Stanford University — CS254: Computational Complexity Handout M
Luca Trevisan April 28, 2010

Midterm

Due on Thursday, May 6, 2010

1. [30/100] *Two properties of* **P**/*poly*

   (a) Prove that the number of languages in **P**/poly is not countable.

   (b) Prove that there is a computable language in **P**/poly that is not in **P**.

2. [30/100] *An unusual business model.*

   Thanks to its breakthrough on closed-timeline quantum computing, Hellaflop Corporation has built a machine that can solve large instances of NP-complete problems. Hellaflop monetizes its technology through a service by which one can submit an NP problem, in the form of a Boolean circuit C, and then the next day, for a fee of $100,000, one gets back an input x such that $C(x) = 1$ if such an input exists. (Or the assurance that no such x exists, otherwise.)

   You set up a reselling business that, for the same service, charges only $2,000 per circuit, but you only guarantee to give back the answer within three weeks. You expect to get about 100 requests per day, and to make a profit. How is it possible?

   Technically, prove the following fact: suppose you are given m circuits $C_1, \ldots, C_m$, and, for each of them, you want to find an x_i such that $C_i(x_i) = 1$, if such an x_i exists. Show that you can solve this problem in time polynomial in the sum of the sizes of the circuits provided that you are given access $\lceil \log_2 m + 1 \rceil + 1$ times to an "oracle" that given a circuit C finds an x such that $C(x) = 1$ if such an x exists, or that tell you that no such x exists otherwise. (For example, given 1,023 circuits, you can solve the circuit sat search problem for all of them if you are given access 11 times to an oracle for the circuit sat search problem.)

3. [40/100] *Applications of an "approximator of circuit probability"*

   Suppose that there is a deterministic polynomial-time algorithm A that on input (the description of) a circuit C produces a number $A(C)$ such that

   $$\mathop{\mathbb{P}}_x[C(x) = 1] - \frac{2}{5} \leq A(C) \leq \mathop{\mathbb{P}}_x[C(x) = 1] + \frac{2}{5} .$$

   (a) [10] Prove that it follows $\mathbf{P} = \mathbf{BPP}$.

(b) [15] Prove that there exists a deterministic algorithm A' that, on input a circuit C and a parameter ϵ, runs in time polynomial in the size of C and in $1/\epsilon$ and produces a value $A'(C, \epsilon)$ such that

$$\mathop{\mathbb{P}}_{x}[C(x) = 1] - \epsilon \leq A'(C, \epsilon) \leq \mathop{\mathbb{P}}_{x}[C(x) = 1] + \epsilon \ .$$

(c) [15] Prove that there exists a deterministic algorithm A'' that, on input a circuit C computing a function $f : \{0,1\}^n \rightarrow \{1, \ldots, k\}$ and a parameter ϵ, runs in time polynomial in the size of C, in $1/\epsilon$ and in k, and produces a value $A''(C, \epsilon)$ such that

$$\mathop{\mathbb{E}}_{x}[f(x)] - \epsilon \leq A''(C, \epsilon) \leq \mathop{\mathbb{E}}_{x}[f(x)] + \epsilon \ .$$

[For this question, you can think of C as being a circuit with $\log k$ outputs, and the outputs of $C(x)$ are the binary representation of $f(x)$.]