Advanced Complexity Theory
6.841/18.405J
Due: Monday, Feb 25, 2001

Madhu Sudan

Problem Set 1

Problems

1. The integer factorization function takes as input an n bit integer X and outputs a list of primes $p_1, \ldots, p_\ell$ such that $X = \prod_{i=1}^{\ell} p_i$. Give a language that is "equivalent" to the integer factorization problem. (Include a precise definition of the notion of "equivalence" in your answer.)

2. Given a language $L \subseteq \{0,1\}^*$, let $L_n = L \cap (\cup_{i=0}^{n}(\{0,1\}^i))$. We say that L is *self-reducible* if there exists a polynomial time oracle Turing machine M such that for every $x \in \{0,1\}^n$,

$$x \in L \quad \Leftrightarrow \quad M^{L_{n-1}}(x) \text{ accepts.}$$

   (a) Given an example of a self-reducible language.

   (b) Prove that if L is self-reducible, then L is in PSPACE.

3. Prove that there exists an oracle A such that $\mathrm{NP}^A \neq \mathrm{co} - \mathrm{NP}^A$.

4. Show that any single-tape, single-head Turing machine recognizing the "palindrome" language $\{xx^R | x \in \{0,1\}^*\}$ (where x^R denotes the reversal of the string x) must take time $\Omega(n^2)$.

5. Let LIN-SPACE be the class of languages recognizable in linear space. Show that LIN-SPACE $\neq$ P.

Instructions (Revised):

- Turn in the solutions to the above problems before lecture on Monday Feb. 25.
- Solutions should be written in latex; and turned in online by email to madhu@mit.edu.
- Collaboration is allowed and encouraged. You may consult (1) the text by Papadimitriou, (2) the text by Sipser, and/or (3) the notes from 6.841 from Spring 2001. But you are not allowed to look at any other sources (previous years psets; papers etc.). And you *must* list all collaborators and sources!
- Correctness, clarity, and succinctness of the solution will determine your score.

Additional Exercises: Not to be turned in!!

The following exercises are recommended if your complexity theory is somewhat rusty. Doing the exercises is not mandatory.

1. Show that a k-tape Turing machine M running in time $t(n)$ can be simulated in time $O(t^2(n))$ on a single-tape Turing machine and in time $O(t(n) \log t(n))$ on a 2-tape machine.

   Open: For every ℓ show that there exists a language L that can be solved in time $t(n)$ by a k tape Turing machine, for some k, but not in time $o(t(n) \log t(n)$ by any ℓ-tape Turing machine.

2. Prove Blum's speedup theorem: Specifically for every Language L decidable in time $t(n) = \omega(n)$ and every constant $\epsilon > 0$, there exists a Turing machine M that decides L in $\epsilon t(n)$ steps.

3. Let ATISP$[a, t, s]$ consist of the set of languages decidable by an alternating Turing machine M that makes $a(n)$ alternations (on inputs of length n), uses $t(n)$ time and $s(n)$ space. Show that
$$\text{ATISP}[0, t^a, s] \subseteq \text{ATISP}[a, ast, ast].$$