USN | | | | | | | | | |

CMRIT CMR INSTITUTE OF TECHNOLOGY, BENGALURU. ACCREDITED WITH A+ GRADE BY NAAC

Internal Assessment Test 3 – November 2019

Sub:	Automata Theory and Computability			Sub Code:	17CS54	Branch:	ISE
Date:	16/11/2019	Duration:	90 min's	Max Marks:	50	Sem / Sec:	V A & B

		MARKS	CO	RBT
	Answer any FIVE FULL Questions			
1 (a)	State and prove pumping lemma for context free languages. Show that $L=\{a^nb^nc^n \mid n\geq1\}$ is not context free.	[10]	L3	CO3
2 (a)	i) If L_1 and L_2 are context free languages then prove that L_1 U L_2, $L_1 \cap L_2$ and L_1 are context free languages ii) If L_1 and L_2 are context free languages then prove that, they are not closed under Intersection, Complement and Difference	[10]	L2	CO3
3 (a)	Define a PDA and Design a PDA that accepts the language $L= \{a^nb^{2n} \mid n\geq1\}$. Draw the graphical representation. Show the moves made by PDA for aabbbb.	[10]	L3	CO4
4 (a)	Convert the following CFG to PDA $E \rightarrow E+E \mid E-E \mid (E) \mid I$ $I \rightarrow Ia \mid Ib \mid I0 \mid I1 \mid a \mid b \mid 0 \mid 1$	[6]	L2	CO3
(b)	List the undecidable questions.	[4]	L1	CO5

			CO	RBT
5 (a)	Obtain a CFG for the following PDA shown below: $\Delta(q_0, a, Z)= (q_0, AZ)$ $\Delta(q_0, a, A)=(q_0, A)$ $\Delta(q_0, b, A)= (q_1, \varepsilon)$ $\Delta(q_0, \varepsilon, A)= (q_2, \varepsilon)$	[10]	L2	CO3
6 (a)	With a neat diagram, explain the working of a basic Turing machine. Obtain a Turing machine to accept the language $L=\{ww^R \mid w \in (0+1)^*\}$	[10]	L3	CO4
7(a)	Write a short note on multi tape Turing machine and non deterministic Turing machine.	[6]	L1	CO1
(b)	Define i) Recursive language ii) Decidable language	[4]	L1	CO1
8(a)	i) Prove that A_{CSG} is decidable. ii) What is Post Correspondence problem? Prove that it is undecidable	[10]	L2	CO1

1) State and prove pumping lemma for context free languages – 6 M
Show that $L=\{a^n b^n c^n \mid n \geq 1\}$ is not context free.- 4 M

Statement:- If L is a context-free language, then:

$\exists K \geq 1$ ($\forall$ strings $w \in L$ where $|w| \geq K$)

$\exists u, v, x, y, z$ ($w = uvxyz$,

$vy \neq \varepsilon$

$|vxy| \leq K$ and

$\forall q \geq 0$ ($uv^q x y^q z$) is in L)

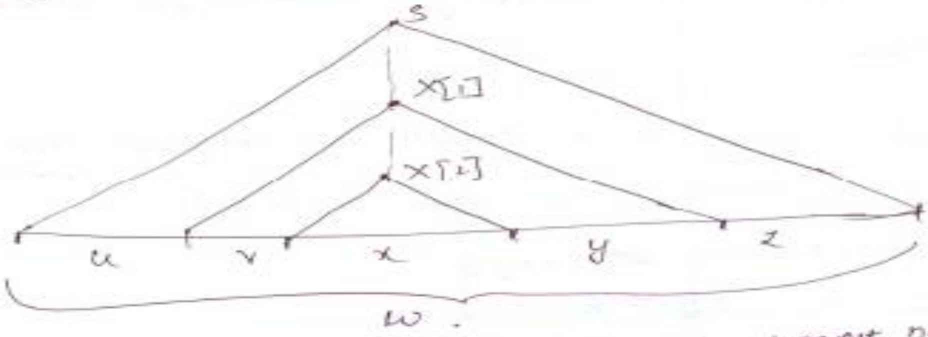
Proof:- If L is context-free then it is generated by some context-free grammar $G = (V, \Sigma, R, S)$ with n nonterminal symbols and branching factor b. Let K be b^{n+1}. Any string that can be generated by G and whose parse tree contains no paths with repeated nonterminals must have length less than or equal to b^n.

Let w be any string in L(G) where $|w| \geq K$.

