Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 98 (2008)

A Hypergraph Dictatorship Test with Perfect Completeness

Victor Chen [*]

Abstract

A hypergraph dictatorship test is first introduced by Samorodnitsky and Trevisan in [21] and serves as a key component in their unique games based PCP construction. Such a test has oracle access to a collection of functions and determines whether all the functions are the same dictatorship, or all their low degree influences are $o(1)$. The test in [21] makes $q \geq 3$ queries and has amortized query complexity $1 + O\left(\frac{\log q}{q}\right)$ but has an inherent loss of perfect completeness. In this paper we give an adaptive hypergraph dictatorship test that achieves both perfect completeness and amortized query complexity $1 + O\left(\frac{\log q}{q}\right)$.

1 Introduction

Linearity and dictatorship testing have been studied in the past decade both for their combinatorial interest and connection to complexity theory. These tests distinguish functions which are linear/dictator from those which are far from being a linear/dictator function. The tests do so by making queries to a function at certain points and receiving the function's values at these points. The parameters of interest are the number of queries a test makes and the completeness and soundness of a test.

In this paper we shall work with boolean functions of the form $f : \{0,1\}^n \to \{\text{-}1, 1\}$. We say a function f is *linear* if $f = (-1)^{\sum_{i \in S} x_i}$ for some subset $S \subseteq [n]$. A *dictator* function is simply a linear function where $|S| = 1$, i.e., $f(x) = (-1)^{x_i}$ for some i. A dictator function is often called a *long code*, and it is first used in [3] for the constructions of probabilistic checkable proofs (PCPs), see e.g., [2, 1]. Since then, it has become standard to design a PCP system as the composition of two verifiers, an outer verifier and an inner verifier. In such case, a PCP system expects the proof to be written in such a way so that the outer verifier, typically based on the verifier obtained from Raz's Parallel Repetition Theorem [17], selects some tables of the proof according to some distribution and then passes the control to the inner verifier. The inner verifier, with oracle access to these tables, makes queries into these tables and ensures that the tables are the encoding of some error-correcting codes and satisfy some joint constraint. The long code encoding is usually employed in these proof constructions, and the inner verifier simply tests whether a collection of tables (functions) are long codes satisfying some constraints. Following this paradigm, constructing a PCP with certain parameters reduces to the problem of designing a long code test with similar parameters.

One question of interest is the tradeoff between the soundness and query complexity of a tester. If a tester queries the functions at every single value, then trivially the verifier can determine all the functions. One would like to construct a dictatorship test that has the lowest possible soundness while making as few

[*] MIT CSAIL. `victor@csail.mit.edu`. Research supported in part by an NSF graduate fellowship and NSF Award CCR-0514915.

ISSN 1433-8092

queries as possible. One way to measure this tradeoff between the soundness s and the number of queries q is *amortized query complexity*, defined as $\frac{q}{\log s^{-1}}$. This investigation, initiated in [25], has since spurred a long sequence of works [22, 20, 11, 6]. All the testers from these works run many iterations of a single dictatorship test by reusing queries from previous iterations. The techniques used are Fourier analytic, and the best amortized query complexity from this sequence of works has the form $1 + O\left(\frac{1}{\sqrt{q}}\right)$.

The next breakthrough occurs when Samorodnitsky [19] introduces the notion of a *relaxed* linearity test along with new ideas from additive combinatorics. In property testing, the goal is to distinguish objects that are very structured from those that are pseudorandom. In the case of linearity/dictatorship testing, the structured objects are the linear/dictator functions, and functions that are far from being linear/dictator are interpreted as pseudorandom. The recent paradigm in additive combinatorics is to find the right framework of structure and pseudorandomness and analyze combinatorial objects by dividing them into structured and pseudorandom components, see e.g. [24] for a survey. One success is the notion of Gowers norm [7], which has been fruitful in attacking many problems in additive combinatorics and computer science. In [19], the notion of pseudorandomness for linearity testing is relaxed; instead of designating the functions that are far from being linear as pseudorandom, the functions having small low degree Gowers norm are considered to be pseudorandom. By doing so, an optimal tradeoff between soundness and query complexity is obtained for the problem of relaxed linearity testing. (Here the tradeoff is stronger than the tradeoff for the traditional problem of linearity testing.)

In a similar fashion, in the PCP literature since [9], the pseudorandom objects in dictatorship tests are not functions that are far from being a dictator. The pseudorandom functions are typically defined to be either functions that are far from all "juntas" or functions whose "low-degree influences" are $o(1)$. Both considerations of a dictatorship test are sufficient to compose the test in a PCP construction. In [21], building on the analysis of the relaxed linearity test in [19], Samorodnitsky and Trevisan construct a dictatorship test (taking the view that functions with arbitrary small "low-degree influences are pseudorandom) with amortized query complexity $1 + O\left(\frac{\log q}{q}\right)$. Furthermore, the test is used as the inner verifier in a conditional PCP construction (based on unique games [12]) with the same parameters. However, their dictatorship test suffers from an inherent loss of perfect completeness. Ideally one would like testers with one-sided errors. One, for aesthetic reasons, testers should always accept valid inputs. Two, for some hardness of approximation applications, in particular coloring problems (see e.g. [10] or [5]), it is important to construct PCP systems with one-sided errors.

In this paper, we prove the following theorem:

Theorem 1.1 (main theorem)**.** *For every $q \geq 3$, there exists an (adaptive) dictatorship test that makes q queries, has completeness 1, and soundness $\frac{O(q^3)}{2^q}$; in particular it has amortized query complexity $1 + O\left(\frac{\log q}{q}\right)$.*

Our tester is a variant of the one given in [21]. Our tester is adaptive in the sense that it makes its queries in two stages. It first makes roughly $\log q$ nonadaptive queries into the function. Based on the values of these queries, the tester then selects the rest of the query points nonadaptively. Our analysis is based on techniques developed in [11, 21, 10, 8].

