SIAM J. COMPUT.
Vol. 35, No. 1, pp. 1–21
© 2005 Society for Industrial and Applied Mathematics

SOME 3CNF PROPERTIES ARE HARD TO TEST*

ELI BEN-SASSON†, PRAHLADH HARSHA‡, AND SOFYA RASKHODNIKOVA§

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 study the query complexity of tests that distinguish (with high probability) between strings in P_φ and strings that are far from P_φ in Hamming distance. We prove that there are 3CNF formulae (with $O(n)$ clauses) such that testing for the associated property requires $\Omega(n)$ queries, even with adaptive tests. This contrasts with 2CNF formulae, whose associated properties are always testable with $O(\sqrt{n})$ queries [E. Fischer et al., *Monotonicity testing over general poset domains*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, ACM, New York, 2002, pp. 474–483]. Notice that for every negative instance (i.e., an assignment that does not satisfy φ) there are three bit queries that witness this fact. Nevertheless, finding such a short witness requires reading a constant fraction of the input, even when the input is very far from satisfying the formula that is associated with the property.

A property is *linear* if its elements form a linear space. We provide sufficient conditions for linear properties to be hard to test, and in the course of the proof include the following observations which are of independent interest:

1. In the context of testing for linear properties, adaptive two-sided error tests have no more power than nonadaptive one-sided error tests. Moreover, without loss of generality, any test for a linear property is a linear test. A linear test verifies that a portion of the input satisfies a set of linear constraints, which define the property, and rejects if and only if it finds a falsified constraint. A linear test is by definition nonadaptive and, when applied to linear properties, has a one-sided error.
2. Random low density parity check codes (which are known to have linear distance and constant rate) are not locally testable. In fact, testing such a code of length n requires $\Omega(n)$ queries.

Key words. sublinear algorithms, lower bounds, property testing, CNF formulae, locally testable codes

AMS subject classification. 68Q17

DOI. 10.1137/S0097539704445445

1. Introduction.

Property testing deals with a relaxation of decision problems, where one must determine whether an input belongs to a particular set, called a *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 [RS96] in the context of linear functions and was

*Received by the editors July 30, 2004; accepted for publication (in revised form) March 17, 2005; published electronically September 8, 2005. A preliminary version of this paper appeared in the *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003 [BHR03].

http://www.siam.org/journals/sicomp/35-1/44544.html

†Computer Science Department, Technion - Israel Institute of Technology, Haifa, Israel (eli@eecs.harvard.edu; elli@cs.technion.ac.il). This work was done while the author was a postdoctoral researcher at the Department of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, and Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. This author's work was supported by NSF grants CCR 0133096, CCR 9877049, CCR 9912342, and CCR 0205390, and by NTT Award MIT 2001-04.

‡Microsoft Research, 1065 La Avenida, Mountain View, CA 94043 (pharsha@microsoft.com). This work was done at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 and was supported in part by NSF Award CCR 9912342 and NTT Award MIT 2001-04.

§Weizmann Institute of Science, Rehovot, Israel (sofya@theory.csail.mit.edu). This work was done at the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

applied to combinatorial objects, especially graphs, by Goldreich, Goldwasser, and Ron [GGR98]. Property testing has recently become quite an active research area; see [Ron01, Fis01] for surveys on the topic.

One of the important problems in property testing is characterizing properties that can be tested with a sublinear[1] number of queries to the input (cf. [GGR98, AKNS01, New02, FN04, AFKS00, Fis01, GT03]; see also section 1.1 for more information). Our paper continues this line of research by trying to relate the testability of properties over the binary alphabet with their formula complexity. We prove a linear lower bound for testing some properties with very small formula complexity, thus showing that the formula complexity of the property does not always help to assess the testability of the property. In section 1.1 several strong lower bounds on the query complexity [GGR98, GT03, BOT02, GR02] are discussed and compared to our lower bound.

Testing kCNFs. A *property* is a collection of strings of a fixed size n. Every property over the binary alphabet can be represented by a conjunctive normal form (CNF) formula, whose set of satisfying assignments equals the set of strings in the property. Testing this property can be viewed as testing whether a given assignment to Boolean variables of the corresponding CNF is close to one that satisfies the formula.[2] Goldreich, Goldwasser, and Ron [GGR98] prove that there exist properties over the binary alphabet that require testing algorithms to read a linear portion of the input. This implies that testing assignments to general CNF formulae is hard. A natural question is whether restricting CNF formulae to a constant number of variables, k, per clause allows for faster testers. Observe that the standard reduction from satisfiability (SAT) to 3SAT does not apply because it introduces auxiliary variables and thus changes the testing problem.

At first glance, there seems to be hope of obtaining good testers for every fixed k because, for any assignment that does not satisfy the formula, there exists a set of k queries that witnesses this fact. Indeed, Fischer et al. [FLN$^+$02] prove that properties expressible as sets of satisfying assignments to 2CNF formulae are testable with $O(\sqrt{n})$ queries, where n is the length of the input. However, we will show that, already for $k = 3$, testing whether an input assignment is close to satisfying a fixed kCNF formula might require a linear number of queries.

Results and techniques. We show the existence of families of 3CNF formulae over n variables (for arbitrarily large n) of size $O(n)$ such that the corresponding properties are not testable with $o(n)$ queries. Thus, we present a gap between 2CNFs and 3CNFs. Our lower bound applies to *adaptive* tests, i.e., tests in which queries might depend on the answers to previous queries. This gives a class of properties which are easy to decide in the standard sense[3] but are hard to test.

Each hard 3CNF property we use is a linear[4] space $V \subseteq \{0,1\}^n$ that can be expressed as the set of solutions to a set of homogeneous linear constraints of weight 3 (i.e., a 3LIN formula). While proving the lower bound, we show that every adaptive

[1] We measure the query complexity as a function of the input length. Thus, linear (sublinear, respectively) query complexity means query complexity that is linear (sublinear) in the input length.

[2] Our problem should not be confused with the problem of testing whether a CNF formula is "close" to being satisfiable (under a proper definition of closeness). In our problem the CNF formula is fixed and known to the tester. See section 1.1 for a discussion of the seemingly related problem.

[3] A property is easy to decide in the following standard sense: given the entire assignment, it can be checked in time linear in the number of variables and size of the formula, whether or not the assignment is a satisfying one.

[4] We work over the field with two elements and our linear space is defined over this field.

two-sided error test for checking membership in a vector space can be converted to a nonadaptive one-sided error test with the same query complexity and essentially identical parameters. This allows us to consider only one-sided error nonadaptive tests. In order to prove that a particular linear space V is hard, 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 (making few queries) fails to reject a random b (selected according to the distribution $\mathcal{B}$) with significant probability.

We now give a rough picture of how to get a vector space V that is hard to test and a distribution $\mathcal{B}$ that shows this hardness (per Yao's minimax principle). Fix $0 < \kappa < 1$ and let V be the set of solutions to a system $\mathcal{A}$ of $m = \kappa n$ random linear constraints over n Boolean variables, where each constraint is the sum of a *constant* number of randomly selected variables, and each variable appears in a *constant* number of constraints. Such linear spaces are called random low density parity check (LDPC) codes.[5] These codes were introduced by Gallager [Gal63], who showed that they have constant rate and (with high probability) large minimal distance. It is possible to show that with high probability the random constraints are linearly independent. Our bad distribution $\mathcal{B}$ is the uniform distribution over vectors that satisfy all but one random constraint of $\mathcal{A}$. Since the constraints are linearly independent, this distribution is well defined. By definition, each input chosen according to $\mathcal{B}$ is not in the property. It is less obvious, but still true, that each such input is *far* from the property. The tricky part is to show that every deterministic test making $o(n)$ queries will fail to reject a random input chosen according to $\mathcal{B}$.

