2017/ODD/08/24/MCS–701 (C)/ CS–701 (O)/MS–101 (C)/128

PG Odd Semester (CBCS) Exam., December—2017

COMPUTER SCIENCE

(7th Semester) / (1st Semester)

Course No. : MCS–701 (C)/MS–101 (C)/CS–701 (O)

(Theory of Computation)

Full Marks : 75

Pass Marks : 30 (for CBCS)
26 (for Old Course)

Time : 3 hours

The figures in the margin indicate full marks for the questions

Answer **five** questions, taking **one** from each Unit

UNIT—I

1. *(a)* Prove by induction that every tree has one more node than its edges. 5

(b) Design a DFA to accept the set of strings that either begin or end (or both) with 01 over the alphabet $\{0, 1\}$. Construct both the transition diagram and the transition table. 5

(c) Construct transition diagram and transition table for a non-deterministic finite automata (NFA) to accept the set of strings over the alphabet $\{0, 1, 2, \cdots, 9\}$ such that the final digit has appeared before. 5

2. *(a)* Prove by induction that every expression has an equal number of left and right parentheses. 5

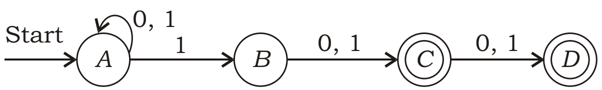
(b) Design a DFA to accept the set of all strings with three consecutive 0s not necessarily at the end over the alphabet $\{0, 1\}$. Construct both the transition diagram and the transition table. 5

(c) What is epsilon-transition? Design an -NFA to accept decimal numbers. Construct both the transition diagram and the transition table. 5

UNIT—II

3. *(a)* Write a regular expression for the set of strings over alphabet $\{a, b, c\}$ containing at least one a and at least one b. 2

(b) Convert the following automata (NFA) which accepts all strings of 0's and 1's such that either the second or the third position from the end has a 1 to a regular expression using state elimination : 6



(c) *(i)* Explain the pumping lemma for regular languages. 4

(ii) Show that the language L_{pr} consisting of all strings of 1's and whose length is a prime is not a regular language. 3

4. *(a)* Write a regular expression for the set of all strings of 0's and 1's such that every pair of adjacent 0's appears before any pair of adjacent 1's. 2

(b) Convert the regular expression $(0\quad 1)\ 1(0\quad 1)$ to an -NFA. Explain the steps of conversion. 6

(c) Explain the table filling algorithm to test the equivalence of the states of a DFA with one example. 7

UNIT—III

5. *(a)* Define context-free language (CFL). Prove that a string w in $\{0, 1\}$ is in $L(G_{\text{pal}})$ if and only if it is a palindrome. Here, G_{pal} is the context-free grammar for palindromes and is given by—

(i) $P \quad ;$

(ii) $P \quad 0;$

(iii) $P \quad 1;$

(iv) $P \quad 0P0;$

(v) $P \quad 1P1.$ 6

(b) What is pushdown automata? Design a pushdown automata to accept the language L_{wwr} such that

$$L_{wwr} \quad \{ww^R | w \text{ is in } (0 \quad 1) \ \}$$

where L_{wwr} represents even length palindromes over alphabet $\{0, 1\}$. Construct the transition diagram for the PDA. 6

(c) State and explain the pumping lemma for the context-free languages. 3

6. *(a)* Design context-free grammar for the following language constituted by the set $\{0^n 1^n | n \quad 1\}$, which is the set of all strings of one or more 0's followed by an equal number of 1's. 6

(b) What is Chomsky normal form of a grammar? Convert the following grammar to Chomsky normal form : 6

$$S \quad ASB\,|$$
$$A \quad aAS\,|\,a$$
$$B \quad SbS\,|\,A\,|\,bb$$

(c) Let w be a terminal string and it is the yield of a parse tree constructed according to a Chomsky normal form grammar $G \quad (V, T, P, S)$. If the length of the shortest path is n, then prove that $|w| \quad 2^{n \quad 1}$. 3

UNIT—IV

7. (a) What is a Turing machine? Explain the structure of a Turing machine and explain how a move is executed by a Turing machine. 5

(b) Explain instantaneous description (IDs) of a Turing machine. 5

(c) Explain a programming technique for a Turing machine. 5

8. (a) State the formal definition of a Turing machine. 3

(b) Consider a function *monus* or proper subtraction (denoted by $\cdot$) defined by

$$m \cdot n \quad \begin{matrix} m \quad n, & \text{if} \quad m \quad n \\ 0, & \text{if} \quad m \quad n \end{matrix}$$

Design a Turing machine to compute the function ‘proper subtraction’. Construct both the transition table and the transition diagram. 7

(c) Discuss the interrelationships among recursive, recursively enumerable and non-recursively enumerable languages. 5

UNIT—V

9. (a) What is universal Turing machine? Explain the structure and operation of a universal Turing machine designed based on universal language. 8

(b) What is multitape Turing machine? Discuss the equivalence of one-tape and multitape Turing machines. 7

10. (a) What do you mean by halting of a Turing machine? Illustrate the significance of halting with an example. 5

(b) What is intractability? Discuss the *P* and *NP* classes of problem solvable in polynomial time by deterministic and non-deterministic Turing machines respectively by taking suitable examples. 10

★ ★ ★