1.1 Future Direction

Unfortunately, the adaptivity of our test is a drawback. The correspondence between PCP constructions and hardness of approximation needs the test to be fully nonadaptive. However, a more pressing issue is that our hypergraph dictatorship test does not immediately imply a new PCP characterization of NP. The reason is that a dictatorship test without "consistency checks" is most easily composed with the unique label cover defined in [12] as the outer verifier in a PCP reduction. As the conjectured NP-hardness of the unique label cover cannot have perfect completeness, the obvious approach in combining our test with the unique games-based outer verifier does not imply a new PCP result. However, there are variants of the unique label cover (e.g., Khot's d to 1 Conjecture) [12] that do have conjectured perfect completeness, and these variants are used to derive hardness of coloring problems in [5]. We hope that our result combined with similar techniques used in [5] may obtain a new conditional PCP construction and will motivate more progress on constraint satisfaction problems with bounded projection .

1.2 Related Works

The problem of linearity testing was first introduced in [4]. The framework of property testing was formally set up in [18]. The PCP Theorems were first proved in [2, 1]; dictatorship tests first appeared in the PCP context in [3], and many dictatorship tests and variants appeared throughout the PCP literature. Dictatorship test was also considered as a standalone property testing in [16]. As mentioned, designing testers and PCPs focusing on amortized query complexity was first investigated in [25], and a long sequence of works [22, 20, 11, 6] followed. The first tester/PCP system focusing on this tradeoff while obtaining perfect completeness was achieved in [10].

The orthogonal question of designing testers or PCPs with as few queries as possible was also considered. In a highly influential paper [9], Håstad constructed a PCP system making only three queries. Many variants also followed. In particular PCP systems with perfect completeness making three queries were also achieved in [8, 13]. Similar to our approach, O'Donnell and Wu [14] designed an optimal three bit dictatorship test with perfect completeness, and later the same authors constructed a conditional PCP system [15].

2 Preliminaries

We fix some notation and provide the necessary background in this section. We let $[n]$ denote the set $\{1, 2, \ldots, n\}$. For a vector $v \in \{0,1\}^n$, we write $|v| = \sum_{i\in[n]} v_i$. We let $\wedge$ denote the boolean AND, where $a \wedge b = 1$ iff $a = b = 1$. For vectors $v, w \in \{0,1\}^n$, we write $v \wedge w$ to denote the vector obtained by applying AND to v and w component-wise. We abuse notation and sometimes interpret a vector $v \in \{0,1\}^n$ as a subset $v \subseteq [n]$ where $i \in v$ iff $v_i = 1$. For a boolean function $f : \{0,1\}^n \to \{0,1\}$, we make the convenient notational change from $\{0,1\}$ to $\{\text{-}1,1\}$ and write $f : \{0,1\}^n \to \{\text{-}1,1\}$.

2.1 Fourier Analysis

Definition 2.1 (Fourier transform)**.** For a real-valued function $f : \{0,1\}^n \to \mathbb{R}$, we define its Fourier transform $\widehat{f} : \{0,1\}^n \to \mathbb{R}$ to be

$$\widehat{f}(\alpha) = \mathop{\mathbb{E}}_{x\in\{0,1\}^n} f(x)\chi_\alpha(x),$$

where $\chi_\alpha(x) = (-1)^{\sum_{i\in[n]} \alpha_i x_i}$. We say $\widehat{f}(\alpha)$ is the *Fourier coefficient* of f at α, and the *characters* of $\{0,1\}^n$ are the functions $\{\chi_\alpha\}_{\alpha\in\{0,1\}^n}$.

It is easy to see that for $\alpha, \beta \in \{0,1\}^n$, $\mathbb{E}\, \chi_\alpha \cdot \chi_\beta$ is 1 if $\alpha = \beta$ and 0 otherwise. Since there are 2^n characters, they form an orthonormal basis for functions on $\{0,1\}^n$, and we have the Fourier inversion formula

$$f(x) = \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)\chi_\alpha(x)$$

and Parseval's Identity

$$\sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^2 = \mathop{\mathbb{E}}_x[f(x)^2].$$

2.2 Influence of Variables

For a boolean function $f : \{0,1\}^n \to \{\text{-}1,1\}$, the *influence* of the i-variable, $\mathrm{I}_i(f)$, is defined to be $\Pr_{x\in\{0,1\}^n}[f(x) \neq f(x+e_i)]$, where e_i is a vector in $\{0,1\}^n$ with 1 on the i-th coordinate 0 everywhere else. This corresponds to our intuitive notion of influence: how likely the outcome of f changes when the i-th variable on a random input is flipped. For the rest of this paper, it will be convenient to work with the Fourier analytic definition of $\mathrm{I}_i(f)$ instead, and we leave it to the readers to verify that the two definitions are equivalent when f is a boolean function.

Definition 2.2. Let $f : \{0,1\}^n \to \mathbb{R}$. We define the influence of the i-th variable of f to be

$$\mathrm{I}_i(f) = \sum_{\alpha \in \{0,1\}^n :\ \alpha_i = 1} \hat{f}(\alpha)^2.$$

We shall need the following technical lemma, which is Lemma 4 from [21], and it gives an upper bound on the influence of a product of functions.

Lemma 2.1 ([21])**.** *Let $f_1, \ldots, f_k : \{0,1\}^n \to [-1,1]$ be a collection of k bounded real-valued functions, and define $f(x) = \prod_{i=1}^k f_i(x)$ to be the product of these k functions. Then for each $i \in [n]$,*

$$\mathrm{I}_i(f) \leq k \cdot \sum_{j=1}^k \mathrm{I}_i(f_j).$$

When $\{f_i\}$ are boolean functions, it is easy to see that $\mathrm{I}_i(f) \leq \sum_{j=1}^k \mathrm{I}_i(f_j)$ by the union bound.

We now define the notion of low-degree influence.

Definition 2.3. Let w be an integer between 0 and n. We define the *w-th degree influence of the i-th variable* of a function $f : \{0,1\}^n \to \mathbb{R}$ to be

$$\mathrm{I}_i^{\leq w}(f) = \sum_{\alpha \in \{0,1\}^n :\ \alpha_i = 1,\ |\alpha| \leq w} \hat{f}(\alpha)^2.$$

While the definition of low-degree influence is standard in the literature, we shall make a few remarks since this definition does not have a clean combinatorial interpretation or an immediate justification. Dictatorship tests (those based on influences) classify functions in the NO instances to be those whose low-degree influences are $o(1)$ for two reasons. One is that large parity functions, which have many variables with influence 1 but no variables with low-degree influence, must be rejected by the test. The second is that if w is fixed, then a bounded function has only a finite number of variables with large w-th degree influence. This easy fact, though we won't need it here, is often needed to lift a dictatorship test to a PCP construction. Both such considerations fail if we substitute the low-degree influence requirement by just influence, thus the need for a thresholded version of influence.