A natural way to test if an input belongs to V is to select a few random constraints in $\mathcal{A}$, query all entries lying in their supports, and accept if and only if all constraints are satisfied. This test always accepts inputs in V. It correctly rejects an input distributed according to $\mathcal{B}$ when the unique random constraint falsified by the input is queried. This method is costly in query complexity because there are $O(n)$ constraints, and only one randomly chosen constraint is not satisfied. A more efficient way to attack the distribution $\mathcal{B}$ would be to use linearity, as follows. If an input satisfies a set of constraints, it must satisfy their sum; if it falsifies exactly one of the constraints in the sum, it must falsify the sum. Thus, one might choose a set of constraints in $\mathcal{A}$, take their sum, and query the entries in the support of the sum. The summation might cancel out some entries (namely, those that appear in an even number of summands) and reduce the query complexity. This suggests the following general test for testing membership in V: query the input in a small subset of entries and accept if and only if these entries satisfy all possible sums (of constraints in $\mathcal{A}$), whose support lies entirely within the small subset. In fact, it can be easily observed that any nonadaptive one-sided error test for membership in V is of the above form. Furthermore, we prove that, without loss of generality, the only possible tests (for membership in a linear space) are of the above mentioned form (see Theorem 3.3). The crux of our proof consists of showing that this general method does not significantly reduce the query complexity. Namely, we show that the sum of *any* large subset of the constraints of $\mathcal{A}$ has large support, and thus results in large query complexity. The reason for this phenomenon is the underlying *expansion* of the random set of constraints. Thus, any deterministic

[5]LDPC codes are linear codes defined as solutions to a system of linear constraints with small support.

testing algorithm making $o(n)$ queries essentially checks only $o(m)$ constraints of $\mathcal{A}$, and thus will reject a random input from $\mathcal{B}$ with subconstant probability.

Connections to coding theory. 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 play a vital role in probabilistically checkable proof (PCP) constructions and are of fundamental importance in theoretical computer science. Recently, Ben-Sasson et al. [BSVW03, BGH$^+$04], following the work of Goldreich and Sudan [GS02], proved the existence of such codes, which achieve linear distance and near constant rate, resulting in better PCP constructions.

As mentioned earlier, the vector spaces we use (which are hard to test) are built upon random LDPC codes, which are heavily studied in coding theory (cf. [Gal63] and [Spi95, Chap. 2] and references therein). It follows from an intermediate step in our proof that this important class of codes is not locally testable. 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. In his thesis, Spielman informally discusses why expander codes might not be locally testable and states that a high level of redundancy among the constraints of the code might be required to make it testable ([Spi95, Chap. 5]. In our proof, we make this argument formal and prove that random (c, d)-regular LDPC codes are not locally testable (see Theorem 3.7). This sheds some light on the question of optimal locally testable codes. The existence of such optimal codes that (i) achieve constant rate, (ii) achieve linear distance, and (iii) are locally testable (or even testable with a sublinear number of queries) remains an interesting open problem.

1.1. Connection to previous results.

Classes of testable properties. One of the important problems in property testing is characterizing properties that can be tested with a sublinear number of queries to the input. A series of works identified classes of properties testable with constant query complexity. Goldreich, Goldwasser, and Ron [GGR98] found many such properties. Alon et al. [AKNS01] proved that all regular languages are testable with constant complexity. Newman [New02] extended their result to properties that can be computed by oblivious read-once constant-width branching programs. Fischer and Newman [FN04] demonstrated a property that requires superconstant query complexity and is computable by a read-twice constant-width branching program, thus showing that Newman's result does not generalize to read-twice branching programs. Several papers [AFKS00, Fis05] worked on the logical characterization of graph properties testable with a constant number of queries. Goldreich and Trevisan [GT03] provide a characterization of properties testable with a constant number of queries and one-sided error in the framework of graph partition properties.

Linear lower bounds. The published linear lower bounds are the aforementioned generic bound due to Goldreich, Goldwasser, and Ron [GGR98], later extended by Goldreich and Trevisan [GT03] to monotone graph properties in NP, and the bound for testing 3-coloring in bounded degree graphs due to Bogdanov, Obata, and Trevisan [BOT02]. In addition, there is a simple and elegant unpublished linear lower bound, observed by Madhu Sudan in a personal communication to the authors. His property consists of polynomials of degree at most $n/2$ over a finite field $\mathbb{F}_n$ of size n, where each polynomial is given by its evaluation on all elements of the field. It is not hard to see that every nonadaptive one-sided error test for this property requires linear query complexity. Since the property of low-degree polynomials is linear, our

reduction from general to nonadaptive one-sided error tests implies a linear lower bound for adaptive two-sided tests for this property. A related property, suggested by Oded Goldreich in a personal communication to the authors, consists of a random linear code. It is not hard to show, using similar reasoning, that with high probability testing this property requires linear query complexity. Observe that both of these properties (low degree polynomials and random codes) are easy to decide once all the input is read, but both cannot be represented by a family of 3CNF formulae.[6]

The aforementioned linear lower bounds of Sudan and Goldreich and of Bogdanov, Obata, and Trevisan 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.[7] But if the property is defined by a kCNF φ, this cannot happen. 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 that 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. A similar phenomenon is exhibited by Goldreich and Ron for testing bipartiteness in 3-regular, n-vertex graphs [GR02]. They showed a lower bound of $\Omega(\sqrt{n})$ on the query complexity; despite this, short witnesses of nonbipartiteness do exist in the form of odd cycles of length $\text{poly}(\log n)$. Our result strengthens this finding, since in our case the query complexity is linear, whereas the witness size is constant.

Testing an input kCNF. A related problem, but very different from ours, is that of testing whether an *input* kCNF formula is satisfiable. (Recall that in our setting the input is an assignment to a fixed kCNF formula.) The exact version of this problem is a classical NP-complete problem. The property testing version was studied by Alon and Shapira [AS03]. They showed that satisfiability of kCNF formulae is testable with complexity independent of the input size.[8] In contrast, our problem is very easy in its exact version but hard in its property testing version for $k \geq 3$.

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 (over some underlying field). In this paper, strings are over a 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 x and x'. 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*, positive error* η_+*, negative error* η_- *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 $1 - \eta_+$, and accepts strings that are ε-far from $\mathcal{P}$ with probability at most η_-, for some $0 \leq \eta_- < 1 - \eta_+ \leq 1$.

[6] In other words, these properties cannot be decided by a family of circuits of depth 2, where the output gate of the circuit is an AND-gate, the next level of gates are all OR-gates of fan-in 3, and the inputs to the OR-gates are either the input bits or their negations.

[7] For example, in Sudan's construction any evaluation of a polynomial on d points can be extended to an evaluation of a polynomial of degree $d' > d$. Thus, $n/2$ values of the polynomial cannot prove or disprove that the polynomial has degree at most $n/2$.

[8] Complexity of a testing problem depends on the definition of distance; in Alon and Shapira's work the distance from the input to a satisfiable formula is defined as the number of clauses that have to be removed to make the input formula satisfiable.

