INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date: .... FN/AN *Time:* 2 Hrs. *Full Marks* ... 45 ... *No. of Students* ... **82** ...
Autumn Semester, 2012-2013 Deptt. ... CSE ... Sub No. ... **CS41001** ...
B. Tech.(Hons.) Sub. Name ... **Theory of Computation** ...

Instructions : **There are four questions. Answer all of them precisely.**

1. (a) Prove that there is no *surjective (onto)* map from a set A to its power set $\mathcal{P}A$.

   (b) Justify that there is a *bijection* from $\mathbb{N}$ to the set R of all *regular languages* over $\{0,1\}$, $R = \{L \subseteq \{0,1\}^* : L \text{ is regular}\}$.

   (c) Use *diagonalisation* to prove that $L_\in = \{< M, x >: \text{Turing machine } M \text{ accepts } x\}$ is not *decidable.* $< M, x >$ is binary encoding of M and x.

   [4 + 4 + 4]

2. (a) Give the definition of (i) *Turing reducibility* and (ii) *mapping reducibility*, of a language L to a language L' ($L \leq_T L'$ and $L \leq_m L'$). Show that *mapping reducibility* implies *Turing reducibility.*

   (b) Prove that every *r.e.* language is *mapping reducible* to $L_{\neg\emptyset} = \{< M >: L(M) \neq \emptyset\}$. Do you have any conclusion about $L_{\neg\emptyset}$?

   (c) Show that $L_{\neg\emptyset} \leq_T L_\in = \{< M, x >: \text{Turing machine } M \text{ accepts } x\}$.

   [4 + 4 + 4]

3. (a) Define the class of *primitive recursive functions.*

   (b) Prove, starting from the definition, that the $(n+1)$-*ary* function f, defined as follows, is *primitive recursive*, whenever the $(n+1)$-*ary* function g is *primitive recursive.*

   $$f(x_1, \cdots, x_n, y) = \prod_{i=0}^{y} g(x_1, \cdots, x_n, i).$$

   (c) Consider the λ-*term* $A = \lambda tor \cdot r(otor)$. Demonstrate that a *fixed-point combinator* B can be formed using A. Show that for any λ-term F, $F(BF) = BF$.

   (d) How do you use your fixed-point combinator in (3c) to define a lambda term corresponding to "*add* m n", to add λ-numerals m and n. Assume that the successor (S), the test-for-zero (Z), and conditional λ-terms are available.

   [4 + 5 + 3 + 3]

4. (a) Prove that the following language is in Δ_2^0 by designing an oracle Turing machine as its decider with a suitable oracle language from Σ_1^0.

   $$L = \{< M, N, x >: M \text{ accepts } x \text{ and } N \text{ does not accept } x\}.$$

   (b) Define the class **NP** and prove that it is closed under language concatenation.

   [3 + 3]