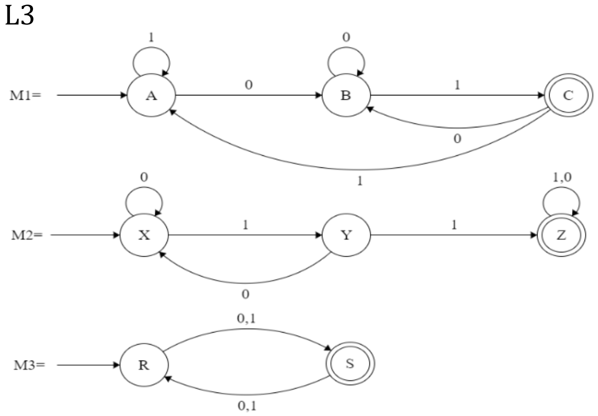
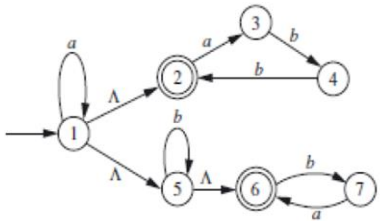
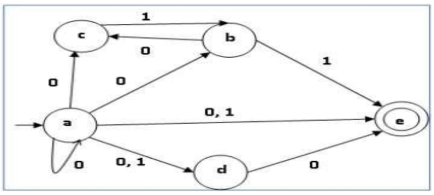
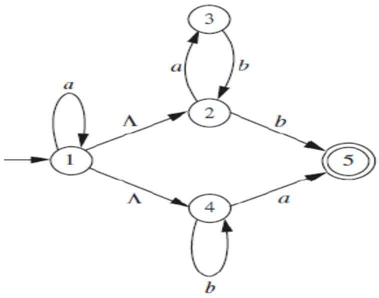
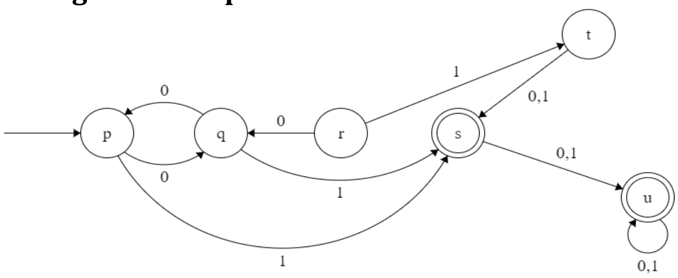
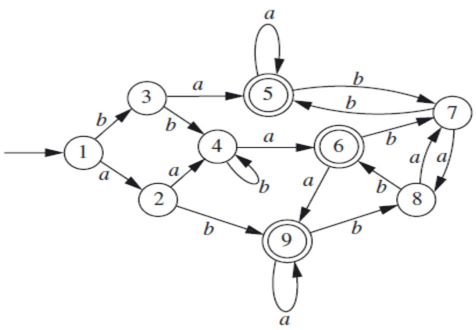
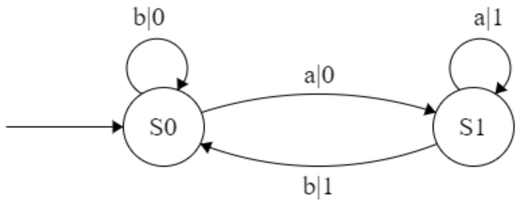
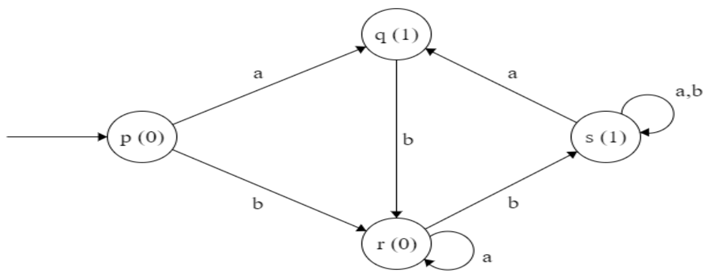
Sr.	Unit No.	Question	BL	CO
1.	1	Define following: 1. Conjunction 2. Disjunction 3. Negation 4. Tautology 5. Composition 6. Inverse of function	R	CO1
2.	1	Describe one-to-one, onto and bijection function.	U	CO1
3.	1	Check whether the function 1. f: R→R, f(x) = x^2 is one to one or onto. 2. f: R→R^+, f(x) = x^2 is one to one or onto. 3. f: R^+→R, f(x) = x^2 is one to one or onto. 4. f: R^+→R^+, f(x) = x^2 is one to one or onto.	U	CO1
4.	1	Define tautology and contradiction	R	CO1
5.	1	Derive truth table for following logic formula: 1. P → (¬P V ¬Q). Is it a tautology? A contradiction? Or neither? Justify your answer. 2. (p v q) ^ ¬(p→q) is tautology? A contradiction? Or neither? Justify your answer.	U	CO1
6.	1	Define reflexivity, symmetry, and transitivity properties of relations.	R	CO1