Note that the positive error η_+ is the maximum error made by the test on the YES-instances (i.e., strings in $\mathcal{P}$), and the negative error η_- is the maximum error made on NO-instances (i.e., strings ε-far from $\mathcal{P}$). Sometimes we refer to $\eta_+ + \eta_-$ as the sum of errors made by the test T. A test is said to have *error* η if $\eta_+, \eta_- \leq \eta$ (for $\eta < \frac{1}{2}$). 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 η is referred to as an (ε, η)-*test* (analogously, $(\varepsilon, \eta_+, \eta_-)$-*test*). An ε-test denotes a test with distance parameter ε. A property is (ε, η, q)-*testable* if it has an (ε, η)-test that makes at most q queries on every input; $(\varepsilon, \eta_+, \eta_-, q)$-*testable* is defined analogously.

Two special classes of tests are of interest. An algorithm is *nonadaptive* if it makes all queries in advance before getting the answers. Namely, a query may not depend on the answers to previous queries. An algorithm has a *one-sided error* if it always accepts an input that has the property. In other words, an algorithm has a one-sided error if the positive error η_+ is 0.

CNF and linear formulae. Recall that a Boolean formula is in 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 needs 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.

Random regular bipartite graphs and LDPC codes. Let $G = \langle L, R, E\rangle$ be a bipartite multigraph, 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 if 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[9] is obtained by selecting a random matching between cn "left" nodes labeled $\{v_1^1, \ldots, v_1^c, v_2^1, \ldots, v_n^c\}$ and $dm = cn$ "right" nodes labeled $\{u_1^1, \ldots, u_m^d\}$, collapsing every c consecutive nodes on the left to obtain n c-regular vertices, and collapsing every d consecutive nodes on the right to obtain m d-regular vertices. Formally, let $L = \{v_1, \ldots, v_n\}$, $R = \{u_1, \ldots, u_m\}$, and connect vertex v_i to u_j if and only if there exist $\alpha \in [c], \beta \in [d]$ such that (v_i^α, u_j^β) is an edge of the random matching. Notice that the resulting graph may be a multigraph (i.e., have multiple edges between two vertices). The code associated with G, called an LDPC code, was first described and analyzed by Gallager [Gal63].

DEFINITION 2.1 (LDPC Code). *Let $G = \langle L, R, E\rangle$ be a bipartite multigraph 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 jth constraint is $A_j(x_1, \ldots, x_n) = \sum_{i \in N(j)} x_i \bmod 2$. (Notice that a variable may appear several times in a constraint because G is a multigraph.) Let $\mathcal{A}(G)$ be the d*LIN *formula*

[9] Typically, one fixes c, d to be constants, whereas n, m are unbounded.

$\mathcal{A}(G) = \bigwedge_{j=1}^{m}\big(A_j(x) = 0\big)$. *The code defined by* G *is the property defined by* $\mathcal{A}(G)$*, namely,*

$$\mathcal{C}(G) = \{x \in \{0,1\}^n | \forall j \in [m] \quad A_j(x) = 0\}.$$

A random (c,d)*-regular LDPC code of length* n *is obtained by taking* $\mathcal{C}(G)$ *for a random* (c,d)*-regular graph* G *with* n *left vertices.*

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

THEOREM 3.1 (main). *There exist* $0 < \delta^\star, \varepsilon^\star, \eta^\star < 1$ *such that, for every sufficiently large* n*, there is a* 3CNF *formula* φ *on* $n^\star = \Theta(n)$ *variables with* $\Theta(n)$ *clauses such that every adaptive* $(\varepsilon^\star, \eta_+, \eta_-, q)$*-test for* φ *with the sum of errors* $\eta_+ + \eta_- \le \eta^\star$ *makes at least* $q = \delta^\star n^\star$ *queries.*

To prove Theorem 3.1, we need to find 3CNF formulae that define hard properties. Our main idea is to work with linear properties (i.e., vector spaces). We prove that, for linear properties, tests of a very simple kind, which we call *linear*, are as powerful as general tests. In particular, linear tests are nonadaptive and have a one-sided error. Working with linear properties allows us to focus on proving a lower bound for this simple kind of tests—bypassing often insurmountable issues of adaptivity and two-sided error.

To explain how we find 3CNF formulae that define hard *linear* properties, we need some linear algebra terminology. Let $\mathbb{F}$ be a field. We say that two vectors $u, v \in \mathbb{F}^n$ are orthogonal to each other (denoted $u \perp v$) if $\sum_{i=1}^{n} u_i \cdot v_i = 0$. Furthermore, we say that a vector u is orthogonal to a subset $S \subseteq \mathbb{F}^n$ (denoted by $u \perp S$) if $u \perp v$ for all vectors $v \in S$. For a linear space $V \subseteq \mathbb{F}^n$ over the field $\mathbb{F}$, the *dual space* $V^\perp$ is defined as the set of all vectors orthogonal to V (i.e., $V^\perp \stackrel{\triangle}{=} \{u \in \mathbb{F}^n : u \perp V\}$). For $I \subseteq [n]$, let $V_I^\perp$ be the subset of $V^\perp$ composed of all vectors with support in I (i.e., $u \in V_I^\perp$ if and only if $u \in V^\perp$ and the indices of nonzero entries of u lie in I).

DEFINITION 3.2 (linear test). *A test for a linear property* $V \subseteq \mathbb{F}^n$ *is called a* linear test *if it is performed by selecting* $I = \{i_1, \ldots, i_q\} \subseteq [n]$ *(according to some distribution), querying* w *at coordinates* I*, and accepting if and only if* $w \perp V_I^\perp$.

Linear tests are by definition nonadaptive and have only a one-sided error (members of V are always accepted). Since the inception of property testing, linear properties have been invariably tested by linear tests (starting with [BLR93]). The following theorem shows this is not a coincidence.

THEOREM 3.3 (linear properties have linear tests). *If a linear property* $V \subseteq \mathbb{F}^n$ *over a finite field* $\mathbb{F}$ *has a two-sided error adaptive* $(\varepsilon, \eta_+, \eta_-, q)$*-test, then it has a linear* $(\varepsilon, 0, \eta_+ + \eta_-, q)$*-test.*

The proof of Theorem 3.3 appears in section 5. The reduction to linear tests does not increase the overall error but rather shifts it from the YES-instances to the NO-instances, maintaining the sum of errors $\eta_+ + \eta_-$. Although stated for general finite fields, this theorem is used in our paper only for linear properties over the binary alphabet, namely, with $\mathbb{F} = GF(2)$.

Consider a vector space $V \subseteq GF(2)^n$ and let $\mathcal{A} = (A_1, \ldots, A_m)$ be a basis for the dual space $V^\perp$. Denote the ith coordinate of $x \in GF(2)^n$ by x_i. For two vectors $x, y \in GF(2)^n$, let $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i \bmod 2$. We can view each vector $A_i \in \mathcal{A}$ as a linear constraint on Boolean variables $x_1, \ldots, x_n$ of the form $\langle x, A_i\rangle = 0$. This gives us a way to see a vector space as a set of vectors satisfying all constraints in the dual

space, or, equivalently, in the basis of the dual space: $V = \{x | \langle x, A_i \rangle = 0 \text{ for all } A_i \in \mathcal{A}\}$. Linear constraints can be thought of as linear formulae.

Let $|x|$ denote the size of the support of vector $x \in GF(2)^n$. 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, to prove Theorem 3.1 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 linear tests. Yao's principle implies that to prove that each low-query *probabilistic* linear test fails on some input, it is enough to give a distribution on the inputs on which each low-query *deterministic* linear test fails. Therefore, we need to exhibit a distribution on vectors that are far from the vector space but are orthogonal with high probability to any fixed set of linear constraints that have support $\leq q$.

