Lower Bounds for Testing Triangle-freeness in Boolean Functions*

Arnab Bhattacharyya† Ning Xie‡

Abstract

Given a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$, we say a triple $(x, y, x+y)$ is a *triangle* in f if $f(x) = f(y) = f(x+y) = 1$. A *triangle-free* function contains no triangle. If f differs from every triangle-free function on at least $\epsilon \cdot 2^n$ points, then f is said to be ϵ-far from triangle-free. In this work, we analyze the query complexity of testers that, with constant probability, distinguish triangle-free functions from those ϵ-far from triangle-free.

Let the *canonical tester* for triangle-freeness denote the algorithm that repeatedly picks x and y uniformly and independently at random from $\mathbb{F}_2^n$, queries $f(x), f(y)$ and $f(x+y)$, and checks whether $f(x) = f(y) = f(x+y) = 1$. Green showed that the canonical tester rejects functions ϵ-far from triangle-free with constant probability if its query complexity is a tower of 2's whose height is polynomial in $1/\epsilon$. Fox later improved the height of the tower in Green's upper bound to $O(\log 1/\epsilon)$. A trivial lower bound of $\Omega(1/\epsilon)$ on the query complexity is immediate. In this paper, we give the first non-trivial lower bound for the number of queries needed. We show that, for every small enough ϵ, there exists an integer $n_0(\epsilon)$ such that for all $n \geq n_0$ there exists a function $f : \mathbb{F}_2^n \to \{0,1\}$ depending on all n variables which is ϵ-far from being triangle-free and requires $\Omega\left((1/\epsilon)^{4.847\cdots}\right)$ queries for the canonical tester. We also show that the query complexity of any general (possibly adaptive) one-sided tester for triangle-freeness is at least square-root of the query complexity of the corresponding canonical tester. Consequently, this means that any one-sided tester for triangle-freeness must make at least $\Omega\left((1/\epsilon)^{2.423\cdots}\right)$ queries.

1 Introduction

Roughly speaking, property testing is concerned with the existence of an efficient algorithm which queries an input object a small number of times and decides correctly with high probability whether the object has a given property or it is "far away" from having the property.

Formally, let D be a finite domain and R be a finite range. Letting $\{D \to R\}$ denote the set of all functions from D to R, a *property* is specified by a family $\mathcal{F} \subseteq \{D \to R\}$ of functions. A *tester* is a randomized algorithm which is given a distance parameter ϵ and has oracle access to an input function $f : D \to R$. It accepts with probability at least $2/3$ if $f \in \mathcal{F}$ and rejects with probability at least $2/3$ if the function is ϵ-*far* from $\mathcal{F}$. Distance between functions $f, g : D \to R$, denoted $\mathrm{dist}(f, g)$, is simply the

*A preliminary version of this article appeared in *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.

†Indian Institute of Science. Email: arnabb@csa.iisc.ernet.in. Most of the work was done when the author was at CSAIL, MIT and supported by a DOE Computational Science Graduate Fellowship and NSF Awards 0514771, 0728645 and 0732334.

‡SCIS, Florida International University, Miami, FL 33199, USA. Email: nxie@cis.fiu.edu. Most of the work was done when the author was at CSAIL, MIT and supported by NSF Awards 0514771, 0728645 and 0732334. Part of the work was done while visiting ITCS, Tsinghua University and supported by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

fraction of the domain where f and g disagree, and $\mathrm{dist}(f, \mathcal{F}) = \min_{g \in \mathcal{F}}\{\mathrm{dist}(f, g)\}$. For $\epsilon \in (0, 1)$, we say f is ϵ-far from $\mathcal{F}$ if $\mathrm{dist}(f, \mathcal{F}) \geq \epsilon$ and ϵ-close otherwise. A tester is *one-sided* if whenever $f \in \mathcal{F}$, the tester accepts with probability 1. The central parameter associated with a tester is its *query complexity*, the number of oracle queries it makes to the function f being tested. In particular, a property is called *strongly testable* if, for every fixed ϵ, there is a tester with query complexity that depends only on the distance parameter ϵ and is independent of the size of the domain. Property testing was formally defined by Rubinfeld and Sudan [35], and the systematic exploration of property testing for combinatorial properties was initiated by Goldreich, Goldwasser, and Ron [21]. Subsequently, a rich collection of properties have been shown to be strongly testable [8, 7, 3, 14, 33, 5, 4, 28, 27].

