Some 3CNF Properties are Hard to Test

Eli Ben-Sasson[*]
DEAS & LCS
Harvard & MIT
Cambridge, MA
eli@eecs.harvard.edu

Prahladh Harsha[†]
LCS
Massachusetts Institute of Technology
Cambridge, MA
prahladh@mit.edu

Sofya Raskhodnikova
LCS
Massachusetts Institute of Technology
Cambridge, MA
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]. Notice that for every bad instance (i.e. an assignment that does not satisfy φ) there is a 3-bit query that witnesses this fact. Nevertheless, finding such a short witness requires a linear number of queries, even for assignments that are very far from satisfying.

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 LDPC codes with linear distance and constant rate are very far from being locally testable.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problems Complexity

General Terms

Algorithms, Theory

Keywords

sublinear algorithms, lower bounds, property testing, CNF formulas, locally testable codes

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 [16] 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 [15, 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 [14] 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 *k*CNF 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 formulae is hard. A natural question is whether restricting CNF formulae to have a constant number of variables k per clause allows for faster testers. At first glance 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 formulae are testable with

[*] Supported by NSF grants CCR-0133096, CCR-9877049, CCR 0205390, and NTT Award MIT 2001-04.

[†] Supported in part by NSF Award CCR 9912342 and NTT Award MIT 2001-04

STOC'03, June 9–11, 2003, San Diego, California, USA.
Copyright 2003 ACM 1-58113-674-9/03/0006 ...$5.00.

$O(\sqrt{n})$ 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 some properties defined by 3CNF formulae requires a *linear* number of queries. Thus, we present a gap between 2CNFs and 3CNFs. We show the existence of families of 3CNF formulae 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.

Each hard 3CNF property we use is a vector space $V \subseteq \{0,1\}^n$ that can be expressed as the set of solutions to a homogeneous 3LIN formula. While proving the lower bound, we show that every adaptive 2-sided error test for checking membership in a vector space can be converted to a non-adaptive 1-sided error test with the same query complexity and essentially identical parameters.

This allows us to consider only 1-sided error non-adaptive tests. In order to prove our lower bound, we need to find for every such test T, a bad vector $b \in \{0,1\}^n$ (that is far from V), such that T accepts b with significant probability (i.e., T fails to reject b, as it should). Yao's minimax principle allows us to switch the quantifiers. In other words, in order to prove our lower bound, it suffices to present a distribution $\mathcal{B}$ over bad vectors such that any deterministic test fails to reject a random b (selected according to the distribution $\mathcal{B}$) with significant probability.

We can now give a rough picture of how to get a vector space V that is hard to test. Take a basis $\mathcal{A}$ for $V^{\perp}$ and define $\mathcal{B}$ to be the uniform distribution over all vectors in $\{0,1\}^n$ that falsify exactly one constraint of $\mathcal{A}$ (and satisfy the rest). It turns out that for good random Low Density Parity Check Codes (LDPC codes), one can pick a basis $\mathcal{A}$ such that the resulting distribution is only over vectors that are far from V. Observe that a 1-sided error test T rejects b only when T detects that b falsifies some constraint in $V^{\perp}$. If T is non-adaptive and deterministic, it is completely determined by the set of variables it queries. If this set is small, T only checks low-weight constraints in $V^{\perp}$. We prove that for random LDPC codes checking low-weight constraints in $V^{\perp}$ allows to check only a small fraction of constraints in $\mathcal{A}$. Therefore, with significant probability, T fails to reject b, selected according to distribution $\mathcal{B}$. This happens for any deterministic T of low query complexity, so by Yao's minimax principle there is no test of small query complexity for the property V.

Our results shed some light on the question of optimal locally testable codes. An infinite family of codes $\{\mathcal{C}\}_n$ is called *locally testable* if the property $\mathcal{C}_n$ is testable with constant query complexity. These codes are in the center of PCP constructions, and are of fundamental importance in theoretical computer science. Recently Goldreich and Sudan proved the existence of such codes which achieve linear distance and near linear rate [12], resulting in better PCP constructions.

The vector spaces we use (which are hard to test) are built upon random (c,d)-regular LDPC 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 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 question of optimal locally testable codes. The existence of such optimal codes that achieve *(i)* constant rate; *(ii)* linear distance; and *(iii)* are locally testable remains an interesting open problem.