DEFINITION 3.4 (hard linear properties). *Let $V \subseteq GF(2)^n$ 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 $x \in GF(2)^n$ that falsifies exactly one constraint in $\mathcal{A}$ has $|x| \geq \varepsilon n$.*
- *$\mathcal{A}$ is* (q, μ)-local *if every $\alpha \in GF(2)^n$ that is a sum of at least μm vectors in $\mathcal{A}$ has $|\alpha| \geq q$.*

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 distribution is over vectors which are ε-far from the vector space. (To see this, notice that if the distance of x from $y \in V$ is less than εn, then $|x + y| < \varepsilon n$ and $x + y$ falsifies exactly one constraint, contradicting the first condition.)

The second condition ensures that the distribution is hard to test. Assume that each deterministic linear test corresponds to some vector $u \in V^\perp, |u| \leq q$. (This is oversimplified because a deterministic linear test may read several dual vectors, whose combined support size is at most q. However, this simple case clarifies our approach and is not far from the formal proof given in section 4.) Vector u can be expressed as a linear combination of vectors in the basis: $u = \sum_{j \in J} A_j$ for some $J \subset [m]$. Let A_k be the (random) constraint falsified by a vector w in our hard distribution. Clearly, u will reject w if and only if $k \in J$. The second condition implies that this will occur with probability at most μ. This intuitive discussion is formalized by the following theorem, proved in section 4.

THEOREM 3.5. *Let $V \subseteq GF(2)^n$ be a linear space. If $V^\perp$ has an ε-separating (q, μ)-local basis $\mathcal{A} = (A_1, \ldots, A_m)$ and $0 < \mu < 1/2$, then every linear ε-test for it with error $\leq 1 - 2\mu$ requires q queries.*

We now turn to constructing linear spaces that are hard to test. In particular, we show that for sufficiently large constants c, d, with high probability a random (c, d)-regular LDPC code (per Definition 2.1) is hard according to Definition 3.4. The proof of this lemma, which uses the probabilistic method, appears in section 6. (We do not attempt to optimize constants.)

LEMMA 3.6 (hard linear properties exist). *Fix odd integer $c \geq 7$ and constants*

$\mu, \varepsilon, \delta, d > 0$ satisfying

$$\mu \leq \frac{1}{100} \cdot c^{-2}; \quad \delta < \mu^c; \quad d > \frac{2\mu c^2}{(\mu^c - \delta)^2}; \quad \varepsilon \leq \frac{1}{100} \cdot d^{-2}.$$

*Then, for all sufficiently large n, with high probability for a random (c,d)-regular graph G with n left vertices, the d*LIN *formula $\mathcal{A}(G)$ (as in Definition 2.1) is linearly independent, ε-separating, and $(\delta n, \mu)$-local.*

We now have an abundance of dLIN formulae that are hard to test for sufficiently large d. As an immediate corollary, we conclude that random LDPC codes are hard to test.

THEOREM 3.7 (random (c,d)-regular LDPC codes are hard to test). *Let $c, d, \mu, \varepsilon, \delta$ satisfy the conditions of Lemma 3.6. For sufficiently large n, with high probability a random (c,d)-regular LDPC code $\mathcal{C}(G)$ of length n satisfies the following: Every adaptive $(\varepsilon, \eta_+, \eta_-, q)$-test for $\mathcal{C}(G)$ with the sum of errors $\eta_+ + \eta_- \leq 1 - 2\mu$ makes at least $q = \delta n$ queries.*

Proof. The proof follows directly from Lemma 3.6 and Theorems 3.3 and 3.5. □

The following reduction brings d down to 3 while preserving the conditions of Definition 3.4 (with smaller constants).

LEMMA 3.8 (reduction to 3CNFs). *Suppose $\mathcal{A} \subseteq \{0,1\}^n$ is a set of $m = \frac{c}{d}n$ vectors, each vector of weight at most d. Suppose furthermore $\mathcal{A}$ is* (i) *linearly independent,* (ii) *ε-separating, and* (iii) *$(\delta n, \mu)$-local. Then there exists a set $\mathcal{A}^\star \subset \{0,1\}^{n^\star}$ of $m^\star$ vectors, each vector of weight at most 3, such that $\mathcal{A}^\star$ is* (i) *linearly independent,* (ii) *$\varepsilon^\star$-separating, and* (iii) *$(\delta^\star n^\star, \mu^\star)$-local, for*

$$m^\star \leq 2dm\ ; \qquad n \leq n^\star \leq (2c+1) \cdot n\ ; \qquad \varepsilon \geq \varepsilon^* \geq \frac{\varepsilon}{(2c+1)};$$

$$\delta \geq \delta^\star \geq \frac{\delta}{d^{\log d + 1} \cdot (2c+1)}\ ; \qquad \mu^\star \leq \mu + \frac{\delta(\log d + 1)}{c}.$$

Lemma 3.8 is proved in section 7. We now complete the proof of our main theorem.

Proof of Theorem 3.1 (main). We start by fixing the following parameters:

$$c = 7; \quad \mu = \frac{1}{100} \cdot c^{-2}; \quad \delta = \frac{\mu^c}{2}; \quad d = \left\lceil \frac{4\mu c^2}{(\mu^c - \delta)^2} \right\rceil = \left\lceil 16 c^2 \mu^{1-2c} \right\rceil; \quad \varepsilon = \frac{1}{100} \cdot d^{-2}.$$

We pick $\mathcal{A}_n \subset \{0,1\}^n$ to be a linearly independent, $(\delta n, \mu)$-local, ε-separating collection of vectors, of weight $\leq d$. By Lemma 3.6, such a set $\mathcal{A}_n$ exists for our setting of $\mu, \varepsilon, \delta, d$ and sufficiently large n.

Next, let $\mathcal{A}^\star_{n^\star} \subset \{0,1\}^{n^\star}$ be a linearly independent, $(\delta^\star n^\star, \mu^\star)$-local, $\varepsilon^\star$-separating set of vectors of weight at most 3, ensured by Lemma 3.8 (where $\delta^\star, \mu^\star, \varepsilon^\star, n^\star$ are as stated in this lemma). Recall that for every 3LIN formula there is an equivalent 3CNF and let φ be the 3CNF formula equivalent to $\mathcal{A}^\star_{n^\star}$. Moreover, because $m^\star, n^\star = \Theta(n)$ and each 3LIN constraint translates into a constant number of 3CNF constraints, we conclude that the number of clauses in φ is linear in $n^\star$.

Notice $\delta^\star, \varepsilon^\star, \mu^\star > 0$ because $\delta, \varepsilon, \mu, d > 0$, and $\delta^\star, \varepsilon^\star < 1$ because $\delta, \varepsilon < 1$. Furthermore, for our setting of constants,

$$\mu^\star \leq \mu + \frac{\delta(\log d + 1)}{c} = \mu + \frac{\mu^c(\log(16c^2\mu^{-(2c-1)}) + 1)}{2c} < \frac{1}{2}.$$

Therefore, $0 < 1 - 2\mu* < 1$. Set $\eta^\star = 1 - 2\mu^\star$. By Theorem 3.5, every linear $\varepsilon^\star$-test for $\mathcal{A}^\star_{n^\star}$ with error $\leq \eta^\star$ requires $\delta^\star n^\star$ queries. Theorem 3.3 implies that every

adaptive $(\varepsilon^\star, \eta_+, \eta_-, q)$-test for $\mathcal{A}^\star_{n^\star}$ with $\eta_+ + \eta_- \leq \eta^\star$ makes at least $\delta^\star n^\star$ queries. This completes the proof of our main theorem. □

4. Lower bounds for linear tests: Proof of Theorem 3.5. We employ Yao's minimax principle. It states that to prove that every q-query randomized linear test fails with probability more than η, it is enough to exhibit a distribution $\mathcal{B}$ on the inputs for which every q-query deterministic linear test fails with probability more than η. 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 the $\mathcal{B}_i$'s. The comment after Definition 3.4 shows that the distribution $\mathcal{B}$ is over strings which are ε-far from V.

