Automata Theory - Assignment - 1

NAGA MANOHAR
2021101128

SECTION - 2 :-

① Given,

L is a CFL over one-symbol alphabet
Say (a) $\Sigma = \{a\}$

RTP :- L is regular.

As L is context free it follows pumping lemma

Hence

let p be the pumping length for L.

Now let $L = L_1 \cup L_2$ s.t.

$L_1 = \{ w : w \in L \text{ and } |w| \leq p \}$

$L_2 = \{ w : w \in L \text{ and } |w| \geq p \}$

Now L_1 is finite $\Rightarrow$ L_1 is regular

we prove that L_2 is regular.

in L_2 as $|w| \geq p$ by pumping lemma

$w = uv^t xy^t z \in L_2 \quad \forall t \geq 0$

as there is only one symbol (a)

let $u = a^u$, $v = a^v$, $x = a^x$, $y = a^y$ and

$z = a^z$

① contd

Now

$$w = a^{u} \cdot a^{vt} \cdot a^{x} \cdot a^{yt} \cdot a^{z} = a^{uxz} \cdot a^{(v+y)t} \in L_2$$

Now

consider a language

$$L_w = \{ w \mid w = a^{uxz} \cdot a^{(v+y)t} \ \forall t \geq 1 \}$$

Now $\boxed{\bigcup_w L_w = L_2}$ where

$$\bigcup_w L_w = L_{w_0} \cup L_{w_1} \cup \cdots$$

$$= \{ L_w(t=1) \cup L_w(t=2) \cup \cdots \}$$

Now

L_w is regular

$\Rightarrow$ L_2 is regular ($\because$ union of regular languages is regular)

$\Rightarrow$ $L_1 \cup L_2$ is regular

$\Rightarrow$ L is regular

Hence, if L is a CFL over one-symbol alphabet then L is regular

(2)

Given, an Infinite regular language A.

Find : a way to split A into 2 Infinite regular languages A_1, A_2

s.t. $A_1 \cup A_2 = A$ and $A_1 \cap A_2 = \emptyset$

W.K.T

Every regular language is accepted by a DFA

DFA $\equiv$ Regular Language

Now

For Language(A) – let DFA(M_A) ~~accepts~~ recognizes A $\Rightarrow$ if $w \in A$ $M_A(w)$ ACCEPTS

if $w \notin A$ $M_A(w)$ REJECTS

~~Now~~ let DFA M_A have a state q

as A is infinite q can be visited more than once.

If q can be visited more than once then it can be visited any number of times

Now define 2 languages A_1, A_2

$A_1 = \{ w \in A : q$ is visited odd number of times$\}$

$A_2 = \{ w \in A : q$ is visited even number of times$\}$

Any string that is accepted by A is accepted by one of A_1 or A_2.

Now state q can be visited 0 times, 1 times 2 times, 3 times ... unlimited number of times...

and from this

① A_1 accepts strings when q is visited 1 times, 3 times, 5 times, ... unlimited times

⇒ A_1 is infinite regular language (A_1 accepts unlimited no. of string)

② A_2 accepts strings when q is visited 0 times, 2 times, 4 times, ... unlimited times

⇒ A_2 accepts unlimited no. of strings

⇒ A_2 is infinite regular language

Now any given string is either in A_1 (or) in A_2 but not in both

⇒ $\boxed{A_1 \cap A_2 = \emptyset}$

and any string $\omega \in L$ must belong to one of A_1 or A_2

⇒ $\boxed{A_1 \cup A_2 = A}$

Hence, we can split A into A_1 & A_2

(3) Given

$s = uvxyz$

RTP: stronger form of pumping lemma when $|vy| > 0$ is replaced by $|v| > 0$ and $|y| > 0$

A is CFL

(i) $\forall i \geq 0$, $uv^ixy^iz \in A$

(ii) $|v| > 0$ and $|y| > 0$

(iii) $|vxy| \leq p$

we have to prove that

if A is CFL,

then there is a pumping length (p) s/t

$s \in A$ and $|s| \geq p$ then $s = uvxyz$

and it follows (i), (ii) & (iii)

let G be a CFG for CFL A.

