REG.NO.:

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2024-2025

SLOT: F1+TF1

Programme Name & Branch : **B.Tech, CSE**
Course Code and Course Name : **BCSE304L – Theory of Computation**
Faculty Name(s) : KANAGARAJ R, MADIAJAGAN M, SATHIYA KUMAR C, KARTHIK G M, BASKARAN P, PARTHASARATHY G, SARITHA MURALI, RAJARAJAN G , SHALINI L, UMA PRIYA D, BOOMINATHAN P, LAKSHMANAN K, BHAWANA TYAGI, BHUVANESWARI M, IYAPPAN P, ISLABUDEEN M, PRAKASH M, SATHYA K, ADRIJA BHATTACHARYA, DEBI PRASANNA ACHARJYA, K.KRISHNA RANI SAMAL,SUGANTHINI C
Class Number(s) : **VL2024250501633, 1615, 1619, 1625, 1637, 1639, 1635, 1653, 1631, 1643, 1617, 1651, 1649, 1645, 1627, 1647, 1629, 1641, 1623, 1613, 1621, 1655**
Date of Examination : **21-Mar-2025, 2:00 PM – 3.30 PM**
Exam Duration : **90 minutes** **Maximum Marks: 50**

General instruction(s):

- Answer All Questions
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)

<table>
<tr><th>Q. No</th><th>Question</th><th>M</th><th>CO</th><th>BL</th></tr>
<tr><td rowspan="2">1.</td><td>a) Prove that the given language $L = \{a^n b^m \mid 1 \le n < m\}$ is regular language or not.
Ans:
If L is regular, it satisfies the Pumping Lemma for pumping length p such that $w \in L$ with $|w| \ge p$.
Split w into xyz, where
1. $xy^iz \in L, \forall i \ge 0$
2. $|y| > 0$
3. $|xy| \le p$.

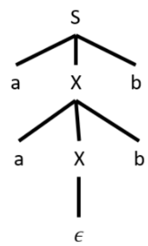
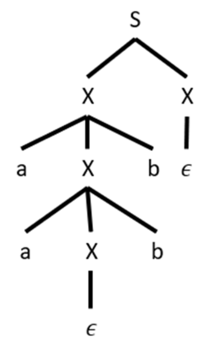
$L = \{a^n b^m \mid 1 \le n < m\}, w = a^l b^l$, i.e $|w| = p$
Choose P = 5, so w = aabbb$\in L$.
Split into xyz for w such that $x = a, y = ab, z = bb$
1. $|y| = |ab| = 2 > 0,\ TRUE$
2. $|xy| = |aab| = 3 \le (p = 5)$, TRUE
3. Check $xy^iz \in L$ for i=1, 2, 3, 4, 5…
i=2: $xy^2z \longrightarrow a(ab)^2bb = aababbb \notin L$
Therefor the given language is not Regular.</td><td>7</td><td rowspan="2">2</td><td rowspan="2">3</td></tr>
<tr><td>b) Consider the homomorphism h from the alphabet {0,1,2} to {a,b} defined by: h(0)=ab, h(1)=b, h(2)=aa.
1. What is h(0210) and h(2201)? 2 marks
2. If L is the language (ab+baa)*bab, what is $h^{-1}(L)$? 1 mark
Ans:
1. h(0210) = abaabab, h(2201) = aaaaabb
2. $h^{-1}(L) = (0 + 12)^*10$</td><td>3</td></tr>
</table>

REG.NO.:

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2024-2025

SLOT: F1+TF1

2.	**a)** Consider the grammar X→ aXb | XX | ϵ. Determine whether the grammar is ambiguous. If it is, prove the ambiguity by deriving two distinct parse trees for the same string. **Ans:** For String aabb $S \overset{lm}{\Rightarrow} aXb$ $\overset{lm}{\Rightarrow} aaXbb$ $\overset{lm}{\Rightarrow} aa\epsilon bb$ $\overset{lm}{\Rightarrow} aabb$  $S \overset{rm}{\Rightarrow} XX$ $\overset{rm}{\Rightarrow} aXbX$ $\overset{rm}{\Rightarrow} aXb\epsilon$ $\overset{rm}{\Rightarrow} aaXbb$ $\overset{rm}{\Rightarrow} aa\epsilon bb$ $\overset{rm}{\Rightarrow} aabb$ 	3	3	2
	b) Design context free grammar for the language $L = \{a^i b^j c^k \mid i = j + k\}$ Ans: L={ $\epsilon, a, ac, ab, aabb, aacc, aabc, aaabbb, aaabbc, aaabcc, aaaccc, \dots$ } $S \rightarrow AB,$ $A \rightarrow aAb \mid aAc \mid \epsilon,$ $B \rightarrow bB \mid cB \mid \epsilon$ String $\boldsymbol{a^4b^3c}$ $S \Rightarrow AB$ $\Rightarrow aAcB$ $\Rightarrow aaAbcB$ $\Rightarrow aaaAbbcB$ $\Rightarrow aaaaAbbbcB$ $\Rightarrow aaaa\epsilon bbbcB$ $\Rightarrow aaaabbbc\epsilon$ $\Rightarrow aaaabbbc$ String accepted	4		
	c) Consider the grammar **G** with $\boldsymbol{S \rightarrow 0S1 \mid 1S0 \mid \epsilon}$. In this grammar, for every **0** there is a **1** and vice versa. Is this grammar generating $\boldsymbol{L = \{w \mid n_0(w) = n_1(w)\}}$**?** Justify. If not, what further modifications are required in the grammar in order to generate the language $\boldsymbol{L}$? Ans: Examples of strings generated by G: • xx • 0x10x1	3		