1.4 Earlier Work

We shortly discuss the connection of this paper to other results. There are two published linear lower bounds for property testing. One is the 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]. There is a simple and elegant unpublished linear lower bound observed by Sudan [Personal Communication]. 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 formulae.

Both linear lower bounds of Sudan and Bogdanov et. al [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[1]. 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. Another relevant result is the lower bound of Goldreich and Ron on testing bipartiteness in 3-regular, n-vertex graphs [11]. They showed a lower bound of $\Omega(\sqrt{n})$ on the query complexity, yet short witnesses of non-bipartiteness do exist, in the form of odd cycles of length poly$(\log n)$. Our result strengthens this finding, since in our case the query complexity is linear whereas the witness size is constant.

1.5 Paper Organization

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

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

[1] 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 $n/2$.

$\min_{x' \in \mathcal{P}} dist(x, x')$, 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, completeness c, soundness s 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 c and accepts strings that are ε-far from $\mathcal{P}$ with probability at most s, for some $0 \leq s < c \leq 1$. A test is said to have *error μ* if $c \geq 1 - \mu$ and $s \leq \mu$ (for $\mu < \frac{1}{2}$).[2] If a test T accepts input x, we say $T(x) = 1$. Otherwise, we say $T(x) = 0$. 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 formulae

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 $\{x | \ x \text{ satisfies } \varphi\}$. For brevity, we refer to a test for this property as a test for φ.

3. MAIN THEOREM

In this section we state and prove the main theorem, saying that some 3CNF properties are hard to test.

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 such that every adaptive (ε, μ)-test for φ requires δn queries.*

Proof. To prove Theorem 1, we find hard 3CNF formulae that define linear properties. Our first step towards proving the main result is the theorem that for linear properties, 1-sided error non-adaptive tests are as powerful as general tests.

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

This theorem is of independent interest as it applies to testing any linear property[3]. The proof of Theorem 2 appears in section 5. The reduction to simpler tests does not increase the error but rather shifts the error from the YES-instances to the NO-instances as it preserves the difference between the completeness and soundness.

Equipped with Theorem 2, we can restrict our attention to proving Theorem 1 for non-adaptive 1-sided error tests, provided that the formulae we work with define linear properties. Indeed, we find linear properties that are hard to test and then represent them by CNFs. Consider a vector space $V \subseteq \{0,1\}^n$. Denote the dual space by $V^\perp$. Let $\mathcal{A} = (A_1, \ldots, A_m)$ be a basis for $V^\perp$. Let $|x|$ denote the weight of vector $x \in \{0,1\}^n$. For two vectors $x, y \in \{0,1\}^n$, let $\langle x, y \rangle = \sum_{i=1}^n x_i y_i \bmod 2$. 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 $d = \max_{A_i \in \mathcal{A}} |A_i|$. We work with an arbitrary constant d and later show how to reduce it to 3. Since each 3LIN formula has an equivalent 3CNF, it is enough to find hard 3LINs.

We now present sufficient conditions for a vector space to be hard to test. To understand the conditions, keep in mind that later we employ Yao's minimax principle to show that all vector spaces satisfying these conditions are hard for 1-sided non-adaptive tests. Yao's principle states that to prove that each low-query probabilistic test fails on some input, it is enough to give a distribution on the inputs on which each low-query deterministic test fails. We are only interested in 1-sided error tests which, by definition, have to accept unless no vector in the tested vector space satisfies the answers to the queries. Therefore, to show that a vector space satisfying our conditions is hard, we need to exhibit a distribution on vectors which are far from the vector space, such that every low-query deterministic non-adaptive test on this distribution fails to determine with non-negligible probability that the input violates the constraints of the vector space.

Definition 1 (Hard Linear Properties). *Let $V \subseteq \{0,1\}^n$ be a vector space 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 $|\alpha| \geq q$.*

Notice that if $\mathcal{A}$ is ε-separating, each string x falsifying exactly one constraint in $\mathcal{A}$ is ε-far from V. To see why, 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, $dist(x, V) \geq \varepsilon n$.

For the proof that every vector space satisfying the above conditions is hard to test, our bad distribution that foils low-query tests is over strings that falsify exactly one constraint. The falsified constraint is chosen uniformly at random. The first condition ensures that the distributions is over vectors which are ε-far from the vector space.

