Electronic Colloquium on Computational Complexity, Report No. 6 (2003)

ECCC

3CNF Properties are Hard to Test

Eli Ben-Sasson*
eli@eecs.harvard.edu

Prahladh Harsha†
prahladh@mit.edu

Sofya Raskhodnikova‡
sofya@mit.edu

Abstract

For a boolean formula φ on n variables, the associated property P_φ is the collection of n-bit strings that satisfy φ. We prove that there are 3CNF properties that require a linear number of queries, even for adaptive tests. This contrasts with 2CNF properties that are testable with $O(\sqrt{n})$ queries [7]. Our results add two novel observations to the recent lower bounds of Bogdanov, Obata and Trevisan [3]. First, notice that deciding P_φ is easy once *all* the input is read. Thus, property testing can be hard even for easily computable properties. Second, for every bad instance (i.e. an assignment that does not satisfy φ) there is a 3-bit query that proves this fact. Nevertheless, we show that finding such a short witness requires a linear number of queries.

We provide a general characterization of linear properties that are hard to test, and in the course of the proof include a couple of observations which are of independent interest.

1. In the context of linear property testing, adaptive 2-sided error tests have no more power than non-adaptive 1-sided error tests.
2. Random linear codes with linear distance and constant rate are very far from being locally testable.

*Department of Engineering and Applied Sciences, Harvard University and Laboratory for Computer Science Massachusetts Institute of Technology Cambridge, MA 02139. Supported by NSF grants CCR 0133096,CCR 9877049, CCR 9912342, CCR 0205390 and NTT Award MIT 2001-04.

†Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. Supported in part by NSF Award CCR 9912342 and NTT Award MIT 2001-04

‡Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

ISSN 1433-8092

1 Introduction

1.1 Property Testing

Property testing deals with a relaxation of decision problems where one must determine whether an input belongs to a particular set, called *property*, or is far from it. "Far" usually means that many characters of the input have to be modified to obtain an element in the set. Property testing was first formulated by Rubinfeld and Sudan [15] in the context of linear functions and applied to combinatorial objects, especially graphs, by Goldreich, Goldwasser and Ron [10]. This has recently become quite an active research area, see [14, 5] for surveys on the topic.

One of the important questions in property testing is characterizing properties that can be tested with a sub-linear number of queries into the input. A series of works identified classes of properties testable with constant query complexity. Goldreich et al. [10] found many such properties. Alon et al. [2] put all regular languages in that category. Their result was extended by Newman [13] to properties that can be computed by oblivious read-once constant-width branching programs. Fischer and Newman [8] demonstrated a property computable by a read-twice constant-width branching program which required super-constant query complexity, thus showing that Newman's result does not generalize to read-twice branching programs. Several papers [1, 6] worked on the logical characterization of graph properties testable with a constant number of queries.

1.2 Testing kCNF Properties

Every property over the binary alphabet can be represented as a Boolean formula, which in turn can be converted to a CNF form. Thus, testing a property over the binary alphabet can be viewed as testing whether a given assignment to Boolean variables is close to one that satisfies a fixed CNF formula. Since we know that there exist properties over the binary alphabet which require testing algorithms to read a linear portion of the input [10], testing assignments to general CNF formulas is hard. A natural question is whether restricting CNF formulas to have a constant number of variables k per clause allows for faster testers. At first sight there is hope for obtaining good testers in this case, because for any assignment that does not satisfy the formula there exists a set of k queries that witnesses this fact. Moreover, reading all the input easily decides the problem. Indeed, Fischer et al. [7] prove that properties expressible as sets of satisfying assignments to 2CNF formulas are testable with $O(\sqrt{n})$ queries, where n is the length of the input. This work left open the question of property testing of kCNFs for $k > 2$.

1.3 Our Results

In this paper we show that testing properties defined by 3CNF formulas requires a *linear* number of queries. Thus, we present a gap between 2CNFs and 3CNFs. We show the existence of families of 3CNF formulas which require a linear number of queries. Our lower bound applies to *adaptive* tests, i.e. tests where queries might depend on the answers to previous queries. This gives a class of properties which are easy to decide exactly (linear time), but are hard to test.

The hard 3CNF property we use is a vector space $V \subset \{0,1\}^n$ that can be expressed as the set of solutions to a homogeneous 3LIN formula. We give a general characterization of spaces that are hard to test (definition 1), which may be of independent interest. Intuitively, V is hard to test if

the dual space $V^\perp$ has a basis under which every vector of $V^\perp$ of small weight is a sum of a tiny fraction of the basis. Thus, querying the input on a vector in the dual space of small weight can only "detect" a small number of falsified constraints.

The vector spaces we use are built upon random (c, d)-regular error correcting codes. These codes, introduced by Gallager [9], are known to achieve constant rate and linear minimal distance. We show that this important class of codes is not locally testable[1] by a long shot. Moreover, the property that makes random codes so good in terms of minimal distance, namely expansion, is also behind the poor testability of these codes. This sheds some light on the important open question of the existence of error correcting codes that achieve *(i)* constant rate; *(ii)* linear distance; and *(iii)* are locally testable (for more information on this question, and the best constructions of locally testable codes, see Goldreich and Sudan [11]).

In the course of proving the lower bound for adaptive tests we show that for linear properties every adaptive 2-sided error test can be converted to a non-adaptive 1-sided error test with the same query complexity and essentially identical error parameters (Theorem 3). This result seems also of independent interest.

We shortly discuss the connection of this paper to other results. There are two published linear lower bounds for property testing. One is the abovementioned generic bound due to Goldreich et al. [10] and the other is for testing 3-coloring in bounded degree graphs due to Bogdanov, Obata and Trevisan [3]. The properties for which the lower bounds are proved are hard to decide even if *all* the input is read. Thus, one might conjecture that a property that is easy to decide is also easy to test. Our result rules this out.

There is a simple and elegant unpublished linear lower bound observed by Sudan [17]. His property consists of polynomials over $\mathbb{F}_n$ of degree at most $n/2$ where each polynomial is given by its evaluation on all elements of the field. It is not hard to see that every non-adaptive 1-sided error test for this property requires linear query complexity. Since the property of low-degree polynomials is linear, our reduction from general to non-adaptive 1-sided error tests implies a linear lower bound for adaptive 2-sided tests for this property. Observe that this property is easy to decide once all the input is read, but is not expressible by a family of 3CNF formulas.

Both linear lower bounds of [17] and [3] capitalize on the existence of inputs that are far from having the property, yet *any* local view of a constant fraction of them can be extended to an element having the property[2]. But if the property is defined by a kCNF φ this cannot happen. For, clearly, any string that does not have the property must falsify at least one clause of φ. Thus, there is some view of the input of size k, that proves the input does not have the property. Our result shows that in certain cases, finding such a falsified clause requires reading a constant fraction of the input, even if the assignment is far from any satisfying one.

1.4 Paper Organization

After required definitions (section 2), we present a self contained proof of the main Theorem in section 3. The proofs of the claims needed for the proof follow in sections 4-7.

[1] A code $\mathcal{C}$ is *locally testable* if the property $\mathcal{C}$ is testable with constant query complexity.

[2] E.g. in Sudan's example any evaluation of a polynomial on d points can be extended to an evaluation of a polynomial of degree $d' > d$. Thus, seeing $n/2 - 1$ values of the polynomial still does not mean the polynomial has degree $n/2$.

2 Definitions

Property testing

A *property* is a collection of strings of a fixed size n. A property is *linear* if it forms a vector space. In this paper, strings are over binary alphabet unless mentioned otherwise. The distance $dist(x, \mathcal{P})$ of a string x to a property $\mathcal{P}$ is $\min_{x' \in \mathcal{P}} dist(x, x')$, where $dist(x, x')$ denotes the Hamming distance between the two strings. The *relative distance* of x to $\mathcal{P}$ is its distance to $\mathcal{P}$ divided by n. A string is ε-far from $\mathcal{P}$ if its relative distance to $\mathcal{P}$ is at least ε.

A *test for property $\mathcal{P}$ with distance parameter ε, error μ and query complexity q* is a probabilistic algorithm that queries at most q bits of the input, accepts strings in $\mathcal{P}$ with probability at least $1-\mu$ and accepts strings that are ε-far from $\mathcal{P}$ with probability at most μ, for some $0 < \mu < \frac{1}{2}$. A test with distance parameter ε and error μ is referred to as an *(ε, μ)-test*. A property is *(ε, μ, q)-testable* if it has an (ε, μ)-test that asks at most q queries on every input.

A couple of special classes of tests are of interest. An algorithm is *non-adaptive* if it asks all queries in advance, before getting the answers. Namely, a query may not depend on the answers to previous queries. An algorithm has *1-sided error* if it always accepts an input that has the property.

CNF and linear formulas

Recall that a Boolean formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, where every clause is a disjunction of literals. (A literal is a Boolean variable or a negated Boolean variable.) If all clauses contain at most three literals, the formula is a 3CNF.

A *linear* (LIN) Boolean formula is a conjunction of constraints, where every constraint is satisfied if and only if the variables in the constraint add up to 0 mod 2. If all constraints contain at most d literals, the formula is a dLIN.

Let φ be a formula on n variables. An n-bit string *satisfies* φ if it satisfies all clauses (constraints) of the formula. An n-bit string is *ε-far* from satisfying φ if at least an ε fraction of the bits need to be changed to make the string satisfy φ. Each formula φ defines a property $SAT(\varphi) = \{x|\ x \text{ satisfies } \varphi\}$. For brevity, we refer to a test for $SAT(\varphi)$ as a test for φ.

