



VIT

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

REG.NO.:

SLOT:

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026**

Programme Name & Branch : B.Tech. & CSE
Course Code and Course Name : BCSE304L & Theory of Computation
Faculty Name(s) : Common to all
Class Number(s) : Common to all
Date of Examination : 20/03/2026
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)
- Course Outcomes: (Type the CO statements covered in this question paper. Use the CO number as per the syllabus copy)
CO2: Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata.
CO3: Identify limitations of some computational models and possible methods of proving them.

| Q. No | Question | M | CO | BL | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------|--|------|-------------|-------------|-------|---|---|---|---|------|---|---|---|------|---|---|---|---|------|---|---|---|---|------|---|------|---|---|---|---|---|---|
| 1. | a) Construct the equivalent Finite Automaton (FA) corresponding to the given grammar by clearly specifying the set of states, input alphabet, start state, final states, and transition function. Finally, draw the state-transition diagram for the constructed automaton. $S \rightarrow aA / bB / cC / a$ $A \rightarrow aS / bD / cA / a$ $B \rightarrow bS / aC / cB / b$ $C \rightarrow aD / bA / cS / c$ $D \rightarrow aB / bC / a$ | 5 | C O 2 | B L 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | b) Use the Pumping Lemma to determine whether the following language is regular or not, and show all the necessary steps to justify your conclusion. $L = \{0^m 1^n 2^q \mid m, n, q \geq 1 \text{ and } m + q = 2n\}$ | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KEY | 1.a) Input Alphabet $\Sigma = \{a, b, c\}$, Start State = S Final State = {F} | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th>State</th> <th>a</th> <th>b</th> <th>c</th> </tr> </thead> <tbody> <tr> <td>S</td> <td>A, F</td> <td>B</td> <td>C</td> </tr> <tr> <td>A</td> <td>S, F</td> <td>D</td> <td>A</td> </tr> <tr> <td>B</td> <td>C</td> <td>S, F</td> <td>B</td> </tr> <tr> <td>C</td> <td>D</td> <td>A</td> <td>S, F</td> </tr> <tr> <td>D</td> <td>B, F</td> <td>C</td> <td>-</td> </tr> <tr> <td>F</td> <td>-</td> <td>-</td> <td>-</td> </tr> </tbody> </table> | | | | State | a | b | c | S | A, F | B | C | A | S, F | D | A | B | C | S, F | B | C | D | A | S, F | D | B, F | C | - | F | - | - | - |
| | State | a | b | c | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | S | A, F | B | C | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | A | S, F | D | A | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | B | C | S, F | B | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | C | D | A | S, F | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D | B, F | C | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F | - | - | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

| | | | | |
|----|---|----|-------------|-------------|
| | <p>1.b) Step 1: Assume L is Regular Assume that L is a regular language. Let the pumping length be $p = 8$. According to the Pumping Lemma, any string $w \in L$ with $w \geq p$ can be written as $w = xyz$ such that: 1. $xy \leq p$ 2. $y > 0$ 3. $xy^i z \in L$ for all $i \geq 0$</p> <p>Step 2: Choose a String in L Choose values satisfying the condition $m + q = 2n$. Let $m = 4, n = 4, q = 4$ Check the condition: $m + q = 4 + 4 = 8$ $2n = 2 \times 4 = 8$ Thus the string belongs to L. $w = 0000 1111 2222$ Length = $12 \geq 8$</p> <p>Step 3: Divide the String Since $xy \leq 8$, the substring y lies in the first eight symbols: 00001111 Assume $x = 000, y = 0, z = 011112222$ where $y > 0$.</p> <p>Step 4: Pump the String Take $i = 2$ xy^2z $= 00000 1111 2222$ New counts: $m' = 5, n' = 4, q' = 4$</p> <p>Step 5: Check the Language Condition Original condition: $m + q = 2n$ For the pumped string: $m' + q' = 5 + 4 = 9$ $2n' = 8$ Since $9 \neq 8$ $xy^2z \notin L$ Thus, the condition does not hold.</p> | | | |
| 2. | Consider the context-free grammar G in Chomsky Normal Form (CNF) given by the following productions: $S \rightarrow AB \mid BC \mid CD$ | 10 | C O 3 | B L 3 |



