USN | | | | | | | | | |

Internal Assessment Test 2 – January 2023

Sub:	**AUTOMATA THEORY AND COMPILER DESIGN**			Sub Code:	**21CS51**	Branch :	**AIDS**	
Date:	**30/01/2024**	Duration:	Max Marks: **50**	Sem	**V**		**OBE**	

		Answer any FIVE Questions	MARKS	CO	RBT
1	a	**Define Regular Expression. Write a short note on Operators of RE.** Language represented in the form of expression (symbols separated by operator) is called regular expression. —--->1 MARK 1. The Union of two languages L and M, denoted L U M, is the set of strings that are in either L or M. eg. if L={001,10,111} and M={ε, 001}, then L U M={ε, 10, 001,111} 2. The Concatenation of Languages L and M is the set of strings that can be formed by taking any string in L and Concatenating it with any string in M. Eg. L={001,10,111} and M={ε, 001} then L.M is { 001,01, 111, 001001, 10001, 111001} 3. The closure(or star, or Kleene Closure) of a language L is denoted L^* and represents the set of those strings that can be formed by taking any number of strings from L, possibly with repetitions and concatenating all of them. Eg. L={0,1} the L^* is all strings of 0's and 1's. —------> 3 MARKS	[4]	CO3	L1

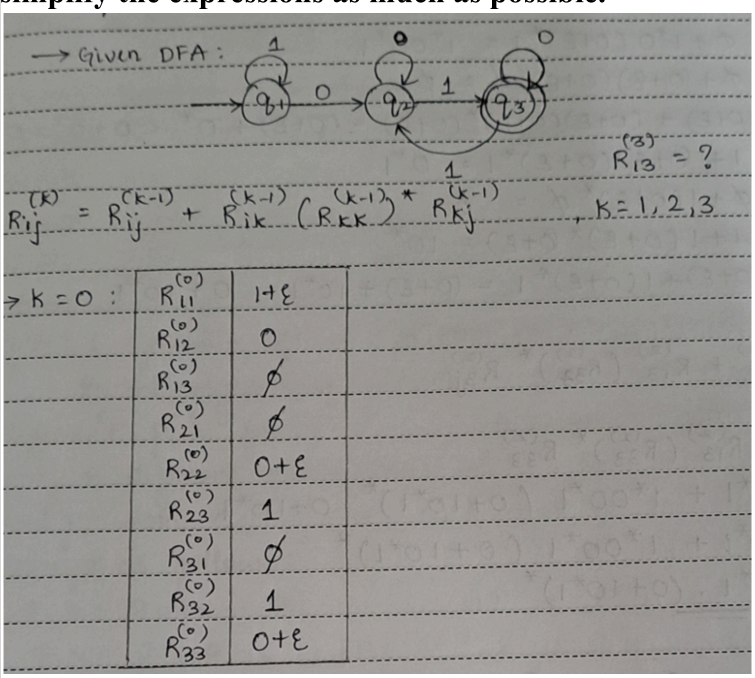
<table>
<tr>
<td></td>
<td>b</td>
<td>
Find the regular expression for the following Languages:
i) To accept strings of 0’s and 1’s having no 2 consecutive 0’s

L={ε, 0, 1, 01, 011, 101, 0101…..}------------->1 MARK
RE=(01+1)*(0+ε)+(0+ε)(10+1)* —---------->2 MARKS

ii) String starts and ends with different alphabet over Σ={0,1}

L={0011, 1010, 01110101, 100010……}---------->1 MARK
RE=0(0+1)*1+1(0+1)*0 -------->2 MARKS
</td>
<td>[6]</td>
<td>CO3</td>
<td>L3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>
Convert DFA to Regular Expression.