3 Main Theorem

In this section we state and prove the main theorem.

Theorem 1 (Main) *There exist $0 < \delta, \varepsilon < 1$, $0 < \mu < \frac{1}{2}$ such that for every sufficiently large n, there is a* 3CNF *formula φ on n variables that requires δn queries for every adaptive (ε, μ)-test.*

The hard properties of Theorem 1 are simply linear spaces over $\{0,1\}^n$. Each of them is the space of solutions of a homogeneous system of 3LIN equations. Consider a vector space V over $\{0,1\}^n$. Denote the dual space by $V^\perp$. Let $\mathcal{A} = (A_1, \ldots, A_m)$ be a basis for $V^\perp$. For $x \in \{0,1\}^n$, let $|x| = \sum_{i=1}^n x_i$ (regular addition, not mod 2). By definition, $V = \{x | \langle x, A_i \rangle = 0 \text{ for all } A_i \in \mathcal{A}\}$. Thus, viewing each A_i as a *constraint*, we can represent V as a dLIN formula where $d = \max_{A_i \in \mathcal{A}} |A_i|$. Since $\mathcal{A}$ is linearly independent, each $\alpha \in V^\perp$ has a unique representation as a sum of vectors in $\mathcal{A}$.

We now present sufficient conditions for a vector space to be hard to test.

Definition 1 (Hard Linear Properties) *Let V be a vector space over $\{0,1\}^n$ and let $\mathcal{A}$ be a basis for $V^\perp$. Fix $0 < \varepsilon, \mu < 1$.*

- *$\mathcal{A}$ is* **ε-separating** *if every $x \in \{0,1\}^n$ that falsifies exactly one constraint in $\mathcal{A}$ has $|x| \geq \varepsilon n$.*
- *$\mathcal{A}$ is* **(q,μ)-local** *if every $\alpha \in \{0,1\}^n$ that is a sum of at least μm vectors in $\mathcal{A}$ has $|\alpha| \geq q$.*

Intuitively, we use the first condition to form a distribution over words that are far from the vector space. The second condition means that to "get information" about a fraction μ of the constraints in $\mathcal{A}$, a test needs at least q queries. **Proof (of Main Theorem 1):** Our first step is to show that any linear space conforming to definition 1 is hard to test by "simple" tests, i.e. ones that have 1-sided error and are non-adaptive.

Theorem 2 (Non-Adaptive 1-Sided Error) *Fix $0 < \varepsilon < 1$, $0 < \mu < \frac{1}{2}$. Let V be a vector space over $\{0,1\}^n$. If $V^\perp$ has an ε-separating (q,μ)-local basis $\mathcal{A} = (A_1, \ldots, A_m)$, then every non-adaptive 1-sided error $(\varepsilon, 1-2\mu)$-test for V requires q queries.*

Theorem 2 is proved in section 4. Next we show that 2-sided error adaptive tests are no more powerful than 1-sided error non-adaptive ones, whenever the property we test is linear. This theorem is of independent interest as it refers to any testing of a linear property.

Theorem 3 (Adaptive 2-Sided Error) *Let V be a vector space over $\{0,1\}^n$. For every adaptive (ε, μ, q)-test T for V, there is a 1-sided error non-adaptive $(\varepsilon, 2\mu, q)$-test T' for V.*

The proof of Theorem 3 appears in section 5. The theorem is proved in two steps. We first reduce 2-sided error adaptive tests to 1-sided error adaptive ones, while doubling the error (Theorem 10). Then we show that 1-sided error adaptive tests are no better than non-adaptive ones (Theorem 11). The reduction doubles the error because it preserves the difference between the completeness (acceptance probability of strings in V, $1-\mu$) and soundness (acceptance probability of strings far from being in V, μ). A different (Fourier based) approach to proving Theorems 2 and 3 appears in appendix B. So far we have shown that any linear property conforming to definition 1 is hard to test, even by adaptive, 2-sided error tests. The following theorem assures us that such linear properties exist. The proof of this theorem, which uses the probabilistic method, appears in section 6.

Theorem 4 (Hard Linear Properties Exist) *There exist integer $d > 0$ and constants μ, ε, δ, such that for all sufficiently large n there is a collection $\mathcal{A}_n \subset \{0,1\}^n$ of vectors of weight at most d which is linearly independent, ε-separating and $(\delta n, \mu)$-local.*

We now have dLIN formulas that are hard to test. The following reduction brings d down to 3 while preserving the properties of definition 1 (with smaller constants).

Theorem 5 (Reduction Theorem) *Every linearly independent, ε-separating, $(\delta n, \mu)$-local $\mathcal{A} \subset \{0,1\}^n$ of vectors of weight at most d can be converted to a linearly independent, $\varepsilon^\star$-separating, $(\delta^\star n^\star, \mu^\star)$-local $\mathcal{A}^\star \subset \{0,1\}^{n^\star}$ of vectors of weight at most 3. If ε, δ, μ are strictly positive constants, so are $\varepsilon^\star, \delta^\star, \mu^\star$.*

Theorem 5 is proved in section 7. Recall that a 3LIN formula can be defined by a 3CNF. This completes the proof of the main Theorem 1. ∎

4 Lower Bounds for Non-Adaptive 1-Sided Error Tests

In this section we give the proof of Theorem 2. **Proof (of Theorem 2):** We employ Yao's minimax principle. It states that to prove that every q-query randomized test fails with probability more than δ it is enough to exhibit a distribution $\mathcal{B}$ on the inputs for which every q-query deterministic test fails with probability more than δ.

For $i = 1 \ldots m$ let $\mathcal{B}_i$ be the uniform distribution over n-bit strings that falsify constraint A_i and satisfy the rest. The distribution $\mathcal{B}$ is the uniform distribution over $\mathcal{B}_i$'s. Lemma 6 shows that distribution $\mathcal{B}$ is over strings which are ε-far from V. Lemma 8 proves that every low complexity deterministic test is likely to fail on $\mathcal{B}$.

Lemma 6 *If $\mathcal{A}$ is ε-separating, each string x falsifying exactly one constraint in $\mathcal{A}$ is ε-far from V.*

Proof: Let $y \in V$. Then $x + y$ falsifies exactly one constraint in $\mathcal{A}$. Since $\mathcal{A}$ is ε-separating, $dist(x, y) = |x + y| \geq \varepsilon n$. By definition, $dist(x, V) \geq \varepsilon n$. ∎

We say that a test *detects a violation* if there is no string in V that is consistent with the answers to the queries. By linearity, we obtain the following claim.

Claim 7 *A test detects a violation if and only if there is a linear combination α of constraints in $\mathcal{A}$ such that $\langle x, \alpha \rangle = 1$ for all $x \in V$ which are consistent with the answers to the queries.*

Observe that an 1-sided error test has to accept a string if it does not detect a violation.

Lemma 8 *Let T be a deterministic 1-sided error non-adaptive test with $< q$ queries. If $\mathcal{A}$ is (q, μ)-local then $\Pr_{x \leftarrow \mathcal{B}}[T(x) = 0] < 2\mu$.*

Note that if $\mathcal{A}$ is not (q, μ)-local, there is a test T with $< q$ queries and $\Pr_{x \leftarrow \mathcal{B}}[T(x) = 0] \geq \mu$. Let $\alpha = \sum_{A \in C} A$ where $C \subseteq \mathcal{A}$, $|C| \geq \mu m$ and $|\alpha| < q$. Then T is the test that queries 1-coordinates of α. It detects a violation whenever the only falsified constraint is in C.

Proof (of Lemma 8): Let Q be the set of queries posed by T. A query to variable x_i is viewed as a vector in $\{0, 1\}^n$ which is 1 at coordinate i and 0 everywhere else. The only constraints in $V^\perp$ that T can check for violation are the ones that can be expressed as a linear combination of the queries. Call this set of constraints $cons(Q)$, i.e. $cons(Q) = \{\alpha \in V^\perp | \alpha \in Span(Q)\}$.

For each $\alpha \in cons(Q)$, let C_α be the set of constraints in $\mathcal{A}$ whose sum is α. This set is well defined as the set of constraints $\mathcal{A}$ is linearly independent. Let Γ be the family of sets C_α, where $\alpha \in cons(Q)$. Thus, the only violations that T detects are the violations caused by falsifying some constraint in $\bigcup_{C \in \Gamma} C$. Namely, $\Pr_{x \leftarrow \mathcal{B}}[T(x) = 0] = \frac{1}{m} \sum_i \Pr_{\mathcal{B}_i}[T(x) = 0] = \frac{1}{m} \left| \bigcup_{C \in \Gamma} C \right|$.