The second condition ensures that the distribution is hard to test. To get the intuition, suppose the second condition is violated. Then a μ fraction of the constraints sums up to a low-weight vector, and the sum represents a constraint on fewer than q variables. Querying variables in the new constraint would allow a test running on our bad distribution to deduce that some constraint is violated with probability at

[2] We state our results using the symmetric error parameter μ, but they can be all stated for arbitrary $s < c$.

[3] The theorem is stated for vector spaces in $\{0,1\}^n$, but our proof can be extended to general vector spaces.

least μ. The second condition disallows this or, intuitively, ensures that to "get information" about a fraction μ of the constraints in $\mathcal{A}$, a test needs at least q queries.

The following theorem, proved in section 4, shows that any linear space conforming to definition 1 is hard for 1-sided error non-adaptive tests.

THEOREM 3 (NON-ADAPTIVE 1-SIDED ERROR). *Fix $0 < \varepsilon < 1$, $0 < \mu < \frac{1}{2}$. Let $V \subseteq \{0,1\}^n$ be a vector space. 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.*

Theorems 2 and 3 show that every linear property conforming to definition 1 is hard even for 2-sided error adaptive 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 formulae 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 TO 3CNFS). *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

This section proves Theorem 3.

PROOF (OF THEOREM 3): 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. The comment after definition 1 shows that distribution $\mathcal{B}$ is over strings which are ε-far from V. Lemma 6 demonstrates that every low complexity deterministic test is likely to fail on $\mathcal{B}$, which completes the proof of Theorem 3. □

LEMMA 6. *Let T be a deterministic 1-sided error non-adaptive test with $< q$ queries. If $\mathcal{A}$ is (q,μ)-local then $\Pr_{x \leftarrow \mathcal{B}}[T(x) = 0] < 2\mu$.*

PROOF. Let Q be the set of queries posed by T. A query to variable x_i is viewed as a vector of weight 1 in $\{0,1\}^n$ which is 1 at coordinate i and 0 everywhere else. Observe that since T has 1-sided error, it has to accept if there is a vector in V consistent with the answers to the queries. By linearity, this is equivalent to saying that T rejects a vector in $\mathcal{B}$ only if the falsified constraint can be expressed as a linear combination of queries and remaining constraints. Thus, we need to show that $< 2\mu$ fraction of constraints in $\mathcal{A}$ can be expressed as a linear combination of queries and remaining constraints.

Let c be such a constraint. Then there is a set $C \subseteq \mathcal{A}$ with $\sum_{c \in C} c \in span(Q)$. We show that fewer than $2\mu m$ constraints in $\mathcal{A}$ are in such sets. Let Γ be the family of such sets, i.e., of subsets of $\mathcal{A}$ that sum up to a vector $\alpha \in span(Q)$.

It remains to show $\left|\bigcup_{C \in \Gamma} C\right| < 2\mu m$. Observe that if $\alpha_1, \alpha_2 \in span(Q)$, so does $\alpha_1 + \alpha_2$. In terms of C's this implies that if $C_1, C_2 \in \Gamma$, so is $C_1 \triangle C_2$[4]. Since $|Q| < q$ and $\mathcal{A}$ is (q,μ)-local, $|C| \leq \mu m$ for all $C \in \Gamma$. We can now apply Lemma 7 to conclude that $\left|\bigcup_{C \in \Gamma} C\right| < 2\mu m$. □

LEMMA 7. *Let $\Gamma = \{C | C \subseteq [m]\}$ be a non-empty family of subsets of $[m]$ such that Γ is closed under symmetric difference and for all sets C in Γ, $|C| \leq w$. Then $\left|\bigcup_{C \in \Gamma} C\right| < 2w$.*

PROOF. Suppose $x \in C$ for some $C \in \Gamma$. 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 of the sets of Γ. Therefore,

$$\frac{|\Gamma|}{2} \cdot \left|\bigcup_{C \in \Gamma} C\right| = \sum_{C \in \Gamma} |C| \leq (|\Gamma| - 1)w < |\Gamma| w.$$

The first inequality holds because the empty set belongs to Γ, and $|C| \leq w$ for all other C in Γ. Since $|\Gamma| > 0$, we conclude that $|\bigcup_{C \in \Gamma} C| < 2w$. □

5. REDUCING 2-SIDED ERROR ADAPTIVE TO 1-SIDED ERROR NON-ADAPTIVE

In this section we prove Theorem 2 by presenting a generic reduction that converts any adaptive 2-sided error test for a linear property to a non-adaptive 1-sided error one without altering the query complexity. 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 9) maintaining the difference between completeness and soundness and then reduce this to a non-adaptive test with 1-sided error (Theorem 11) maintaining both completeness and soundness[5]. The second reduction was suggested by Madhu Sudan.