Let T be any smallest parse tree for w.

~~Lets~~ T must have height at least $n+1$.

Choose some path T of length at least $n+1$.
Let X be the bottommost repeated non terminal along that path. Then w can be rewritten as uvxyz as shown below.



The tree rooted at [1] has height at most $n+1$. Thus its yield vxy, has length less than or equal to b^{n+1} which is K, and $vy \neq \varepsilon$ ($S \overset{*}{\Rightarrow} uXz \overset{*}{\Rightarrow} uxz$) (rule 1)
Finally v and y can be pumped: uxz must be in L. And, for any $q \geq 1$, $uv^q x y^q z$ must be in L because,
$S \overset{*}{\Rightarrow} uXz \overset{*}{\Rightarrow} uvXyz \overset{*}{\Rightarrow} uvxyz$ (rule 2) could have been immediately used at X[1]. And for any $q \geq 1$ $uv^q x y^q z$ must be in L because rule 1 could have been used q times before finally using rule 2.

So, if L is a context-free language, every "long" ② string in L must be pumpable.
The pumped region can be pumped out once or pumped in any number of times, in all cases resulting in another string that is also in L.

Let L be a context free, Let $w = a^n b^n c^n \in L$.

Note, $|w| \geq n$ and we split w into $uvxyz$ such that

$|vxy| \leq n$ and $|vy| \geq 1$ or $vy \neq \epsilon$

and so $uv^q w x^q y \in L$ for $i = 0, 1, 2$ must satisfy according to pumping lemma.

Consider case: $u = a^j$, $v = a^k$ where $|vy| = j + k \geq 1$ and $|vxy| \leq n$. which is $\underbrace{\overbrace{a \cdots a}^{n} \overbrace{b \cdots b}^{n}}_{uvxy} \underbrace{\overbrace{c \cdots c}^{n}}_{z}$,

Pump one: $uv^2 w x^2 y \in L$ for $q = 2$ the language is $a^{n+j+k} b^n c^n \notin L$. Hence by contradiction the given language is not context free.

---

2) i) If L_1 and L_2 are context free languages then prove that $L_1 \cup L_2$, $L_1 \cap L_2$ and L_1 are context free languages

$\Rightarrow$ The context-free languages are closed under union:

If L_1 and L_2 are context free languages, then there exists context free grammars $G_1 = (V_1, \Sigma_1, R_1, S_1)$ & $G_2 = (V_2, \Sigma_2, R_2, S_2)$ such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$.

Build a new grammar G such that $L(G) = L(G_1) \cup L(G_2)$.

G will contain all the rules of both G_1 and G_2

we add to G a new symbol, S, and two new rules $S \to S_1$ and $S \to S_2$.

The two new rules allow G to generate a string iff at least one of G_1 or G_2 generate it.

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

$\Rightarrow$ The context free languages are closed under concatenation.

If L_1 and L_2 are CFLs then there exists CFGs $G_1 = (V_1, \Sigma_1, R_1, S_1)$ and $G_2 = (V_2, \Sigma_2, R_2, S_2)$ such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$.

Build a new grammar G such that $L(G) = L(G_1) L(G_1)$.

G will contain rules of both G_1 and G_2.

Add to G a new start symbol S and new rule $S \to S_1 S_2$.

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

The Context-free languages are closed under Kleene star.

If L_1 is a context-free language, then there exists a context-free grammar $G_1 = (V_1, \Sigma_1, R_1, S_1)$ such that $L_1 = L(G_1)$.

Build a new grammar G such that $L(G) = L(G_1)^*$.

G will contain all the rules of G_1,

add to G a new start state S and two new rules

$S \to \varepsilon$ and $S \to SS_1$,

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

Proof * The context-free languages are not closed under intersection.

The proof is by counterexample

$n = 3$, $m = 2$

$L_1 = \{a^n b^n c^m : n, m \geq 0\}$ $\quad L_1 = \{a^3 b^3 c^2\}$

$L_2 = \{a^m b^n c^n : n, m \geq 0\}$ $\quad L_2 = \{a^2 b^3 c^3\}$

Both L_1 and L_2 are context-free, since CFG exist for these.

$L_1 \cap L_2 = \{a^2 b^3 c^2\}$

But now consider. $L = L_1 \cap L_2$.

$= \{a^n b^n c^n : n \geq 0\}$.

As it can be proved through pumping lemma that $\{a^n b^n c^n\}$ is not context-free. Context free languages are not closed under intersection.