It remains to show $\left| \bigcup_{C \in \Gamma} C \right| < 2\mu m$. We observe that if $\alpha_1, \alpha_2 \in cons(Q)$, so does $\alpha_1 + \alpha_2$. In terms of C_α's this implies that if $C_1, C_2 \in \Gamma$, so is $C_1 \triangle C_2$[3]. Since $|Q| < q$ and $\mathcal{A}$ is (q, μ)-local, $|C_\alpha| \leq \mu m$. We can now apply Lemma 9 to conclude that $\left| \bigcup_{C \in \Gamma} C \right| < 2\mu m$. ∎

Theorem 2 follows from Lemmas 6 and 8. ∎

Lemma 9 *Let $\Gamma = \{C | C \subseteq [m]\}$ be a family of subsets of $[m]$ such that Γ is closed under symmetric difference and for all sets C in Γ, $|C| \leq w$. Then $\left| \bigcup_{C \in \Gamma} C \right| < 2w$.*

[3]For sets A, B, the symmetric difference of A and B, $A \triangle B = \{x | x \in A \text{ and } x \notin B\} \cup \{x | x \notin A \text{ and } x \in B\}$.

Proof: Suppose $x \in C$ for some $C \in \Gamma$. We then observe that for any set C' in Γ (including C) either $x \in C'$ or $x \in C \triangle C'$ but not both. Since Γ is closed under symmetric difference and $C' = C \triangle (C \triangle C')$, each element in $\bigcup_{C \in \Gamma} C$ occurs in exactly half the sets of Γ. Therefore,

$$\frac{|\Gamma|}{2} \cdot \left| \bigcup_{C \in \Gamma} C \right| = \sum_{C \in \Gamma} |C|.$$

Since $|C| \leq w$ for any $C \in \Gamma$ and the empty set belongs to Γ, we conclude that the number of elements in $\bigcup_{C \in \Gamma} C$ is less than $2w$. ∎

5 Reducing 2-Sided Error Adaptive to 1-Sided Error Non-Adaptive

In this section we prove Theorem 3 by presenting a generic reduction that converts any adaptive test with 2-sided error for linearity checking to a non-adaptive one with 1-sided error. We perform this reduction in two stages, we first reduce an adaptive test with 2-sided error to an adaptive test with 1-sided error (Theorem 10) and then reduce this to a non-adaptive test with 1-sided error (Theorem 11). The second reduction was suggested by Madhu Sudan.

Theorem 10 (2-sided to 1-sided error) *Let V be a vector space over $\{0,1\}^n$. For every adaptive (ε, μ, q)-test T for V, there is a 1-sided error adaptive $(\varepsilon, 2\mu, q)$-test T' for V.*

Theorem 11 (Adaptive to non-adaptive for 1-sided error) *Let V be a vector space over $\{0,1\}^n$. For every 1-sided error adaptive (ε, μ, q)-test T for V, there is a 1-sided error non-adaptive (ε, μ, q)-test T' for V.*

Assume without loss of generality, that associated with every test T, there exists a set of decision trees $\Upsilon_T = \{\Gamma_1, \Gamma_2, \dots\}$ and a distribution $\mathcal{D}_T$ on this set of decision trees such that on input x, T chooses a decision tree Γ with probability $\mathcal{D}_T(\Gamma)$ and then answers according to $\Gamma(x)$.

The reduction in Theorem 10 is carried out by relabeling the leaves of each decision tree Γ in $\mathcal{D}_T$ to obtain an "optimal" tree Γ', where every leaf l is labelled 0 if the path reaching l violates some constraint in $V^\perp$ and is labelled 1 otherwise. On input x, the new test picks a decision tree Γ according to $\mathcal{D}_T$ and outputs $\Gamma'(x+v)$ where v is a random vector in V. It is easy to see that the new test has 1-sided error and the same query complexity as the old one. Moreover, the modification might only increase the difference between the acceptance probability of the strings in V and that of strings far from V.

The reduction from a 1-sided error adaptive test to a non-adaptive test with 1-sided error in Theorem 11 is performed in the following manner: The non-adaptive test T' chooses a decision tree Γ from Υ_T according to the distribution $\mathcal{D}_T$ just as the adaptive test T would do. However, instead of querying according to Γ, the test T' chooses a random branch of the tree Γ and non-adaptively queries all the variables that occur along this branch. It then accepts the answers iff they do not violate any constraint in the dual space $V^\perp$.

We leave the finer details of the reductions and the proofs of Theorems 10 and 11 to Appendix A. The linearity of the space V plays a crucial role in the correctness of both reductions.

6 Random Codes Require Linear Size Queries

In this section we prove Theorem 4. In particular, we show that a random (c, d)-regular code with high probability obeys definition 1, for large enough constants c, d. We start by defining such codes, originally introduced and analyzed by Gallager [9].

6.1 Random Regular Codes

Let $G = \langle L, R, E \rangle$ be a bipartite multi-graph, with $|L| = n, |R| = m$, and let $d(v)$ be the degree of a vertex v. G is called (c, d)-*regular* if for all $v \in L$, $d(v) = c$, and for all $v \in R$, $d(v) = d$. A random (c, d)-regular graph with n left vertices and $m = \frac{c}{d}n$ right vertices, is obtained by selecting a random matching between cn "left" nodes, and $dm = cn$ "right" nodes. Collapse c consecutive nodes on the left to obtain n c-regular vertices, and collapse d consecutive nodes on the right to obtain m d-regular vertices. Notice that the resulting graph may be a multi-graph (i.e. have multiple edges between two vertices). The code associated with G is obtained by letting R define $\mathcal{C}^{\perp}$, as in the following definition.

Definition 2 *Let $G = \langle L, R, E \rangle$ be a bipartite multi-graph, with $|L| = n, |R| = m$. Associate a distinct Boolean variable x_i with any $i \in L$. For each $j \in R$, let $N(j) \subseteq L$ be the set of neighbors of j. The j'th constraint is $A_j = \sum_{i \in N(j)} x_i \bmod 2$. Let $\mathcal{A}(G)$ be the $m \times n$ matrix where the jth row of $\mathcal{A}(G)$ is A_j. The code defined by G is*

$$\mathcal{C}(G) = (\mathcal{A}(G))^{\perp} = \{x \in \{0,1\}^n | \mathcal{A}(G) \cdot x = \vec{0}\}.$$

A random (c, d)-regular code is obtained by taking $\mathcal{C}(G)$ as in the previous definition, for G a random (c, d)-regular graph.

6.2 Some Expansion Properties of Random Regular Graphs

To prove $\mathcal{C}(G)$ obeys definition 1, we use standard arguments about expansion of the random graph G. We reduce each requirement on $\mathcal{A}(G)$ to a requirement on G, and then show that the expansion of a random G implies that it satisfies the requirements. We need the following notions of neighborhood and expansion.

Definition 3 (Neighbors) *Let $G = \langle V, E \rangle$ be a graph. For $S \subset V$, let*

- *$N(S)$ be the set of neighbors of S.*
- *$N^1(S)$ be the set of* unique *neighbors of S, i.e. vertices with exactly one neighbor in S.*
- *$N^{odd}(S)$ be the set of neighbors of S with an* odd number *of neighbors in S.*

Notice that $N^1(S) \subseteq N^{odd}(S)$.

Definition 4 (Expansion) *Let $G = \langle L, R, E \rangle$ be a bipartite graph with $|L| = n, |R| = m$.*

- *G is called an* (λ, γ)-right expander *if*

$$\forall S \subset R,\ |S| \leq \gamma n,\ |N(S)| > \lambda \cdot |S|.$$

- *G is called an* (λ, γ)-right unique neighbor expander *if*

$$\forall S \subset R,\ |S| \leq \gamma n,\ |N^1(S)| > \lambda \cdot |S|.$$

- *G is called an* (λ, γ)-right odd expander *if*

$$\forall S \subset R,\ |S| \geq \gamma n,\ |N^{odd}(S)| > \lambda \cdot |S|.$$

Notice that expanders and unique neighbor expanders discuss subsets of size *at most* γn, whereas odd expanders discuss subsets of size *at least* γn. Left expanders (all three of them) are defined analogously by taking $S \subset L$ in definition 4.

The following lemmas are proved using standard techniques for analysis of expansion of random graphs, such as those appearing in e.g. [4, 16]. We defer the proofs to appendix C.

Lemma 12 *There exists a constant $r > 0$ such that for any integers $c \geq 5, d \geq 2$, a random (c,d)-regular graph is with high probability a $(1, r \cdot d^{-2})$-left unique neighbor expander.*

Lemma 13 *For any odd integer c, any constants $\mu > 0, \delta < \mu^c$, and any integer $d > \frac{2\mu c^2}{(\mu^c - \delta)^2}$, a random (c,d)-regular graph is with high probability a (δ, μ)-right odd expander.*

6.3 Random Codes Require Large Query Complexity

We are ready to prove Theorem 4.

Theorem 14 *For any odd integer $c \geq 5$, there exists an integer $d > c$, and constants $\varepsilon, \delta, \mu > 0$, such that for a random (c,d)-regular graph G, the set $\mathcal{A}(G)$ is with high probability (i) linearly independent, (ii) $(\delta n, \mu)$-local, and (iii) ε-separating.*

Proof (of Theorem 4): Fix $c = 5$. Let $d, \varepsilon, \delta, \mu$ be as in Theorem 14. The theorem follows. ∎

Proof (of Theorem 14): Given odd $c \geq 5$ we will define the constants $d, \varepsilon, \delta, \mu$ throughout the course of the proof.

(i) We need to show that adding up any subset of $\mathcal{A}(G)$ cannot yield $\vec{0}$. Since we are working modulo 2, this is equivalent to proving

$$\forall T \subseteq R, \quad N^{odd}(T) \neq \emptyset.$$

For small T we use unique neighbor expansion, and for large T we use odd neighbor expansion.

