B.Tech Third Year Mid-Semester Examination
Department: Computer Science and Engineering
Course Name: Formal Language and Automata Theory
Course Code: CS 303
Full Marks-40 **Time: 2 hours**
Answer ALL the Questions

Make reasonable assumptions as and whenever necessary. You can answer the questions in any order. However answers to all the parts of any particular question should appear together. The notations carry the usual meanings.

(Q1). Construct DFA for each of the following languages. The underlying alphabet is $\Sigma = \{a,b\}$.

(a). { w | w begins with ab but does not end with ab}

(b). The set L of strings in which every a is immediately followed by at least two b's.

(c). {w | w has exactly one occurrence of the substring ab and does not end with a}

2+3+2

(Q2). (a). Construct a DFA for the regular expression (1 | 0) 1* (1 | 0) (construct syntax tree; compute *firstpos*, *lastpos*, *followpos* etc).

(Q2). (b). Determine the minimal number of states in the DFA corresponding to the regular expression (a + b)* (ba). **5+5**

(Q3). Construct a PDA for the following language:

$L = \{ wcw^R \mid w \in \{a,b\}^*, C \text{ is a marker}\}$
Make usual assumptions as and whenever required. **8**

(Q4). Design a PDA that recognises balanced parenthesis (*including the nested parenthesis*) by empty stack. Convert this to a final-state PDA (i.e. PDA accepting in final state). Explain each step with examples, whenever required.

8

(Q5). (a). Consider the CFG with following productions

s->aSbScS | aScSbS | bSaScS | bScSaS | cSaSbS | cSbSaS | ε

Does this grammar generate the language $\{x \in \{a,b,c\}^* \mid n_a(x) = n_b(x) = n_c(x)$, $n_a(x)$, $n_b(x)$ and $n_c(x)$ denote number of a, b and c, respectively, in the given string x}? Justify your claim with sufficient evidences and steps.

(Q5). (b). Show with appropriate examples, and their derivation trees that the CFG with productions

S→ a | Sa | bSS | SSb| SbS is ambiguous.

4+3