<table>
<tr><td>δ</td><td>0</td><td>1</td></tr>
<tr><td>→q1</td><td>q2</td><td>q1</td></tr>
<tr><td>q2</td><td>q2</td><td>q3</td></tr>
<tr><td>*q3</td><td>q3</td><td>q2</td></tr>
</table>

Give all the regular expressions for $R_{ij}^{(0)}$, $R_{ij}^{(1)}$, $R_{ij}^{(2)}$. Try to simplify the expressions as much as possible.


—---> 3 MARKS
</td>
<td>[10]</td>
<td>CO3</td>
<td>L3</td>
</tr>
</table>

<table>
<tr>
<td>3</td>
<td>a</td>
<td>
$\rightarrow k=1: \; R_{ij}^{(1)} = R_{ij}^{(0)} + R_{i1}^{(0)} (R_{11}^{(0)})^* R_{1j}^{(0)}$
$R_{11}^{(1)}$ | $(1+\epsilon)+(1+\epsilon)(1+\epsilon)^*(1+\epsilon) = 1+1^* = 1$
$R_{12}^{(1)}$ | $0+(1+\epsilon)(1+\epsilon)^*0 = 1^*0$
$R_{13}^{(1)}$ | $\phi+(1+\epsilon)(1+\epsilon)^*\phi = \phi$
$R_{21}^{(1)}$ | $\phi+\phi(1+\epsilon)^*(1+\epsilon) = \phi$
$R_{22}^{(1)}$ | $(0+\epsilon)+\phi(1+\epsilon)^*0 = (0+\epsilon)$
$R_{23}^{(1)}$ | $1+\phi(1+\epsilon)^*\phi = 1$
$R_{31}^{(1)}$ | $\phi+\phi(1+\epsilon)^*(1+\epsilon) = \phi$
$R_{32}^{(1)}$ | $1+\phi(1+\epsilon)^*0 = 1$
$R_{33}^{(1)}$ | $(0+\epsilon)+\phi(1+\epsilon)^*\phi = (0+\epsilon)$
$\rightarrow k=2: \; R_{ij}^{(2)} = R_{ij}^{(1)} + R_{i2}^{(1)} (R_{22}^{(1)})^* R_{2j}^{(1)}$
——----> **3** MARKS
k=2: $R_{11}^{(2)}$ | $1^* + 1^*0(0+\epsilon)^*\phi$
$R_{12}^{(2)}$ | $1^*0 + 1^*0(0+\epsilon)^*(0+\epsilon) = 1^*00^*$
$R_{13}^{(2)}$ | $\phi + 1^*0(0+\epsilon)^*1 = 1^*00^*1$
$R_{21}^{(2)}$ | $\phi + (0+\epsilon)(0+\epsilon)^*\phi = \phi$
$R_{22}^{(2)}$ | $(0+\epsilon)+(0+\epsilon)(0+\epsilon)^*(0+\epsilon) = (0+\epsilon)+0^* = 0+0^* = 0^*$
$R_{23}^{(2)}$ | $1+(0+\epsilon)(0+\epsilon)^*1 = 0^*1$
$R_{31}^{(2)}$ | $\phi+1(0+\epsilon)^*\phi = \phi$
$R_{32}^{(2)}$ | $1+1(0+\epsilon)^*(0+\epsilon) = 10^*$
$R_{33}^{(2)}$ | $(0+\epsilon)+1(0+\epsilon)^*1 = (0+\epsilon)+10^*1 = 0+10^*1$
$k=3: \; R_{ij}^{(3)} = R_{ij}^{(2)} + R_{i3}^{(2)} (R_{33}^{(2)})^* R_{3j}^{(2)}$
$\therefore R_{13}^{(3)} = R_{13}^{(2)} + R_{13}^{(2)} (R_{33}^{(2)})^* R_{33}^{(2)}$
$= 1^*00^*1 + 1^*00^*1\,(0+10^*1)^* \; 0+10^*1$
$= 1^*00^*1 + 1^*00^*1\,(0+10^*1)^*$
$= 1^*00^*1\,.\,(0+10^*1)^*$
——-------> 3 MARKS
RE——-----------------> 1 MARK
State and prove Pumping Lemma Theorem and

Theorem 4.1: (The *pumping lemma for regular languages*) Let L be a regular language. Then there exists a constant n (which depends on L) such that for every string w in L such that $|w| \geq n$, we can break w into three strings, $w = xyz$, such that:
1. $y \neq \epsilon$.
2. $|xy| \leq n$.
3. For all $k \geq 0$, the string xy^kz is also in L.