A deterministic linear test T is identified by a subset $I \subseteq [n], |I| = q$ and rejects the input w only if w is not orthogonal to $V_I^\perp$ (see Definition 3.2). Write each vector $u \in V^\perp$ in the basis $\mathcal{A}$ as $u = \mathcal{A} \cdot b_u$, where $\mathcal{A}$ is interpreted as the $m \times n$ matrix whose ith row is A_i and $b_u \in GF(2)^m$. Let $J_T = \cup_{V_I^\perp} \mathrm{supp}(b_u)$. We claim T rejects w distributed according to $\mathcal{B}$ if and only if the index of the unique constraint falsified by w belongs to J_T. This is because w is orthogonal to all but one $A_i \in \mathcal{A}$, so the subspace of $V^\perp$ that is orthogonal to w is precisely the span of $\mathcal{A} \setminus \{A_i\}$. Summing up, the probability that T rejects w is precisely $|J_T|/m$. We now give an upper bound on $|J_T|$. Notice that $V_I^\perp$ is a vector space, so the set $V' \stackrel{\triangle}{=} \{b_u : u \in V_I^\perp\}$ is also a vector space (it is the image of the vector space $V_I^\perp$ under the linear map $\mathcal{A}^{-1}$). Since $\mathcal{A}$ is (q, μ)-local, we know $|b_u| \leq \mu m$ for every $u \in V_I^\perp$. Thus, V' is a vector space over $GF(2)$ in which each element has support size $\leq \mu m$. We claim that $|J_T| = |\cup_{v' \in V'} \mathrm{supp}(v')| \leq 2\mu m$. To see this, pick a uniformly random vector of $v \in V'$. We claim that

$$\mathrm{E}_{v' \in V'} |v| = |\cup_{v'' \in V'} \mathrm{supp}(v'')|/2.$$

This follows from the linearity of expectation and the fact that the projection of V' onto any $j \in \cup_{V'} \mathrm{supp}(v')$ is a linear function, so the expected value of v'_j is $1/2$. Since $\mathcal{A}$ is (q, μ)-local, we know $\mathrm{E}_{v' \in V'} |v'| \leq \mu m$, which means that $|J_T| \leq 2\mu m$. This implies that our deterministic test (reading q entries of w) will detect a violation with probability at most $|J_T|/m \leq 2\mu$. The proof of Theorem 3.5 is complete. □

5. Reducing general tests to linear ones. In this section we prove Theorem 3.3 by presenting a generic reduction that converts any adaptive two-sided error test for a linear property to a linear test, as in Definition 3.2. We perform this reduction in two stages: we first reduce an adaptive test with two-sided error to an adaptive linear test (Theorem 5.3), maintaining the sum of the positive and negative errors $(\eta_+ + \eta_-)$, and then remove the adaptivity and maintain all other parameters (Theorem 5.6). The second reduction was suggested by Madhu Sudan. We state and prove these reductions for the general case when the linear spaces V considered are over any finite field $\mathbb{F}$, though we require them only for the case $\mathbb{F} = GF(2)$.

Preliminaries. 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 $\Upsilon_T = \{\Gamma_1, \Gamma_2, \ldots\}$ 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 $\Gamma(x)$.

We say that a test *detects a violation* if no string in V is consistent with the answers to the queries. By linearity, it is equivalent to having a constraint α in $V^\perp$

such that $\langle x, \alpha \rangle \neq 0$ for all $x \in \mathbb{F}^n$, 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 $x \in \mathbb{F}^n$, let V_l^x be the set of all vectors in $x + V$ that are consistent with the answers along the path leading to l.

CLAIM 5.1. *Let $\mathbb{F}$ be a finite field and $V \subseteq \mathbb{F}^n$ be a vector space. Let $x \in \mathbb{F}^n$. For any decision tree Γ and a leaf l in Γ, if both V_l and V_l^x are nonempty, then $|V_l| = |V_l^x|$.*

Proof. Let U be the set of all strings in V which have the element 0 in all the positions queried along the path leading to l. Since $0^n \in U$, we have that U is nonempty. 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. Reduction from adaptive two-sided to adaptive linear.

DEFINITION 5.2 (adaptive linear test). *A test for a linear property $V \subseteq \mathbb{F}^n$ is called* adaptive linear *if it is performed by making adaptive queries $I = \{i_1, \ldots, i_q\}$ (according to some distribution) to w and accepting if and only if $w \perp V_I^\perp$.*

Notice that adaptive linear tests have a one-sided error: every $w \in V$ is always accepted.

THEOREM 5.3. *Let $\mathbb{F}$ be a finite field and $V \subseteq \mathbb{F}^n$ a vector space. If V has an adaptive $(\varepsilon, \eta_+, \eta_-, q)$-test T, then it has a (one-sided error) adaptive linear $(\varepsilon, 0, \eta_+ + \eta_-, q)$-test T'.*

Proof. Let $T = (\Upsilon_T, \mathcal{D}_T)$ be a two-sided error (adaptive) (ε, η, q)-test for V. To convert T to an adaptive linear one, 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 labeled *optimally* if its label is 0 when the query answers on the path to l falsify some constraint in $V^\perp$, and its label is 1 otherwise. We relabel the leaves of each tree Γ in Υ_T *optimally* to obtain the tree Γ_{opt}.

Relabeling produces a one-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 T'.

DEFINITION 5.4. *Given a two-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' is adaptive linear and has the same query complexity as T. It remains to check that T' has error $\eta_+ + \eta_-$ on negative instances.

For any $x \in \mathbb{F}^n$ and any test T, let ρ_x^T be the average acceptance probability of test T over all strings in $x + V$, i.e., $\rho_x^T = \text{average}_{y \in x+V}\big(\Pr[T(y) = 1]\big)$. For notational brevity, we denote $\rho_{0^n}^T$, the average acceptance probability of strings in V, by ρ^T. Observe that, for the new test T', for each input x, $\Pr[T'(x) = 1] = \rho_x^{T'}$.

Claim 5.5 below shows that the transformation to a one-sided error test given by Definition 5.4 increases the acceptance probability of any string not in V by at most $\rho^{T'} - \rho^T$. Notice that all vectors in $x + V$ have the same distance to V. Therefore if x is ε-far from V, then $\rho_x^T \leq \eta_-$. Together with Claim 5.5, it implies that for all vectors x that are ε-far from V, the error is low:

$$\Pr[T'(x) = 1] = \rho_x^{T'} \leq \rho^{T'} - \rho^T + \rho_x^T \leq 1 - (1 - \eta_+) + \eta_- = \eta_+ + \eta_-.$$

This completes the proof of Theorem 5.3 ∎

CLAIM 5.5. *$\rho^T - \rho_x^T \le \rho^{T'} - \rho_x^{T'}$ for any vector $x \in \mathbb{F}^n$.*

Proof. Let $x \in \mathbb{F}^n$. It is enough to prove that relabeling one leaf l of a decision tree Γ in Υ_T *optimally* does not decrease $\rho^T - \rho_x^T$. Then we obtain the claim by relabeling 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 relabeled 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, ρ_x^T does not increase. Hence, $\rho^T - \rho_x^T$ does not decrease.