	• 1x01x0 • 0(0x1)10(0x1)1 → 00x1100x11 • 1(0x1)01(0x1)0 → 10x01 The justification is as follows: 1. Each production rule ensures that for every 0, there is a matching 1, and vice versa. 2. The recursive structure ensures that the balance is maintained at every step of the derivation. 3. The base case x provides a single valid terminal string, ensuring $n_a(w)=n_b(w)=0$ (a valid element of L). **Base Case:** For S → x, the string x has no 0s or 1s, so $n_a(w)=n_b(w)=0$, which satisfies the condition. **Inductive Step:** Assume that a string ww derived from S satisfies $n_a(w)=n_b(w)$. When applying S → 0S1 or S → 1S0, the added 0 and 1 (or 1 and 0) maintain the balance. Therefore, any string derived from SS satisfies $n_a(w)=n_b(w)$. Hence, the grammar is correct for generating L.			
3.	**a)** Consider the grammar $S \rightarrow ASA \mid aB,\ A \rightarrow B, B \rightarrow b \mid \epsilon$. Convert into Chomsky Normal Form Ans: 1. S appears in RHS, S→ASA so P:{ S’ → S, S → ASA | aB, A → B, B → b | ϵ} 2. Remove ϵ- production. B → ϵ : P:{ S’ → S, S → ASA | aB | a, A → B | ϵ, B → b} A → ϵ : P:{ S’ → S, S → ASA | AS | SA | S | aB | a, A → B, B → b} 3. Remove unit production S → S : Recursive so remove from P:{ S’ → S, S → ASA | AS | SA | aB | a, A → B, B → b} A → B : P:{ S’ → S, S → ASA | AS | SA | aB | a, A → b, B → b} S’ → S : P:{ S’ → ASA | AS | SA | aB | a, S → ASA | AS | SA | aB | a, A → b, B → b} 4. Production more than 2 not terminals in RHS S’ → ASA : S’ → AX, X →SA, Similarly for S → ASA. So P: {S’ → AX | AS | SA | aB | a, S → AX | AS | SA | aB | a, A → b, X → SA, B → b} 5. RHS should have only one Terminal S’→aB : S’ → YB, Y → a. So P: {S’ → AX | AS | SA | YB | a, S → AX | AS | SA | YB | a, A → b, X → SA, Y → b B → b} 6. Hence given grammar is in CNF	6	3	4
	b) Consider the context-free grammar whose rules are $S \rightarrow SS \mid Sa \mid a$. Convert into equivalent grammar in GNF.	4		

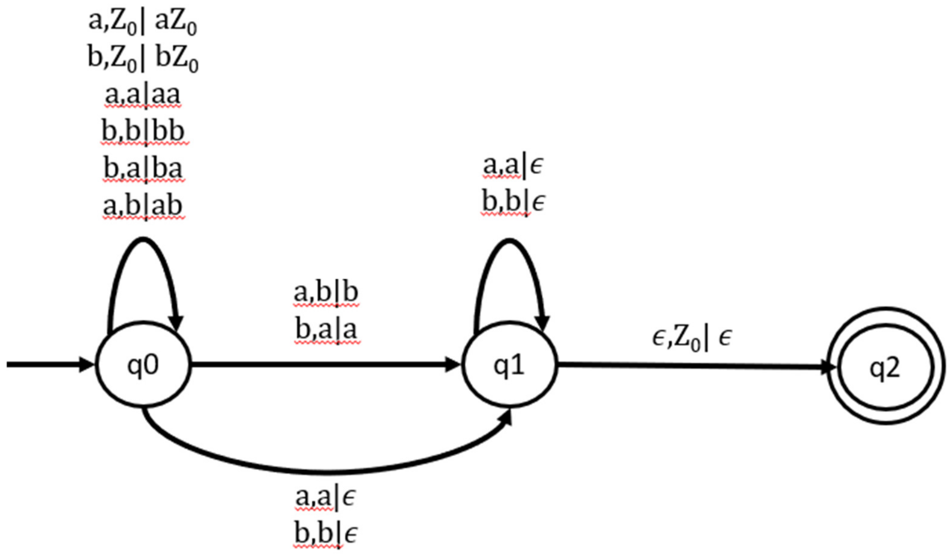
<table>
<tr><td></td><td>Ans:
S→SS|Sa|a
1. Remove Left Recursion S → SS substitute S → a then S → aS
$P: \{S \rightarrow aS, \quad S \rightarrow Sa, \quad S \rightarrow a\}$
2. Remove Left Recurison S → Sa substitute S → a
$P: \{S \rightarrow aS, \quad S \rightarrow aa, \quad S \rightarrow a\}$
Given grammar has RHS starts with a terminal. So it is now in GNF.</td><td></td><td></td><td></td></tr>
<tr><td>4.</td><td>Given the context-free grammar $\boldsymbol{G}: \boldsymbol{S} \rightarrow \boldsymbol{XY}, \boldsymbol{X} \rightarrow \boldsymbol{YY} \mid \mathbf{0}, \boldsymbol{Y} \rightarrow \boldsymbol{XY} \mid \mathbf{1}$. Use the CYK algorithm to determine whether the strings 00110 is in the language generated by the given grammar.