That is, we can always find a nonempty string y not too far from the beginning of w that can be "pumped"; that is, repeating y any number of times, or deleting it (the case $k = 0$), keeps the resulting string in the language L.
</td>
<td>[5]</td>
<td>CO3</td>
<td>L3</td>
</tr>
</table>

		PROOF: Suppose L is regular. Then $L = L(A)$ for some DFA A. Suppose A has n states. Now, consider any string w of length n or more, say $w = a_1a_2 \cdots a_m$, where $m \geq n$ and each a_i is an input symbol. For $i = 0, 1, \ldots, n$ define state p_i to be $\hat{\delta}(q_0, a_1a_2 \cdots a_i)$, where δ is the transition function of A, and q_0 is the start state of A. That is, p_i is the state A is in after reading the first i symbols of w. Note that $p_0 = q_0$. By the pigeonhole principle, it is not possible for the $n + 1$ different p_i's for $i = 0, 1, \ldots, n$ to be distinct, since there are only n different states. Thus, we can find two different integers i and j, with $0 \leq i < j \leq n$, such that $p_i = p_j$. Now, we can break $w = xyz$ as follows: 1. $x = a_1a_2 \cdots a_i$. 2. $y = a_{i+1}a_{i+2} \cdots a_j$. 3. $z = a_{j+1}a_{j+2} \cdots a_m$. —------->3 MARKS **Prove L={a^n | n is prime} is not regular** L is a Regular Language 'n' is an integer constant Select a string 'w' from L such that L={aa, aaa, aaaaa, aaaaaaa……} Let n=3 aaa x=a y=a z=a For k=1= xy^k**z= aaa** $\in$ L k=2= aaaa $\notin$ L **contradiction** —------> 2 MARKS			

	b	**Write a short note on the definition of CFG.** There are four important components in a grammatical description of a language: 1. There is a finite set of symbols that form the strings of the language being defined. This set was $\{0, 1\}$ in the palindrome example we just saw. We call this alphabet the *terminals*, or *terminal symbols*. 2. There is a finite set of *variables*, also called sometimes *nonterminals* or *syntactic categories*. Each variable represents a language; i.e., a set of strings. In our example above, there was only one variable, P, which we used to represent the class of palindromes over alphabet $\{0, 1\}$. 3. One of the variables represents the language being defined; it is called the *start symbol*. Other variables represent auxiliary classes of strings that are used to help define the language of the start symbol. In our example, P, the only variable, is the start symbol. 4. There is a finite set of *productions* or *rules* that represent the recursive definition of a language. Each production consists of: (a) A variable that is being (partially) defined by the production. This variable is often called the *head* of the production. (b) The production symbol $\rightarrow$. (c) A string of zero or more terminals and variables. This string, called the *body* of the production, represents one way to form strings in the language of the variable of the head. In so doing, we leave terminals unchanged and substitute for each variable of the body any string that is known to be in the language of that variable. The four components just described form a *context-free grammar*, or just *grammar*, or *CFG*. We shall represent a CFG G by its four components, that is, $G = (V, T, P, S)$, where V is the set of variables, T the terminals, P the set of productions, and S the start symbol. —---> 3 MARKS **Construct a CFG for L={$a^n b^{2n}$ | $n \geqslant 1$}.** S->aSbb| abb CFG, G={{S}, {a,b},P, S} —---> 2 MARKS	[5]	CO3	L3

<table>
<tr>
<td>4</td>
<td>a</td>
<td>

Obtain the LFD and RMD for a+a+a, using the following production rules

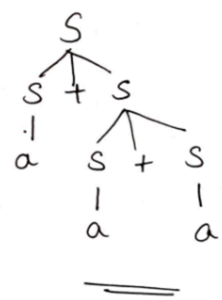
$S \rightarrow S+S|a$

Show whether the grammar is ambiguous or not. If ambiguous, design grammar to be unambiguous.

