Computational complexity: Assignment 3

Due date: October 30, 2013

General instructions:

- Write your solutions by furnishing all relevant details (you may assume the results already covered in the class).
- You are strongly urged to solve the problems by yourself.
- If you discuss with someone else or refer to any material (other than the course notes) then please put a reference in your answer script stating clearly whom or what you have consulted with and how it has benifited you. We would appreciate your honesty.
- If you need any clarification, please ask the instructor.

Total: 50 points

In the following problems, n stands for length of input string (unless mentioned otherwise in the problem).

1. (**5 points**) [Excercise-5.6 from Arora-Barak's book] Adapt the proof of 'Time/space trade-off for SAT' to show that SAT $\in \mathsf{TISP}(n^c, n^d)$ for every constants c and d such that $c(c+d) < 2$.

2. (**4 points**) [Excercise-5.10 from Arora-Barak's book] Suppose A is some language such that $\mathsf{P}^A = \mathsf{NP}^A$. Then, show that $\mathsf{PH}^A \subseteq \mathsf{P}^A$ (in other words, the proof of 'PH collapse', done on the class, *relativizes*).

3. (**5 points**) [Excercise-6.1 from Arora-Barak's book] In this execise, you'll prove Shanon's result that every function $f : \{0,1\}^v \to \{0,1\}$ can be computed by a circuit of size $O(2^n/n)$.

   (a) (**2 points**) Prove that every such function f can be computed by a circuit of size $O(2^n)$.

   (b) (**3 points**) Improve this bound to show that any such function f can be computed by a circuit of size $O(2^n/n)$.

4. (**4 points**) [Excercise-6.3 from Arora-Barak's book] Describe a *decidable* language in $\mathsf{P}_{/\mathsf{poly}}$ that is not in P.

5. (**8 points**) [Excercise-6.5 from Arora-Barak's book] Show that for every $k > 0$, PH contains languages whose boolean circuit complexity is $\Omega(n^k)$.

6. (**9 points**) [Exercise-6.12 from Arora-Barak's book] In this exercise, you'll prove that $\mathsf{NL} \subseteq \mathsf{NC}$.

   (a) (**2 points**) Describe an NC circuit for the problem of computing the product of two given $n \times n$ matrices A, B over a finite field $\mathbb{F}$ of size at most polynomial in n.

   (b) (**3 points**) Describe an NC circuit for computing, given an $n \times n$ matrix, the matrix A^n over a finite field $\mathbb{F}$ of size at most polynomial in n.

   (c) (**4 points**) Conclude that the PATH problem (and hence every NL language) is in NC.

7. (**7 points**) [Exercise-6.15 from Arora-Barak's book] Prove that if a language L is P-complete then $L \in \mathsf{NC}$ (respectively, L) if and only if $\mathsf{P} = \mathsf{NC}$ (respectively, L).

8. (**8 points**) [Exercise-6.19 from Arora-Barak's book] Show that if linear programming has a fast parallel algorithm then $\mathsf{P} = \mathsf{NC}$.

9. (**Bonus problem**) [Exercise-6.16 from Arora-Barak's book] Show that the problem

$$\{< M, k >: M \text{ is a matrix (having integer entries) with determinant } k\}$$

   is in NC.