2.3 Gowers norm

In [7], Gowers uses analytic techniques to give a new proof of Szeméredi's Theorem [23] and in particular, initiates the study of a new norm of a function as a measure of pseudorandomness. Subsequently this norm is termed the *Gowers uniformity norm* and has been intensively studied and applied in additive combinatorics, see e.g. [24] for a survey. The use of the Gowers norm in computer science is initiated in [19, 21].

Definition 2.4. Let $f : \{0,1\}^n \to \mathbb{R}$. We define the *$d$-th dimension Gowers uniformity norm* of f to be

$$||f||_{U_d} = \left(\mathop{\mathbb{E}}_{x,\, x_1, \ldots, x_d} \left[\prod_{S \subseteq [d]} f\left(x + \sum_{i \in S} x_i \right) \right] \right)^{1/2^d}.$$

For a collection of 2^d functions $f_S : \{0,1\}^n \to \mathbb{R}, S \subset [d]$, we define the *$d$-th dimension Gowers inner product* of $\{f_S\}_{S \subseteq d}$ to be

$$\left\langle \{f_S\}_{S \subseteq [d]} \right\rangle_{U_d} = \mathop{\mathbb{E}}_{x,\, x_1, \ldots, x_d} \left[\prod_{S \subseteq [d]} f_S\left(x + \sum_{i \in S} x_i \right) \right].$$

When f is a boolean function, one can interpret the Gowers norm as simply the expected number of "affine parallelepipeds" of dimension d. While this expression may look cumbersome at first glance, the use of the Gowers norm is in some sense to control expectations over some other expressions. For instance, to count the number of $d+1$-term progressions of the form $x, x+y, \ldots, x+d \cdot y$ in a subset, one may be interested in approximating expressions of the form $\mathbb{E}_{x,y}[f_1(x) f_2(x+y) \cdots f_d(x+d \cdot y)]$, where $f_1, \ldots, f_d$ are some bounded functions over some appropriate domain. In fact, as shown by Gowers, these expectations are upper bounded by the Gowers inner product of f_i, which is also upper bounded by $\min_{i \in [d]} ||f_i||_{U_d}^{2^d}$. Thus, in a rough sense, questions regarding progressions are then reduced to questions regarding the Gowers norms, which are more amenable to analytic techniques.

The proof showing that $\mathbb{E}_{x,y}[f_1(x) f_2(x+y) \cdots f_d(x+d \cdot y)]$ is upper bounded by the minimum Gowers norm of all the functions f_i is not difficult; it proceeds by repeated applications of the Cauchy-Schwarz inequality and substitution of variables. Collectively, statements saying that certain expressions are governed by the Gowers norm are coined *von-Neumann type theorems* in the literature.

For the analysis of hypergraph-based dictatorship test, we shall encounter the following expression.

Definition 2.5. Let $\{f_S\}_{S \subseteq [d]}$ be a collection of functions where $f_S : \{0,1\}^n \to \mathbb{R}$. We define the *$d$-th dimension Gowers linear inner product* of $\{f_S\}$ to be

$$\left\langle \{f_S\}_{S \subseteq [d]} \right\rangle_{LU_d} = \mathop{\mathbb{E}}_{x_1, \ldots, x_d} \left[\prod_{S \subseteq [d]} f_S\left(\sum_{i \in S} x_i \right) \right].$$

This definition is a variant of the Gowers inner product and is in fact upper bounded by the square root of the Gowers inner product as shown in [21]. Furthermore they showed that if a collection of functions has large Gowers inner product, then two functions must share an influential variable. Thus, one can infer the weaker statement that large linear Gowers inner product implies two functions have an influential variable.

For our purposes, we can encapsulate all the prior discussion into the following statement, which is Lemma 16 from [21]. This is the only fact on the Gowers norm that we explicitly need.

Lemma 2.2 ([21])**.** *Let $\{f_S\}_{S\subseteq[d]}$ be a collection of bounded functions of the form $f_S : \{0,1\}^n \to [-1,1]$. Suppose $\left\langle \{f_S\}_{S\subseteq[d]} \right\rangle_{LU_d} \geq \epsilon$ and $\mathbb{E}\, f_{[d]} = 0$. Then there exists some variable i, some subsets $S \neq T \subseteq [d]$ such that the influences of the i-th variable in both f_S and f_T are at least $\frac{\epsilon^4}{2^{O(d)}}$.*

3 Dictatorship Test

Definition 3.1 (dictatorship)**.** For $i \in [n]$, the *i-th dictator* is the function $f(x) = (-1)^{x_i}$.

In the PCP literature, the i–th dictator is also known as the long code encoding of i, $\langle(-1)^{x_i}\rangle_{x\in\{0,1\}^n}$, which is simply the evaluation of the i-th dictator function at all points.

Now let us define a t-function dictatorship test. Suppose we are given oracle access to a collection of boolean functions $f_1, \ldots, f_t$. We want to make as few queries as possible into these functions to decide if all the functions are the same dictatorship, or no two functions have some common structure. More precisely, we have the following definition:

Definition 3.2. We say that a test $T = T^{f_1,\ldots,f_t}$ is a *t–function dictatorship test* with *completeness* c and *soundness* s if T is given oracle access to a family of t functions $f_1, \ldots, f_t : \{0,1\}^n \to \{\text{-}1, 1\}$, such that

- if there exists some variable $i \in [n]$ such that for all $a \in [t]$, $f_a(x) = (-1)^{x_i}$, then T accepts with probability at least c, and
- for every $\epsilon > 0$, there exist a positive constant $\tau > 0$ and a fixed positive integer w such that if T accepts with probability at least $s + \epsilon$, then there exist two functions f_a, f_b where $a, b \in [t], a \neq b$ and some variable $i \in [n]$ such that $\mathrm{I}_i^{\leq w}(f_a), \mathrm{I}_i^{\leq w}(f_b) \geq \tau$.