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--------------|---|--------------|-------------|-------------|----------|--|--|--------------|------|------|--|--|--|--------------|---|------|------|--|--|--------------|---|------|------|---|--|--------------|------|------|------|------|------|--|----------|----------|----------|----------|----------|--------------|---|--|--|--|--|--------------|---|------|--|--|--|--------------|------|---|------|--|--|--------------|------|------|------|------|--|--|----------|----------|----------|----------|--|--|--|--|
| | $B \rightarrow CC \mid b$ $C \rightarrow AB \mid AD \mid a$ $D \rightarrow BB \mid b$ <p>Using the CYK algorithm, determine whether the following strings belong to the language generated by the grammar G:</p> <ol style="list-style-type: none"> 1. bbabb 2. baab <p>Construct the CYK parsing table step by step for both strings and identify which string is generated by the grammar and which is not. Clearly show all intermediate computations used to conclude.</p> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KEY | <p>i) String 1: bbabb</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">L = 5</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>L = 4</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td></td> <td></td> <td></td> </tr> <tr> <td>L = 3</td> <td style="border: 1px solid black; padding: 5px;">A</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td></td> <td></td> </tr> <tr> <td>L = 2</td> <td style="border: 1px solid black; padding: 5px;">D</td> <td style="border: 1px solid black; padding: 5px;">A, S</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td style="border: 1px solid black; padding: 5px;">D</td> <td></td> </tr> <tr> <td>L = 1</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td style="border: 1px solid black; padding: 5px;">A, C</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> </tr> <tr> <td></td> <td style="text-align: center;">b</td> <td style="text-align: center;">b</td> <td style="text-align: center;">a</td> <td style="text-align: center;">b</td> <td style="text-align: center;">b</td> </tr> </table> <p>Since S is present in the top cell, the string babba belongs to the grammar</p> <p>ii) String 2: baab</p> <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;">L = 5</td> <td style="border: 1px solid black; padding: 5px;">D</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>L = 4</td> <td style="border: 1px solid black; padding: 5px;">D</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td></td> <td></td> <td></td> </tr> <tr> <td>L = 3</td> <td style="border: 1px solid black; padding: 5px;">A, S</td> <td style="border: 1px solid black; padding: 5px;">B</td> <td style="border: 1px solid black; padding: 5px;">S, C</td> <td></td> <td></td> </tr> <tr> <td>L = 2</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td style="border: 1px solid black; padding: 5px;">A, C</td> <td style="border: 1px solid black; padding: 5px;">A, C</td> <td style="border: 1px solid black; padding: 5px;">B, D</td> <td></td> </tr> <tr> <td></td> <td style="text-align: center;">b</td> <td style="text-align: center;">a</td> <td style="text-align: center;">a</td> <td style="text-align: center;">b</td> <td></td> </tr> </table> <p>Since S is not present in the top cell, the string baab is not belongs to the grammar</p> | L = 5 | S, C | | | | | L = 4 | S, C | S, C | | | | L = 3 | A | S, C | S, C | | | L = 2 | D | A, S | S, C | D | | L = 1 | B, D | B, D | A, C | B, D | B, D | | b | b | a | b | b | L = 5 | D | | | | | L = 4 | D | B, D | | | | L = 3 | A, S | B | S, C | | | L = 2 | B, D | A, C | A, C | B, D | | | b | a | a | b | | | | |
| L = 5 | S, C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 4 | S, C | S, C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 3 | A | S, C | S, C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 2 | D | A, S | S, C | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 1 | B, D | B, D | A, C | B, D | B, D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | b | b | a | b | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 5 | D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 4 | D | B, D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 3 | A, S | B | S, C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| L = 2 | B, D | A, C | A, C | B, D | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | b | a | a | b | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | <p>a) Let $L = \{ a^n b^m c^p \mid n \geq 0, m \geq 0, p \geq 0 \}$ be a language over the alphabet $\Sigma = \{a, b, c\}$. Let $h: \Sigma \rightarrow \{0,1\}^*$ be a homomorphism defined as $h(a) = 01$, $h(b) = 10$, and $h(c) = 00$. Determine the homomorphic image $h(L)$ and express the resulting language in terms of patterns over the alphabet $\{0, 1\}$. Further, analyze whether the language $h(L)$ is a regular language, and justify your answer.</p> | 5 | C O 2 | B L 4 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | <p>b) A context-free grammar may contain both null (ϵ) productions and unit productions. In such cases, specify the correct order in which these productions should be eliminated during grammar simplification, and justify why this order is</p> | 5 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

