Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering

Foundations of Computing Science (CS60005) **Autumn Semester, 2022-2023**

Class Test 2 **Date: 05-Nov-2022 (Saturday) 10:30AM – 11:30AM** **Marks: 20**

Instructions:

- There are THREE questions. Answer ALL questions.
- Write your answers in the answer booklet provided to you in the examination hall.
- Keep you answers brief and precise. Write solutions for all parts of a question together.
- Precisel state all assumptions you make.
- Sketchy proofs and claims without proper reasoning will be given no credit.

1. Prove or disprove the following statements.

   (a) Every infinite regular set contains a subset that is not recursively enumerable. [4]

   Solution: An infinite regular set is countable. But it has uncountably many subsets. Number of Turing machines is countable since each Turing machine can be encoded (uniquely) as a natural number. A subset of the set of all Turing machines corresponds to the set of all *r.e.* langauges, which is countable. Hence at least one of the subsets of a regular set must be non *r.e.* .

   (b) Every infinite *r.e.* set contains an infinite recursive subset. [6]

   Hint: A set is recursive iff there exists an enumeration machine enumerating its strings in non-decreasing order of length.

   Solution: We know that a set is recursive iff there exists an enumeration machine enumerating its strings in *lexicographic order*. (Here, lexicographic order of strings in Σ^* is an arrangement such that strings are in non-decreasing order of length and strings of the same length are in lexicographic order.)

   Let A be an infinite *r.e.* set over alphabet Σ and let $\mathcal{M}$ be an enumeration machine that enumerates A. Let $\mathcal{N}$ be an enumeration machine that simulates $\mathcal{M}$ and does the following whenever $\mathcal{M}$ enters the enumeration state:

   - Suppose x is the first string that $\mathcal{M}$ enumerates. Enumerate x and continue simulating $\mathcal{M}$, remembering x.
   - Repeat: if $\mathcal{M}$ enumerates a string y such that x precedes y in a lexicographic order of strings, then enumerate y; set $x \leftarrow y$ (i.e., replace x on the tape with y). Otherwise, ignore y and continue simulating $\mathcal{M}$.

   Observe that, for any string x, $\mathcal{M}$ always enumerates a string y that comes after x in the lexicographic order as A is infinite.

   The strings enumerated by $\mathcal{N}$ are in lexicographic order and therefore $L(\mathcal{N})$ is recursive.

2. Consider the languge $\{(\mathcal{M}, x, p) \mid \mathcal{M}$ on input x visits state p during the computation$\}$. (Here, $p \in Q$ with Q being the set of states of $\mathcal{M}$ and $x \in \Sigma^*$ where Σ is the input alphabet of the Turing machine $\mathcal{M}$.) Is this language decidable? Justify. [5]

   Solution: Let $L = \{(\mathcal{M}, x, p) \mid \mathcal{M}$ on input x visits state p during the computation$\}$. We show that L is undecidable by reducin MP to L. The reduction maps an instance $(\mathcal{M}, x)$ of MP to $(\mathcal{M}, x, t)$ where t is the accept state of $\mathcal{M}$. Now, $(\mathcal{M}, x) \in \mathsf{MP} \implies \mathcal{M}$ accepts $x \implies \mathcal{M}$ enters state t on input $x \implies (\mathcal{M}, x, t) \in L$; and $(\mathcal{M}, x) \notin \mathsf{MP} \implies \mathcal{M}$ does not accept $x \implies \mathcal{M}$ never enters state t on input $x \implies (\mathcal{M}, x, t) \notin L$.

3. Let $\mathsf{REG} = \{\mathcal{M} \mid \mathcal{M}$ is a TM and $L(\mathcal{M})$ is a regular set$\}$. One of the following is true. Identify which one and justify your answer. [5]

   (a) REG is recursive.

(b) REG is *r.e.* and $\neg\mathsf{REG}$ is not *r.e.*

(c) REG is not *r.e.* and $\neg\mathsf{REG}$ is *r.e.*

(d) Neither REG nor $\neg\mathsf{REG}$ is *r.e.*

Solution: Answer is (d).

Let P_{REG} denote the property on r.e. sets defined as

$$P_{\mathsf{REG}}(A) = \begin{cases} \mathsf{T} & \text{if } A \text{ is regular} \\ \mathsf{F} & \text{otherwise} \end{cases}$$

Then deciding this property is equivalent to deciding REG. If a set A is regular it is not necessary that all its supersets are regular. In other words, there exist A, B such that $A \subseteq B$ and $P_{\mathsf{REG}}(A) = \mathsf{T}$, $P_{\mathsf{REG}}(B) = \mathsf{F}$. For instance, we can take $A = \phi$ and $B = \{0^n1^n \mid n \geq 0\}$. This shows that P_{REG} is a non-monotone property. By Rice's theorem, P_{REG} is not r.e. or equivalently REG is not r.e. Similarly, we can show that $\neg\mathsf{REG}$ or equivalently,

$$P_{\neg\mathsf{REG}}(A) = \begin{cases} \mathsf{T} & \text{if } A \text{ is not regular} \\ \mathsf{F} & \text{otherwise} \end{cases}$$

is not r.e. by proving that $P_{\neg\mathsf{REG}}$ is a non-monotone property. We only need to exhibit two sets A, B with $A \subseteq B$ such that $P_{\neg\mathsf{REG}}(A) = \mathsf{T}$ and $P_{\neg\mathsf{REG}}(B) = \mathsf{F}$. Taking $A = \{0^n1^n \mid n \geq 0\}$ and $B = \{0^*1^*\}$ suffices.