7.	1	Check Equivalence Relation for given Examples. 1. A={1,2,3}, R = {(1, 3), (3, 1), (2, 2)} 2. A={1,2,3}, R = {(1, 1), (2, 2), (3, 3), (1, 2)} 3. R = Ø 4. A={1,2,3}, R = {(1,2), (1,1), (2,1), (2,2), (3,2), (3,3)}	U	CO1
8.	1	Prove that √2 is irrational by method of contradiction.	A	CO1
9.	1	Prove that for every n >= 1, using PMI $\sum_{i=0}^{n} i = n(n+1)/2$	A	CO1
10.	1	Prove that for every n >= 1, 7 + 13 + 19 + . . . + (6n + 1) = n(3n +4) using PMI	A	CO1
11.	1	Prove that for every n ≥ 1, using PMI $\sum_{i=1}^{n} 1/i(i+1) = n/(n+1)$	A	CO1
12.	1	Prove that For every n >= 1, using PMI $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$	A	CO1
13.	1	Prove that 1 +3 +5 + ... +r = n^2 for all n>0 where r is an odd integer (Note : r = 2n-1) using PMI	A	CO1
14.	1	Show that $2^n > n^3$ for n >= 10 by Mathematical Induction.	A	CO1
15.	1	Prove that for every n >= 0, n (n^2 +5) is divisible by 6 using PMI	A	CO1
16.	2	Draw Finite Automata (FA) for following languages: L1 = {x / 11 is not a substring of x, x Є {0,1}*} L2 = {x / x ends with 10, x Є {0,1}* } Find FA accepting languages (i) L1 ∩ L2 and (ii) L1 – L2 (iii) L1 U L2	A	CO2