| | | | | |
|-------------|---|--|--|--|
| | <p>appropriate. Then consider the following grammar and simplify it by applying the transformations in the correct order:</p> $S \rightarrow AB / C$ $A \rightarrow aA / \epsilon$ $B \rightarrow bB / A$ $C \rightarrow B / c$ | | | |
| K e y | <p>3.a)</p> <p>Step 1: Apply Homomorphism A general string in L is $a^n b^m c^p$</p> <p>Applying the homomorphism h: $h(a^n b^m c^p) = h(a)^n h(b)^m h(c)^p$</p> <p>Substitute the mappings: $= (01)^n (10)^m (00)^p$</p> <p>Step 2: Homomorphic Image Thus the resulting language is $h(L) = \{(01)^n (10)^m (00)^p \mid n \geq 0, m \geq 0, p \geq 0\}$</p> <p>Step 3: Pattern Description The strings consist of:</p> <ul style="list-style-type: none"> • n repetitions of 01 • followed by m repetitions of 10 • followed by p repetitions of 00 <p>Example strings: ϵ 01 0110 011010 01101000 0110101000</p> <p>Step 4: Regularity Analysis The language can be written as the regular expression $(01)^* (10)^* (00)^*$</p> <p>Each part:</p> <ul style="list-style-type: none"> • $(01)^*$ is regular • $(10)^*$ is regular • $(00)^*$ is regular <p>The concatenation of regular languages is also regular. Therefore $h(L)$</p> <p>can be recognized by a finite automaton.</p> <p>Final Conclusion $h(L) = \{(01)^n (10)^m (00)^p \mid n, m, p \geq 0\}$</p> <p>and</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">$h(L)$ is a regular language</div> <p>because it can be expressed using the regular expression</p> | | | |



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026**

| | | | | | |
|-------------|--|--|-------------|-------------|--|
| | <p>3.b) Step 1: Correct Order of Removal When a grammar contains both null (ϵ) productions and unit productions, the correct order of elimination is:</p> <ol style="list-style-type: none"> 1. Remove null (ϵ) productions first 2. Remove unit productions next <p>Justification Removing ϵ-productions first is necessary because eliminating ϵ may create new unit productions. If unit productions were removed first, new ones could reappear after eliminating ϵ-productions, making the simplification incomplete. Therefore, ϵ-productions are removed first, followed by unit productions.</p> <p>Step 2: Remove ϵ-Productions</p> <p>$S \rightarrow AB \mid B \mid C$ $A \rightarrow aA \mid a$ $B \rightarrow bB \mid A$ $C \rightarrow B \mid c$</p> <p>Step 4: Remove Unit Productions Unit productions are:</p> <p>$S \rightarrow B$ $S \rightarrow C$ $B \rightarrow A$ $C \rightarrow B$</p> <p>$S \rightarrow AB \mid bB \mid aA \mid a \mid c$ $A \rightarrow aA \mid a$ $B \rightarrow bB \mid aA \mid a$ $C \rightarrow bB \mid aA \mid a \mid c$</p> | | | | |
| 4. | <p>Consider the following context-free grammar (CFG) G:</p> $S \rightarrow SA \mid aB \mid b$ $A \rightarrow cA \mid d$ $B \rightarrow bB \mid c$ <p>a). Convert the above grammar into Chomsky Normal Form (CNF) by performing the necessary transformations.</p> <p>b). Convert the resulting grammar (converted in 4.a) into Greibach Normal Form (GNF), and show all intermediate steps clearly.</p> | 10 | C 0 3 | B L 3 | |
| K E Y | <p>CNF Form</p> $S \rightarrow SA \mid XB \mid b$ $A \rightarrow ZA \mid d$ $B \rightarrow YB \mid c$ $X \rightarrow a$ $Y \rightarrow b$ $Z \rightarrow c$ | <p>Case $i = j$(Left recursion)</p> $A_1 \rightarrow A_1A_2 \mid aA_4 \mid b$ <p>Here</p> $\beta = A_2$ $\alpha = aA_4, b$ $B \rightarrow bB \mid c$ <p>Introduce new variable B_1</p> $B_1 \rightarrow A_2B_1 \mid A_2$ | | | |



