Pokhara University

|  |  |  |
| --- | --- | --- |
| Level: Bachelor | Semester – Fall | Year : 2011 |
| Programme: BE | | Full Marks: 100 |
| Course: Theory of Computation | | Pass Marks: 45 |
| Time : 3hrs. |

|  |
| --- |
| *Candidates are required to give their answers in their own words as far as practicable.* |
| *The figures in the margin indicate full marks.* |
| Attempt all the questions. |

|  |  |  |
| --- | --- | --- |
|  | 1. Define DFA and NFA. 2. Convert the NFA M=({q0,q1,q2,q3},{0,1}, δ, q0,{ q3}) to its equivalent DFA. δ is given by      1. Find the regular expression for the following NFA. | 5  5  5 |
|  | 1. When the grammar is ambiguous? Show that the given grammar is ambiguous: S->aB/ab, A->aAB/a, B->ABb/b. 2. State and prove the pumping lemma for regular sets.   **OR**   1. Show that regular sets are closed under intersection. 2. Show that the language L=anbncn / n ≥1 is not context free. | 7  8  7  8 |
|  | 1. Reduce the following grammar G to CNF. S→aAD, A→aB/bAB, B→b , D→d 2. Define push down automata. Construct a push down automata A equivalent to the following context free grammar: S->0BB, BK->0S|1S|0. Test whether 0100 is in N(A).   **OR**   1. Design a pushdown automata which accepts the language L = {W E{a,b)\*/w has equal number of a’s and b’s.} 2. What do you mean by pushdown automata? Discuss its power in accepting languages in relation to finite automata. | 7  8  8  7 |
|  | 1. What do you mean by recursive and recursively enumerable language? Give some properties of recursively enumerable language. 2. Design a Turing machine that accepts the language.   L = 1n2n3n / n ≥ 0 | 7  8 |
|  | 1. What are two computational complexities that matter in designing efficient program? 2. What are NP complete problems? Give some examples of NP complete problem. | 7  8 |
|  | 1. Prove that for every context free grammar there is an equivalent grammar G2 in Chomsky Normal From (CNF). 2. Write CFG which generates palindrome of binary numbers. | 8  7 |
|  | Write short notes on **any two:**   1. Universal Turing machine 2. Derivation tree 3. Decision algorithms for regular set | 2×5 |