INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
End-Autumn Semester 2018-19

Date of Examination : 20-11-2018 Session(FN/AN) AN Duration 3 hrs Total Marks 100
Subject No : CS41001 Subject Name : THEORY OF COMPUTATION
Department/Centre/School : Computer Science and Engineering
Specific charts, graph paper,log book etc. required No
Special Instructions (if any) Answer all questions. If required, you may make assumptions and state them upfront. No credit will be given to sketchy proofs and claims without proper reasoning.

1. Let $\sigma : \mathbb{N} \to \mathbb{N}$ be any total computable function. Prove that σ has infinitely many fixed points i.e., there are infinitely many e such that $L(\mathcal{M}_e) = L(\mathcal{M}_{\sigma(e)})$. Here, $\mathcal{M}_e$ is the Turing machine with description e. [15]

   Hint: Recursion theorem ensures there is atleast one fixed point for 'any' total computable function. If the set of fixed points is finite, does it contradict recursion theorem?

2. Show that the following functions/predicates are primitive recursive. You may assume that basic comparison predicates are primitive recursive. State any additional assumptions you make.

   (a) $\text{div}(x, y) = \begin{cases} 1 & \text{if } y \text{ divides } x \\ 0 & \text{otherwise} \end{cases}$. [3]

   (b) The greatest common divisor function gcd : $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined as

   $$\gcd(a, b) = \begin{cases} 0 & \text{if } ab = 0 \\ d & \text{if } d \text{ is the largest number } \leq x \text{ that divides both } a \text{ and } b \end{cases}$$

   [3]

   (c) The predicate coprime : $\mathbb{N} \times \mathbb{N} \to \{0, 1\}$ given by $\text{coprime}(a, b) = \begin{cases} 1 & \text{if } \gcd(a, b) = 1 \\ 0 & \text{otherwise} \end{cases}$. [3]

   (d) The Euler's totient function $\phi : \mathbb{N} \to \mathbb{N}$ is defined as

   $$\phi(n) = \begin{cases} n & \text{if } n \leq 1 \\ k & \text{if } k = |\{a \mid 0 < a < n \text{ and } a \text{ is coprime to } n\}| \end{cases}$$

   i.e., $\phi(n)$ is the number of integers $< n$ that are coprime to n. [6]

3. (a) A Hamiltonian path/cycle in an undirected graph is a path/cycle that visits every vertex in the graph exactly once. Let

   $$\begin{aligned} \mathsf{HAMPATH} &= \{\langle G \rangle : \text{ undirected graph } G \text{ contains a Hamiltonian path}\}, \\ \mathsf{HAMCYCLE} &= \{\langle G \rangle : \text{ undirected graph } G \text{ contains a Hamiltonian cycle}\}. \end{aligned}$$

   Assuming HAMPATH is **NP**-complete, prove that HAMCYCLE is **NP**-complete. [6]

   (b) Show that the travelling salesman problem (TSP) defined as

   $$\begin{aligned} \mathsf{TSP} = \{\langle G, w, k \rangle : & \text{ undirected graph } G = (V, E) \text{ with weight } w : E \to \mathbb{N} \text{ contains} \\ & \text{a cycle visiting every vertex exactly once with total weight } \leq k\}, \end{aligned}$$

   is **NP**-complete. Here, total weight is the sum of weights on the edges that form the cycle. [9]

4. The 0-1-KNAPSACK problem is defined as follows: Let $\{a_i\}_{i=1}^n, b$ be positive integers. The problem is to decide whether there is is a solution to $\sum_{i=1}^n a_i x_i = b$ with $x_i \in \{0,1\}$. It is known that 0-1-KNAPSACK is **NP**-complete.

   (a) Show that if we remove the constraints that $x_i \in \{0,1\}$ and allow x_i's to be arbitrary integers, then the problem is in **P**. **[6]**

   Hint: All integer linear combinations of $z_1, \ldots, z_n$ are multiples of $d = \gcd(z_1, \ldots, z_n)$.

   (b) Let 2INTSOLN be the problem of deciding, given a multivariate polynomial $p(x_1, x_2, \ldots, x_n)$ of degree 2 with integer coefficients, whether or not the equation $p(x_1, \ldots, x_n) = 0$ has an integer solution. (A multivariate polynomial of degree 2 in $x_1, \ldots, x_n$ consists of terms of the form $c \cdot x_i^a x_j^b$ where $c \in \mathbb{Z}$, $a, b \in \{0,1,2\}$ and $a + b \leq 2$.) Show that 2INTSOLN is **NP**-hard. **[10]**

   Hint: Think of a reduction from 0-1-KNAPSACK.

   (c) Can you say whether or not 2INTSOLN is **NP**-complete? Justify. **[4]**

5. (a) Is the union of two **NL**-complete languages is **NL**-complete? Justify your answer. **[3]**

   (b) If a language A is **PSPACE**-hard, then A is **NP**-hard. True or false? Justify. **[2]**

   (c) Prove that the language of properly nested parantheses *and* (square) brackets is in **L**. For instance, `[]([()])` is properly nested while `[([)]]` is not. **[4]**

   (d) Two Boolean formulae ψ, ϕ are equivalent if they are defined over the same set of variables and for all possible assignments z, $\psi(z) = \phi(z)$. Show that the language MINF is in **PSPACE**.

   $$\text{MINF} = \{\phi \mid \phi \text{ is a Boolean formula not equivalent to any smaller Boolean formula}\}$$

   [6]

6. A map $h : \Sigma^* \to \Gamma^*$ is a homomorphism if $h(xy) = h(x)h(y)$ for all strings $x, y \in \Sigma^*$ (here, xy denotes concatenation of x and y). It follows that $h(\epsilon) = \epsilon$. A homomorphism is *non-erasing* if $h(a) \neq \epsilon$ for all $a \in \Sigma$. For a language $A \subset \Sigma^*$, define $h(A) = \{h(x) \mid x \in A\} \subseteq \Gamma^*$. Prove the following statements.

   (a) **NP** is closed under non-erasing homomorphisms i.e., $A \in \mathbf{NP} \Rightarrow h(A) \in \mathbf{NP}$. **[8]**

   Hint: Think lengths of strings.

   (b) **P** is closed under non-erasing homomorphisms iff $\mathbf{P} = \mathbf{NP}$. **[12]**

   Hint: (for $\Rightarrow$ direction) Suppose **P** is closed under non-erasing homomorphisms. Can you define a homomorphism h and exhibit a language L such that $L \in \mathbf{P}$ but $h(L)$ is **NP**-complete?