LIB
1.7.13 AN

Reg. No. : | | | | | | | | | | | | |

Question Paper Code : 65021

5 Year M.Sc. DEGREE EXAMINATION, MAY/JUNE 2013.

Fifth Semester

Information Technology

XCS 352/10677 SW 505 — THEORY OF COMPUTATION

(Common to 5 Year M.Sc. Computer Technology and 5 Years M.Sc. Software Engineering)

(Regulation 2003/2010)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Write the difference between DFA and NFA.
2. Give a DFA which accepts even number of '*a*' and '*b*'.
3. Write CFG for $L = \{w \in (a,b)^* | n_a(w) = n_b(w)\}$.
4. What is ambiguous grammar?
5. Write the applications of pumping lemma.
6. What is Turing acceptable language?
7. What is context sensitive grammar? Give example.
8. List the different types of Turing machine.
9. Give example for unsolvable problem.
10. What is meant by polynomial time reducible problem?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Convert the regular expression $(ab|\ ac)^* b$ into NFA-$\in$ and then transform it into DFA. (10)

(ii) Write the applications of automata. (6)

Or

(b) (i) Write an algorithm to convert NFA into DFA. (6)

(ii) Construct DFA for language to recognize strings over $\Sigma = \{a, b, c\}$ in which no string contain a substring '*aa*' or '*bb*' or '*cc*' (i.e no symbols ever follow itself). Generate regular grammar from the constructed DFA. (10)

12. (a) (i) Construct PDA 'M' to accept the strings of odd and even palindrome over the alphabets $\Sigma = \{a, b\}$. Also find ID for the input string w = babab from 'M'. (12)

(ii) Prove that $L = \{a^n b^n c^n | n > 0\}$ is not CFG. (4)

Or

(b) (i) Construct PDA to accept the strings of language $L = \{a^n b^n | n > 0, w = (a, b)*\}$. (8)

(ii) Prove that the equivalence of PDA and CEG. (8)

13. (a) Construct TM to perform multiplication over $\Sigma = \{0\}$ where $L = \{0^{mn} | m, n, > 0\}$. (16)

Or

(b) (i) Construct TM for $L = \{a^n b^n c^n | n > 0\}$ without using any special tape symbol other than blank symbol 'B'. (12)

(ii) Write short notes on random access turing machine. (4)

14. (a) (i) Describe halting problem with suitable example. (8)

(ii) How TM determines that the problem is solvable or unsolvable? (8)

Or

(b) (i) What is meant by undecidable problems? Give some example. (8)

(ii) Say True or False and justify your answer – "Every computable problem can be described using TM". (8)

15. (a) Describe P and NP problems in detail. (16)

Or

(b) (i) How to you determine that the problem is NP hard or NP complete? (10)

(ii) What is the significant of classifying the problems as P and NP classes? (6)