<table>
<tr><td>17.</td><td>2</td><td>Draw Finite Automata (FA) for following languages:
L1 = {x / 00 is not a substring of x }
L2 = {x / x ends with 01 }
Find FA accepting languages (i) L1 ∩ L2 and (ii) L2 – L1 (iii) L1 U L2</td><td>A</td><td>CO2</td></tr>
<tr><td>18.</td><td>2</td><td>Let M1, M2 and M3 be the FAs pictured in Figure, recognizing languages L1, L2 and L3, respectively. Draw FAs recognizing the following languages.
a. L1 U L2
b. L1 ∩ L3

</td><td>A</td><td>CO2</td></tr>
<tr><td>19.</td><td>2</td><td>Define δ* for FA , NFA and NFA- Λ.</td><td>R</td><td>CO2</td></tr>
<tr><td>20.</td><td>2</td><td>Explain how to convert NFA – Λ into NFA and FA with suitable example OR
Convert following NFA- Λ to NFA and FA.
1.
<table>
<tr><td>Q</td><td>δ (q, Λ)</td><td>δ (q,0)</td><td>δ (q,1)</td></tr>
<tr><td>A</td><td>{B}</td><td>{A}</td><td>Ø</td></tr>
<tr><td>B</td><td>{D}</td><td>{C}</td><td>Ø</td></tr>
<tr><td>C</td><td>Ø</td><td>Ø</td><td>{B}</td></tr>
<tr><td>D</td><td>Ø</td><td>{D}</td><td>Ø</td></tr>
</table>
2.
<table>
<tr><td></td><td>Λ</td><td>a</td><td>b</td><td>c</td></tr>
<tr><td>→p</td><td>Ø</td><td>{p}</td><td>{q}</td><td>{r}</td></tr>
<tr><td>q</td><td>{p}</td><td>{q}</td><td>{r}</td><td>Ø</td></tr>
<tr><td>*r</td><td>{q}</td><td>{r}</td><td>Ø</td><td>{p}</td></tr>
</table>
3.
<table>
<tr><td>Q</td><td>δ (q, Λ)</td><td>δ (q,0)</td><td>δ (q,1)</td></tr>
<tr><td>A</td><td>{B,D}</td><td>{A}</td><td>Ø</td></tr>
<tr><td>B</td><td>Ø</td><td>{C}</td><td>{E}</td></tr>
<tr><td>C</td><td>Ø</td><td>Ø</td><td>{B}</td></tr>
<tr><td>D</td><td>Ø</td><td>{E}</td><td>{D}</td></tr>
<tr><td>E</td><td>Ø</td><td>Ø</td><td>Ø</td></tr>
</table>
4.
<table>
<tr><td rowspan="2">Current State</td><td colspan="2">Input Symbol</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>→Q_0</td><td>Q_1</td><td>Q_0 , Q_2</td></tr>
<tr><td>Q_1</td><td>Q_2</td><td>Q_0</td></tr>
<tr><td>Q_2</td><td>Q_0</td><td>---</td></tr>
</table></td><td>A</td><td>CO2</td></tr>
</table>

<table>
<tr><td></td><td></td><td>5.

6.

7.
</td><td></td><td></td></tr>
<tr><td>21.</td><td>2</td><td>For the following Regular Expression draw an NFA- Λ recognizing the corresponding languages.
1. (00 + 1)* (10)*
2. 001*0*11
3. (0 + 1)* (10+01)* 11
4. (0 + 1)*1(0 + 1)
5. (0 + 1)* (10+110)* 1</td><td>A</td><td>CO2</td></tr>
<tr><td>22.</td><td>2</td><td>Prove: Any Regular Language can be accepted by a finite automaton (Kleene’s Theorem, Part - I)</td><td>U</td><td>CO2</td></tr>
<tr><td>23.</td><td>2</td><td>Minimize DFA for given example.

<table><tr><th>State / Transition</th><th>a</th><th>b</th></tr><tr><td>→ 1</td><td>{3}</td><td>{2}</td></tr><tr><td>2</td><td>{4}</td><td>{1}</td></tr><tr><td>3</td><td>{5}</td><td>{4}</td></tr><tr><td>4</td><td>{4}</td><td>{4}</td></tr><tr><td>5</td><td>{3}</td><td>{2}</td></tr></table>
</td><td>A</td><td>CO2</td></tr>
</table>

<table>
<tr><td>24.</td><td>2</td><td>Explain Pumping Lemma and its applications.</td><td>U</td><td>CO2</td></tr>
<tr><td>25.</td><td>2</td><td>Explain ‘finite state machines with outputs. Discriminate between Mealy and Moore machines.</td><td>U</td><td>CO2</td></tr>
<tr><td>26.</td><td>2</td><td>Design a Moore machine to find residue number 3 for binary number
Design Moore machine for 1’s complement of binary number.</td><td>A</td><td>CO2</td></tr>
<tr><td>27.</td><td>2</td><td>Convert Mealy machine to Moore machine.

</td><td>A</td><td>CO2</td></tr>
<tr><td>28.</td><td>2</td><td>Convert Moore machine to Mealy machine.
1.


