CSE 531: Computability and Complexity — Autumn 2004

Problem Set #2 — Instructor: Venkatesan Guruswami

Due on Tuesday, **November 2, 2004** in class.

Instructions: Same as Problem Set 1. There are **SIX** problems, each worth 10 points.

---

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. Prove that telling if the intersection of two context-free languages is context-free is undecidable. Formally, prove that the language

   $$\text{INT}_{\text{CFG}} = \{\langle G_1, G_2\rangle \mid G_1, G_2 \text{ are context-free grammars and } L(G_1) \cap L(G_2) \text{ is context-free}\}$$

   is undecidable.

3. In this problem, you will show that a certain "tiling" problem is undecidable. An instance of the tiling problem is a collection of square tiles $t_1, t_2, \ldots, t_q$, together with a list L_h of horiontally compatible, and a list L_v of vertically compatible pairs. An $n \times n$ *tiling* is a placement of tiles into an $n \times n$ grid so that every pair of horizontally adjacent tiles appears in the list L_h, and every pair of vertically adjacent tiles appears in the list L_v; in addition the tile in the upper left corner must be t_1. The language TILE consists of those instances for which there is an $n \times n$ tiling for all $n \geq 0$.

   (a) Describe the above tiling problem formally by giving a precise definition of the language TILE.

   (b) Prove that the language TILE is undecidable.
   Suggestion: Try reducing from the halting problem and "naming" some of your tiles with triplets of symbols in your reduction.

4. Prove that the set of incompressible strings contains no infinite Turing-recognizable subset.

5. Suppose M is a single-tape Turing machine that decides the language of palindromes defined as

   $$\text{PAL} = \{ww^R \mid w \in \{0,1\}^*\} ,$$

   where w^R stands for the reverse of the string w. Prove that the worst-case number of steps that M takes on inputs of length n grows as $\Omega(n^2)$.
   (Note that PAL can be trivially decided in $O(n)$ steps on a 2-tape Turing machine. Proving such a superlinear lower bound against 2-tape Turing machines, for *any explicit* language, remains an open problem.)

<u>Hint</u>: Let M be a single-tape TM deciding PAL. Consider the computation of M on input $x0^{2n}x^R$ where x is some *incompressible* string of length n. Show that if M takes fewer than cn^2 steps for some small enough $c > 0$, then the Kolmogorov complexity of x will be much less than n, a contradiction. To show the latter, try encoding x by the relevant information about M during the times it *crosses* some point in the middle 0^{2n} portion of the input $x0^{2n}x^R$.

6. Define a relation $R \subseteq (\Sigma^*)^k$ to be decidable if the language

$$L_R = \{\langle x_1, x_2, \ldots, x_k\rangle \mid (x_1, x_2, \ldots, x_k) \in R\}$$

is decidable. Define Σ_k, for $k \geq 0$, to be the class of all languages L for which there is a decidable $(k+1)$-ary relation R such that

$$L = \{x \mid \exists x_1 \forall x_2 \cdots Q_k x_k \; R(x_1, x_2, \ldots, x_k, x)\} ,$$

where the quantifier Q_k is $\exists$ if k is odd and $\forall$ if k is even. We define $\Pi_k = \mathrm{co}\Sigma_k$, i.e. Π_k is the set of all complements of languages in Σ_k.

In this notation, clearly Σ_0 and Π_0 equal the set of decidable languages. Now to your exercises:

(a) Show that Π_1 is precisely the class of co-Turing-recognizable languages.

(b) Prove that the languages

$$ALL_{\mathrm{TM}} = \{\langle M\rangle \mid M \text{ is a Turing machine and } L(M) = \Sigma^*\} ,$$

and

$$INFINITE_{\mathrm{TM}} = \{\langle M\rangle \mid M \text{ is a Turing machine and } L(M) \text{ is infinite}\}$$

both belong to Π_2.

(c) Prove that ALL_{TM} is "complete" for Π_2 in the sense that every language $A \in \Pi_2$ mapping reduces to ALL_{TM}.

(d) * **(Extra credit)** In class, we showed that $ALL_{\mathrm{TM}} \notin \Pi_1$ (by showing $A_{\mathrm{TM}} \leq_m ALL_{\mathrm{TM}}$). Use part (c) to show that $ALL_{\mathrm{TM}} \notin \Sigma_1$, i.e., ALL_{TM} is not Turing-recognizable.