* The context-free languages are not closed under intersection.

Given any sets L_1 and L_2.

$L_1 \cap L_2 = \neg(\neg L_1 \cup \neg L_2)$.

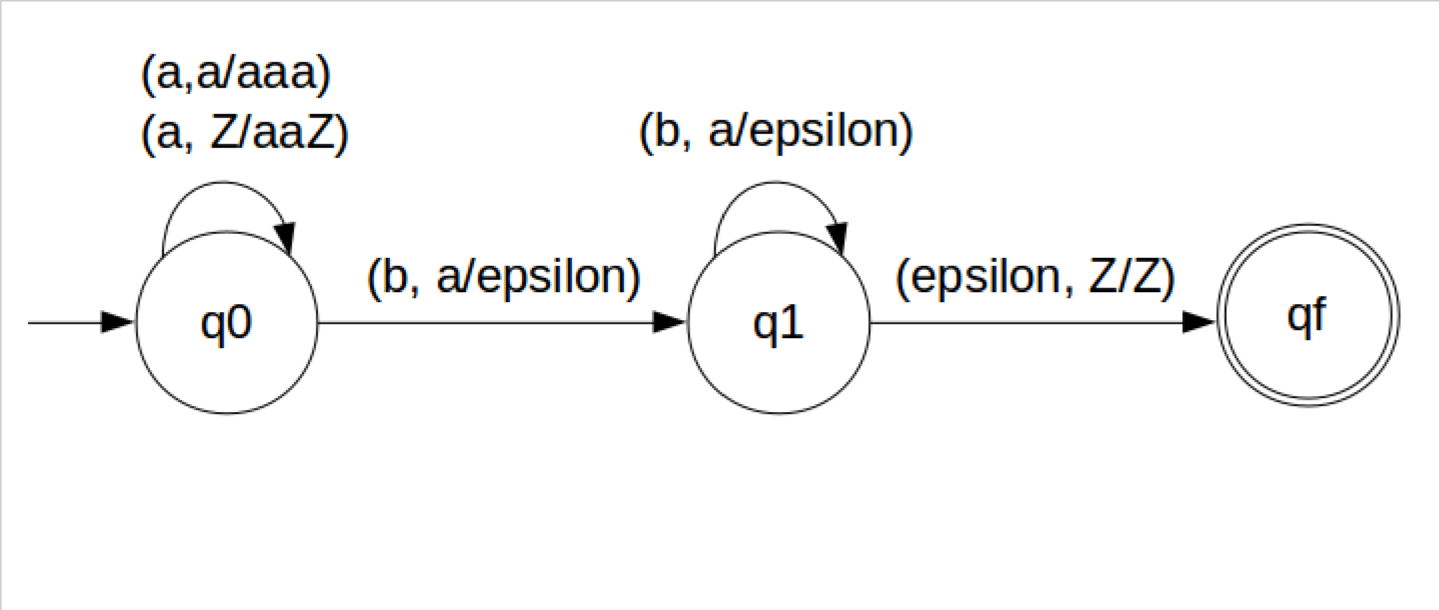
The context free languages are closed under union.
If they were also closed under complement, they would be necessarily be closed under intersection.
But they are not closed under intersection.
Thus they are not closed under complement either.

The context-free languages are not closed under difference (subtraction): given any language L,
$\neg L = \Sigma^* - L$.

Σ^* is context-free.
So if the context-free languages were closed under difference, the complement of any context-free language would necessarily be context-free. But complement of context-free language is not closed. Hence context free languages are not closed under difference.

3) Define a PDA and Design a PDA that accepts the language L= $\{a^n b^{2n} \mid n \geq 1\}$. Draw the graphical representation. Show the moves made by PDA for aabbbb.

```
δ(q0, a, Z) = (q0, aaZ)
δ(q0, a, a) = (q0, aaa)
δ(q0, b, a) = (q1, ε)
δ(q1, b, a) = (q1, ε)
δ(q1, ε, Z) = (qf, Z)
```



4a) Convert the following CFG to PDA $E \rightarrow E+E \mid E-E \mid (E) \mid I$

$I \rightarrow Ia \mid Ib \mid I0 \mid I1 \mid a \mid b \mid 0 \mid 1$

$\delta(q_0, \varepsilon, Z) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, \varepsilon, E) \rightarrow (q_1, E+E)$

$\delta(q_0, \varepsilon, E) \rightarrow (q_1, E-E)$

$\delta(q_0, \varepsilon, E) \rightarrow (q_1, (E))$

$\delta(q_0, \varepsilon, E) \rightarrow (q_1, I)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, Ia)$.