Consider a parse tree T_w^G of G that yields string w

let Total No. of variables in $G = |V|$

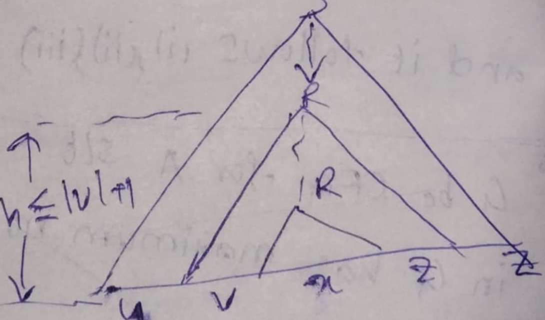
if $w \in L$ s/t $|w| = p = d^{|V|+1}$, then the parse tree corresponding to it will have height $\geq |V|+1$

- The longest path from the start variable S to a terminal is $\geq |V|+1$

Now consider the lowest $|V|+1$ variables in that path

here, atleast one state is repeated (by pigeon hole principle)

Then any string w s/t $|w| \geq p$ can be split as $w = uvxyz$ s/t $|vxy| \leq p$



- from the figure

the upper most R falls within the bottom |V|+1 variables in the longest path

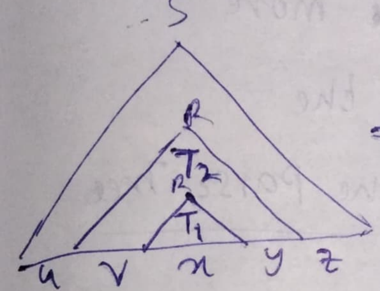
$\Rightarrow$ length of string generated $\Rightarrow l \le d^{|V|+1} = p$

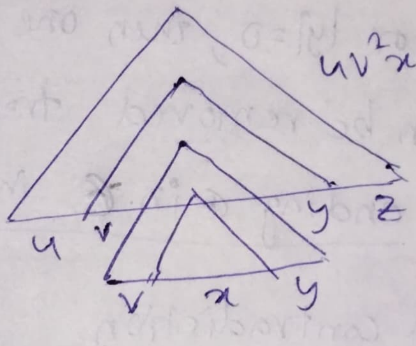
$\Rightarrow l \le p$

$\Rightarrow \boxed{|vxy| \le p}$ (iii)

Now we prove $\boxed{uv^ixy^iz \in L \quad \forall i \ge 0}$ (i)

① $\forall i > 0$ - Replace the subtree rooted at T_1 with the subtree rooted at T_2



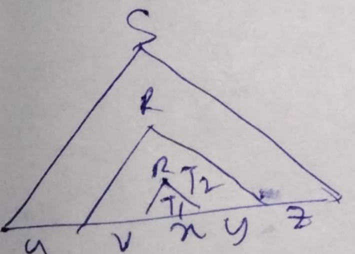


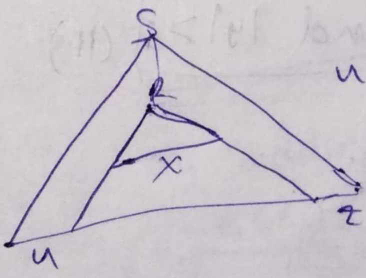
$uv^2xy^2z \in L$

$(i = 2)$

and we can do this $\forall i > 0$

② $i = 0$ - replace the subtree rooted at T_2 with the subtree rooted at T_1





$uv^0xy^0z \in L$

But if G is ambigous $\Rightarrow$ More than 1 parse Tree exists

$\Rightarrow$ consider the parse tree with the smallest number of nodes

So $\boxed{T_w^G \text{ is the smallest parse Tree generating } w}$

Now we prove that

(ii) $|v| > 0$ and $|y| > 0$

If $|v| = 0$ or $|y| = 0$, then one or more vertices can be removed from the Sub-Trees ending in ϵ in the parse Tree

- This is a contradiction as it is not possible to remove vertices from the tree T_w^G ($\because T_w^G$ is smallest)

$\Rightarrow$ $\boxed{|v| > 0 \text{ and } |y| > 0}$ (II)

Hence, proved.