Fix c, and reverse the roles of left and right in lemma 12. We conclude the existence of constant $r > 0$, such that for any $d \geq 5$, G is whp a $(1, r \cdot c^{-2})$-right unique neighbor expander. This implies that if $|T| \leq r \cdot c^{-2} \cdot |R|$, then $N^{odd}(T) \neq \emptyset$, because $N^{odd}(T) \supseteq N^1(T)$ and $N^1(T) \neq \emptyset$.

Lemma 13 says that for any $\mu > 0$, and large enough d, all sets of size at least μm have nonempty odd neighborhood. (Actually, the lemma shows that the odd neighborhood is of linear size, which is more than what we need here.) Fixing μ, δ, d to the following values completes the proof of the first claim:

$$\mu = r \cdot c^{-2}; \quad \delta = \mu/2; \quad d > \frac{2\mu c^2}{(\mu^c - \delta)^2}.$$

(ii) Notice that if $T \subseteq R$, then $N^{odd}(T)$ is exactly the support of $\sum_{j \in T} A_j$. Thus, it suffices to show that $N^{odd}(T)$ is large for large subsets T.

By the definition of d, μ, δ from part (ii) and by lemma 13 G is whp a $(\delta n, \mu)$-right odd expander. This means $\mathcal{A}(G)$ is $(\delta n, \mu)$-local. Part *(ii)* is proved.

(iii) Let G_{-j} be the graph obtained from G by removing vertex $j \in R$ and all edges touching it. Since $\mathcal{A}(G)$ is linearly independent, it is sufficient to show that $\mathcal{C}(G_{-j})$ has no element of Hamming weight $< \varepsilon n$.

Let x be a non-zero element of $\mathcal{C}(G_{-j})$, and let $S_x \subseteq L$ be the set of coordinates at which x is 1. Consider the graph G_{-j}. In this graph, the set of unique neighbors of S_x is empty because $x \in \mathcal{C}(G_{-j})$ (otherwise, some $j' \in N^1(S_x)$, so $\langle A_{j'}, x \rangle = 1$, a contradiction.) Thus,

$$N^1(S_x) \subseteq \{j\} \tag{1}$$

where $N^1(S_x)$ is the set of unique neighbors of S_x in G. Clearly, $|S_x| > 1$, because the left degree of G is $c > 1$. But if $|S_x| \leq r \cdot d^{-2} \cdot n$ then by lemma 12 $|N^1(S_x)| \geq |S_x| > 1$, in contradiction to equation (1). We conclude that for any $x \in \mathcal{C}(G_{-j})$, $|x| \geq r \cdot d^{-2}$, so $\mathcal{A}(G)$ is ε-separating for ε satisfying:

$$\varepsilon \leq r \cdot d^{-2}.$$

Part *(iii)* is completed, and with it the theorem. ∎

7 Reducing dLIN to 3LIN

In this section we prove Theorem 5 which directly follows from the final theorem of this section. The randomized construction from section 6 produces d-linear formulas which are hard to test for some constant d. We would like to make d as small as possible. This section obtains 3-linear hard to test formulas. First we give a reduction from d-linear to $\lceil \frac{d}{2} \rceil + 1$-linear formulas, and then apply it $\log d$ times to get 3-linear formulas.

Let φ be a d-linear formula on variables in $X = \{x_1, \ldots, x_n\}$. The reduction maps φ to a $(\lceil \frac{d}{2} \rceil + 1)$-linear formula on variables $X \cup Z$ where Z is a collection of new variables $\{z_1, \ldots, z_m\}$. For each constraint c_i, say $x_1 \oplus \ldots \oplus x_d = 0$, in φ, two constraints, c_i^1 and c_i^2 are formed: $x_1 \oplus \ldots \oplus x_{\lceil \frac{d}{2} \rceil} \oplus z_i = 0$ and $x_{\lceil \frac{d}{2} \rceil + 1} \oplus \ldots \oplus x_t \oplus z_i = 0$. Let V be the vector space over $\{0,1\}^n$ of vectors satisfying φ, and let $\mathcal{A}$ be an m-dimensional basis for the vector space $V^\perp$ of constraints. Define $\mathcal{R}(\mathcal{A})$ to be the collection of $2m$ vectors over $\{0,1\}^{n+m}$ formed by splitting every constraint in $\mathcal{A}$ in two, as described above. The following three lemmas show that the reduction preserves the properties which make the formula hard to test.

Lemma 15 *$\mathcal{R}(\mathcal{A})$ is independent.*

Proof: It is enough to prove that no set of constraints in $\mathcal{R}(\mathcal{A})$ sums up to 0. Let $C \in \mathcal{R}(\mathcal{A})$. If only one of the two constraints involving a new variable z appears in C, then the sum of vectors in C has 1 in z's position. If, on the other hand, all constraints appear in pairs, then the sum of vectors in C is equal to the sum of the constraints in $\mathcal{A}$ from which C's constraints were formed. By independence of old constraints, this sum is not 0.

Lemma 16 *If $\mathcal{A}$ is ε-separating, then $\mathcal{R}(\mathcal{A})$ is ε'-separating where $\varepsilon' = \frac{\varepsilon}{1+m/n}$.*

Proof: Let x' be a vector in $\{0,1\}^{n+m}$ that falsifies exactly one constraint, say c_i^1, in $\mathcal{R}(\mathcal{A})$. Namely, $\langle x', c_i^1 \rangle = 1$ and $\langle x', c' \rangle = 0$ for all $c' \in \mathcal{R}(\mathcal{A}), c' \neq c_i^1$. Let $x = x'_1 \dots x'_n$. Then $\langle x, c_i \rangle = \langle x', c_i^1 + c_i^2 \rangle = \langle x', c_i^1 \rangle + \langle x', c_i^2 \rangle = 1$, and similarly, $\langle x, c \rangle = 0$ for all $c \in \mathcal{A}, c \neq c_i$. Thus, x falsifies exactly one constraint in $\mathcal{A}$. Since $\mathcal{A}$ is ε-separating, $|x| \geq \varepsilon n$. It follows that $|x'| \geq \varepsilon n$, implying that $\mathcal{R}(\mathcal{A})$ is $\frac{\varepsilon n}{n+m}$-separating. ∎

Lemma 17 *If $\mathcal{A}$ is (q,μ)-local, then $\mathcal{R}(\mathcal{A})$ is (q',μ')-local where $q' = \frac{2q}{d+2}$ and $\mu' = \mu + \frac{q'}{2m}$.*

Proof: Let $\alpha' \in \{0,1\}^{m+n}$ be the sum of a subset T of $\mu' \cdot 2m$ constraints in $\mathcal{R}(\mathcal{A})$. Let T_2 be the subset of constraints in T that appear in pairs. Namely, for every new variable z, both constrains with z are either in T_2 or not in T_2. Let $T_1 = T \setminus T_2$.

Case 1: $|T_1| \geq q'$. For every constraint in T_1, the new variable z from that constraint does not appear in any other constraint in T. Therefore, α' is 1 on z's coordinate. Hence, $|\alpha'| \geq |T_1| \geq q'$.

Case 2: $|T_1| < q'$. Then $|T_2| = |T| - |T_1| \geq \mu' m - q' = 2\mu m$. Let S be the set of constraints in $\mathcal{A}$ that gave rise to constraints in T_2. Then $|S| = |T_2|/2 \geq \mu m$. Old variables appear in the same number of constraints in S and in T_2. Thus,

$$\left| \sum_{c \in T_2} c \right| \geq \left| \sum_{c \in S} c \right| \geq r.$$

The last inequality follows from the fact that $\mathcal{A}$ is (q,μ)-local. When constraints from T_1 are added to $\sum_{c \in T_2} c$, each T_1 constraint zeroes out at most $\lceil \frac{d}{2} \rceil$ coordinates. It also adds at least 1 to the weight of the sum since it contains a new variable that does not appear in any other constraints in T. Hence,

$$|\alpha'| \geq \left| \sum_{c \in T_2} c \right| - \frac{d}{2} \left| \sum_{c \in T_1} c \right| \geq q - \frac{d}{2} q' = q'. \qquad \blacksquare$$

Now we study what happens if the reduction is applied a few times until d becomes 3.

Theorem 18 *Let V be a vector space over $\{0,1\}^n$ and let $\mathcal{A}$ be an m-dimensional basis for $V^\perp$ containing vectors of weight at most d. Let $\mathcal{A}^\star$ be a set of $m^\star$ vectors over $\{0,1\}^{n^\star}$, obtained by applying the reduction $\mathcal{R}$ $\log d$ times, until the weight of every vector is 3. If $\mathcal{A}$ is ε-separating (q,μ)-local, then $\mathcal{A}^\star$ is $\varepsilon^\star$-separating and $(q^\star,\mu^\star)$-local, where*

$$m^\star = dm; \quad n^\star = n + (d-1)m; \quad \varepsilon^\star = \frac{\varepsilon}{1 + (d-1)m/n}; \quad q^\star = \frac{2q}{d+2}; \quad \mu^\star = \mu + \frac{q}{m} \cdot \frac{d+2}{d+1}.$$

Proof: The theorem follows from lemmas 15, 16, 17.

Acknowledgements

We thank Madhu Sudan for *(i)* many helpful conversations; *(ii)* suggesting the reductions of section 5; and *(iii)* allowing us to include the "hard to test" properties based on Reed-Muller codes. We are thankful to Piotr Indyk for referring us to Azuma's inequality to simplify the analysis and to Michael Sipser for helpful discussions.