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

It suffices to show that $|V_l| \ge |V_l^x|$. Since the path leading to l does not falsify any constraint, V_l is nonempty. If V_l^x is empty, we are done. Otherwise, both V_l and V_l^x are nonempty, and by Claim 5.1, $|V_l| = |V_l^x|$. □

5.2. Reduction to linear tests. In this section, we remove the adaptivity from the linear tests. 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 until it reads the final variable. Hence, any adaptive decision based on such a partial view does not help.

THEOREM 5.6. *If $V \subseteq \mathbb{F}^n$ is a vector space over a finite field $\mathbb{F}$ that has an adaptive linear $(\varepsilon, 0, \eta, q)$-test, then it has a (nonadaptive) linear $(\varepsilon, 0, \eta, q)$-test.*

Proof. Let T be an adaptive linear $(\varepsilon, 0, \eta, q)$-test for V. Let Υ_T and $\mathcal{D}_T$ be the associated set of decision trees and the corresponding distribution, respectively.

DEFINITION 5.7. *Given an adaptive linear 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, and reject if a violation is detected; otherwise accept.*

T' makes the same number of queries as T. Moreover, the queries depend only on the random $v \in V$ and not on the input x. Hence, the test T' is nonadaptive. The following claim relates the acceptance probability of T' to the average acceptance probability of T.

CLAIM 5.8. *Let T be an adaptive linear test and T' the nonadaptive version of T (as in Definition 5.7). Then, for any string $x \in \mathbb{F}^n$,*

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

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 $\mathrm{var}(l)$ denote the set of variables queried along the path leading to l in the tree Γ. Following the notation of Claim 5.1, 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 if and only if x does not violate any constraint in $V^\perp$ involving only the variables $\mathrm{var}(l)$. 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 $I_l^x = 1$ if and only if $T'(x) = 1$.

The quantity "$\operatorname{average}_{v \in V} (\Pr[T(x+v) = 1])$" can be obtained as follows: First,

choose a decision tree $\Gamma \in \Upsilon_T$ according to the distribution $\mathcal{D}_T$. 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 $\operatorname{average}_{v \in V}(\Pr[T(x+v)=1])$. Thus,

$$\operatorname*{average}_{v \in V}(\Pr[T(x+v)=1]) = \sum_{\Gamma \in \Upsilon_T} \mathcal{D}_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l^x|}{|V|} \right). \tag{5.1}$$

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 $\operatorname{var}(l)$, accepts if $I_l^x = 1$, and rejects otherwise. This gives us the following expression for $\Pr[T'(x)=1]$:

$$\Pr[T'(x)=1] = \sum_{\Gamma \in \Upsilon_T} \mathcal{D}_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l|}{|V|} \cdot I_l^x \right). \tag{5.2}$$

From (5.1) and (5.2), 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 claim.

Observe that $|V_l|$ is nonempty since l is labeled 1. Hence, by Claim 5.1, $|V_l| = |V_l^x|$ if V_l^x is also nonempty. It now suffices to show that V_l^x is nonempty if and only if $I_l^x = 1$.

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

Now, for the other direction, suppose $I_l^x = 1$. Then the values of the variables $\operatorname{var}(l)$ 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 $\operatorname{var}(l)$. Let $v \in V_l$. Then, the vector $x - u + v \in x + V$ has the same values for the variables $\operatorname{var}(l)$ as v. Hence, V_l^x is nonempty. This concludes the proof of the claim. □

The above claim proves that T' inherits its acceptance probability from T. As mentioned earlier, T' inherits its query complexity from T. Hence T' is a linear $(\varepsilon, 0, \eta, q)$-test for V. □

6. Random codes require a linear number of queries. In this section we prove Lemma 3.6. We start by analyzing the expansion properties of random regular graphs.

6.1. Some expansion properties of random regular graphs. To prove that a random $\mathcal{C}(G)$ obeys Definition 3.4 with high probability, 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 graph G implies that it satisfies the requirements. We need the following notions of neighborhood and expansion.

DEFINITION 6.1 (neighborhood). *Let $G = \langle V, E \rangle$ be a graph. For $S \subset V$, let*

(i) *$N(S)$ be the set of neighbors of S.*
(ii) *$N^1(S)$ be the set of* unique *neighbors of S, i.e., vertices with exactly one neighbor in S.*
(iii) *$N^{odd}(S)$ 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 6.2 (expansion). *Let $G = \langle L, R, E\rangle$ be a bipartite graph with $|L| = n, |R| = m$.*

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

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

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

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

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

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

Notice that the definitions of an expander and a unique neighbor expander deal with subsets of size *at most* γn, whereas the definition of an odd expander deals with subsets of size *at least* γn. Left expanders (all three of them) are defined analogously by taking $S \subset L$ in Definition 6.2.

Lemmas 6.3 and 6.6 are proved using standard techniques for analysis of expansion of random graphs, such as those appearing in, for example, [CS88, Spi95].

LEMMA 6.3. *For any integers $c \geq 7, d \geq 2$ and sufficiently large n, a random (c, d)-regular graph with n left vertices is with high probability a $(1, \frac{1}{100} \cdot d^{-2})$-left unique neighbor expander.*

Proof. We need the following claims, the proofs of which will follow.

CLAIM 6.4. *For any integers $c \geq 2, d$, any constant $\alpha < c - 1$, and sufficiently large n, a random (c, d)-regular bipartite graph with n left vertices is with high probability a (α, ε)-left expander for any ε satisfying*

$$\varepsilon \leq \left(2e^{(1+\alpha)} \cdot \left(\frac{\alpha d}{c} \right)^{(c-\alpha)} \right)^{-\frac{1}{c-\alpha-1}}. \tag{6.1}$$

CLAIM 6.5. *Let G be a (c, d)-regular bipartite graph with n left vertices. If G is a (α, ε)-left expander, then G is a $(2\alpha - c, \varepsilon)$-left unique neighbor expander.*

Set $\alpha = \frac{c+1}{2}$. Then $\frac{c}{2} < \alpha < c - 1$ for $c \geq 7$. Let G be a random (c, d)-regular bipartite graph. By Claim 6.4, with high probability G is an (α, ε)-right expander for any ε satisfying (6.1).

The following inequalities hold for our selection of α and any $c \geq 7$:

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

$$\frac{\alpha}{c} > \frac{1}{2},$$

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

Hence, $\varepsilon = \frac{1}{100} \cdot d^{-2}$ satisfies (6.1). Claim 6.5 completes the proof of Lemma 6.3. □

Proof of Claim 6.4. Let BAD be the event in which 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 in which all edges leaving S land inside T. We upper-bound the probability of this bad event:

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

The inequality above holds as long as $\alpha d s < cn$. In the following, 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 $\kappa = e^{1+\alpha} \cdot \left(\frac{\alpha d}{c}\right)^{c-\alpha}$.

$$\begin{aligned}
\Pr[BAD] &\leq \sum_{s=1}^{\varepsilon n} \binom{n}{s} \cdot \binom{m}{\alpha s} \cdot \Pr[B_s] \\
&\leq \sum_{s=1}^{\varepsilon n} \left(\frac{en}{s}\right)^{s} \cdot \left(\frac{em}{\alpha s}\right)^{\alpha s} \cdot \left(\frac{\alpha d s}{cn}\right)^{cs} \\
&= \sum_{s=1}^{\varepsilon n} \left[e^{1+\alpha} \cdot \left(\frac{\alpha d}{c}\right)^{c-\alpha} \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s} \\
&= \sum_{s=1}^{\varepsilon n} \left[\kappa \cdot \left(\frac{s}{n}\right)^{c-\alpha-1} \right]^{s}. \qquad (6.2)
\end{aligned}$$

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

$$\varepsilon \leq (2\kappa)^{\frac{-1}{(c-\alpha-1)}} = \left(2e^{(1+\alpha)} \cdot \left(\frac{\alpha d}{c}\right)^{(c-\alpha)}\right)^{-\frac{1}{c-\alpha-1}}. \qquad (6.3)$$

For this value of ε, each term of the sum (6.2) is at most $\frac{1}{2}$.
Set $\lambda = \min\{\frac{1}{3}, \frac{c-\alpha-1}{2}\}$ and split the sum (6.2) into two subsums:

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

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

Proof of Claim 6.5. Let $S \subset L, |S| \leq \varepsilon |L|$. Then by expansion,

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

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

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

Combining the two equations, we get our claim. □

LEMMA 6.6. *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.*

Proof. In the proof, we make use of the following theorem (see [MR95]).