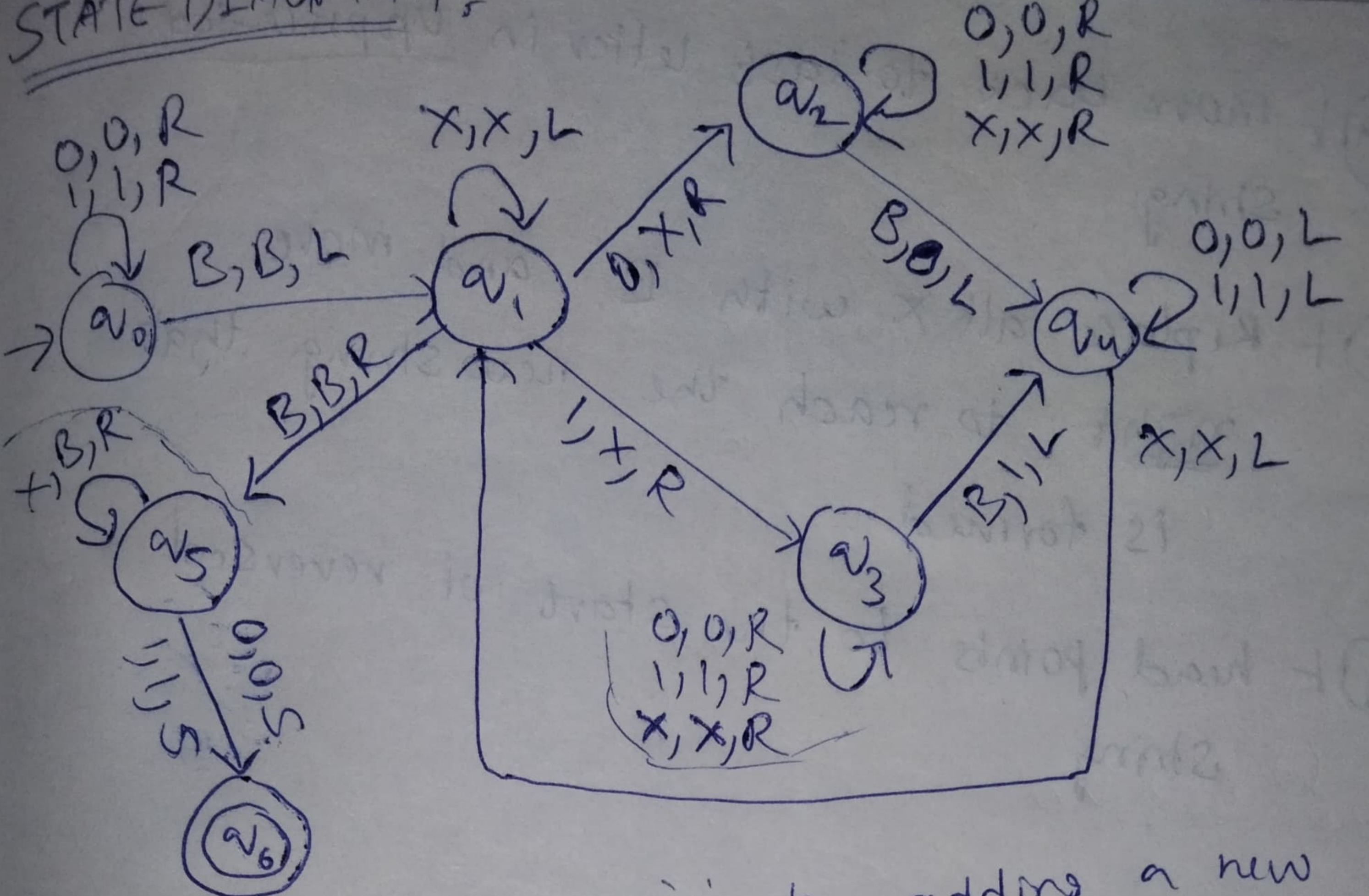
(4) Given to construct a turing machine s/t

given a string $w \in \{0,1\}^n$, $n \in N$ outputs the reverse string

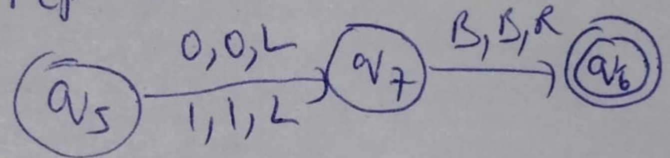
Here we use a Lazy Turing Machine (L, R, S)

Stay put

STATE DIAGRAM:



we can always replace S by adding a new state q_7 as



<u>Idea:</u> we want to copy the string after reversing to the right of it. then delete original string