References

[1] Alon, N., Fischer, E., Krivelevich, M., and Szegedy, M. Efficient testing of large graphs. *Combinatorica 20*, 4 (2000), 451–476.

[2] Alon, N., Krivelevich, M., , Newman, I., and Szegedy, M. Regular languages are testable with a constant number of queries. *SIAM Journal of Computing 30*, 6 (2001), 1842–1862.

[3] Bogdanov, A., Obata, K., and Trevisan, L. A lower bound for testing 3-colorability in bounded-degree graphs. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science* (Vancouver, Canada, 16–19 Nov. 2002). (to appear).

[4] Chvátal, V., and Szemerédi, E. Many hard examples for resolution. *Journal of the ACM 35*, 4 (Oct. 1988), 759–768.

[5] Fischer, E. The art of uninformed decisions: A primer to property testing. *Bulletin of the European Association for Theoretical Computer Science 75* (Oct. 2001), 97–126. The Computational Complexity Column.

[6] Fischer, E. Testing graphs for colorability properties. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms* (New York, 7–9 Jan. 2001), pp. 873–882.

[7] Fischer, E., Lehman, E., Newman, I., Raskhodnikova, S., Rubinfeld, R., and Samorodnitsky, A. Monotonicity testing over general poset domains. In *Proc. 34th ACM Symp. on Theory of Computing* (New York, 19–21 May 2002), pp. 474–483.

[8] Fischer, E., and Newman, I. Functions that have read-twice, constant width, branching programs are not necessarily testable. In *Proc. 17th Conference on Computational Complexity* (Montréal, Québec, Canada, 21–24 May 2002), pp. 73–77.

[9] Gallager, R. G. *Low Density Parity Check Codes.* MIT Press, Cambridge, MA, 1963.

[10] Goldreich, O., Goldwasser, S., and Ron, D. Property testing and its connection to learning and approximation. *Journal of the ACM 45*, 4 (July 1998), 653–750.

[11] Goldreich, O., and Sudan, M. Locally testable codes and PCPs of almost linear length. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science* (Vancouver, Canada, 16–19 Nov. 2002). (to appear).

[12] Motwani, R., and Raghavan, P. *Randomized Algorithms.* Cambridge University Press, Cambridge, 1995.

[13] Newman, I. Testing membership in languages that have small width branching programs. *SIAM Journal of Computing 31*, 5 (2002), 1557–1570.

[14] Ron, D. Property testing (a tutorial). In *Handbook of Randomized Computing* (2001), S. Rajasekaran, P. M. Pardalos, J. H. Reif, and J. D. Rolim, Eds., vol. 9 of *Combinatorial Optimization*, Kluwer Academic Publishers, pp. 597–649.

[15] Rubinfeld, R., and Sudan, M. Robust characterizations of polynomials with applications to program testing. *SIAM Journal of Computing 25*, 2 (Apr. 1996), 252–271.

[16] Spielman, D. *Computationally Efficient Error-Correcting Codes and Holographic Proofs.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[17] Sudan, M. Personal Communication.

A Proofs of Theorems 10 and 11

We can assume the following regarding the behavior of any adaptive test, without loss of generality. Associated with T, there exists a set of decision trees $\Upsilon_T = \{\Gamma_1, \Gamma_2, \dots\}$ and a distribution $\mathcal{D}_T$ on this set of decision trees such that on any input x, T chooses a decision tree Γ with probability $\mathcal{D}_T(\Gamma)$ and then answers according to $\Gamma(x)$. Theorem 10 is proved in Section A.1 while Theorem 11 is proved in Section A.2.

A.1 2-Sided to 1-Sided Error

Let $V \subseteq \{0,1\}^n$ be a vector space. As before, let $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$ be a basis for the dual space $V^\perp$. The space $\{0,1\}^n$ can be partitioned into 2^m sets as follows: For each $S \subset \mathcal{A}$, let V_S be the set of vectors that violate all the constraints in S and satisfy all other constraints in $\mathcal{A}$. In this notation, $V_\emptyset = V$. From the above definitions, it follows that if $x \in V_S$ for some $S \subseteq \mathcal{A}$, then $V_S = x + V$.

For any subset S of $\mathcal{A}$ and any test T, define $\rho_T(S)$ as follows:

$$\rho_T(S) = \operatorname*{average}_{y \in V_S} \big(\Pr[T(y) = 1]\big)$$

For notational brevity, we denote $\rho_T(\emptyset)$, the average acceptance probability of strings in V, by ρ_T.

Let T be 2-sided error (adaptive) (ε, μ, q)-test for V. Let Υ_T and $\mathcal{D}_T$ be the associated set of decision trees and the corresponding distribution respectively. For any decision tree Γ in Υ_T, we define an optimal labeling of its leaves which converts the 2-sided error test T into one with 1-sided.

Definition 5 *Let Γ be a decision tree. Γ' is the decision tree with the same queries as Γ, and the following leaf label for any leaf l: l is labelled 0 if the path reaching l violates some constraint in $V^\perp$ and is labelled 1 otherwise.*

We now define the 1-sided error test T' corresponding to T.

Definition 6 (1-Sided Error Test) *Given a 2-sided error (adaptive) test T for V, define the test T' as follows: On input x, T' chooses a decision tree Γ according to the distribution $\mathcal{D}_T$ as T does. It then chooses a random $v \in V$ and answers according to $\Gamma'(x+v)$.*

Clearly, T' is a 1-sided error test as it never rejects unless it detects a violation. Also, T' has the same query complexity as T. The only fact that needs to be verified is that the error of T' is at most 2μ.

Lemma 19 *For any non-empty set $S \subseteq \mathcal{A}$, $\rho_T - \rho_T(S) \leq \rho_{T'} - \rho_{T'}(S)$.*

Proof: If every decision tree Γ in Υ_T is labelled optimally as mentioned in Definition 5, then the lemma is true. Otherwise, there exists a decision tree Γ_1 in Υ_T and a leaf l in Γ_1 such that l is not labelled optimally. Let $\tilde{\Gamma_1}$ be the decision tree formed by changing the label of label l in Γ_1. And let $\tilde{T}$ be the test defined as follows: On input x, $\tilde{T}$ chooses a decision tree Γ according to the distribution $\mathcal{D}_T$ as T does. It then chooses a random $v \in V$. If $\Gamma = \Gamma_1$ it answers according to $\tilde{\Gamma_1}(x+v)$ else it answers according to $\Gamma(x+v)$.

It suffices to show that $\rho_T - \rho_T(S) \leq \rho_{\tilde{T}} - \rho_{\tilde{T}}(S)$ for any set S. The lemma then follows by inductively modifying the label of every leaf in Υ_T that is not labeled optimally.

Let S be any non-empty subset of $\mathcal{A}$. There are two cases to consider.

Case(i) The path reaching l falsifies some constraint in $V^{\perp}$.

In this case, l is labelled 1 in Γ_1 and 0 in $\tilde{\Gamma_1}$. ρ_T is unaltered by this label change. (i.e., $\rho_T = \rho_T(\emptyset) = \rho_{\tilde{T}}(\emptyset) = \rho_{\tilde{T}}$). On the other hand, $\rho_T(S)$ can only decrease. (i.e., $\rho_{\tilde{T}}(S) \leq \rho_T(S)$.) Hence, $\rho_T - \rho_T(S) \leq \rho_{\tilde{T}} - \rho_{\tilde{T}}(S)$.

Case(ii) The path reaching l does not falsify any constraint in $V^{\perp}$.

Here, l is labelled 0 in Γ_1 and 1 in $\tilde{\Gamma_1}$. Let U be the set of variables queried along the branch leading to l and $b \in \{0,1\}^*$ be the answers corresponding to these queries. Let X be the set of vectors x in V such that $x|_U = b$. Thus, every string in X is rejected by Γ_1 but is accepted by $\tilde{\Gamma_1}$ while the acceptance of the remaining strings in V is unaltered. Hence, the probability ρ_T increases by the quantity $\mathcal{D}_T(\Gamma) \cdot |X|/2^{n-m}$, i.e.,

$$\rho_{\tilde{T}} = \rho_T + \mathcal{D}_T(\Gamma) \cdot \frac{|X|}{2^{n-m}}$$

Similarly, if Y is the set of vectors y in V_S such that $y|_U = b$, we have that

$$\rho_{\tilde{T}}(S) = \rho_T(S) + \mathcal{D}_T(\Gamma) \cdot \frac{|Y|}{2^{n-m}}$$

Either Y is empty, in which case $\rho_T(S)$ does not increase. Suppose Y is non-empty. Let W be the set of variables not queried along the branch leading to l. Any string y in Y can be decomposed into two parts - the portion b actually seen by the test (i.e., involving the variables in U) and the portion y' not seen by the test (i.e., involving variables in W). Hence, y can be

written as $y = (b, y')$. Let $Y' = \{y' | (b, y') \in Y\}$. Similarly, define $X' = \{x' | (b, x') \in X\}$. Let $y = (b, y')$ be any string in Y. Then, $Y' = y' + X'$. Hence, $|Y| = |Y'| = |X'| = |X|$. Thus, the increase in $\rho_T(S)$ is the same as the increase in ρ_T. Hence, in either case, we have,

$$\rho_T - \rho_T(S) \leq \rho_{\tilde{T}} - \rho_{\tilde{T}}(S)$$

∎

Thus, the difference between the average acceptance probabilities of strings in V and strings ε-far from V only increases in the transformation from T to T'. Since the error of test T is μ, this difference for the test T is at least $(1 - \mu) - \mu = 1 - 2\mu$. Hence, this difference for T' is also at least $1 - 2\mu$. However, we know that the acceptance probability of strings in V for the 1-sided error test T' is 1. Hence, the acceptance probability of strings ε-far from V for test T' is at most 2μ. This proves Theorem 10.