A natural test for checking membership in a linear subspace V is one that is determined by a distribution over sets of constraints in the dual space $V^\perp$. This test chooses a set of constraints from the dual space $V^\perp$ according to this distribution, queries all variables that appear in this set of constraints and accepts or rejects depending on whether the constraints are satisfied or not. Clearly, this is a 1-sided error non-adaptive test. The proofs of Theorem 9 and Theorem 11 demonstrate that any test can be converted into one of the above form maintaining the query complexity and the difference between completeness and soundness.

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

[5] These reductions are stated for linear spaces V over the field $GF(2)$. However, they naturally extend to larger fields.

Any probabilistic test can be viewed as a distribution over deterministic tests and each deterministic test can be represented by a decision tree. Thus, any test T can be represented by an ordered pair $(\Upsilon_T, \mathcal{D}_T)$ where $\Upsilon_T = \{\Gamma_1, \Gamma_2, \dots\}$ is a set of decision trees and $\mathcal{D}_T$ is a distribution on this set 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 following terminology and lemma will be useful in analyzing the reductions. 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, it is equivalent to having a constraint α in $V^\perp$ such that $\langle x, \alpha\rangle = 1$ for all $x \in \{0,1\}^n$ which are consistent with the answers to the queries.

Let V be a vector space. For any leaf l of decision tree Γ, let V_l be the set of all vectors in V that are consistent with the answers along the path leading to l. Similarly, for any string $x \in \{0,1\}^n$, let V_l^x be the the set of all vectors in $x + V$ that are consistent with the answers along the path leading to l.

Lemma 8. *Let $V \subseteq \{0,1\}^n$ be a vector space and $x \in \{0,1\}^n$. For any decision tree Γ and a leaf l in Γ, if both V_l and V_l^x are non-empty, then $|V_l| = |V_l^x|$.*

Proof. Let U be the set of all strings in V which have the bit 0 in all the positions queried along the path leading to l. Since $0^n \in U$, we have that U is non-empty. Observe that if $u \in U$ and $v \in V_l$, then $u + v \in V_l$. In fact, if $V_l \neq \emptyset$, $V_l = v + U$ for any $v \in V_l$. Hence, $|V_l| = |U|$. Similarly, if $V_l^x \neq \emptyset$, we have that $V_l^x = y + U$ for any $y \in V_l^x$. Hence, $|V_l^x| = |U|$ and the lemma follows. □

5.1 2-Sided to 1-Sided Error

In this section, we reduce a 2-sided error (adaptive) test to a 1-sided error (adaptive) test maintaining the difference between completeness and soundness and without altering the query complexity.

Theorem 9. *Let $V \subseteq \{0,1\}^n$ be a vector space. 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.*

Proof. Let $T = (\Upsilon_T, \mathcal{D}_T)$ be a 2-sided error (adaptive) (ε, μ, q)-test for V. To convert T to a 1-sided error test, we modify the test so that it rejects if and only if it observes that a constraint in $V^\perp$ has been violated. We say that a leaf l is labelled *optimally* if its label is 0 when the query answers on the path to l falsify some constraint in $V^\perp$, and 1 otherwise. We relabel the leaves of each tree Γ in Υ_T *optimally* to obtain the tree Γ_{opt}.

Relabelling produces a 1-sided error test with unchanged query complexity. However, the new test performs well only on "average". To get good performance on every string, we randomize the input x by adding a random vector v from V to it and perform the test on $x + v$ instead of x. Now we formally define the 1-sided error T' corresponding to T.

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

Clearly, T' has 1-sided error as it rejects only if it detects a violation. Also, T' has the same query complexity as T. It remains to verify the soundness of T'.

First, let us introduce some notation. 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 $S \subseteq \mathcal{A}$, let V_S be the set of vectors that violate all constraints in S and satisfy all other constraints in $\mathcal{A}$. In this notation, $V_\emptyset = V$. It follows that if $x \in V_S$ for some $S \subseteq \mathcal{A}$, then $V_S = x + V$. Note that $dist(x, V) = dist(y, V)$ for all $x, y \in V_S$. Hence, the set of strings that are ε-from V is a union of sets of the form V_S. For any subset S of $\mathcal{A}$ and any test T, let $\rho_T(S)$ be the average acceptance probability of test T over all strings in V_S, i.e., $\rho_T(S) = \text{average}_{y \in V_S}\big(\Pr[T(y) = 1]\big)$. For notational brevity, we denote $\rho_T(\emptyset)$, the average acceptance probability of strings in V, by ρ_T. Observe that for the new test T', for each input x, $\Pr[T'(x) = 1] = \rho_{T'}(S)$, where $V_S = V + x$.

