Exercise 1 : Topics in Pseudorandomness (Fall 2005-2006)

Ronen Shaltiel

November 14, 2005

1. (The class BPP and Chernoff's inequality)
   Let BPP_ϵ be the class of all languages L such that there exists a polynomial p and a polynomial time Turing machine A such that for every x,

   $$\Pr_{y \in_R \{0,1\}^{p(|x|)}}[A(x,y) \neq 1_L(x)] < \epsilon$$

   Here $1_L(x)$ denotes the characteristic function of L which answers 1 on x iff $x \in L$. The class BPP (which is often referred to as languages accepted by two-sided error polynomial time probabilistic algorithms) is formally defined to be $\mathrm{BPP}_{1/3}$. The purpose of this exercise is to show that the constant $1/3$ can be replaced by any other constant smaller than $1/2$. Another purpose is to exercise using *Chernoff's inequality*:

   Chernoff's inequality: Let $X_1, \cdots, X_n$ be independent random variables taking values in $\{0,1\}$. Let $X = \sum X_i$ and denote the expectation of X by μ. The following holds for every $0 < \delta \leq 1$:

   $$\Pr[|X - \mu| > \delta\mu] < e^{-\frac{\delta^2\mu}{6}}$$

   Prove that $\mathrm{BPP}_{2^{-|x|}} = \mathrm{BPP}_{1/3} = \mathrm{BPP}_{1/2-1/|x|}$.

2. ($\mathrm{BPP} \subseteq \mathrm{P/poly}$) We first define nonuniform Turing machines.[1]

   Definition 1 (nonuniform Turing machines). *A nonuniform Turing machine C is a pair:*

   - *M a Turing machine.*
   - *A series of strings $\{a_n\}$.*

   The output of C given an input x is defined to be $M(x, a_{|x|})$. We say that C is of size $t(n)$ if for every input x, $M(x, a_{|x|})$ runs in time $t(|x|)$.

   Prove that any language in BPP there is a polynomial $p(n)$ such that the language can be computed by a nonuniform Turing machine of size $p(n)$. (Hint: use the previous question).

[1] For students who saw Boolean circuits we remark that the class of functions computed by polynomial size nonuniform Turing machines is exactly the class of functions computed by polynomial size Boolean circuits (and is often denoted by P/poly).

3. (Program Checking and Correcting - Another Interesting Application of Randomness)[2]

We are given a program that implements a function f that takes as inputs two $n \times n$ matrices A and B over $GF(2)$. The programmer claims that his program implements very efficient matrix multiplication. Namely that $f(A, B) = AB$, the product of A and B over $GF(2)$.[3] But we feel the need to be a bit cautious ... The following questions will provide methods for checking and possibly correcting the program.

(a) Given three matrices A, B and C (where $C = f(A, B)$) we want to verify that $AB = C$. Give a simple procedure M that performs $O(n^2)$ basic operations such that if $AB = C$ then $\Pr[M(A, B, C) = \text{“accept”}] = 1$ and if $AB \neq C$ then $\Pr[M(A, B, C) = \text{“accept”}] \leq 1/2$.

Note: One simple possible solution could have been to recompute the product AB. But this will require n^3 basic operations and we are losing the benefit of the fast matrix multiplication we were promised.

Hint: think of the protocol we saw in class (in the context of communication complexity) for verifying equality of strings, using the Hadamard codes.

(b) Assume we know that the procedure works well on average. That is, $\Pr_{A,B}[f(A, B) = AB] \geq 7/8$. Show how to correct the program to a program that evaluates f' such that *for every inputs* A and B, $\Pr[f'(A, B) = AB] \geq 1/2$.

You are allowed to invoke the procedure for f a constant number of times and to perform $O(n^2)$ additional basic operations.

Hint1: First show how obtain a programm that works well for every matrix A, when B is uniformly selected. E.g., $\forall A, \Pr_B[f(A, B) = AB] \geq 1/2$.

Hint2: To implement the previous hint, think how to reduce the problem of computing AB to a constant number of other matrix multiplications where the second matrix is uniformly selected.

[2] This question may be a bit harder than the previous ones if you didn't see similar ideas in the past.

[3] This simply means that the entries of A and B are in $\{0, 1\}$ and that the computations are done $\bmod 2$.