A.2 Adaptive to Non-Adaptive

Let V be a vector space over $\{0,1\}^n$. We first show how to transform any 1-sided error deterministic adaptive test for V into a 1-sided error (probabilistic) non-adaptive test and then extend this transformation to work for probabilistic adaptive tests too.

A q-query deterministic adaptive test for V is a decision tree of height at most q. Let Γ be any q-query 1-sided error deterministic adaptive test (decision tree) for V. Since Γ has 1-sided error, it must accept if it does not detect any violation. We may also assume that Γ rejects on detecting a violation as this only decreases the acceptance probability of strings not in V. Furthermore, we may also assume that Γ rejects as soon as it detects a violation. Recall from Claim 7, that a test detects a violation if and only if it observes a constraint $\alpha \in V^{\perp}$ that is violated by the answers to the queries.

Let $\alpha = \alpha_0 \alpha_1 \ldots \alpha_n$ be any constraint that is checked by Γ. Then there exists a path from the root along which Γ queries all variables x_i such that $\alpha_i = 1$. Let x_j be the last variable (starting from the root) that is queried among the above and v the corresponding non-leaf node at which it is queried. We call such nodes *constraint nodes* as it is at these nodes that the test detects if a constraint is violated or not. We call the remaining non-leaf nodes of Γ as *query nodes*. We label the query nodes by the variable being queried at that node and constraint nodes by the constraint being checked at that node. From the observations made in the earlier paragraph, we conclude that every constraint node has a child that is a leaf node labeled 0 and conversely, any leaf node labeled 0 is the child of a constraint node.

We now define test T_Γ, which is the non-adaptive test corresponding to the deterministic adaptive test Γ.

Definition 7 (Non Adaptive test for 1-sided error deterministic test) *Given a 1-sided error deterministic (adaptive) test Γ for V, define the test T_Γ as follows:*
On input x,

1. *$v_0 \leftarrow$ root node, $i \leftarrow 0$*
2. *While v_i is not a leaf node do,*

(a) *Query the variable corresponding to node v_i.*

(b) *If v_i is a query node, then set v_{i+1} to be one of the children of v_i with probability $\frac{1}{2}$ each.*

(c) *If v_i is a constraint node labeled α, check if x satisfies the constraint α. If it satisfies, then set v_{i+1} to be the child of v_i that is not a leaf labeled 0 else set v_{i+1} to be the child of v_i that is a leaf labeled 0.*

(d) *Increment i.*

3. *If the leaf node v_i is labeled 1, accept, else reject.*

T_Γ reaches a leaf labeled 0 if and only if the constraint corresponding to the parent of this leaf node is violated. Hence, T_Γ is 1-sided error test. It is not obvious that T_Γ is a non-adaptive test. This is because in step $2(c)$, the test chooses the path corresponding to the leaf labeled 0 iff the constraint is not satisfied. This is the only "adaptive" portion of test T_Γ. However, this can be easily made non-adaptive in the following manner without altering the acceptance probability of the test: At any constraint node, the test assumes that the constraint has been satisfied and proceeds along the branch which is not a leaf labeled 0. Then, at the end it checks if all the constraints along the branch have been satisfied. If not, it rejects, else it accepts.

The query complexity of T_Γ is no more than the height of tree Γ. The following lemma relates the accepting probability of T_Γ to the average acceptance probability of Γ.

Lemma 20 *Let Γ be any 1-sided error deterministic adaptive test for V and T_Γ the corresponding non-adaptive test (as in Definition 7). Then for any string $x \in \{0,1\}^n$,*

$$\Pr[T_\Gamma(x) = 1] = \operatorname*{average}_{v \in V} (\Gamma(x))$$

Proof: For any string x and v a node of decision tree Γ, let $\pi_x(v)$ denote the fraction of strings in $x + V$ that follow a branch containing the node v when queried according to the decision tree Γ and $\wp_x(v)$ be the probability that v is one of the nodes v_i for some i, when x is tested by the test T_Γ. We can easily check that

$$\begin{aligned} \Pr[T_\Gamma(x) = 1] &= \sum \wp_x(v) \\ \operatorname*{average}_{v \in V}(\Gamma(x)) &= \sum \pi_x(v) \end{aligned}$$

where the summation in both cases is over all leaf nodes of Γ labeled 1. Thus it is sufficient if we prove that for all nodes v of Γ and all strings x, $\wp_x(v) = \pi_x(v)$.

We prove this result by induction on the depth of the node v. If v is the root node, then $\wp_x(v) = \pi_x(v) = 1$. Assume that $\wp_x(u) = \pi_x(u)$ for all strings x and all nodes u of depth $d-1$. Let v be any node of the tree Γ of depth $d > 1$. Let u be the parent of v. By the induction hypothesis, $\wp_x(u) = \pi_x(u)$. If $\wp_x(u) = \pi_x(u) = 0$, then $\wp_x(v) = \pi_x(v) = 0$. Otherwise two cases arise.

Case (i) u is a query node. In this case, it can be easily checked that $\pi_x(v) = \pi_x(u)/2$ and $\wp_x(v) = \wp_x(u)/2$. Hence, $\wp_x(v) = \pi_x(v)$.

Case (ii) u is a constraint node. Let the label of u be α. We first note that every string in $x+V$ satisfies exactly the same set of constraints as x. Suppose x satisfies the constraint α. If v is the child of u that is not a leaf labeled 0, then $\pi_x(v) = \pi_x(u)$ and $\wp_x(v) = \wp_x(u)$. If v is the child of u that is a leaf labeled 0, then $\wp_x(v) = \pi_x(v) = 0$. Hence, in both case we have $\wp_x(v) = \pi_x(v)$. The case when x does not satisfy the constraint α is similar.

∎

Let T be 1-sided error (adaptive) (ε, μ, q)-test for V. Let Υ_T and $\mathcal{D}_T$ be the associated set of decision trees and the corresponding distribution respectively. We now define the non-adaptive test T' corresponding to T.

Definition 8 (1-sided error Adaptive Test) *Given a 1-sided error adaptive test Γ for V, define the test T' as follows: On input x, T' chooses a decision tree Γ according to the distribution $\mathcal{D}_T$, it then performs non-adaptive test T_Γ on the string x and answers according to $T_\Gamma(x)$.*

Since T_Γ is non-adaptive, so is T'. T' inherits its query complexity from T_Γ which in turn inherits it from T. Thus, the query complexity of T' is at most q. The fact that T' inherits its error also from T is given by the following lemma.

Lemma 21 *Let T be any 1-sided error (adaptive) test and T' the non-adaptive version of T (as in Definition 8). Then, for any string $x \in \{0,1\}^n$,*

$$\Pr[T'(x) = 1] = \operatorname*{average}_{v \in V} \left(\Pr[T(x+v) = 1]\right)$$

Proof: This lemma follows from Lemma 20 as follows.

$$\begin{aligned}
\Pr[T'(x) = 1] &= \Pr\big[T_\Gamma(x) = 1 | \Gamma \leftarrow \mathcal{D}_T\big] \cdot \Pr\big[\Gamma \leftarrow \mathcal{D}_T\big] \\
&= \left(\operatorname*{average}_{v \in V}(\Gamma(x+v))\right) \cdot \Pr\big[\Gamma \leftarrow \mathcal{D}_T\big] \qquad \text{(From Lemma 20)} \\
&= \operatorname*{average}_{v \in V}\left(\Gamma(x+v) \cdot \Pr\big[\Gamma \leftarrow \mathcal{D}_T\big]\right) \\
&= \operatorname*{average}_{v \in V}\left(\Pr[T(x+v) = 1]\right)
\end{aligned}$$

∎

B Alternative Proof using Fourier Analysis

In this section we prove the following theorem, that combines Theorems 2 and 3.

Theorem 22 *Fix $0 < \varepsilon < 1$, $0 < \mu < \frac{1}{2}$. If a vector space V over $\{0,1\}^n$ has an ε-separating (q, μ)-local basis $\mathcal{A} = (A_1, \ldots, A_m)$, then every adaptive 2-sided error $(\varepsilon, \frac{1}{2} - \mu)$-test requires q queries.*

Our proof is based on Fourier analysis. For $f : \{0,1\}^n \to R$ a Boolean function, the *Fourier representation* of f is

$$f(x) = \sum_{\alpha \in \{0,1\}^n} \hat{f}_\alpha \cdot (-1)^{\langle \alpha, x \rangle}$$

where $\langle \alpha, x \rangle = \sum_{i=1}^n \alpha_i \cdot x_i \bmod 2$, and

$$\hat{f}_\alpha = 2^{-n} \sum_{x \in \{0,1\}^n} f(x) \cdot (-1)^{\langle x, \alpha \rangle}$$

Proof (of Theorem 22): Let $V = \mathcal{A}^\perp$, and let $\mathcal{G}$ be the uniform distribution over words in V. Let $\mathcal{B}_i$ be the uniform distribution over words that falsify constraint A_i and satisfy the rest, and let $\mathcal{B}$ be the uniform distribution over $\mathcal{B}_i$. For a decision tree Γ, define its *distinguishing factor* to be

$$\Delta(\Gamma, \mathcal{A}) = |\Pr_{\mathcal{G}}[\Gamma = 1] - \Pr_{\mathcal{B}}[\Gamma = 1]|$$

If the error probability of the test is μ, then the distinguishing probability of the test[4] is at least $(1 - (\frac{1}{2} - \mu)) - (\frac{1}{2} - \mu) = 2\mu$. Thus, by Yao's minimax principle, in order to prove Theorem 22, it suffices to prove the following lemma.