Ex: | B | a | b | B | B | B | => | B | a | b | b | a | B | => | B | B | B | b | a | B |

head, copy here, remove, head

STATE Description

q_0 : used to reach the end of the string – transitions to q_1 on reaching B and head points to last character of string

q_1 : Skip all x and point to the (Left) letter just left of leftmost x

q_2 : skip all x and already reversed part (Right) then add the new symbol to the reversed string – for '0'

q_3 : same as q_2 but for '1'

q_4 : move back to last letter in unprocessed string.

q_5 : Replace all x with B and move right to reach the new string that is formed

q_6 : head points to the start of reversed string

(5) Given a language

$L = \{a^n b^n c^n \mid n \in N\}$

(1) No, L is not context free

proof: Assume that L is context free

=> By pumping lemma let $\underline{p}$ be the pumping length. let $w = a^p b^p c^p \in L$ ($|w| = 3p (\geq p)$)

=> $w = uvxyz$ s/t $|vxy| \leq p$ and $|vy| \geq 1$

$\forall t \geq 0$ $uv^t xy^t z \in L$

First observe that in $w = a^p b^p c^p$ is separated the last $\underline{a}$ and the first $\underline{c}$ is separated by p 1's as $|vxy| \leq p$

the substring $|vxy|$ has atmost 2 distinct symbols. Now

consider the string $w' = uv^2 xy^2 z$

~~$|w| = 3p (\geq p)$~~

case 1: $\underline{vxy = a^k \text{ or } b^k \text{ or } c^k}$ $k \leq p$

then $w' \in L$

Case if $\underline{vxy = a^k}$ then, w' will have more a's then b's and c's.

~~NOW~~

Case 2:

$vxy = a^m b^n$ or $b^m c^n$, $m+n \le p$

Again, $w' \in L$

Ex: If $vxy = a^m b^n$, then w' will have less c's then a's and b's.

By case 1, case 2 we arrive at contradiction

hence

$L \notin CFL$

2) Yes, Recursive

consider the languages

$L_1 = \{ a^i b^i c^j \mid i, j \in N \}$

$L_2 = \{ a^i b^j c^j \mid i, j \in N \}$

$L_1 \rightarrow$ It can be built be ~~contac~~ concatenation of 2 CFL's and L_1 has CFG as follows:

$S \rightarrow S_1 S_2$

$S_1 \rightarrow a S_1 b \mid ab$

$S_2 \rightarrow c S_2 \mid c$

is CFL $\Rightarrow$ L_1 is recursive language

($\because$ CFL $\subseteq$ Recursive languages) in L_2 : the CFG for L_2 is :

$S \to S_1 S_2$
$S_1 \to aS_1 \mid a$
$S_2 \to bS_2c \mid bc$

as it has a CFG L_2 is CFL $\Rightarrow$ L_2 is recursive

$\therefore L_1, L_2$ are recursive languages

Now consider

$L = L_1 \cap L_2 = \{a^i b^i c^i \mid i \in N\}$

$L_1 \cap L_2$ is Recursive ($\because$ Recursive languages are closed under intersection)

Hence
L is Recursive language

Now Recursive Languages $\subseteq$ Recursive Enumerable Languages

(3) L is Recursively enumerable

6

1. Recursively Enumerable but NOT Recursive:

The set of halting turing machines

is ~~recu~~ an example

2. ~~A Language~~ There does NOT exist a language that is recursive but not recursively enumerable.

as

Recursive Languages $\subseteq$ Recursively Enumerable Languages (by defn)

~~R~~ that is Given a recursive language we can always build a recursively enumerable language s.t

for some Language(L) $\begin{cases} w \in L & M(w) \text{ Accepts} \\ w \notin L & M(w) \text{ rejects. } \cancel{\text{or loops}} \end{cases}$ $\boxed{\text{RE}}$

just that it won't loop, and it is not mandatory when $w \in L \rightarrow$ ~~Accepts~~ rejects

Here L is both Recursive and recursively enumerable

(4) Given $\Sigma = \{0,1\}$

RTP:

(1) Set of all strings is countably infinite

(2) Set of all Languages in uncountably infinite

(1) we prove this by building a bijection with $\mathbb{N}$ = Natural numbers set

let

X be the set of all strings

s/t (1) They are sorted according to length of string first and

for a particular length (p), the set of strings with length p are sorted in lexigographical order

