INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Mid-Autumn Semester Examination 2022-23

Date of Examination : 21-11-2022 **Session(FN/AN)** AN **Duration** 3 hrs **Total Marks** 100
Subject No : CS41001 **Subject:** 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. Keep your solutions brief and precise. State all assumptions you make. Sketchy proofs and claims without proper reasoning will be given no credit.

1. For the following languages, identify whether the language is recursive, *r.e.* but not recursive or not *r.e.* . Justify your answer.
   (a) $\{\mathcal{M} \mid \mathcal{M}$ is a TM with more than 100 states$\}$.
   (b) $\{\mathcal{M} \mid \mathcal{M}$ is a TM and $L(\mathcal{M})$ is recursive$\}$.
   (c) $\{\mathcal{M} \mid \mathcal{M}$ is a TM and $|L(\mathcal{M})| \geq 100\}$.
   (d) $\{G \mid G$ is a CFG and $L(G) = L(G)^{\mathbf{R}}\}$. [$L(G)^{\mathbf{R}} = \{x^{\mathbf{R}} \mid x \in L(G)\}$, with $x^{\mathbf{R}}$ denoting reverse of string x.]

$\boxed{4 \times 5 = 20}$

2. (a) BICON is the language containing all undirected graphs G that are biconnected, i.e., for any pair of vertices u, v in the graph G, at least 2 edges have to be removed from G to disconnect u from v. Prove that BICON is in **NL**.
   (b) Prove or disprove: The union of 2 **NL**-complete languages is also **NL**-complete.

$\boxed{12 + 8 = 20}$

3. (a) 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 $h(A) = \{h(x) | x \in A\} \subseteq \Gamma^*$. Prove that $\mathbf{\Sigma}_i^p$ is closed under non-erasing homomorphisms, i.e, $A \in \mathbf{\Sigma}_i^p \implies h(A) \in \mathbf{\Sigma}_i^p$.
   (b) A restriction of a Boolean circuit is obtained by setting some of the variables to constant values in $\{0, 1\}$. In other words, if C is a Boolean circuit over a set of variables $U = \{u_1, \ldots, u_n\}$ and some of the variables are set to 0 or 1, then the resulting circuit is called a restriction of C. The language CKT-RESTRICTION consists of pairs of Boolean circuits (C_1, C_2) such that some restriction of C_1 computes the same truth table as C_2. Show that CKT-RESTRICTION is in $\mathbf{\Sigma}_2^p$.

$\boxed{12 + 8 = 20}$

4. A language $L \in$ **PSPACE** is said to be **PSPACE1**-complete if for any $L' \in$ **PSPACE**, there is a reduction algorithm from L' to L that uses polynomial space. We denote by $L' \leq^1 L$. L is said to be **PSPACE2**-complete if for any $L' \in$ **PSPACE**, there is a reduction algorithm from L' to L that on n-length inputs use $c \cdot n^i$ space, for some universal constant c.
   (a) Show that any language in **PSPACE** is **PSPACE1**-complete.
   (b) Show that there cannot be a **PSPACE2**-complete language.

$\boxed{10 + 10 = 20}$

5. Assume that SAT cannot have an algorithm running in time $o(2^n)$, where n is the number of variables in the input instance. Prove that under this assumption there are infinitely many languages that are neither in **P** nor **NP**-complete. $\boxed{20}$