Lemma 23 *For any Γ of depth less than q, $\Delta(\Gamma, \mathcal{A}) < 2\mu$.*

¿From here on we concentrate on proving lemma 23. The following claim is proved by a similar proof to that of Theorem 10, and hence we omit it.

Claim 24 *For any decision tree Γ, and its optimal 1-sided error labeling Γ' from definition 5*

$$\Delta(\Gamma, \mathcal{A}) \leq \Delta(\Gamma', \mathcal{A})$$

By the previous claim we may assume wlog Γ has optimal labeling, and has 1-sided error.

Claim 25 *For any 1-sided error decision tree Γ and any $\alpha \in V^\perp$:*

$$\hat{\Gamma}_\alpha \geq 0$$

Proof: For ℓ a leaf of Γ, let $f_\ell : \{0,1\}^n \to \{0,1\}$ be the Boolean function that is one if x reaches ℓ in Γ, and zero otherwise. let $L_1(\Gamma)$ be the set of leaves of Γ that are labeled one. For any $x \in \{0,1\}^n$, x leads to exactly one leaf of Γ, so:

$$\Gamma(x) = \sum_{\ell \in L_1(\Gamma)} f_\ell(x)$$

By linearity of the Fourier transform, we get:

$$\hat{\Gamma}_\alpha = \sum_{\ell \in L_1(\Gamma)} (\hat{f}_\ell)_\alpha$$

Thus, in order to prove the lemma, we only need to show that for any $\ell \in L_1(\Gamma)$, and any $\alpha \in V^\perp$, $(\hat{f}_\ell)_\alpha \geq 0$. Let $Supp(\ell)$ be the variables queried along the branch leading to ℓ, and let $Supp(\alpha)$ be the set of variables set to 1 in α. There are two cases to consider.

[4]The distinguishing probability of the test is the expected distinguishing probability of an individual test (decision tree).

Supp$(\alpha) \subseteq$ *Supp*(ℓ): Since $\ell \in L_1(\Gamma)$, and Γ has 1-sided error, we know that the path reaching ℓ does not falsify any constraint in $V^\perp$. Since *Supp*$(\alpha) \subseteq$ *Supp*(ℓ), we conclude that any $x \in f_\ell^{-1}(1)$ must obey $\langle \alpha, x \rangle = 0$. Hence

$$(\hat{f}_\ell)_\alpha = 2^{-n} \sum_{x \in f_\ell^{-1}(1)} (-1)^{\langle \alpha, x \rangle} = 2^{-n} \cdot |f_\ell^{-1}(1)| \geq 0$$

Supp$(\alpha) \not\subseteq$ *Supp*(ℓ): $(\hat{f}_\ell)_\alpha$ measures the correlation of f_ℓ with the linear function $(-1)^{\langle x, \alpha \rangle}$. As *Supp*$(\alpha) \not\subseteq$ *Supp*(ℓ) we immediately get $(\hat{f}_\ell)_\alpha = 0$, because exactly half of $f_\ell^{-1}(1)$ sets $(-1)^{\langle \alpha, x \rangle}$ to zero and the other half sets it to one.

∎

We calculate the Fourier representation of the good and bad distributions.

$$\hat{\mathcal{G}}_\alpha = 2^{-n} \sum_{x \in \{0,1\}^n} \mathcal{G}(x)(-1)^{\langle x, \alpha \rangle} = \begin{cases} 2^{-n} & \alpha \in V^\perp \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} \hat{\mathcal{B}}_{i\alpha} &= 2^{-n} \sum_{x \in \{0,1\}^n} \mathcal{B}_i(x)(-1)^{\langle x, \alpha \rangle} \\ &= \sum_{x + v_i \in C} 2^{-k}(-1)^{\langle x + v_i, \alpha \rangle} = \begin{cases} (-1)^{\langle \alpha, v_i \rangle} \cdot 2^{-n} & \alpha \in V^\perp \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

For any linearly independent set $\mathcal{A} = \{A_1, \ldots, A_m\}$ let $\eta_A(\alpha)$ be the following function:

$$\eta_{\mathcal{A}}(\alpha) = \begin{cases} |I| & \exists I \subseteq [m] \;\; \alpha = \sum_{i \in I} A_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Notice that for $\alpha \notin V^\perp$ we define $\eta_{\mathcal{A}}(\alpha) = 0$. This will simplify our exposition. Notice that $\eta_{\mathcal{A}}$ is well defined for *any* set of vectors $\mathcal{A}$ that is linearly independent.

Let $\mathcal{F} = \mathcal{G} - \mathcal{B}$. We calculate $\hat{\mathcal{F}}_\alpha$.

Claim 26 $\hat{\mathcal{F}}_\alpha = \frac{2\eta_{\mathcal{A}}(\alpha)}{2^n m}$

Proof: By linearity of the Fourier transform we get:

$$\hat{\mathcal{F}}_\alpha = \hat{\mathcal{G}}_\alpha - \frac{1}{m} \sum_{i=1}^{m} \hat{\mathcal{B}}_{i\alpha}$$

Since both $\hat{\mathcal{G}}$ and $\hat{\mathcal{B}}_i$ are non-zero only on $\alpha \in V^\perp$, we fix our attention there. The key observation is that $\langle \alpha, v_i \rangle = 1$ if A_i is one of the rows needed to sum up in order to reach α, and $\langle \alpha, v_i \rangle = 0$ otherwise. This means that if $\alpha = \sum_{i \in I} A_i$ then exactly $|I|$ elements $\mathcal{B}_{i\alpha}$ contribute negatively to $\mathcal{B}_\alpha$ whereas $m - |I|$ elements contribute positively. The claim follows because $\hat{\mathcal{F}}_\alpha = \hat{\mathcal{G}}_\alpha - \hat{\mathcal{B}}_\alpha$. ∎

We calculate $\Delta(\Gamma, \mathcal{A})$ for Γ of depth $< q$ using Fourier representations.

$$\begin{aligned}
\Delta(\Gamma, \mathcal{A}) &= \sum_x \mathcal{G}(x) \cdot \Gamma(x) - \sum_x \mathcal{B}(x) \cdot \Gamma(x) \\
&= \sum_x \mathcal{F}(x) \cdot \Gamma(x) \\
&= \sum_x \left(\sum_\alpha \hat{\mathcal{F}}_\alpha (-1)^{\langle x, \alpha \rangle} \right) \cdot \left(\sum_\beta \hat{\Gamma}_\beta (-1)^{\langle x, \beta \rangle} \right) \\
&= \sum_\alpha \sum_\beta \hat{\mathcal{F}}_\alpha \hat{\Gamma}_\beta \sum_x (-1)^{\langle x, \alpha + \beta \rangle} \\
&= \sum_\alpha \hat{\mathcal{F}}_\alpha \hat{\Gamma}_\alpha \cdot 2^n \\
&= \sum_\alpha \frac{2\eta_{\mathcal{A}}(\alpha)}{m} \cdot \hat{\Gamma}_\alpha
\end{aligned}$$

By claim 25, $\hat{\Gamma}_\alpha \geq 0$ for any $\alpha \in V^\perp$. If $\alpha \notin V^\perp$, then $\eta_{\mathcal{A}}(\alpha) = 0$. Since $\eta_{\mathcal{A}}(\alpha) \geq 0$ we conclude that all elements in the sum above are non-negative. If $|\alpha| \geq q$ then $\hat{\Gamma}_\alpha = 0$ because Γ reads less than q bits and so has zero correlation with α. If $|\alpha| < q$, then $\eta_{\mathcal{A}}(\alpha) < \mu m$. This is simply the definition of $\mathcal{A}$ being (q, μ)-local. Thus we conclude:

$$\Delta(\Gamma, \mathcal{A}) < 2\mu \sum_{\alpha \in V^\perp} \hat{\Gamma}_\alpha \tag{3}$$

Claim 27 $\sum_{\alpha \in V^\perp} \hat{\Gamma}_\alpha \leq 1$.