The following lemma shows that the transformation to a 1-sided error test given by Definition 2 increases the acceptance probability of any string not in V by at most $\rho_{T'} - \rho_T$.

Lemma 10. *For any non-empty set $S \subseteq \mathcal{A}$,*

$$\rho_T - \rho_T(S) \le \rho_{T'} - \rho_{T'}(S).$$

Proof. Let S be a non-empty subset of $\mathcal{A}$. It is enough to prove that relabeling one leaf l of a decision tree Γ in Υ_T *optimally* does not decrease $\rho_T - \rho_T(S)$. Then we obtain the lemma by relabelling one leaf at a time to get T' from T. There are two cases to consider.

Case (i) The path to l falsifies some constraint in $V^\perp$. Then l is relabelled from 1 to 0. This change preserves ρ_T because it only affects strings that falsify some constraint. Moreover, it can only decrease the acceptance probability for such strings. Therefore, $\rho_T(S)$ does not increase. Hence, $\rho_T - \rho_T(S)$ does not decrease.

Case(ii) The path to l does not falsify any constraint in $V^\perp$. Then l is relabelled from 0 to 1. Let X and Y respectively be the set of vectors in V and V_S that are consistent with the answers observed along the path to l. Thus, every string in $X \cup Y$ was rejected before relabeling, but is accepted now. The behavior of the algorithm on the remaining strings in V and V_S is unaltered. Hence, the probability ρ_T increases by the quantity $\mathcal{D}_T(\Gamma_1) \cdot \frac{|X|}{|V|}$. Similarly, $\rho_T(S)$ increases by $\mathcal{D}_T(\Gamma_1) \cdot \frac{|Y|}{|V|}$.

It suffices to show that $|X| \ge |Y|$. Since the path leading to l does not falsify any constraint, X is non-empty. If Y is empty, we are done. Otherwise, suppose Y is non-empty. Let $x \in Y$. Then $X = V_l$ and $Y = V_l^x$ in the notation of Lemma 8. Since both $X = V_l$ and $Y = V_l^x$ are non-empty, by Lemma 8, $|X| = |Y|$, which concludes the proof of the lemma. □

Thus, the above transformation to a 1-sided error test does not decrease the difference between the completeness and the soundness. As the completeness increases from $1-\mu$ to 1, the soundness increases from μ to at most 2μ. This completes the proof of Theorem 9. □

5.2 Adaptive to Non-Adaptive

In this section, we argue that adaptivity does not help to check linear constraints. The intuition behind this is as follows: To check if a linear constraint is satisfied, a test needs to query all the variables that participate in that constraint. Based on any partial view involving some of the variables, the test cannot guess if the constraint is going to be satisfied or not till it reads the final variable. Hence, any adaptive decision based on such a partial view does not help.

THEOREM 11. *Let $V \subseteq \{0,1\}^n$ be a vector space. 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.*

PROOF. Let T be a 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. Since T is of 1-sided error, T accepts if it does not detect a violation. Furthermore, we may assume that T rejects if it detects a violation since this can only decrease the acceptance probability of strings not in V. This implies that all the trees in Υ_T are *optimally* labeled. We now define the non-adaptive test T' corresponding to T.

DEFINITION 3 (1-SIDED ERROR NON ADAPTIVE TEST). *Given a 1-sided error (adaptive) test T for V, define the test T' as follows: On input x, choose a random $v \in V$, query x on all variables that T queries on input v, reject if a violation is detected, otherwise accept.*

T' has 1-sided error because it rejects only if it detects a violation. The query complexity of T' is the same as that of for T. Moreover, the queries depend only on the random $v \in V$ and not on the input x. Hence, the test T' is non-adaptive. The following lemma relates the acceptance probability of T' to the average acceptance probability of T.

