CSE P531: Computability and Complexity Theory (Spring, 2016)

Homework 6 Out: Thursday, 5-May. **Due**: Saturday, 14-May (9pm in the **Dropbox**)

Reading:

Sipser, 8.1-8.4, 9.1; Arora-Barak 4.1-4.3

Instructions:

Your proofs and explanations should be clear, well-organized and as concise as possible.

You are allowed to discuss the problems with fellow students taking the class. However, you must write up your solutions completely on your own. Moreover, if you do discuss the problems with someone else, I am asking, on your honor, that you do not take any written material away from the discussion. In addition, for each problem on the homework, I ask that you acknowledge the people you discussed that problem with, if any.

Most of the problems require only one or two key ideas for their solution – spelling out these ideas should give you most of the credit for the problem even if you err in some finer details. So, make sure you clearly write down the main idea(s) behind your solution.

A final piece of advice: Begin work on the problem set early and don't wait until the deadline is only a few days away.

1. Checking for integer roots

A **monomial** in variables $x_1, x_2, \dots, x_n$ is a product $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, where the a_i's are natural numbers. An **integral polynomial** in $x_1, x_2, \dots, x_n$ is a sum of monomials with integer coefficients. For instance,

$$p(x_1, x_2, x_3) = 4x_1x_2 - 7x_1x_3^2 + 11\, x_1^2x_2^2x_3 + 2$$

A root $(z_1, z_2, \dots, z_n)$ of a polynomial p in n variables is a sequence of numbers such that $p(z_1, z_2, \dots, z_n) = 0$. A root is integral if all the z_i's are integers.

Consider the language:

$$\text{INTEGRAL-ROOT} = \{ \langle p \rangle : \ p \text{ is a polynomial with an integer root} \}$$

a) Show that 3-SAT $\leq_P$ INTEGRAL-ROOT.
b) Does this imply that INTEGRAL-ROOT is NP-complete? What's the difficulty?

2. EXPTIME vs. NEXPTIME

Recall that EXPTIME $= \bigcup_k \text{TIME}\left(2^{n^k}\right)$. Let NEXPTIME = $\bigcup_k \text{NTIME}\left(2^{n^k}\right)$ be the class of languages decidable on a non-deterministic TM with exponential time. Your goal in this problem is to show that if EXPTIME ≠ NEXPTIME, then P ≠ NP.

To accomplish this, it will help to use the function

$$\text{pad} : \ \Sigma^* \times \mathbb{N} \to \Sigma^*\#^*$$

defined by $\text{pad}(s, l) = s\#^j$ where $j = \max(0, l - \text{length}(s))$. In other words, the function pad adds enough # characters to the end of s so that it has length exactly l (and it just returns s if $\text{length}(s) > l$).

For a language A and a function $f : \mathbb{N} \to \mathbb{N}$, we define a new language

$$\text{pad}\big(A, f(n)\big) = \Big\{\text{pad}\Big(s, f\big(\text{length}(s)\big)\Big) : s \in A\Big\}$$

Prove that if $A \in TIME(n^{10})$, then $\text{pad}(A, n^2) \in TIME(n^5)$.

Prove that if EXPTIME ≠ NEXPTIME, then P ≠ NP.

EXTRA CREDIT [each worth two regular problems]

1. Graph acyclicity

You can read in Sipser (Section 8.4) about the complexity class $L = \mathrm{SPACE}(\log n)$ of languages that can be decided using only $O(\log n)$ space. This is the set of languages where inputs of length n can be decided using only $O(\log n)$ bits of memory. (Note that for Turing machines, this is modeled by having two tapes: The input tape is **read only** and your TM is only allowed to access $O(\log n)$ cells on the working tape.)

Consider the language

$$\mathrm{ACYCLIC} = \{ \langle G \rangle : \ G \text{ is an undirected graph with no cycles } \}$$

(Note that G could be disconnected.) Prove that $\mathrm{ACYCLIC} \in L$.

2. Bounded acceptance

For any natural number $k \geq 1$, define the language

$$A_{TM}^{k} = \{ \langle M, w \rangle : M \text{ is a TM with at most } k \text{ states and } M \text{ accepts } w \}$$

Prove that for some $k \geq 1$, the language A_{TM}^{k} is undecidable.