Ans:
<table>
<tr><td>-</td></tr>
<tr><td>-</td><td>-</td></tr>
<tr><td>-</td><td>-</td><td>-</td></tr>
<tr><td>-</td><td>S</td><td>X</td><td>-</td></tr>
<tr><td>X</td><td>X</td><td>Y</td><td>Y</td><td>X</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
</table>

String is not accepted.</td><td>10</td><td>3</td><td>3</td></tr>
<tr><td>5.</td><td>a) Consider the PDA for accepting palindrome strings over $\{\boldsymbol{a}, \boldsymbol{b}\}$. For the below incomplete trace sequence (instantaneous description – ID) and moves for the given string $\boldsymbol{ababababa}$, identify the required state or partial input string or stack content in the “fill in the blanks” entry. Note $\boldsymbol{q_0}$ is the start state and $\boldsymbol{q_f}$ is the final state, $\boldsymbol{Z_0}$ is the start stack symbol.

$(q_0, ababababa, Z_0)$
$\vdash (q_0, babababa, aZ_0)$
$\vdash (q_0, abababa, baZ_0)$
$\vdash (q_0, bababa, -)$
$\vdash (q_0, ababa, -)$
$\vdash (-, baba, babaZ_0)$
$\vdash (q_1, -, abaZ_0)$
$\vdash (q_1, ba, baZ_0)$
$\vdash (q_1, a, aZ_0)$
$\vdash (-, \epsilon, Z_0)$
$\vdash (q_f, \epsilon, Z_0)$

Ans:</td><td>5</td><td>1</td><td>4</td></tr>
</table>



Trace “ababababa”:

$$\delta(q_0, ababababa, Z_0) \vdash (q_0, babababa, aZ_0)$$
$$\vdash (q_0, abababa, baZ_0)$$
$$\vdash (q_0, bababa, abaZ_0)$$
$$\vdash (q_0, ababa, babaZ_0)$$
$$\vdash (q_1, baba, babaZ_0)$$
$$\vdash (q_1, aba, abaZ_0)$$
$$\vdash (q_1, ba, baZ_0)$$
$$\vdash (q_1, a, aZ_0)$$
$$\vdash (q_1, \epsilon, Z_0)$$
$$\vdash (q_2, \epsilon)$$

String is accepted

b) Consider the language $L = \{a^{2n}c^m b^n \mid n, m \geq 1\}$. The objective is to construct a (pushdown automata) PDA for this language and below are the partial transition rules. The main idea is to push a **0** in the stack whenever ***a*** is read and when ***b***s are read, they need to be cancelled against **0**s such a way that for every one ***b*** is read, two **0**s need to be cancelled out in the stack. You are expected to fill-in the remaining rules which are left un-attempted (denoted with - in the rules) and complete the rules such that the constructed PDA will accept the given language ***L***. FYI, the transition function **δ** of the PDA is defined as: $\boldsymbol{\delta : Q \times \Sigma \cup \{\lambda\} \times \Gamma \rightarrow finite\ subset\ of\ Q \times \Gamma^*}$. You are allowed to use only one (new) additional state which is $\boldsymbol{q_f}$, the final state. Write only incomplete rules 2,4,6,7,8 for answering the question. ***Z*** is the start stack symbol. 5

a. $\delta(q0, a, Z) = (q1, 0Z)$
b. $\delta(q1, a, -) = (q1, 00)$

	c. $\delta(q1, c, 0) = (q2, 0)$ d. $\delta(q2, c, -) = (-, -)$ e. $\delta(q2, b, 0) = (q3, \lambda)$ f. $\delta(q3, -, -) = (-, \lambda)$ g. $\delta(q4, b, -) = (-, -)$ a. $\delta(q4, \lambda, -) = (-, -)$ Ans: **a)** $\delta(q0, a, z) = (q1, 0z)$ **b)** $\delta(q1, a, 0) = (q1, 00)$ **c)** $\delta(q1, c, 0) = (q2, 0)$ **d)** $\delta(q2, c, 0) = (q2, 0)$ **e)** $\delta(q2, b, 0) = (q3, \lambda)$ **f)** $\delta(q3, \lambda, 0) = (q4, \lambda)$ **g)** $\delta(q4, b, 0) = (q3, \lambda)$ **h)** $\delta(q4, \lambda, z) = (qf, z)$			

****