Left Most: —--------->4 MARKS

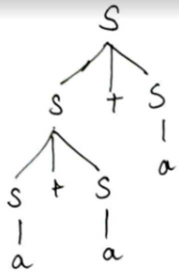
RightMost—---------> 4 MARKS

S
$\underset{lm}{\Rightarrow} S+S \quad S\rightarrow S+S$
$\underset{lm}{\Rightarrow} a+S \quad S\rightarrow a.$
$\underset{lm}{\Rightarrow} a+S+S \quad S\rightarrow S+S$
$\underset{lm}{\Rightarrow} a+a+S \quad S\rightarrow a$
$\underset{lm}{\Rightarrow} a+a+a \quad S\rightarrow a$



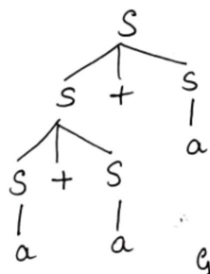
$\underset{lm}{\Rightarrow} S$
$\underset{lm}{\Rightarrow} S+S$
$\underset{lm}{\Rightarrow} S+S+S$
$\Rightarrow a+S+S$
$\Rightarrow a+a+S$
$\Rightarrow a+a+a$

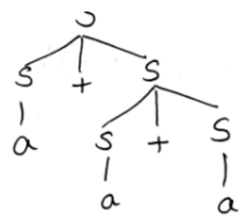
</td>
<td>[10]</td>
<td>CO2</td>
<td>L5</td>
</tr>
</table>



$\xRightarrow{rm} S$
$\xRightarrow{rm} S+S$
$\xRightarrow{rm} S+a$
$\xRightarrow{rm} S+S+a$
$\xRightarrow{rm} S+a+a$
$\xRightarrow{rm} a+a+a$

$\xRightarrow{rm} S$
$\xRightarrow{rm} S+S$
$\xRightarrow{rm} S+S+S$
$\xRightarrow{rm} S+S+a$
$\xRightarrow{rm} S+a+a$
$\xRightarrow{rm} a+a+a$





Given Grammar is Ambiguous.

S->S+a|a

Design LMD and RMD and prove new grammar is unambiguous

—----> 2 MARKS

<table>
<tr>
<td>5</td>
<td>a</td>
<td>

Give First and Follow for following Production rules.

S→ABCDE
A→a|ε
B→b|ε
C→c
D→d|ε
E→e|ε

<table>
<tr><th>Production</th><th>FIRST</th><th>FOLLOW</th></tr>
<tr><td>S</td><td>{ a, b, c }</td><td>{ $ }</td></tr>
<tr><td>A</td><td>{ a, ε }</td><td>{ b, c }</td></tr>
<tr><td>B</td><td>{ b, ε }</td><td>{ c }</td></tr>
<tr><td>C</td><td>{ c }</td><td>{ d, e, $ }</td></tr>
<tr><td>D</td><td>{ d, ε }</td><td>{ e, $ }</td></tr>
<tr><td>E</td><td>{</td><td>{</td></tr>
</table>

</td>
<td>[4]</td>
<td>CO2</td>
<td>L3</td>
</tr>
</table>

			e, ε }	$ }				

<table>
<tr>
<td></td>
<td>b</td>
<td>

Convert the following automata into RE by state elimination method.