LEMMA 12. *Let T be a 1-sided error (adaptive) test and T' the non-adaptive version of T (as in Definition 3). Then, for any string $x \in \{0,1\}^n$,*

$$\Pr[T'(x) = 1] = \underset{v \in V}{\text{average}} \left(\Pr[T(x+v) = 1] \right).$$

PROOF. For any decision tree Γ, let $l_1(\Gamma)$ denote the set of leaves in Γ that are labeled 1. For any leaf l in a decision tree Γ, let $\text{var}(l)$ denote the set of variables queried along the path leading to l in the tree Γ. Following the notation of Lemma 8, let V_l and V_l^x be the set of all vectors in V and $x+V$ respectively that are consistent with the answers along the path leading to l. Also let I_l^x be a binary variable which is set to 1 iff x does not violate any constraint in $V^\perp$ involving only the variables $\text{var}(l)$. Observe that if test T' chooses the decision tree $\Gamma \in \Upsilon_T$ and the vector $v \in V$ such that $v \in V_l$ for some leaf l labeled 1 in the tree Γ, then $I_l^x = 1$ iff $T'(x) = 1$.

The quantity "$\text{average}_{v \in V} \left(\Pr[T(x+v) = 1] \right)$" can be obtained as follows: First choose a decision tree $\Gamma \in \Upsilon_T$ according to the distribution $\mathcal{D}_T$ and then for each leaf l labeled 1 in Γ, find the fraction of vectors in $x+V$ that follow the path leading to l. The weighted sum of these fractions is $\text{average}_{v \in V} \left(\Pr[T(x+v) = 1] \right)$. Thus,

$$\underset{v \in V}{\text{average}} \left(\Pr[T(x+v) = 1] \right) = \sum_{\Gamma \in \Upsilon_T} \mathcal{D}_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l^x|}{|V|} \right). \tag{1}$$

Now consider the quantity "$\Pr[T'(x) = 1]$". Test T' can be viewed in the following fashion: On input x, T' chooses a random decision tree $\Gamma \in \Upsilon_T$ according to the distribution $\mathcal{D}_T$, it then chooses a leaf l labeled 1 in Γ with probability proportional to the fraction of vectors $v \in V$ that are accepted along the path leading to l (i.e., $|V_l|/|V|$), queries x on all variables in $\text{var}(l)$ and accepts if $I_l^x = 1$ and rejects otherwise. This gives us the following expression for $\Pr[T'(x) = 1]$.

$$\Pr[T'(x) = 1] = \sum_{\Gamma \in \Upsilon_T} \mathcal{D}_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l|}{|V|} \cdot I_l^x \right) \tag{2}$$

From Equations (1) and (2), we obtain that it suffices to prove that $|V_l^x| = I_l^x \cdot |V_l|$ for all leaves l labeled 1 in order to prove the lemma.

Observe that $|V_l|$ is non-empty since l is labeled 1. Hence, by Lemma 8, $|V_l| = |V_l^x|$ if V_l^x is also non-empty. It now suffices to show that V_l^x is non-empty iff $I_l^x = 1$.

Suppose V_l^x is non-empty. Then there exists $y \in x + V$ that does not violate any constraint involving only the variables $\text{var}(l)$. But y and x satisfy the same set of constraints. Hence, x also does not violate any constraint involving only the variables $\text{var}(l)$. Thus, $I_l^x = 1$.

Now, for the other direction, suppose $I_l^x = 1$. Then the values of the variables $\text{var}(l)$ of x do not violate any constraint in $V^\perp$. Hence, there exists $u \in V$ that has the same values as x for the variables $\text{var}(l)$. Let $v \in V_l$. Then, the vector $x - u + v \in x + V$ has the same values for the variables $\text{var}(l)$ as v. Hence, V_l^x is non-empty. This concludes the proof of the lemma. □

The above lemma proves that T' inherits its acceptance probability from T. As mentioned earlier, T' inherits its query complexity from T. Thus, the query complexity of T' is at most q. Hence T' is a 1-sided error non-adaptive (ε, μ, q)-test for V. □

6. RANDOM CODES REQUIRE A LINEAR NUMBER OF 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 4. *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. Notice that a variable may appear several times in a constraint.

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 5 (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 6 (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*
$$\forall S \subset R,\ |S| \le \gamma n,\ |N(S)| > \lambda \cdot |S|.$$
- *G is called an (λ, γ)-*right unique neighbor expander *if*
$$\forall S \subset R,\ |S| \le \gamma n,\ |N^1(S)| > \lambda \cdot |S|.$$
- *G is called an (λ, γ)-*right odd expander *if*
$$\forall S \subset R,\ |S| \ge \gamma n,\ |N^{odd}(S)| > \lambda \cdot |S|.$$

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 6.

