CS 225 - Pseudorandomness Prof. Salil Vadhan

Take-Home Final

Harvard SEAS - Spring 2015 *Due: Wed. May 13, 2015*

Your problem set solutions must be typed (in e.g. LaTeX) and submitted electronically to `cs225-hw@seas.harvard.edu`. There are no late days for this problem set. Please name your file `final-`lastname`.*`.

Collaboration or use of references other than the course materials (the text, problem sets, solutions, your notes) is not allowed. You may only discuss these problems with course staff. You may use any result proved in the text, on problem sets, or in section, as long as you state it clearly. *ed problems are extra credit.

Problem 2.11. (Consequences of Derandomizing prBPP) Even though **prBPP** is a class of decision problems, it also captures many other types of problems that can be solved by randomized algorithms:

2. (**NP** Search Problems) An **NP** search problem is specified by a polynomial-time verifier V and a polynomial p; the problem is, given an input $x \in \{0,1\}^n$, find a string $y \in \{0,1\}^{p(n)}$ such that $V(x,y) = 1$. Suppose that such a search problem can be solved in probabilistic polynomial time, i.e. there is a probabilistic polynomial-time algorithm A such that for every input $x \in \{0,1\}^n$, outputs $y \in \{0,1\}^{p(n)}$ such that $V(x,y) = 1$ with probability at least $2/3$ over the coin tosses of A. Show that if **prBPP** = **prP**, then there is a deterministic polynomial-time algorithm B such that for every $x \in \{0,1\}^n$, $B(x)$ always outputs $y \in \{0,1\}^{p(n)}$ such that $V(x,y) = 1$. (Hint: consider a promise problem whose instances include pairs (x,r) where r is a prefix of the coin tosses of A, and use it to approximate the Method of Conditional Probabilities.)

4. Use Part 2, the Prime Number Theorem (see Problem 2.4), and the fact that Primality is in **BPP** (Problem 2.6) to show that if **prBPP** = **prP**, then there is a deterministic polynomial-time algorithm that given a number N, outputs a prime in the interval $[N, 2N)$ for all sufficiently large N.

Problem 6.4. (Rényi extractors) Call a function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ a (k,ε) *Rényi extractor* if for every source X on $\{0,1\}^n$ of Rényi entropy at least k, it holds that $\mathrm{Ext}(X, U_d)$ has Rényi entropy at least $m - \varepsilon$.

1. Prove that a (k,ε) Rényi extractor is also a $(k, \sqrt{\varepsilon})$ extractor.

2. Show that for every $n, k, m \in \mathbb{N}$ with $m \leq n$, $k \geq m/2$, and $\varepsilon > 0$, there exists a (k,ε) Rényi extractor $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = \min\{O(n - k + \log(1/\varepsilon)), m/2 + O(\log(n/\varepsilon))\}$. (Hint: Problem 3.4 may be useful.)

3. Show that if $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k,1)$ Rényi extractor, then $d \geq \min\{n - k, m/2\} - O(1)$. (Hint: consider a k-source that is uniform over $\{x : \exists y \mathrm{Ext}(x,y) \in T\}$ for an appropriately chosen set T of size $\lfloor M/2D^2 \rfloor$.)

Problem 7.7. (Better Local Decoding of Reed–Muller Codes) Show that for every constant $\varepsilon > 0$, there is a constant $\gamma > 0$ such that there is a local $(1/2-\varepsilon)$-decoding algorithm for the q-ary Reed-Muller code of degree d and dimension m, provided that $d \leq \gamma q$. (Here we are referring to unique decoding, not list decoding.) The running time of the decoder should be $\mathrm{poly}(m, q)$.

Problem 7.8. (Hitting-Set Generators) A set $H_m \subset \{0,1\}^m$ is a (t, ε) *hitting set* if for every nonuniform algorithm T running in time t that accepts greater than an ε fraction of m-bit strings, T accepts at least one element of H_m.

1. Show that if, for every m, we can construct an $(m, 1/2)$ hitting set H_m in time $s(m) \geq m$, then $\mathbf{RP} \subset \bigcup_c \mathbf{DTIME}(s(n^c))$. In particular, if $s(m) = \mathrm{poly}(m)$, then $\mathbf{RP} = \mathbf{P}$.

2. Show that if there is a (t, ε) pseudorandom generator $G_m : \{0,1\}^d \to \{0,1\}^m$ computable in time s, then there is a (t, ε) hitting set H_m constructible in time $2^d \cdot s$.

3. (*) Show that if, for every m, we can construct an $(m, 1/2)$ hitting set H_m in time $s(m) = \mathrm{poly}(m)$, then $\mathbf{BPP} = \mathbf{P}$. This can be proven in at least two ways: one uses Problem 3.1 and the other uses a variant of Problem 7.1 together with Corollary 7.64. How do the parameters for general $s(m)$ compare between these two approaches?

4. Define the notion of a (t, k, ε) black-box construction of hitting set-generators (similar to Definition 7.65), and show that, when $t = \infty$, such constructions are equivalent to constructions of *dispersers* (Definition 6.19).