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, Ib)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, I0)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, I1)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, a)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, b)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, 0)$

$\delta(q_0, \varepsilon, I) \rightarrow (q_1, 1)$.

$\delta(q_0, 0, 0) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, 1, 1) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, a, a) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, b, b) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, +, +) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, -, -) \rightarrow (q_1, \varepsilon)$

$\delta(q_0, (, () \rightarrow (q_1, \varepsilon)$

$\delta(q_0,),)) \rightarrow (q_1, \varepsilon)$

b) List the undecidable questions

* Given a context-free language language L is $L = \Sigma^*$?
* Given a context-free language L, is the complement of L context-free?
* Given a context-free language L is L regular?
* Given two context-free languages L_1 & L_2 is $L_1 = L_2$?
* Given two context-free languages L_1 & L_2, is $L_1 \subseteq L_2$?
* Given two context-free languages L_1 and L_2 is $L_1 \cap L_2 = \phi$
* Given a context-free language L, is L inherently ambiguous?
* Given a context free grammar G, is G ambiguous.

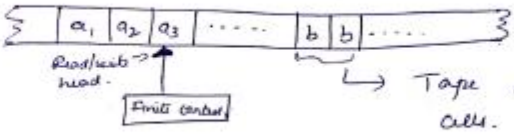
6) With a neat diagram, explain the working of a basic Turing machine.

Turing Machine Model

The Turing machine consists of

→ Finite control connected to a R/w head

→ Tape which is devided into number of cells.



Each cell can store only one symbol. The input to and output from the FSA are affected by the R/w head. In one move, the machine examines present symbol and present state to determine

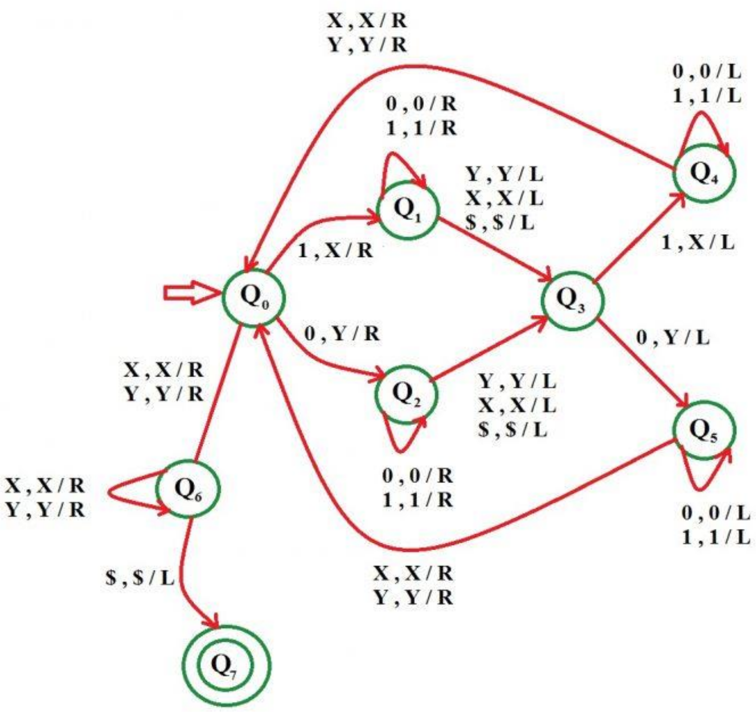
→ a new symbol to be written on the tape

→ a motion of R/w head along the tape (either left or right).

→ next state of the automaton and

→ Whether to halt or not.

Obtain a Turing machine to accept the language $L=\{ww^R \mid w \in (0+1)^*\}$



7)a) Write a short note on multi tape Turing machine and non deterministic Turing machine

Multitape Turing Machines

A multitape TM has.

Q - set of states.

q_0 - Initial state

$F \subseteq Q$ - set of final states.