A q-function dictatorship test making q queries, with soundness $\frac{q+1}{2^q}$ was proved in [21], but the test suffers from imperfect completeness. We obtain a $(q - O(\log q))$–dictatorship test that makes q queries, has completeness 1, soundness $\frac{O(q^3)}{2^q}$, and in particular has amortized query complexity $1 + O\left(\frac{\log q}{q}\right)$, the same as the test in [21]. By a simple change of variable, we can more precisely state the following:

Theorem 3.1 (main theorem restated)**.** *For infinitely many t, there exists an adaptive t-function dictatorship test that makes $t + \log(t+1)$ queries, has completeness 1, and soundness $\frac{(t+1)^2}{2^t}$.*

Our test is adaptive and selects queries in two passes. During the first pass, it picks an arbitrary subset of $\log(t+1)$ functions out of the t functions. For each function selected, our test picks a random entry y and queries the function at entry y. Then based on the values of these $\log(t+1)$ queries, during the second pass, the test selects t positions nonadaptively, one from each function, then queries all t positions at once. The adaptivity is necessary in our analysis, and it is unclear if one can prove an analogous result with only one pass.

3.1 Folding

As introduced by Bellare, Goldreich, and Sudan [3], we shall assume that the functions are "folded" as only half of the entries of a function are accessed. We require our dictatorship test to make queries in a special manner. Suppose the test wants to query f at the point $x \in \{0,1\}^n$. If $x_1 = 1$, then the test queries $f(x)$ as usual. If $x_1 = 0$, then the test queries f at the point $\vec{1} + x = (1, 1 + x_2, \ldots, 1 + x_n)$ and negates the value it receives. It is instructive to note that folding ensures $f(\vec{1} + x) = -f(x)$ and $\mathbb{E}\, f = 0$.

3.2 Basic Test

For ease of exposition, we first consider the following simplistic scenario. Suppose we have oracle access to just one boolean function. Furthermore we ignore the tradeoff between soundness and query complexity. We simply want a dictatorship test that has completeness 1 and soundness $\frac{1}{2}$. There are many such tests in the literature; however, we need a suitable one which our hypergraph dictatorship test can base on. Our basic test below is a close variant of the one proposed by Guruswami, Lewin, Sudan, and Trevisan [8].

BASIC TEST T: with oracle access to f,

1. Pick x_i, x_j, y, z uniformly at random from $\{0,1\}^n$.
2. Query $f(y)$.
3. Let $v = \frac{1-f(y)}{2}$. Accept iff

$$f(x_i)f(x_j) = f(x_i + x_j + (v\vec{1} + y) \wedge z).$$

Lemma 3.2. *The test T is a dictatorship test with completeness 1.*

Proof. Suppose f is the ℓ-th dictator, i.e., $f(x) = (-1)^{x_\ell}$. First note that

$$v + y_\ell = \frac{1-(-1)^{y_\ell}}{2} + y_\ell,$$

which evaluates to 0. Thus by linearity of f

$$\begin{aligned} f(x_i + x_j + (v\vec{1} + y) \wedge z) &= f(x_i)f(x_j)f((v\vec{1} + y) \wedge z) \\ &= f(x_i)f(x_j)(-1)^{(v+y_\ell)\wedge z_\ell} \\ &= f(x_i)f(x_j) \end{aligned}$$

and the test always accepts. □

To analyze the soundness of the test T, we need to derive a Fourier analytic expression for the acceptance probability of T.

Proposition 3.3. *Let p be the acceptance probability of T. Then*

$$p = \frac{1}{2} + \frac{1}{2}\sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3\, 2^{-|\alpha|}\left(1 + \sum_{\beta \subseteq \alpha} \widehat{f}(\beta)\right).$$

For sanity check, let us interpret the expression for p. Suppose $f = \chi_\alpha$ for some $\alpha \neq \vec{0} \in \{0,1\}^n$, i.e., $\widehat{f}(\alpha) = 1$ and all other Fourier coefficients of f are 0. Then clearly $p = \frac{1}{2} + 2^{-|\alpha|}$, which equals 1 whenever f is a dictator function as we have just shown. If $|\alpha|$ is large, then T accepts with probability close to $\frac{1}{2}$. We shall first analyze the soundness and then derive this analytic expression for p.

Lemma 3.4. *The test T is a dictatorship test with soundness $\frac{1}{2}$.*

Proof. Suppose the test T passes with probability at least $\frac{1}{2}+\epsilon$, for some $\epsilon > 0$. By applying Proposition 3.3, Cauchy-Schwarz, and Parseval's Identity, respectively, we obtain

$$\begin{aligned}\epsilon &\leq \frac{1}{2} \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3\, 2^{-|\alpha|} \left(1 + \sum_{\beta \subseteq \alpha} \widehat{f}(\beta)\right) \\ &\leq \frac{1}{2} \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3\, 2^{-|\alpha|} \left(1 + \left(\sum_{\beta \subseteq \alpha} \widehat{f}(\beta)^2\right)^{\frac{1}{2}} \cdot 2^{\frac{|\alpha|}{2}}\right) \\ &\leq \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3\, 2^{-\frac{|\alpha|}{2}}.\end{aligned}$$

Pick the least positive integer w such that $2^{-\frac{w}{2}} \leq \frac{\epsilon}{2}$. Then by Parseval's again,

$$\begin{aligned}\frac{\epsilon}{2} &\leq \sum_{\alpha \in \{0,1\}^n : |\alpha| \leq w} \widehat{f}(\alpha)^3 \\ &\leq \max_{\alpha \in \{0,1\}^n : |\alpha| \leq w} \left|\widehat{f}(\alpha)\right|.\end{aligned}$$

So there exists some $\beta \in \{0,1\}^n$, $|\beta| \leq w$ such that $\frac{\epsilon}{2} \leq \left|\hat{f}(\beta)\right|$. With f being folded, $\beta \neq \vec{0}$. Thus, there exists an $i \in [n]$ such that $\beta_i = 1$ and

$$\frac{\epsilon^2}{4} \leq \widehat{f}(\beta)^2 \leq \sum_{\alpha \in \{0,1\}^n : \alpha_i = 1, |\alpha| \leq w} \widehat{f}(\alpha)^2.$$

□

Now we give the straightforward Fourier analytic calculation for p.