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026**

| | | | | | |
|-----|--|---|-------------|-------------|--|
| | <p>GNF FORM</p> <p>$A_1 = S$ $A_2 = A$ $A_3 = X$ $A_4 = B$ $A_5 = Z$ $A_6 = Y$</p> <p>Step 1: Modified Grammar:</p> <p>$A_1 \rightarrow A_1A_2 \mid A_3A_4 \mid b$ $A_2 \rightarrow A_5A_2 \mid d$ $A_4 \rightarrow A_6A_4 \mid c$ $A_3 \rightarrow a$ $A_5 \rightarrow c$ $A_6 \rightarrow b$</p> <p>Step 2: Apply GNF Conversion Rules</p> <p>Case $i < j$</p> <p>$A_1 \rightarrow A_3A_4$ Since $A_3 \rightarrow a$ $A_1 \rightarrow aA_4$</p> | <p>Rewrite</p> <p>$A_1 \rightarrow aA_4B_1 \mid bB_1 \mid aA_4 \mid b$</p> <p>Convert B_1 to GNF</p> <p>Since</p> <p>$A_2 \rightarrow cA_2 \mid d$</p> <p>Substitute</p> <p>$B_1 \rightarrow cA_2B_1 \mid dB_1 \mid cA_2 \mid d$</p> <p>Remaining Productions</p> <p>$A_2 \rightarrow cA_2 \mid d$ $A_4 \rightarrow bA_4 \mid c$</p> <p>Final Grammar in GNF</p> <p>$S \rightarrow aBB_1 \mid bB_1 \mid aB \mid b$ $B_1 \rightarrow cAB_1 \mid dB_1 \mid cA \mid d$ $A \rightarrow cA \mid d$</p> | | | |
| 5. | <p>Give a Pushdown Automaton (PDA) that recognizes a language over the alphabet $\Sigma = \{a, b, c, d\}$ in which the strings consist of one or more a's followed by b's, then c's, and finally d's, such that the number of b's is exactly twice the number of a's and the number of d's is exactly three times the number of c's. For example, the strings abbcddd, aabbbccddd, aaabbbbbccdddddd belong to the language, whereas abbd, abbcdd, aabbbccddd, abbbccddd do not belong to the language.</p> <p>a) Construct a Pushdown Automaton (PDA) that recognizes this pattern of language.</p> <p>b) Illustrate its operation by deriving the sequence of Instantaneous Descriptions (ID) showing the step-by-step stack configurations for the input strings</p> <p>i) aabbbccdddddd</p> <p>ii) abbcddd</p> | 10 | C O 1 | B L 3 | |
| KEY | <p>Language Description</p> <p>Alphabet $\Sigma = \{a, b, c, d\}$ Strings have the form : $a^n b^{2n} c^m d^{3m}$ where $n, m \geq 1$</p> <p>Transition Function</p> <p>Processing a's, push two A's for each a</p> <p>$\delta(q_0, a, Z_0) = (q_0, AAZ_0)$ $\delta(q_0, a, A) = (q_0, AAA)$</p> | <p>(i) String 1: aabbbccdddddd</p> <p>$(q_0, aabbbccdddddd, Z_0)$ $(q_0, abbbccdddddd, AAZ_0)$ $(q_0, bbbccdddddd, AAAAZ_0)$ $(q_1, bbbccdddddd, AAAZ_0)$ $(q_1, bbccdddddd, AAZ_0)$ $(q_1, bccdddddd, AZ_0)$ $(q_1, cccdddddd, Z_0)$ $(q_2, cccdddddd, CCCZ_0)$ $(q_2, dccccdd, CCCCCZ_0)$</p> | | | |



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
CONTINUOUS ASSESSMENT TEST - II
WINTER SEMESTER 2025-2026

| | |
|---|---|
| <p>Processing b's: Pop one A for each b</p> $\delta(q_0, b, A) = (q_1, \epsilon)$ $\delta(q_1, b, A) = (q_1, \epsilon)$ <p>Processing c's: Push three C's for each c</p> $\delta(q_1, c, Z_0) = (q_2, CCCZ_0)$ $\delta(q_1, c, C) = (q_2, CCCC)$ <p>Processing d's: Pop one C for each d</p> $\delta(q_2, d, C) = (q_3, \epsilon)$ $\delta(q_3, d, C) = (q_3, \epsilon)$ <p>Accept</p> $\delta(q_3, \epsilon, Z_0) = (q_3, \epsilon)$ | $(q_3, dddd, CCCCZ_0)$ $(q_3, dddd, CCCCZ_0)$ $(q_3, ddd, CCCZ_0)$ (q_3, dd, CCZ_0) (q_3, d, CZ_0) (q_3, ϵ, Z_0) $(q_3, \epsilon, \epsilon)$ <p>Stack empty → Accepted</p> <p>(ii) String 2: abbbcdddd</p> $(q_0, abbbcdddd, Z_0)$ $(q_0, bbbcdddd, AAZ_0)$ $(q_1, bbcdddd, AZ_0)$ $(q_1, bcdddd, Z_0)$ <p>Next symbol b, but stack top is Z₀</p> <p>No valid transition. Thus, computation halts → Rejected</p> |
|---|---|