The following lemmas are proved using standard techniques for analysis of expansion of random graphs, such as those appearing in e.g. [4, 17]. We defer the proofs to appendix A.

Lemma 13. *There exists a constant $r > 0$ such that for any integers $c \ge 5, d \ge 2$, a random (c,d)-regular graph is with high probability a $(1, r \cdot d^{-2})$-left unique neighbor expander.*

Lemma 14. *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 Are Hard to Test

We are ready to prove Theorem 4.

Lemma 15. *For any odd integer $c \ge 5$, 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 Lemma 15. The theorem follows. □

Proof (of Lemma 15): Given odd $c \ge 5$ 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 13. We conclude the existence of constant $r > 0$, such that for any $d \ge 5$, G is with high probability a $(1, r \cdot c^{-2})$-right unique neighbor expander. This implies that if $|T| \le r \cdot c^{-2} \cdot |R|$, then $N^{odd}(T) \neq \emptyset$ because $N^{odd}(T) \supseteq N^1(T)$ and $N^1(T) \neq \emptyset$.

Lemma 14 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 14 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,

$$N^1(S_x) \subseteq \{j\} \tag{3}$$

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 $|S_x| \le r \cdot d^{-2} \cdot n$ then by lemma 13 $|N^1(S_x)| \ge |S_x| > 1$, in contradiction to equation (3). We conclude that for any $x \in \mathcal{C}(G_{-j})$, $|x| \ge r \cdot d^{-2}$, so $\mathcal{A}(G)$ is ε-separating for ε satisfying:

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

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

7. REDUCING dLIN TO 3LIN

This section proves Theorem 5 which directly follows from the final theorem of this section. The randomized construction from section 6 produces d-linear formulae 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 formulae. First we give a reduction from d-linear to $\lceil \frac{d}{2} \rceil + 1$-linear formulae, and then apply it $\log d$ times to get 3-linear formulae.

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 $x_{\lceil \frac{d}{2} \rceil + 1} \oplus \ldots \oplus x_d \oplus z_i = 0$. Let $V \subseteq \{0,1\}^n$ be the vector space 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 in $\{0,1\}^{n+m}$ 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 16. *$\mathcal{R}(\mathcal{A})$ 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. $\square$

LEMMA 17. *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 $x = x'_1 \ldots x'_n$. 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. $\square$

LEMMA 18. *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,

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

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

THEOREM 19. *Let $V \subseteq \{0,1\}^n$ be a vector space 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 in $\{0,1\}^{n^\star}$, 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*

$$m^\star = dm \ ; \qquad n^\star = n + (d-1)m \ ;$$

$$\varepsilon^\star = \frac{\varepsilon}{1 + (d-1)m/n} \ ; \qquad q^\star = \frac{2q}{d+2} \ ;$$

$$\mu^\star = \mu + \frac{q}{m} \cdot \frac{d+2}{d+1} \ .$$

PROOF. The theorem follows from lemmas 16, 17, 18. $\square$

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.

8. REFERENCES

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

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

[3] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science*, pages 93–102, Vancouver, Canada, 16–19 Nov. 2002.

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

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

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

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

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

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

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

[11] O. Goldreich and D. Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002.

[12] O. Goldreich and M. Sudan. Locally testable codes and PCPs of almost linear length. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science*, pages 13–22, Vancouver, Canada, 16–19 Nov. 2002.

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

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

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

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

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

APPENDIX

A. PROOFS FROM SECTION 6

PROOF (OF LEMMA 13): We need a couple of lemmas, the proof of which will follow.

LEMMA 20. *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 21. *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 20, G is a (α, ε)-right expander for any ε satisfying equation (20).

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 (20). Finally, by lemma 21, we get that G is whp a $(1, rd^{-2})$-left unique neighbor expander. □

PROOF (OF LEMMA 20): 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.

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

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 $\kappa = e^{1+\alpha} \cdot \left(\frac{\alpha d}{c}\right)^{c-\alpha}$.

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

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

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

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.

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

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

Proof (of Lemma 21): 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 14): In the proof, we make use of the following theorem (see [13])

Theorem 22 (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 $T \subseteq R \quad |T| = t \geq \mu m$. 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 22) 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:

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

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