~~Ex~~ let p=3

then $\{000, 001, 010, 011, 100 \ldots\}$

Now   let $\boxed{\text{length} = l}$

$$X = \{ \underbrace{\epsilon}, \overbrace{0, 1}^{\text{length}=1}, \overbrace{00, 01, 10, 11}^{l=2}, \overbrace{000, 001, 010, 011, \ldots}^{l=3} \}$$

$$N = \{ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 \ldots \}$$

~~Also~~ Here X and N have one-to one correspondence (Bijection)

let $f : X \to N$ map strings in X to numbers in N as shown

then 'f' is bijection

$\Rightarrow$ Set X is countably infinite

---

(2) Assume that the set of all languages is countably infinite and be denoted by L

$\Rightarrow L = \{ L_1, L_2, L_3 \ldots \}$

$L_1, L_2, L_3 \ldots$ are all possible languages over Σ

Now By diagonalisation process

build a matrix with

row indices (i) — language L_i

column indices (j) — string S_j

S_j = set of all strings as in X in (1)

- each cell at row index(i) and column index(j) is set to 1 if L_i accepts S_j else it is set to 0

$$
\begin{array}{c|cccccc}
 & S_1 & S_2 & S_3 & S_4 & \cdots \\
\hline
L_1 & 0 & 1 & 1 & 0 & \cdots \\
L_2 & 1 & 1 & 0 & 1 & \cdots \\
L_3 & 1 & 0 & 0 & 1 & \cdots \\
L_4 & 0 & 1 & 1 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

Now we flip all the diagonal entries of the matrix.
Then collect all strings S_j which have their corresponding diagonal cell as 1 after flipping

$$
\begin{array}{c|cccccc}
 & S_1 & S_2 & S_3 & S_4 & \cdots \\
\hline
L_1 & \boxed{1} & 1 & 1 & 0 & \cdots \\
L_2 & 1 & \boxed{0} & 0 & 1 & \cdots \\
L_3 & 1 & 0 & \boxed{1} & 1 & \cdots \\
L_4 & 0 & 1 & 1 & \boxed{0} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}
$$

let

Language

$L_{diag} = S_1, S_3 \ldots$

Ldiag has a special property, it is different from all languages $L_i \in L$

thus $\boxed{L_{diag} \notin L}$

but Ldiag is a language over Σ

$\Rightarrow L_{diag} \in L$ should hold

This is a contradiction.

Hence L is not countably infinite

$\Rightarrow$ L is Uncountably infinite

Hence proved.

(8) Given,

a language L. TM = Turing Machine
RE = Recursive enumerable

RTP: $L = \{ \langle M \rangle \mid M$ is a Turing Machine and M does not accept $\langle M \rangle \}$

is undecidable.

Let A be TM that recognizes L
(Assume L is decidable)

let $\{M_1, M_2, \ldots\}$ be the set of Turing machines as a turing machine can be encoded as $\langle M \rangle$

consider the follow matrix

$$
\begin{array}{c|cccc}
 & \langle M_1 \rangle & \langle M_2 \rangle & \langle M_3 \rangle & \cdots \\
\hline
M_1 & 1 & 0 & 0 & \cdots \\
M_2 & 0 & 1 & 0 & \cdots \\
M_3 & 0 & 1 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \\
A & \cdots & \cdots & \cdots & x \text{ — TM-A}
\end{array}
$$

for a cell ith row jth col $mat[i][j] = 1$ if M_i accepts $\langle M_j \rangle$
else $mat[i][j] = 0$

Now by defn for A we have opposite values in diagonal entries

~~But~~ If we take opposite values on diagonal then we get a contradiction ~~what initial diagonal values~~ should be for x – what should its value be!!

x = diagonal value for ~~that~~ TM – A

⇒ our Assumption is wrong

L is NOT decidable

Now consider $\bar{L}$

$\bar{L}$ is recursively enumerable

~~L can~~ TM for $\bar{L}$ can be made by following

⇒ Simulate M with input <M>

↳ if ~~M~~ ACCEPTS the ACCEPT

K → TM for $\bar{L}$ ⇒ { else REJECTS

as a TM K recognizes $\bar{L}$ ⇒ $\bar{L}$ is recursively enumerable

Now

① L is ~~...~~ undecidable

② $\bar{L}$ is RE

⇒ L is partially undecidable

⇒ L form CO-RE

⇒ L is NOT recursively Enumerable (NOT RE)

Hence, L is undecidable and NOT recursively enumerable