the church-Turing thesis was introduced.

Any algorithm process can be simulated efficiently by a Turing Machine. But a challange to the strong church turing thesis ade no from analog computation. certain types of analog computer solved some problems efficiently whereas these problems had no efficient solution on a Turing machine.

But when the presence of noise was taken into account, the power of the analog computer disappeared.

which led to the modification of the church thesis

"Any algorithm process can be simulated efficiently using a non-deterministic TM"

6. <u>classes of P and NP problems</u>

A Turing machine M is said to be of time complexity T(n) if following rules

Given an input w of length n, M halts after moving at most T(n) moves.

In this case M eventually halts. Recall that the standard TM is called as a deterministic TM

A Language L is in class P if there exists some polynomial T(n) such that L=T(M) for some deterministic TM M of time complexity T(n)

A language L is in class NP if there is a non-deterministic TM and a polynomial time complexity T(n) such that L = T(M) and M executes at most T(n) moves for every input w of length n.

7. Post Correspondance problem

The problem has many application in the theory of formal languages. The problem over an alphabet Σ belongs to a class of yes/no problems and is stated as follows.

Consider two lists $x = (x_1, \ldots x_n)$ and

$y = (y_1, \ldots y_n)$ of

non-empty string over alphabet $\Sigma = \{0, 1\}$

The PCP is to determine whether or not there exists $i_1, \ldots i_m$ where $1 \le i_j \le n$ such that

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

The indexes i_j's need to be distinct and m may be greater than n. Also if there exists a solution to PCP, there exists infinitely many solutions.

Example : $x = (b, bab^3, ba)$ and

$y = (b^3, ab, a)$

$x \rightarrow bab^3\ bb\ ba$

$y \rightarrow bab^3\ b^3a$.

Thus the PCP has a solution.