2.
<table>
<tr><td rowspan="2">Present State</td><td colspan="2">Next State</td><td rowspan="2">Output</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>→p_0</td><td>r</td><td>q_0</td><td>ε</td></tr>
<tr><td>p_1</td><td>r</td><td>q_0</td><td>1</td></tr>
<tr><td>q_0</td><td>p_1</td><td>s_0</td><td>0</td></tr>
<tr><td>q_1</td><td>p_1</td><td>s_0</td><td>1</td></tr>
<tr><td>r</td><td>q_1</td><td>p_1</td><td>0</td></tr>
<tr><td>s_0</td><td>s_1</td><td>r</td><td>0</td></tr>
<tr><td>s_1</td><td>s_1</td><td>r</td><td>1</td></tr>
</table>
3.
<table>
<tr><td rowspan="2">Old state</td><td>After input a</td><td>After input b</td><td rowspan="2">Output</td></tr>
<tr><td>New state</td><td>New state</td></tr>
<tr><td>$-q_0$</td><td>q_1</td><td>q_2</td><td>0</td></tr>
<tr><td>q_1</td><td>q_3</td><td>q_2</td><td>1</td></tr>
<tr><td>q_2</td><td>q_2</td><td>q_3</td><td>0</td></tr>
<tr><td>q_3</td><td>q_3</td><td>q_3</td><td>1</td></tr>
</table></td><td>A</td><td>CO2</td></tr>
<tr><td>29.</td><td>3</td><td>Explain Chomsky hierarchy,
Find CFG for the following languages.
1. $L = \{ a^i b^j a^k \mid j > i + k \}$
2. $L = \{ a^i b^j c^k \mid i = j \text{ or } j = k \}$
3. $L = \{ 0^i 1^j 0^k \mid j > i + k \}$
4. $(011 + 1)^* (01)^*$
5. $L=(0+1)1^*(1+(01)^*)$
6. $L = a^*b^*$
7. $\{a^i b^j c^k \mid i=j+k\}$</td><td>A</td><td>CO3</td></tr>
</table>

<table>
<tr><td>30.</td><td>3</td><td>1. For string aaabbabbba, find Left most derivation and Right most derivation.
S→ aB|bA
A→a|aS|bAA
B→b|bS|aBB
2. Derive left most and right most derivation of string aabbaa using given grammar.
S→aAS|a
A→SbA|SS|ba</td><td>A</td><td>CO3</td></tr>
<tr><td>31.</td><td>3</td><td>Define CFG. When a CFG is called an ‘ambiguous CFG’? Explain with example.<table><tr><td>S →a|Sa|bSS|SSb|SbS</td><td>S→ABA,
A→aA | ε
B→bB | ε</td><td>S →ABA
A →aA | Λ
B →bB | Λ</td></tr><tr><td>S → A | B
A →aAb | aabb
B →abB | Λ</td><td colspan="2">S → S + S | S * S | a | b
Write the unambiguous CFG based on precedence rules for the above grammar.
Derive the parse tree for expression (a + a)*b</td></tr></table></td><td>U</td><td>CO3</td></tr>
<tr><td>32.</td><td>3</td><td>Explain Union Rule and Concatenation Rule for Context-Free Grammar.</td><td>U</td><td>CO3</td></tr>
<tr><td>33.</td><td>3</td><td>Define CNF and Convert following CFG to equivalent Chomsky Normal Form.
S→ AACD | ACD | AAC | CD | AC | C
A→ aAb | ab
C→ aC | a
D→ aDa | bDb | aa | bb</td><td>A</td><td>CO3</td></tr>
<tr><td>34.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ ASB | Λ
A→ aAS | a
B→ SbS | A | bb</td><td>A</td><td>CO3</td></tr>
<tr><td>35.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ ASA | aB
A→ B | S
B→ b | Λ</td><td>A</td><td>CO3</td></tr>
<tr><td>36.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ bA|aB
A→ bAA|aS|a
B→ aBB|bS|b</td><td>A</td><td>CO3</td></tr>
<tr><td>37.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ aAbB
A→ Ab | b
B→ Ba | a</td><td>A</td><td>CO3</td></tr>
<tr><td>38.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→0A0 | 1B1 | BB
A→C
B→S | A
C→S | ε</td><td>A</td><td>CO3</td></tr>
<tr><td>39.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ aY | Ybb | Y
X→ Λ | a
Y→ aXY | bb | Xxa</td><td>A</td><td>CO3</td></tr>
<tr><td>40.</td><td>3</td><td>Convert following CFG to equivalent Chomsky Normal Form.
S→ AaA | CA | BaB</td><td>A</td><td>CO3</td></tr>
</table>