Proof of Proposition 3.3. As usual, we first arithmetize p. We write

$$\begin{aligned}p \;=\; &\mathop{\mathbb{E}}_{x_i, x_j, y, z} \left(\frac{1 + f(y)}{2}\right) \left(\frac{1 + \mathrm{Acc}(x_i, x_j, y, z)}{2}\right) + \\ &\mathop{\mathbb{E}}_{x_i, x_j, y, z} \left(\frac{1 - f(y)}{2}\right) \left(\frac{1 + \mathrm{Acc}(x_i, x_j, \vec{1} + y, z)}{2}\right),\end{aligned}$$

where

$$\mathrm{Acc}(x_i, x_j, y, z) = f(x_i) f(x_j) f(x_i + x_j + (y \wedge z)).$$

Since f is folded, $f(\vec{1} + y) = -f(y)$. As y and $\vec{1} + y$ are both identically distributed in $\{0,1\}^n$, we have

$$p = 2 \mathop{\mathbb{E}}_{x_i, x_j, y, z} \left(\frac{1 + f(y)}{2}\right) \left(\frac{1 + \mathrm{Acc}(x_i, x_j, y, z)}{2}\right).$$

Since $\mathbb{E} f = 0$, we can further simplify the above expression to be

$$p = \frac{1}{2} + \frac{1}{2} \mathop{\mathbb{E}}_{x_i,x_j,y,z} \left[(1 + f(y)) \operatorname{Acc}(x_i, x_j, y, z)\right].$$

It suffices to expand out the terms $\mathbb{E}_{x_i,x_j,y,z}[\operatorname{Acc}(x_i, x_j, y, z)]$ and $\mathbb{E}_{x_i,x_j,y,z}[f(y) \operatorname{Acc}(x_i, x_j, y, z)]$.

For the first term, it is not hard to show that

$$\mathop{\mathbb{E}}_{x_i,x_j,y,z}[\operatorname{Acc}(x_i, x_j, y, z)] = \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3 \ 2^{-|\alpha|},$$

by applying the Fourier inversion formula on f and averaging over x_i and x_j and then averaging over y and z over the AND operator.

Now we compute the second term. Applying the Fourier inversion formula to the last three occurrences of f and averaging over x_i and x_j, we obtain

$$\mathop{\mathbb{E}}_{x_i,x_j,y,z}[f(y) \operatorname{Acc}(x_i, x_j, y, z)] = \sum_{\alpha \in \{0,1\}^n} \widehat{f}(\alpha)^3 \mathop{\mathbb{E}}_{y,z} \left[f(y)\chi_\alpha(y \wedge z)\right].$$

It suffices to expand out $\mathbb{E}_{y,z} \left[f(y)\chi_\alpha(y \wedge z)\right]$. By grouping the z's according to their intersection with different possible subsets β of α, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{y,z} \left[f(y)\chi_\alpha(y \wedge z)\right] \\
&= \sum_{\beta \subseteq \alpha} \Pr_{z \in \{0,1\}^n} \left[z \cap \alpha = \beta\right] \mathop{\mathbb{E}}_{y} \left[f(y) \prod_{i:\, \alpha_i = 1} (-1)^{y_i \wedge z_i}\right] \\
&= \sum_{\beta \subseteq \alpha} 2^{-|\alpha|} \mathop{\mathbb{E}}_{y} \left[f(y) \prod_{i:\, \beta_i = 1} (-1)^{y_i}\right] \\
&= 2^{-|\alpha|} \sum_{\beta \subseteq \alpha} \widehat{f}(\beta).
\end{aligned}$$

Putting everything together, it is easy to see that we have the Fourier analytic expression for p as stated in the lemma. □

3.3 Hypergraph Dictatorship Test

We prove the main theorem in this section. The basis of our hypergraph dictatorship test will be very similar to the test in the previous section. We remark that we did not choose to present the exact same basic test for hopefully a clearer exposition.

We now address the tradeoff between query complexity and soundness. If we simply repeat the basic test a number of iterations independently, the error is reduced, but the query complexity increases. In other words, the amortized query complexity does not change if we simply run the basic test for many independent iterations. Following Trevisan [25], all the dictatorship tests that save query complexity do so by reusing queries made in previous iterations of the basic test. To illustrate this idea, suppose test T queries f at the points $x_1 + h_1$, $x_2 + h_2$, $x_1 + x_2 + h_{1,2}$ to make a decision. For the second iteration, we let T query f at the points $x_3 + h_3$ and $x_1 + x_3 + h_{1,3}$ and reuse the value $f(x_1 + h_1)$ queried during the first run of T. T then uses the three values to make a second decision. In total T makes five queries to run two iterations.

We may think of the first run of T as parametrized by the points x_1 and x_2 and the second run of T by x_1 and x_3. In general, we may have k points $x_1, \dots, x_k$ and a graph on $[k]$ vertices, such that each edge e of the graph corresponds to an iteration of T parametrized by the points $\{x_i\}_{i\in e}$. We shall use a complete hypergraph on k vertices to save on query complexity, and we will argue that the soundness of the algorithm decreases exponentially with respect to the number of iterations.

