CSE 531: Computability and Complexity — Autumn 2003

Problem Set #2 — Instructor: Venkatesan Guruswami

Due on Wednesday, **October 29, 2003** in class.

Instructions: Same as Problem Set 1.
There are **seven** problems, including an optional problem.

---

1. In class, we have discussed various undecidability results relating to Turing machines that deal with the nature of language they accept. In this problem, we consider questions concerning the actual runtime behavior of Turing machines. Which of the following problems about Turing machines are decidable and which are not? Briefly justify your answers.

   (a) To determine, given a Turing machine M and a string w, whether M ever moves it head to the left when it is run on input w.

   (b) To determine, given a Turing machine M, whether the tape ever contains four consecutive 1's during the course of M's computation when it is run on input 01.

2. (a) Prove that a language A is Turing recognizable if and only if A is mapping reducible to A_{TM}.

   (b) Prove that a language B is decidable if and only if B is mapping reducible to $\{0^n 1^n \mid n \geq 1\}$.

3. This problem investigates the solvability of the Post Correspondence problem (PCP) over small alphabets.

   (a) Prove that PCP over a unary alphabet $\Sigma = \{1\}$ is decidable.

   (b) Prove that PCP over a binary alphabet $\Sigma = \{0, 1\}$ is undecidable.

4. Problem 5.20, Sipser's book (Acceptance and emptiness problems for two headed finite automata)

   - Suggestion: Reading Theorems 5.8 and 5.9 on linear bounded automata will help.

5. In light of the fact that CFLs are not closed under complementation, it is a natural question whether the problem of deciding whether a particular grammar generates a language whose complement is also a CFL can be algorithmically solved. To study this, define the language

$$COMPL_{\mathrm{CFG}} = \{\langle G\rangle \mid G \text{ is a context-free grammar and } \overline{L(G)} \text{ is context-free}\}$$

   where $\overline{L(G)}$ denotes the complement of $L(G)$. Prove that $COMPL_{\mathrm{CFG}}$ is undecidable.

   (Suggestion: Use an approach based on computation histories of Turing machines similar to the proof that ALL_{CFG} is undecidable.)

6. Let $\mathcal{P}$ be a property (subset) of Turing-recognizable languages. Suppose that there exists an infinite language $L \in \mathcal{P}$ such that no finite subset of L belongs to $\mathcal{P}$. Then, prove that the language

$$\mathcal{P}_{\mathrm{TM}} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) \in \mathcal{P}\}$$

   is not *Turing-recognizable.*

7. * **(Optional Problem)** A famous conjecture in Number Theory, called the "Twin Primes" conjecture, states that there are infinitely many positive integers n for which both n and $n+2$ are prime numbers. (For instance $(3, 5)$, $(5, 7)$, $(11, 13)$, $(17, 19)$ are all twin prime pairs and the conjecture states that there are infinitely many such pairs.) This conjecture is still unsolved (see http://mathworld.wolfram.com/TwinPrimeConjecture.html for more background if you are curious).

   Suppose A_{TM} were decidable by a TM H. Then, use H to describe a Turing Machine that is guaranteed to halt and correctly state whether or not the "Twin Primes" conjecture holds.