A central quest of research in property testing has been to characterize properties according to their query complexity. One can ask, for example, whether a large class of properties are all strongly testable, and how the query complexity of a strongly testable property depends on the distance parameter ϵ. Such broad understanding of testability has been achieved for graph and hypergraph properties. For graph properties, it is known exactly ([3, 14]) which properties are strongly testable in the dense graph model. Furthermore, for an important class of properties, H-freeness for fixed subgraphs H, it is known exactly for which H, testing H-freeness requires the query complexity to be super-polynomial in $1/\epsilon$ (ϵ being the distance parameter) and for which only a polynomial number of queries suffice: This was proved by Alon [1] for one-sided testers and by Alon and Shapira [6] for general (two-sided) testers. Progress toward similar understanding has also been made for hypergraph properties [34, 9, 7].

Somewhat ironically, algebraic properties, the main objects of study in the seminal work of Rubinfeld and Sudan [35], are not as well understood as (hyper)graph properties from a high-level perspective. On the one hand, there has been a lot of work in constructing low-query testers for specific algebraic properties, such as linearity and membership in various error-correcting codes. However, the systematic study of the query complexity of algebraic properties began only recently with the work of Kaufman and Sudan [29]. Formally, the class of properties under consideration here are linear-invariant properties. In this setting[1], the domain $D = \mathbb{F}_2^n$ and range $R = \{0, 1\}$, where $\mathbb{F}_2$ is the finite field with two elements. A property $\mathcal{F}$ is said to be *linear-invariant* if for every $f \in \mathcal{F}$ and linear map $L : \mathbb{F}_2^n \to \mathbb{F}_2^n$, it holds that $f \circ L \in \mathcal{F}$. Roughly speaking, Kaufman and Sudan showed strong testability of any locally-characterized linear-invariant and *linear* property[2]. Moreover, the query complexity of all such properties is only $O(1/\epsilon)$. Nonlinear linear-invariant properties were studied formally in [12] where the authors isolated a particular class of nonlinear linear-invariant properties, $\mathcal{M}$-freeness for some fixed binary matroids $\mathcal{M}$, and showed an infinitely large set of strongly testable $\mathcal{M}$-freeness properties.[3] Subsequently, Shapira [36] and Král *et al* [31] independently showed that, in fact for any fixed binary matroid $\mathcal{M}$, $\mathcal{M}$-freeness is strongly testable, mirroring the analogous result of subgraph-freeness testing. However, unlike the case of graphs where it is known exactly which subgraph-freeness properties can be tested in time $\mathrm{poly}(1/\epsilon)$ and which cannot, there are no similar results known for matroid-freeness properties. Indeed, to the best of our knowledge, prior to our work, there were no non-trivial lower bounds known for the query complexity (in terms of ϵ) for any

[1] [29] considers linear invariance over general fields. In this paper, we restrict ourselves to $\mathbb{F}_2^n$ for simplicity.

[2] A property $\mathcal{F}$ is linear if for any f and g that are in $\mathcal{F}$ necessarily implies that $f + g$ is in $\mathcal{F}$.