Formally, consider a hypergraph $H = ([k], E)$. Let $\{f_a\}_{a\in[k]\cup E}$ be a collection of boolean functions of the form $f_a : \{0,1\}^n \to \{\text{-}1, 1\}$. We assume all the functions are folded, and so in particular, $\mathbb{E}\, f_a = 0$. Consider the following test:

HYPERGRAPH H-TEST: with oracle access to $\{f_a\}_{a\in[k]\cup E}$,

1. Pick $x_1, \dots, x_k, y_1, \dots, y_k$, and $\{z_a\}_{a\in[k]\cup E}$ independently and uniformly at random from $\{0,1\}^n$.
2. For each $i \in [k]$, query $f_i(y_i)$.
3. Let $v_i = \frac{1-f_i(y_i)}{2}$.
   Accept iff for every $e \in E$,

$$\prod_{i\in e}\Big[f_i(x_i + (v_i\vec{1} + y_i) \wedge z_i)\Big] = f_e\left(\sum_{i\in e} x_i + \Big(\Sigma_{i\in e}(v_i\vec{1} + y_i)\Big) \wedge z_e\right).$$

We make a few remarks regarding the design of H-Test. The hypergraph test by Samorodnitsky and Trevisan [21] accepts iff for every $e \in E$, $\prod_{i\in e} f_i(x_i + \eta_i)$ equals $f_e(\sum_{i\in e} x_i + \eta_e)$, where the bits in each vector η_a are chosen independently to be 1 with some small constant, say 0.01. The noise vectors η_a rule out the possibility that linear functions with large support can be accepted. To obtain a test with perfect completeness, we use ideas from [8, 16, 10] to simulate the effect of the noise perturbation.

Note that for y, z chosen uniformly at random from $\{0,1\}^n$, the vector $y \wedge z$ is a $\frac{1}{4}$–noisy vector. As observed by Parnas, Ron, and Samorodnitsky [16], the test $f(y \wedge z) = f(y) \wedge f(z)$ distinguishes between dictators and linear functions with large support. One can also combine linearity and dictatorship testing into a single test of the form $f(x_1 + x_2 + y \wedge z)(f(y) \wedge f(z)) = f(x_1)f(x_2)$ as Håstad and Khot demonstrated [10]. However, iterating this test is too costly for us. In fact, Håstad and Khot also consider an adaptive variant that reads $k^2 + 2k$ bits to obtain a soundness of 2^{-k^2}, the same parameters as in [20], while achieving perfect completeness as well. Without adaptivity, the test in [10] reads $k^2 + 4k$ bits. While both the nonadaptive and adaptive tests in [10] have the same amortized query complexity, extending the nonadaptive test by Håstad and Khot to the hypergraph setting does not work for us. So to achieve the same amortized query complexity as the hypergraph test in [21], we also exploit adaptivity in our test.

Theorem 3.5 (main theorem restated)**.** *For infinitely many t, there exists an adaptive t-function dictatorship test with $t + \log(t+1)$ queries, completeness 1, and soundness $\frac{(t+1)^2}{2^t}$.*

Proof. Take a complete hypergraph on k vertices, where $k = \log(t+1)$. The statement follows by applying Lemmas 3.6 and 3.7. □

Lemma 3.6. *The H-Test is a $(k+|E|)$-function dictatorship test that makes $|E|+2k$ queries and has completeness* 1.

Proof. The test makes k queries $f_i(y_i)$ in the first pass, and based on the answers to these k queries, the test then makes one query into each function f_a, for each $a \in [k] \cup E$. So the total number of queries is $|E|+2k$.

Now suppose all the functions are the ℓ-th dictator for some $\ell \in [n]$, i.e., for all $a \in [k] \cup E$, $f_a = f$, where $f(x) = (-1)^{x_\ell}$. Note that for each $i \in [k]$,

$$v_i + y_i(\ell) = \frac{1-(-1)^{y_i(\ell)}}{2} + y_i(\ell),$$

which evaluates to 0. Thus for each $e \in E$,

$$\begin{aligned}\prod_{i\in e} f_i(x_i + (v_i\vec{1} + y_i)\wedge z_i) &= f\left(\sum_{i\in e} x_i\right)\cdot \prod_{i\in e} f((v_i\vec{1}+y_i)\wedge z_i)\\ &= f\left(\sum_{i\in e} x_i\right)\cdot \prod_{i\in e}(-1)^{(v_i+y_i(\ell))\wedge z_i(\ell)}\\ &= f\left(\sum_{i\in e} x_i\right),\end{aligned}$$

and similarly,

$$f_e\left(\sum_{i\in e} x_i + \left(\Sigma_{i\in e}(v_i\vec{1}+y_i)\right)\wedge z_e\right) = f\left(\sum_{i\in e} x_i\right).$$

Hence the test always accepts. □

Lemma 3.7. *The H-Test has soundness* $2^{k-|E|}$.

Before proving Lemma 3.7 we first prove a proposition relating the Fourier transform of a function perturbed by noise to the function's Fourier transform itself.

Proposition 3.8. *Let $f : \{0,1\}^n \to \{\text{-}1,1\}$. Define $g : \{0,1\}^{2n} \to [-1,1]$ to be*

$$g(x;y) = \mathop{\mathbb{E}}_{z\in\{0,1\}^n} f(c' + x + (c+y)\wedge z),$$

where c, c' are some fixed vectors in $\{0,1\}^n$. Then

$$\widehat{g}(\alpha;\beta)^2 = \widehat{f}(\alpha)^2\, 1_{\{\beta\subseteq\alpha\}} 4^{-|\alpha|}.$$

Proof. This is a straightforward Fourier analytic calculation. By definition,

$$\widehat{g}(\alpha;\beta)^2 = \left(\mathop{\mathbb{E}}_{x,y,z\in\{0,1\}^n} f(c' + x + (c+y)\wedge z)\chi_\alpha(x)\chi_\beta(y)\right)^2.$$

By averaging over x it is easy to see that

$$\widehat{g}(\alpha;\beta)^2 = \widehat{f}(\alpha)^2\left(\mathop{\mathbb{E}}_{y,z\in\{0,1\}^n} \chi_\alpha((c+y)\wedge z)\chi_\beta(y)\right)^2.$$

Since the bits of y are chosen independently and uniformly at random, if $\beta\backslash\alpha$ is nonempty, the above expression is zero. So we can write

$$\widehat{g}(\alpha;\beta)^2 = \widehat{f}(\alpha)^2\, 1_{\{\beta\subseteq\alpha\}} \left(\prod_{i\in\alpha\backslash\beta} \mathop{\mathbb{E}}_{y_i,z_i} (-1)^{(c_i+y_i)\wedge z_i} \cdot \prod_{i\in\beta} \mathop{\mathbb{E}}_{y_i,z_i} (-1)^{(c_i+y_i)\wedge z_i+y_i} \right)^2 .$$

It is easy to see that the term $\mathbb{E}_{y_i,z_i}(-1)^{(c_i+y_i)\wedge z_i}$ evaluates to $\frac{1}{2}$ and the term $\mathbb{E}_{y_i,z_i}(-1)^{(c_i+y_i)\wedge z_i+y_i}$ evaluates to $(-1)^{c_i}\frac{1}{2}$. Thus

$$\widehat{g}(\alpha;\beta)^2 = \widehat{f}(\alpha)^2 \;\; 1_{\{\beta\subseteq\alpha\}}\, 4^{-|\alpha|}$$

as claimed. □

Now we prove Lemma 3.7.