	0	**1**
→q1	**q2**	**q1**
q2	**q3**	**q1**
***q3**	**q3**	**q2**

Elemenate q_2

1) $R = \phi, \ Q = 0, \ S = \phi, \ P = 0$

$R + QS^*P$

$\phi + 0(\phi)^*0$

00

$\rightarrow q_1 \xrightarrow{00} q_3$

2) $q_3 \rightarrow q_2 \rightarrow q_1$

$R = \phi, \ Q = 1, \ S = \phi, \ P = 1$

$R + QS^*P$

$\phi + 1(\phi)^*1$

$= 11$

$\rightarrow q_1 \xleftarrow{11} q_3$

</td>
<td>[6]</td>
<td>CO3</td>
<td>L3</td>
</tr>
</table>

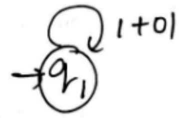
iii) $\rightarrow q_1 \longrightarrow q_2 \longrightarrow q_1$

$R = 1, \quad Q = 0, \quad S = \phi, \quad P = 1$

$R + Q(S)^{*}P$

$1 + 0(\phi)^{*}1$

$\underline{\underline{1 + 01}}$

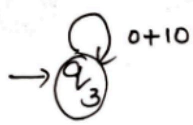


iv) $q_3 \rightarrow q_2 \rightarrow q_3$

$R = 0, \quad Q = 1, \quad S = \phi, \quad P = 0$

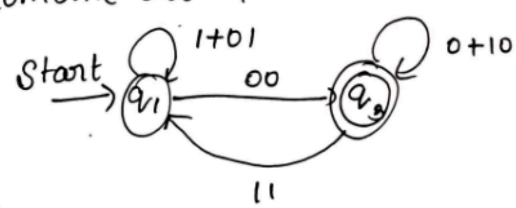
$0 + 1(\phi)^{*}0$

$= \underline{\underline{0 + 10}}$



Eliminating q2 —--> 6 MARKS
Final RE —------> 4 MARKS

Combine all 4 states



$R = 1+01, \quad S = 00, \quad U = 0+10, \quad T = 11$

$RE = (R + SU^{*}T)^{*} SU^{*}$

$= \underline{\underline{((1+01) + 00(0+10)^{*}11)^{*}\, 00\, (0+10)^{*}}}$

<table>
<tr>
<td>6</td>
<td></td>
<td>

Consider the grammar below

E→E+T|T
T→T*F|F
F→(E)|id

1. **Remove left recursion**
2. **Find FIRST and FOLLOW**
3. **Construct a parsing table and show whether the grammar is accepted by LL(1) parser or not.**

$$
\begin{array}{lcl}
E & \rightarrow & T\ E' \\
E' & \rightarrow & +\ T\ E' \mid \epsilon \\
T & \rightarrow & F\ T' \\
T' & \rightarrow & *\ F\ T' \mid \epsilon \\
F & \rightarrow & (\ E\ ) \mid \textbf{id}
\end{array}
$$

------> 2 MARKS

<table>
<tr><td></td><td>FIRST</td><td>FOLLOW</td></tr>
<tr><td>E</td><td>{id,(}</td><td>{$,)}</td></tr>
<tr><td>E'</td><td>{+,ε}</td><td>{$,)}</td></tr>
<tr><td>T</td><td>{id,(</td><td>{+,</td></tr>
</table>

</td>
<td>[10]</td>
<td>CO2</td>
<td>L3,L5</td>
</tr>
</table>

	}	$,) }
T'	{ *, ε }	{ + , $,) }
F	{id, (}	{ + , $,) , * }

—---> 4 MARKS

Parsing table —---> 3 MARKS

	id	(	)	*	+	$
E	E → TE'	E → TE'				
E'			E' → ε		E' → +TE'	E' → ε
T	T → FT'	T → FT'				
T'			T' →)	T' → *FT'	T' → ε	T' → ε
F	F → id	F → (E)				

Final statement —---> 1 MARK

Grammar is accepted by LL(1)

CCI SIGNATURE HOD SIGNATURE