

**Continuous Assessment Test (CAT) – I - August 2024**

|               |  |            |                           |
|---------------|--|------------|---------------------------|
| Programme     | : B.Tech CSE, AI ML, CPS,AIR, ECM  | Semester   | : Fall Semester 2024-2025 |
| Course Code   | : BCSE307L   | Class Nos: | : CH2024250101301         |
| Course Title: | : Compiler Design  |            | : CH2024250101295         |
|               |  |            | : CH2024250101299         |
|               |  |            | : CH2024250101297         |
| Faculty(s)    | : Dr. SUGANYA R<br>Dr. NAGARAJ S V<br>Dr. SELVAM D<br>Dr. SURESHKUMAR WI | Slot       | : G2+TG2                  |
| Time          | : 90 Minutes   | Max. Marks | : 50                      |

*Answer all the Questions*

**General Instructions:**

- Write only your registration number on the question paper in the box provided and do not write other information.
- Use statistical tables supplied from the exam cell as necessary
- Use graph sheets supplied from the exam cell as necessary
- Only non-programmable calculator without storage is permitted

| Q. No. | Sub-division | Question  | Marks |
|--------|--------------|---|-------|
| 1.     | a)           | Provide a detailed explanation of the compilation process using the given example to demonstrate the output of each phase of compilation for the input:<br>$sum = sum + i * 10;$ (7marks)   | 15    |
|        | b)           | Construct an non deterministic Finite Automata (NFA) for the following expression $(a b)^*a(a b)^*ab(a b)$ (8marks)   |       |
| 2.     |              | Convert the regular expression " $(aa b)^*a(ab c)^*$ " into a minimized deterministic finite automaton using the direct method while indicating all steps with appropriate explanation during the conversion process.   | 10    |
| 3.     |              | Construct an LL(1) parsing table for the given grammar :<br>G : A -> BDC<br>D -> +A   $\epsilon$<br>B -> (A)   aFC<br>F -> *B   $\epsilon$<br>C -> $\epsilon$<br>Check whether the given input string " $(a+a*a)+(a*a)$ " is acceptable or not by using the constructed LL(1) parsing table.  | 15    |
| 4.     |              | Construct Operator Precedence parser table for the given grammar<br>S -> WbS   W<br>W -> L*W   L<br>L -> a<br>With the help of the Operator Precedence parser table, parse the input string " $a*aba$ " based on operator precedence parsing technique.<br>Note: Non Terminals -> S-Sentence, W-word, L-letter<br>Terminals -> b-blank, a-identifier, and * | 10    |