Proof of Lemma 3.7. Let p be the acceptance probability of H-test. Suppose that $2^{k-|E|}+\epsilon \leq p$. We want to show that there are two functions f_a and f_b such that for some $i\in[n]$, some fixed positive integer w, some constant $\epsilon'>0$, it is the case that $\mathrm{I}_i^{\leq w}(f_a), \mathrm{I}_i^{\leq w}(f_b) \geq \epsilon'$. As usual we first arithmetize p. We write

$$p = \sum_{v\in\{0,1\}^k} \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \prod_{i\in[k]} \frac{1+(-1)^{v_i} f_i(y_i)}{2} \prod_{e\in E} \frac{1+\mathrm{Acc}(\{x_i,y_i,v_i,z_i\}_{i\in e}, z_e)}{2},$$

where

$$\begin{aligned}\mathrm{Acc}(\{x_i,y_i,v_i,z_i\}_{i\in e}, z_e) \;&=\; \prod_{i\in e}\left[f_i(x_i+(v_i\vec{1}+y_i)\wedge z_i)\right] \\ &\quad\cdot f_e\left(\sum_{i\in e} x_i + \left(\Sigma_{i\in e}(v_i\vec{1}+y_i)\right)\wedge z_e\right).\end{aligned}$$

For each $i\in[k]$, f_i is folded, so $(-1)^{v_i}f_i(y_i) = f_i(v_i\vec{1}+y_i)$. Since the vectors $\{y_i\}_{i\in[k]}$ are uniformly and independently chosen from $\{0,1\}^n$, for a fixed $v\in\{0,1\}^k$, the vectors $\{v_i\vec{1}+y_i\}_{i\in[k]}$ are also uniformly and independently chosen from $\{0,1\}^n$. So we can simplify the expression for p and write

$$p = \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \left[\prod_{i\in[k]} (1+f_i(y_i)) \prod_{e\in E} \frac{1+(\mathrm{Acc}\{x_i,y_i,\vec{0},z_i\}_{i\in e}, z_e)}{2} \right].$$

Instead of writing $\mathrm{Acc}(\{x_i,y_i,\vec{0},z_i\}_{i\in e},z_e)$, for convenience we shall write $\mathrm{Acc}(e)$ to be a notational shorthand. Observe that since $1+f_i(y_i)$ is either 0 or 2, we may write

$$p \leq 2^k \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \left[\prod_{e\in E} \frac{1+\mathrm{Acc}(e)}{2} \right].$$

Note that the product of sums $\prod_{e\in E}\frac{1+\mathrm{Acc}(e)}{2}$ expands into a sum of products of the form

$$2^{-|E|}\left(1+\sum_{\emptyset\neq E'\subseteq E}\;\prod_{e\in E'}\mathrm{Acc}(e)\right),$$

so we have

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \left[2^{-|E|} \sum_{\emptyset \neq E' \subseteq E} \prod_{e \in E'} \mathrm{Acc}(e) \right].$$

By averaging, there must exist some nonempty subset $E' \subseteq E$ such that

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \left[\prod_{e \in E'} \mathrm{Acc}(e) \right].$$

Let Odd consists of the vertices in $[k]$ with odd degree in E'. Expanding out the definition of $\mathrm{Acc}(e)$, we can conclude

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{\{x_i\},\{y_i\},\{z_a\}} \left[\prod_{i \in \mathrm{Odd}} f_i(x_i + y_i \wedge z_i) \cdot \prod_{e \in E'} f_e \left(\sum_{i \in e} x_i + \left(\sum_{i \in e} y_i \right) \wedge z_e \right) \right].$$

We now define a family of functions that represent the "noisy versions" of f_a. For $a \in [k] \cup E$, define $g'_a : \{0,1\}^{2n} \to [-1,1]$ to be

$$g'_a(x;y) = \mathop{\mathbb{E}}_{z \in \{0,1\}^n} f_a(x + y \wedge z).$$

Thus we have

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{\{x_i\},\{y_i\}} \left[\prod_{i \in \mathrm{Odd}} g'_i(x_i; y_i) \cdot \prod_{e \in E'} g'_e \left(\sum_{i \in e} x_i; \sum_{i \in e} y_i \right) \right].$$

Following the approach in [11, 21], we are going to reduce the analysis of the iterated test to one hyperedge. Let d be the maximum size of an edge in E', and without loss of generality, let $(1, 2, \ldots, d)$ be a maximal edge in E'. Now, fix the values of $x_{d+1}, \ldots, x_k$ and $y_{d+1}, \ldots, y_k$ so that the following inequality holds:

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{x_1,y_1,\ldots,x_d,y_d} \left[\prod_{i \in \mathrm{Odd}} g'_i(x_i; y_i) \cdot \prod_{e \in E'} g'_e \left(\sum_{i \in e} x_i; \sum_{i \in e} y_i \right) \right]. \tag{3.1}$$

We group the edges in E' based on their intersection with $(1, \ldots, d)$. We rewrite Inequality 3.1 as

$$\frac{\epsilon}{2^k} \leq \mathop{\mathbb{E}}_{(x_1,y_1),\ldots,(x_d,y_d) \in \{0,1\}^{2n}} \left[\prod_{S \subseteq [d]} \prod_{a \in \mathrm{Odd} \cup E' : a \cap [d] = S} g_a \left(\sum_{i \in S} x_i; \sum_{i \in S} y_i \right) \right], \tag{3.2}$$

where for each $a \in [k] \cup E$, $g_a(x;y) = g'_a(c'_a + x; c_a + y)$, with $c'_a = \sum_{i \in a \setminus [d]} x_i$ and $c_a = \sum_{i \in a \setminus [d]} y_i$ fixed vectors in $\{0,1\}^n$.