THEOREM 6.7 (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:

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

We conclude by linearity of expectation:

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

We now use 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 $d > \frac{2\mu c^2}{(\mu^c-\delta)^2}$, Azuma's inequality (Theorem 6.7) gives us

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

where $\varepsilon = \frac{d(\mu^c-\delta)^2}{2\mu c^2} - 1 > 0$. Since there are at most 2^m possible sets $T \subseteq R$, by the union bound,

$$\begin{aligned} \Pr\left[\exists T \subset R\ |T| \geq \mu m, \left|\sum_{j\in T} A_j\right| \leq \delta n\right] &\leq 2^m \cdot 2e^{-(1+\varepsilon)m} \\ &= o(1). \end{aligned}$$

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

6.2. Proof of Lemma 3.6. Let G be a random (c,d)-regular graph G with n left vertices. We prove that $\mathcal{A}(G)$ is with high probability (i) linearly independent, (ii) $(\delta n, \mu)$-local, and (iii) ε-separating.

(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 6.3. We conclude that, for any $d \geq 7$ and for our setting of μ, G is with high probability a $(1,\mu)$-right unique neighbor expander. This implies that if $|T| \leq \mu|R|$, then $N^{odd}(T) \neq \emptyset$ because $N^{odd}(T) \supseteq N^1(T)$ and $N^1(T) \neq \emptyset$.

Lemma 6.6 says that for any $\mu > 0$, and for our selection of d, all sets of size at least μm have a nonempty odd neighborhood. (Actually, the lemma shows that the odd neighborhood is of linear size, which is more than we need here.) This completes the proof of the first claim.

(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 definition of d, μ, δ and by Lemma 6.6, with high probability G is 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 nonzero element of Hamming weight $< \varepsilon n$.

Let x be a nonzero 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,

$$N^1(S_x) \subseteq \{j\}, \tag{6.4}$$

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 $|S_x| \leq \frac{1}{100} \cdot d^{-2} \cdot n$, then by Lemma 6.3 $|N^1(S_x)| \geq |S_x| > 1$, in contradiction to (6.4). We conclude that for any $x \in \mathcal{C}(G_{-j})$, $|x| \geq \frac{1}{100} \cdot d^{-2}$, so $\mathcal{A}(G)$ is ε-separating for our selection of ε. Part (iii) is complete.

This completes the proof of Lemma 3.6. □

7. Reducing *d*LIN to 3LIN. This section proves Lemma 3.8. 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 A_i, say $x_1 \oplus \cdots \oplus x_d = 0$, in φ, two constraints, A_i^1 and A_i^2, are formed: $x_1 \oplus \cdots \oplus x_{\lceil \frac{d}{2} \rceil} \oplus z_i = 0$ and $x_{\lceil \frac{d}{2} \rceil + 1} \oplus \cdots \oplus x_d \oplus z_i = 0$. 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 claims show that the reduction preserves the properties which make the formula hard to test. A parameter followed by a prime denotes the value of the parameter after one application of the reduction: for example, $m' = 2m$, $n' = m + n$, and $d' = \lceil \frac{d}{2} \rceil + 1$.

CLAIM 7.1. *$\mathcal{R}(\mathcal{A})$ is independent.*

Proof. It is enough to prove that no set of constraints in $\mathcal{R}(\mathcal{A})$ sums up to 0. Let C be a subset of constraints 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. □

CLAIM 7.2. *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 A_i^1, in $\mathcal{R}(\mathcal{A})$. Namely, $\langle x', A_i^1 \rangle = 1$ and $\langle x', A' \rangle = 0$ for all $A' \in \mathcal{R}(\mathcal{A}), A' \neq A_i^1$. Let $x = x'_1 \dots x'_n$. Then $\langle x, A_i \rangle = \langle x', A_i^1 + A_i^2 \rangle = \langle x', A_i^1 \rangle + \langle x', A_i^2 \rangle = 1$, and similarly, $\langle x, A \rangle = 0$ for all $A \in \mathcal{A}, A \neq A_i$. 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. □

CLAIM 7.3. *If $\mathcal{A}$ is (q, μ)-local, then $\mathcal{R}(\mathcal{A})$ is (q', μ')-local, where $q' = \frac{q}{d'}$ and $\mu' = \mu + \frac{q'}{m'}$.*

Proof. Let $\alpha' \in \{0,1\}^{m+n}$ be the sum of a subset T of $\mu' \cdot m'$ constraints in $\mathcal{R}(\mathcal{A})$. Let T_2 be the subset of constraints in T which appear in pairs. Namely, for every new variable z, both constraints 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 $|T_2| = |T| - |T_1| \geq \mu' \cdot m' - q' = \mu \cdot m' = 2\mu m$. 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,

$$\left| \sum_{A' \in T_2} A' \right| \geq \left| \sum_{A \in S} A \right| \geq q.$$

The last inequality follows from the fact that $\mathcal{A}$ is (q, μ)-local. When constraints from T_1 are added to $\sum_{A' \in T_2} A'$, each T_1 constraint zeros out at most $\lceil \frac{d}{2} \rceil = d' - 1$ coordinates:

$$|\alpha'| \geq \left| \sum_{A' \in T_2} A' \right| - \frac{d}{2} \left| \sum_{A' \in T_1} A' \right| \geq q - (d' - 1)q' = q'. \qquad \square$$

If the reduction is applied $\lceil \log(d-2) \rceil$ times, the number of terms in a constraint drops to 3. To see this, think of applying the reduction i times to a formula with $d \leq 2^i + 2$ terms per constraint. Successive iterations will decrease the clause size to $\leq 2^{i-1} + 2$, $\leq 2^{i-2} + 2$, etc. We apply the reduction $\lceil \log d \rceil$ times to obtain hard 3LIN formulae from hard dLIN formulae, as shown in Lemma 3.8.

Proof of Lemma 3.8. For $\mathcal{A} \subset \{0,1\}^n$ as in the statement of our lemma, let $\mathcal{R}^{(0)}(\mathcal{A}) = \mathcal{A}$, and for $i \geq 1$ let $\mathcal{R}^{(i)}(\mathcal{A}) = \mathcal{R}(\mathcal{R}^{(i-1)}(\mathcal{A}))$. Let $\mathcal{A}^\star = \mathcal{R}^{(\lceil \log d \rceil)}(\mathcal{A})$. As explained above, each constraint in $\mathcal{A}^\star$ has weight at most 3. We now calculate the remaining parameters of $\mathcal{A}^\star$. In doing so, we denote the value of a parameter in

$\mathcal{R}^{(i)}(\mathcal{A})$ by the superscript $^{(i)}$, and the superscript * signifies the final value of the parameter in $\mathcal{A}^\star$.