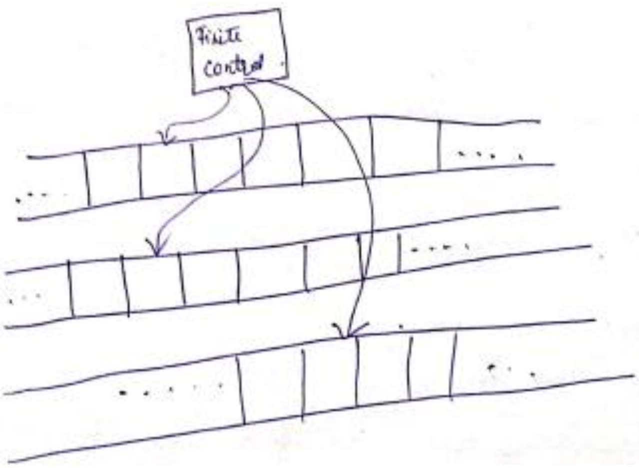
Γ set of tape symbols.

b - blank symbol.

Σ - set of symbols (input).

δ - is a partial function from.

$Q \times \Gamma^k$ into $Q \times \Gamma^k \times \{L, R, S\}^k$.



In a typical move

(i) M enter a new state

(ii) on each tape, a new symbol is written in the cell under the head.

(iii) Each tape head moves to the left or right or remains stationary. The heads move independently some move to the left, some to the right and remaining heads do not move.

Nondeterministic Turing machine.

1. Q is a finite nonempty set of states
2. Γ is a finite nonempty set of tape symbol
3. $b \in \Gamma$ is called the blank symbol.
4. Σ is a nonempty subset of Γ, called the set of input symbols and $b \notin \Sigma$.
5. q_0 is the initial state.
6. $F \subseteq Q$ is the set of final states
7. δ is a partial function from $Q \times \Gamma$ into the power set of $Q \times \Gamma \times \{L, R\}$.

If $q \in Q$ and $x \in \Gamma$, and

$\delta(q, x) = \{(q_1, y_1, D_1), (q_2, y_2, D_2 \ldots (q_n, y_n, D_n)\}$.

then NTM can choose any one of the actions defined by (q_i, y_i, D_i) for $i = 1, 2, \ldots n$.

b) Define i) Recursive language ii) Decidable language

(i) Recursive :- A Language $L \subseteq \Sigma^*$ is recursive if there exists some TM M that satisfies the following two conditions.

(a) If $w \in L$ then M accepts w and halts.

(b) If $w \notin L$, then M eventually halts, without reaching an accepting state.

Decidable language :- A problem with two answers (yes/no) is decidable if the corresponding language ~~language~~ is recursive.

8) i) Prove that A_{CSG} is decidable.

Theroem 10.3 A_{CSG} is decidable.

Proof The proof is a modification of the proof of Theorem 10.2. In Theorem 10.2, we considered derivations with $2k - 1$ steps for testing whether an input string of length k was in $L(G)$. In the case of context-sensitive grammar we construct $W_i = \{\alpha \in (V_N \cup \Sigma)^* \mid S \overset{*}{\underset{G}{\Rightarrow}} \alpha$ in i or fewer steps and $|\alpha| \leq n\}$. There exists a natural number k such that $W_k = W_{k+1} = W_{k+2} = \ldots$ (refer to proof of Theorem 4.3).

So $w \in L(G)$ if and only if $w \in W_k$. The construction of W_k is the key idea used in the construction of a TM accepting A_{CSG}. Now we can design a Turing machine M as follows:

1. Let G be a context-sensitive grammar and w an input string of length n. Then (G, w) is an input for TM.
2. Construct $W_0 = \{S\}$. $W_{i+1} = W_i \cup \{\beta \in (V_N \cup \Sigma)^* \mid$ there exists $\alpha_i \in W_i$ such that $\alpha \Rightarrow \beta$ and $|\beta| \leq n\}$. Continue until $W_k = W_{k+1}$ for some k. (This is possible by Theorem 4.3.)
3. If $w \in W_k$, $w \in L(G)$ and M accepts (G, w); otherwise M rejects (G, w). ∎

ii)What is Post Correspondence problem? Prove that it is undecidable

The Post Correspondence Problem (PCP) was first introduced by Emil Post in 1946. Later, the problem was found to have many applications 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 the two lists $x = (x_1 \ldots x_n)$, $y = (y_1 \ldots y_n)$ of nonempty strings over an alphabet $\Sigma = \{0, 1\}$. The PCP is to determine whether or not there exist $i_1, \ldots, i_m$, 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 be distinct and m may be greater than n. Also, if there exists a solution to PCP, there exist infinitely many solutions.

We have to determine whether or not there exists a sequence of substrings of x such that the string formed by this sequence and the string formed by the sequence of 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

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

Thus the PCP has a solution.