By grouping the edges based on their intersection with $[d]$, we can rewrite Inequality 3.2 as

$$\begin{aligned} \frac{\epsilon}{2^k} &\leq \mathop{\mathbb{E}}_{(x_1,y_1),\ldots,(x_d,y_d) \in \{0,1\}^{2n}} \left[\prod_{S \subseteq [d]} G_S \left(\sum_{i \in S} (x_i; y_i) \right) \right] \\ &= \left\langle \{G_S\}_{S \subseteq [d]} \right\rangle_{LU_d}, \end{aligned}$$

where G_S is simply the product of all the functions g_a such that $a \in \mathrm{Odd} \cup E'$ and $a \cap [d] = S$.

Since $(1, \ldots, d)$ is maximal, all the other edges in E' do not contain $(1, \ldots, d)$ as a subset. Thus $G_{[d]} = g_{[d]}$ and $\mathbb{E}\, G_{[d]} = 0$. By Lemma 2.2, the linear Gowers inner product of a family of functions $\{G_S\}$ being positive implies that two functions from the family must share a variable with positive influence. More precisely, there exist $S \neq T \subseteq [d]$, $i \in [2n]$, $\tau > 0$, such that $\mathrm{I}_i(G_S), \mathrm{I}_i(G_T) \geq \tau$, where $\tau = \frac{\epsilon^4}{2^{O(d)}}$.

Note that $G_\emptyset$ is the product of all the functions g'_a that are indexed by vertices or edges outside of $[d]$. So $G_\emptyset$ is a constant function, and all of its variables clearly have influence 0. Thus neither S nor T is empty. Since G_S and G_T are products of at most 2^k functions, by Lemma 2.1 there must exist some $a \neq b \in [d] \cup E'$ such that $\mathrm{I}_i(g_a), \mathrm{I}_i(g_b) \geq \frac{\tau}{2^{2k}}$. Recall that we have defined $g_a(x; y)$ to be $\mathbb{E}_z\, f_a(c'_a + x + (c_a + y) \wedge z)$. Thus we can apply Proposition 3.8 to obtain

$$\begin{aligned} \mathrm{I}_i(g_a) &= \sum_{(\alpha,\beta)\in\{0,1\}^{2n}; i\in(\alpha,\beta)} \widehat{g}_a(\alpha;\beta)^2 \\ &= \sum_{\alpha\in\{0,1\}^n; i\in\alpha} \sum_{\beta\subseteq\alpha} \hat{f}_a(\alpha)^2\, 4^{-|\alpha|} \\ &= \sum_{\alpha\in\{0,1\}^n; i\in\alpha} \hat{f}_a(\alpha)^2\, 2^{-|\alpha|}. \end{aligned}$$

Let w be the least positive integer such that $2^{-w} \leq \frac{\tau}{2^{2k+1}}$. Then it is easy to see that $\mathrm{I}_i^{\leq w}(f_a) \geq \frac{\tau}{2^{2k+1}}$. Similarly, $\mathrm{I}_i^{\leq w}(f_b) \geq \frac{\tau}{2^{2k+1}}$ as well. Hence this completes the proof. □

4 Acknowledgments

I am grateful to Alex Samorodnitsky for many useful discussions and his help with the Gowers norm. I also thank Madhu Sudan for his advice and support and Swastik Kopparty for an encouraging discussion during the initial stage of this research.

References

[1] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[2] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[3] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing*, 27(3):804–915, 1998.

[4] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[5] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 344–353, New York, NY, USA, 2006. ACM.

[6] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum csp. In *STACS*, pages 194–205, 2005.

[7] W. T. Gowers. A new proof of Szemerédi's theorem. *Geom. Funct. Anal.*, 11(3):465–588, 2001.

[8] Venkatesan Guruswami, Daniel Lewin, Madhu Sudan, and Luca Trevisan. A tight characterization of NP with 3 query PCPs. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 8, Washington, DC, USA, 1998. IEEE Computer Society.

[9] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[10] Johan Håstad and Subhash Khot. Query efficient PCPs with perfect completeness. *Theory of Computing*, 1(7):119–148, 2005.

[11] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Struct. Algorithms*, 22(2):139–160, 2003.

[12] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 767–775, New York, NY, USA, 2002. ACM.

[13] Subhash Khot and Rishi Saket. A 3-query non-adaptive PCP with perfect completeness. In *CCC '06: Proceedings of the 21st Annual IEEE Conference on Computational Complexity*, pages 159–169, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Ryan O'Donnell and Yi Wu. 3-bit dictator testing: 1 vs. 5/8. In *SODA '09: Proceedings of the Nineteenth Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 365–373, Philadelphia, PA, USA, 2009. Society for Industrial and Applied Mathematics.

[15] Ryan O'Donnell and Yi Wu. Conditional hardness for satisfiable-3csps. In *STOC '09: Proceedings of the forty-first annual ACM symposium on Theory of computing*, page To appear, 2009.

[16] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2002.

[17] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[18] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[19] Alex Samorodnitsky. Low-degree tests at large distances. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 506–515, New York, NY, USA, 2007. ACM.

[20] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 191–199, New York, NY, USA, 2000. ACM.

[21] Alex Samorodnitsky and Luca Trevisan. Gowers uniformity, influence of variables, and PCPs. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 11–20, New York, NY, USA, 2006. ACM.

[22] Madhu Sudan and Luca Trevisan. Probabilistically checkable proofs with low amortized query complexity. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 18, Washington, DC, USA, 1998. IEEE Computer Society.

[23] Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arith.*, 27:199–245, 1975.

[24] Terence Tao. Structure and randomness in combinatorics. In *FOCS '07: Proceedings of the forty-eighth annual ACM symposium on Foundations of computer science*, pages 3–15, New York, NY, USA, 2007. ACM.

[25] Luca Trevisan. Recycling queries in PCPs and in linearity tests (extended abstract). In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 299–308, New York, NY, USA, 1998. ACM.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/