Since each application of the reduction doubles the dimension, $m^\star = 2^{\lceil \log d \rceil} m \le 2dm$. To calculate $n^\star$, observe that the reduction does not change $m - n$ and recall that $dm = cn$. Therefore,

$$n^\star = n + m^\star - m \le n + 2dm = (2c+1) \cdot n.$$

Claim 7.1 guarantees that $\mathcal{A}^\star$ is independent. By Claim 7.2, $\varepsilon' = \frac{\varepsilon}{1+(m/n)} = \varepsilon \frac{n}{n'}$. Thus,

$$\varepsilon \ge \varepsilon^\star = \varepsilon \cdot \frac{n}{n^{(1)}} \cdot \frac{n^{(1)}}{n^{(2)}} \cdots \frac{n^{(\lceil \log d \rceil - 1)}}{n^\star} = \varepsilon \frac{n}{n^\star} \ge \frac{\varepsilon}{2c+1}.$$

Let $q = \delta n$. Applying Claim 7.3 $\lceil \log d \rceil$ times, we obtain

$$\begin{aligned}
q \ge q^\star &= \frac{q}{d^{(1)} \times d^{(2)} \times \cdots \times d^\star} \ge \frac{q}{d^{\lceil \log d \rceil}} \ge \frac{q}{d^{\log d + 1}}; \\
\delta \ge \delta^\star &= \frac{q^\star}{n^\star} \ge \frac{q}{d^{\log d+1} \cdot (2c+1)n} = \frac{\delta}{d^{\log d+1} \cdot (2c+1)}; \\
\mu^\star &= \mu + \frac{q^{(1)}}{m^{(1)}} + \frac{q^{(2)}}{m^{(2)}} + \cdots + \frac{q^\star}{m^\star} \\
&< \mu + \frac{q}{m} \left(\frac{1}{2d^{(1)}} + \frac{1}{4d^{(1)}d^{(2)}} + \cdots + \frac{1}{d \cdot d^{(1)} d^{(2)} \cdots d^\star} \right) \\
&\le \mu + \frac{q}{m} \frac{\lceil \log d \rceil}{d} \le \mu + \frac{d\delta n}{cn} \cdot \frac{\log d + 1}{d} = \mu + \frac{\delta(\log d + 1)}{c}.
\end{aligned}$$

This completes the proof of Lemma 3.8. □

Acknowledgments. 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 thank Piotr Indyk for referring us to Azuma's inequality to simplify the analysis and thank Michael Sipser for helpful discussions. We are grateful to Oded Goldreich for allowing us to include the "hard to test" properties based on random linear codes. We thank Oded Goldreich and the anonymous referees for several useful suggestions which improved the presentation of the paper.

REFERENCES

[AFKS00] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy, *Efficient testing of large graphs*, Combinatorica, 20 (2000), pp. 451–476. (A preliminary version appears in Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 1999, pp. 656–666.)

[AKNS01] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy, *Regular languages are testable with a constant number of queries*, SIAM J. Comput., 30 (2001), pp. 1842–1862. (A preliminary version appears in Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 1999, pp. 645–655.)

[AS03] N. Alon and A. Shapira, *Testing satisfiability*, J. Algorithms, 47 (2003), pp. 87–103. (A preliminary version appears in Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, ACM, New York, 2002, pp. 645–654).

[BGH+04] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan, *Robust PCPs of proximity, shorter PCPs and applications to coding*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, June 13–15, 2004, pp. 1–10.

[BHR03] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova, *Some* 3CNF *properties are hard to test*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego. CA, June 9–11, 2003, pp. 345–354.

[BSVW03] E. Ben-Sasson, M. Sudan, S. Vadhan, and A. Wigderson, *Randomness-efficient low degree tests and short PCPs via epsilon-biased sets*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego. CA, June 9–11, 2003, pp. 612–621.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld, *Self-testing/correcting with applications to numerical problems*, J. Comput. System Sci., 47 (1993), pp. 549–595. (A preliminary version appears in Proceedings of the 22nd Annual Symposium on Theory of Computing (STOC), ACM, New York, 1990, pp. 73–83.)

[BOT02] A. Bogdanov, K. Obata, and L. Trevisan, *A lower bound for testing* 3*-colorability in bounded-degree graphs*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science, Vancouver, Canada, Nov. 16–19, 2002, pp. 93–102.

[CS88] V. Chvátal and E. Szemerédi, *Many hard examples for resolution*, J. ACM, 35 (1988), pp. 759–768.

[Fis01] E. Fischer, *The art of uninformed decisions: A primer to property testing*, Bull. European Assoc. Theoret. Comput. Sci., 75 (2001), pp. 97–126.

[Fis05] E. Fischer, *Testing graphs for colorability properties*, Random Structures Algorithms, 26 (2005), pp. 289–309. (A preliminary version appears in Proceeding of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, ACM, New York, 2001, pp. 873–882.)

[FLN+02] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky, *Monotonicity testing over general poset domains*, in Proceedings of the 34th Annual ACM Symposium on Theory of Computing, New York, May 19–21, 2002, pp. 474–483.

[FN04] E. Fischer and I. Newman, *Functions that have read-twice constant width branching programs are not necessarily testable*, Random Structures Algorithms, 24 (2004), pp. 175–193. (A preliminary version appears in Proceedings of the 17th Annual Conference on Computational Complexity, IEEE, Los Alamitos, CA, 2002, pp. 55–61.)

[Gal63] R. G. Gallager, *Low Density Parity Check Codes*, MIT Press, Cambridge, MA, 1963.

[GGR98] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750. (A preliminary version appears in Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 1996, pp. 339–348.)

[GR02] O. Goldreich and D. Ron, *Property testing in bounded degree graphs*, Algorithmica, 32 (2002), pp. 302–343. (A preliminary version appears in Proceedings of the 29th Annual Symposium on Theory of Computing (STOC), ACM, New York, 1997, pp. 406–415.)

[GS02] O. Goldreich and M. Sudan, *Locally testable codes and PCPs of almost linear length*, in Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computers Science, Vancouver, Canada, Nov. 16–19, 2002, pp. 13–22.

[GT03] O. Goldreich and L. Trevisan, *Three theorems regarding testing graph properties*, Random Structures Algorithms, 23 (2003), pp. 23–57. (A preliminary version appears in Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 2001, pp. 460–469.)

[MR95] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.

[New02] I. Newman, *Testing membership in languages that have small width branching programs*, SIAM J. Comput., 31 (2002), pp. 1557–1570. (A preliminary version appears in Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS), IEEE, Los Alamitos, CA, 2000, pp. 251–258.)

[Ron01] D. Ron, *Property testing (a tutorial)*, in Handbook of Randomized Computing, Comb. Optim. 9, S. Rajasekaran, P. M. Pardalos, J. H. Reif, and J. D. P. Rolim, eds., Kluwer Academic Publishers, Dordrecht 2001, pp. 597–649.

[RS96] R. Rubinfeld and M. Sudan, *Robust characterizations of polynomials with applications to program testing*, SIAM J. Comput., 25 (1996), pp. 252–271. (Preliminary versions appear in Proceedings of the 23rd Symposium on Theory of Computing (STOC), ACM, New York, 1991, pp. 33–42 and Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, ACM, New York, 1992, pp. 23–32.)

[Spi95] Daniel A. Spielman, *Computationally Efficient Error-Correcting Codes and Holographic Proofs*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.