		A→ aaBa | CDA | aa | DC B→ bB | bAB | bb | aS C→ Ca | bC | D D→ bD | Λ		
41.	3	Convert following CFG to equivalent Chomsky Normal Form. S → TU | V T → aTb | Λ U→ cU | Λ V →aVc | W W →bW | Λ	**A**	**CO3**
42.	**4**	Define PDA & its application also explain acceptance of a string by empty stack.	**U**	**CO4**
43.	**4**	Describe the pushdown automata for language $\{0^n1^n \mid n \geq 0\}$.	**A**	**CO4**
44.	**4**	**Design PDA for** Equal number of a's and b's OR Equal no of 0's and 1's.	**A**	**CO4**
45.	**4**	$L = \{ x \in \{ a, b\}^* \mid n_a(x) > n_b(x) \}$ **OR** Design PDA accepting strings with more a's than b's. Trace it for the string "abbabaa".	**A**	**CO4**
46.	**4**	$L = \{ xcx^r / x \in \{a,b\}^* \}$ design a PDA and trace it for string "abcba".	**A**	**CO4**
47.	**4**	Design PDA accepting all odd-even length strings over {a, b}	**A**	**CO4**
48.	**4**	Design PDA accepting the language: $\{ a^i b^j c^k \mid i, j, k \geq 0$ and $j = i$ or $j = k\}$	**A**	**CO4**
49.	**4**	Design PDA accepting the language: $\{ a^i b^j c^k \mid i, j, k \geq 0$ and $i=j+k\}$	**A**	**CO4**
50.	**4**	Design PDA accepting the language: $\{ a^n b^{n+m} a^m \mid n, m \geq 0\}$	**A**	**CO4**
51.	**4**	Design and draw a deterministic PDA accepting "Balanced strings of Brackets" which are accepted by following CFG. S → SS | [S] | {S} | Λ **OR** Design a pushdown automata to check well-formed parenthesis.	**A**	**CO4**
52.	**4**	Design PDA for $L = \{x \in \{a, b\}^* \mid n_a(x) \neq n_b(x)\}$ Trace it for the string abbaababbb (no of a's and b's are not same)	**A**	**CO4**
53.	**4**	Design a PDA, M to accept $L = \{a^n b^{2n} \mid n \geq 1\}$	**A**	**CO4**
54.	**4**	Convert given CFG to PDA. S→ 0AB A→ 1A | 1 B→ 0B | 1A | 0 Trace the string 01011 using PDA.	**A**	**CO4**
55.	**4**	Convert given CFG to PDA. S → 0B| 1A A → 0S|1AA | 0 B → 1S| 0BB | 1	**A**	**CO4**
56.	**4**	Convert given CFG to PDA. I → a | b | Ia | Ib | I0 | I1 E→ I | E * E | E + E | (E)	**A**	**CO4**
57.	**4**	Convert given CFG to PDA. S → AB A → BB | a B → AB | a | b	**A**	**CO4**
58.	**5**	Draw TM to accept Palindromes over {a, b}. (Even as well as Odd Palindromes)	**A**	**CO5**
59.	**5**	Draw TM to accept the language: $\{0^n 1^n \mid n \geq 1\}$	**A**	**CO5**
60.	**5**	Draw TM to accept the language: $\{ a^n b^n c^n \mid n \geq 1 \}$	**A**	**CO5**
61.	**5**	Draw the TM to copy string and delete a symbol.	**A**	**CO5**
62.	**5**	Design a Turing machine to reverse the string over alphabet {0, 1}	**A**	**CO5**

63.	**5**	Design a Turing machine which accepts the language consisting string which contain aba as a substring over alphabets {a, b}	**A**	**CO5**
64.	**5**	Draw the TM for L = { ss | s ∈ (a, b)*} OR Draw the TM for L = {xx | x ∈ {a, b}∗} also trace string aa.	**A**	**CO5**
65.	**5**	Draw a transition diagram for a Turing machine accepting the following language. $L = \{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}$.	**A**	**CO5**
66.	**5**	Explain Universal TM and Church Turing Thesis describes its capabilities.	**U**	**CO5**