Proof: Let $H(\alpha)$ be the characteristic function of $V^\perp$, i.e. the function that is one on α if $\alpha \in V^\perp$ and zero otherwise. It's Fourier representation is

$$\hat{H}_x = 2^{-n} \sum_{\alpha \in \{0,1\}^n} H(\alpha)(-1)^{\langle x, \alpha \rangle} = \begin{cases} 2^{-k} & x \in V \\ 0 & \text{otherwise} \end{cases}$$

So

$$\begin{aligned}
\sum_{\alpha \in V^\perp} \hat{\Gamma}_\alpha &= \sum_\alpha \hat{\Gamma}_\alpha \cdot V^\perp \\
&= \sum_\alpha H(\alpha) \cdot \left(2^{-n} \cdot \sum_x \Gamma(x) \cdot (-1)^{\langle \alpha, x \rangle} \right) \\
&= \sum_x \Gamma(x) \cdot \left(2^{-n} \cdot \sum_\alpha H(\alpha) \cdot (-1)^{\langle \alpha, x \rangle} \right) \\
&= \sum_x \Gamma(x) \cdot \hat{H}_x \\
&= 2^{-k} \sum_{x \in V} \Gamma(x) \leq 1
\end{aligned}$$

The last inequality follows because $\Gamma(x) \leq 1$ for all x, and $|V| = 2^k$. ∎

Plugging claim 27 into equation (3) we get:

$$\Delta(\Gamma, \mathcal{A}) < 2\mu$$

We have proved lemma 23, and with it the proof of Theorem 22 is completed. ∎

C Proofs from Section 6

Proof (of Lemma 12): We need a couple of lemmas, the proof of which will follow.

Lemma 28 *For any integers $c \geq 2, d$, and any constant $\alpha < c-1$, a random (c,d)-regular bipartite graph with n left vertices, is with high probability a (α, ε)-left expander, for any ε satisfying*

$$\varepsilon \leq \left(2e^{(1+\alpha)} \cdot \left(\frac{\alpha d}{c} \right)^{(c-\alpha)} \right)^{-\frac{1}{c-\alpha-1}} \tag{4}$$

Lemma 29 *Let G be a (c,d)-regular bipartite graph. If G is an (α, ε)-left expander, then G is an $(2\alpha - c, \varepsilon)$-left unique neighbor expander.*

We do not try to optimize constants. Let $\alpha = \frac{c+1}{2}$, Noticing that for $c \geq 5$, $\frac{c}{2} < \alpha < c-1$. By lemma 28, G is a (α, ε)-right expander for any ε satisfying equation (28).

For our selection of α, and any $c \geq 5$, the following inequalities can be verified:

$$\frac{(1+\alpha)}{(c-\alpha-1)} \leq 3$$

$$\frac{\alpha}{c} \leq 2/3$$

$$\frac{(c-\alpha)}{(c-\alpha-1)} \leq 2$$

Hence setting $\varepsilon = (100 \cdot d)^{-2}$ satisfies equation (28). Finally, by lemma 29, we get that G is whp a $(1, rd^{-2})$-left unique neighbor expander. ∎

Proof (of Lemma 28): Let BAD be the event that the random graph is *not* an expander. This means there is some $S \subset L, |S| \leq \varepsilon n$ such that $|N(S)| \leq \alpha \cdot |S|$.

Fix sets $S \subset L, T \subset R$, $|S| = s \leq \varepsilon n, |T| = \alpha s$, and let B_s be the event that all edges leaving S land inside T. We upper-bound the probability of this bad event.

$$\Pr[B_s] \quad = \quad \prod_{i=0}^{c \cdot s - 1} \frac{\alpha d s - i}{cn - i} \leq \left(\frac{\alpha d s}{cn} \right)^{cs}$$

The inequality follows as long as $\alpha ds < cn$. We now use a union bound over all sets $S \subset L \;\; |S| = s \leq \varepsilon n$ and all sets $T \subset R, \;\; |T| = \alpha s$. Let κ be the constant $\kappa = e^{1+\alpha} \cdot \left(\frac{\alpha d}{c}\right)^{c-\alpha}$.

$$\begin{aligned}
\Pr[BAD] &\leq \sum_{s=1}^{\varepsilon n} \binom{n}{s} \cdot \binom{m}{\alpha s} \cdot \Pr[B_s] \\
&\leq \sum_{s=1}^{\varepsilon n} \left(\frac{en}{s}\right)^{s} \cdot \left(\frac{em}{\alpha s}\right)^{\alpha s} \cdot \left(\frac{\alpha ds}{cn}\right)^{cs} \\
&= \sum_{s=1}^{\varepsilon n} \left[e^{1+\alpha} \cdot \left(\frac{\alpha d}{c}\right)^{c-\alpha} \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} \\
&= \sum_{s=1}^{\varepsilon n} \left[\kappa \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} \qquad (5)
\end{aligned}$$

By definition of $\alpha, \;\; c - \alpha - 1 > 0$, hence $\left(\frac{s}{n}\right)^{c-\alpha-1} \leq 1$. Set

$$\varepsilon \leq (2\kappa)^{\frac{-1}{(c-\alpha-1)}} = \left(2e^{(1+\alpha)} \cdot \left(\frac{\alpha d}{c}\right)^{(c-\alpha)} \right)^{-\frac{1}{c-\alpha-1}} \qquad (6)$$

For this value of ε, each term of the sum (5) is at most $1/2$. Set $\lambda = \min\{\frac{1}{3}, \frac{c-\alpha-1}{2}\}$, and split the sum (5) into two sub-sums.

$$\begin{aligned}
\Pr[BAD] &\leq \sum_{s=1}^{\varepsilon n} \left[\kappa \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} \\
&\leq \sum_{s=1}^{n^{\lambda}} \left[\kappa \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} + \sum_{s=n^{\lambda}}^{\varepsilon n} \left[\kappa \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} \\
&\leq n^{\lambda} \cdot \kappa \cdot n^{(\lambda-1)2\lambda} + n \cdot 2^{-n^{\lambda}} = \kappa \cdot n^{-\lambda+2\lambda^2} + n \cdot 2^{-n^{\lambda}} \\
&\leq \kappa \cdot n^{-1/9} + n \cdot 2^{-n^{\lambda}} = o(1)
\end{aligned}$$

We conclude that with high probability, G is an (α, ε)-left expander. ∎

Proof (of Lemma 29): Let $S \subset L, |S| \leq \varepsilon |L|$. Then by expansion we get

$$\alpha \cdot |S| < |N(S)|$$

Any neighbor of S that is not a unique neighbor, must be touched by at least 2 edges leaving S. Since the left degree of G is c, we get

$$|N(S)| \leq |N^1(S)| + \frac{c \cdot |S| - |N^1(S)|}{2} = \frac{c \cdot |S| + |N^1(S)|}{2}$$

Combining the two equations, we get our claim. ∎

Proof (of Lemma 13): In the proof, we make use of the following theorem (see [12])

Theorem 30 (Azuma's Inequality) *If $X_0, \ldots, X_t$ is a martingale sequence such that $|X_i - X_{i+1}| \leq 1$ for all i, then*

$$Pr[|X_t - X_0| \geq \lambda\sqrt{t}] \leq 2e^{-\lambda^2/2}$$

Fix $T \subseteq R$ $|T| = t \geq \mu m$. Let $X = |N^{odd}(T)|$. We start by computing $E[X]$. For $i = 1 \ldots n$, let X_i be the random variable indicating whether vertex $i \in L$ is in $N^{odd}(T)$. Clearly $X = \sum_{i=1}^{n} X_i$, so by the linearity of expectation, we need only compute $E[X_i]$. Recall that $cn = dm$, Let $\text{odd}(c) = \{1, 3, 5, \ldots, c\}$ be the set of positive odd integers $\leq c$, and notice that $c \in \text{odd}(c)$ because c is odd.

$$\begin{aligned} E[X_i] &= \frac{\sum_{i \in \text{odd}(c)} \binom{\mu dm}{i} \cdot \binom{(1-\mu)dm}{c-i}}{\binom{cn}{c}} \\ &\geq \frac{\binom{\mu cn}{c}}{\binom{cn}{c}} = \mu^c - O(\frac{1}{n}) \end{aligned}$$

We conclude by linearity of expectation:

$$E[X] \geq \mu^c \cdot n - O(1)$$

We now make use of the following edge-exposure martingale to show concentration of X around its expectation. Fix an ordering on the μdm edges leaving T, and define a sequence of random variables $Y_0, \ldots Y_{\mu dm}$ as follows: Y_i is the random variable that is equal to the expected size of $N^{odd}(T)$ after the first i edges leaving T have been revealed. By definition, $Y_{\mu dm} = X$, $Y_0 = E[X]$, and the sequence is a martingale, where $|Y_i - Y_{i+1}| \leq 1$ for all $i \leq \mu dm$. Since $d > \frac{2\mu c^2}{(\mu^c - \delta)^2}$, we apply Azuma's inequality (Theorem 30) and get:

$$\begin{aligned} \Pr[X \leq \delta n] &\leq \Pr[|Y_{\mu dm} - Y_0| \geq (\mu^c - \delta)n] \\ &= \Pr[|Y_{\mu dm} - Y_0| \geq (\mu^c - \delta)\frac{d}{c}m] \\ &\leq 2e^{-\frac{d(\mu^c - \delta)^2}{2\mu c^2} \cdot m} \leq 2e^{-(1+\varepsilon)m} \end{aligned}$$

Where $\varepsilon = \frac{d(\mu^c - \delta)^2}{2\mu c^2} - 1 > 0$. There are at most 2^m possible sets $T \subseteq R$, so a union bound gives:

$$\Pr[\exists T \subset R \; |T| \geq \mu m \; |\sum_{j \in T} A_j| \leq \delta n] \leq 2^m \cdot 2e^{-(1+\varepsilon)m} = o(1)$$

We conclude that $\mathcal{A}(G)$ is whp a (δ, μ)-right odd expander. ∎

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'