[3] For the purpose defining properties studied in this paper, matroid is simply a synonym for a collection of binary vectors. Given a matroid $\mathcal{M}$ represented by vectors $(v_1, \ldots, v_k)$ with each $v_i \in \mathbb{F}_2^r$, the property of $\mathcal{M}$-freeness is the family of Boolean functions $\mathcal{F}_{\mathcal{M}} = \{f : \mathbb{F}_2^n \to \{0, 1\} \mid \forall \text{ linear map } L : \mathbb{F}_2^r \to \mathbb{F}_2^n, (f(L(v_1)), \ldots, f(L(v_k))) \neq 1^k\}$. Clearly properties such defined are linear-invariant properties. The matroid corresponds to triangle-freeness is $\mathcal{M} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_1 + \boldsymbol{e}_2)$. To see this, note that $\mathcal{M}$-freeness requires that for any (non-singular) linear map L defined by $L(\boldsymbol{e}_1) = x$ and $L(\boldsymbol{e}_2) = y$ (hence $L(\boldsymbol{e}_1 + \boldsymbol{e}_2) = x + y$), where x and y are two arbitrary (distinct) vectors in $\mathbb{F}_2^n$, it is the case that $(f(x), f(y), f(x + y)) \neq 1^3$. This is just the definition of triangle-freeness property.

natural linear-invariant algebraic property.

1.1 Our Results

We are interested in the property of triangle-freeness for Boolean functions. Let $f : \mathbb{F}_2^n \to \{0,1\}$ be a Boolean function. We say a triple $(x, y, x+y)$ is a *triangle* in f if $f(x) = f(y) = f(x+y) = 1$. The function f is said to be *triangle-free* if it contains no triangle. The *canonical tester* for triangle-freeness repeatedly picks x and y uniformly and independently at random and checks if $f(x) = f(y) = f(x+y) = 1$.

In this paper we give the first non-trivial query lower bounds for testing triangle-freeness in Boolean functions. In particular, we show that, for every small enough (but constant) ϵ there exists an integer $n_0(\epsilon)$ such that for all $n \geq n_0$ there exists a function $f : \mathbb{F}_2^n \to \{0,1\}$ depending on all the n variables which is ϵ far from being triangle-free and requires $\Omega\left((\frac{1}{\epsilon})^{4.847\cdots}\right)$ queries for the canonical tester (Theorem 3.15 in Section 3.4), and $\Omega\left((\frac{1}{\epsilon})^{2.423\cdots}\right)$ queries for any one-sided tester (Theorem 4.10 in Section 4). We discuss more about the background of our results below.

Green [23] showed that it suffices for the canonical tester to make only a constant number of queries, so that the property of triangle-freeness is strongly testable. Green's analysis is quite different from that of typical algebraic tests and is more reminiscent of the analysis for tests of graph properties. In particular, Green developed an algebraic regularity lemma for the Boolean cube (his result is much more general – in fact, it works for any abelian group). The query complexity upper bound proved by Green has a very bad dependency on ϵ: it is a tower of 2's whose height is polynomial in $1/\epsilon$. A more combinatorial way to state Green's result is that, for any function ϵ-far from being triangle-free, there are at least $\delta(\epsilon)2^{2n}$ triangles in the function, though this $\delta(\epsilon)$ is only proved to be super tiny. More recently, Fox [19] gave a new proof of the so-called "graph removal lemma". His proof does not use Szemerédi's regularity lemma and gives a better bound. Combining with Král', Serra and Vena's proof [30] (of Green's removal lemma using the directed graph removal lemma), Fox improved the height of the tower of 2's in Green's query complexity upper bound from polynomial in $1/\epsilon$ to $O(\log(1/\epsilon))$ (see [25] for a direct Fourier analytical proof of this latter bound). A trivial lower bound of $\Omega(1/\epsilon)$ is straightforward to show. But, to the best of our knowledge, there is no non-trivial lower bound for testing triangle-freeness in Boolean functions. This question was left open in [23].

It is interesting to compare the testability of algebraic triangle-freeness and graphic triangle-freeness. Using Szemerédi's regularity lemma, triangle-freeness in graphs is known to be testable with a tower-type query complexity upper bound. Alon [1] gave a super-polynomial query complexity lower bound and it is the strongest query lower bound for a natural strongly testable property known to date. However, the proof technique in [1] does not seem to directly apply to the algebraic setting due to the inherent additive structures of the Boolean cubes. More generally, it seems to us that proving lower bounds for the Boolean function case is more challenging than that of the graphic case.

Proving lower bounds for the canonical tester translates to a clearly defined algebraic question. This is because a canonical tester is a one-sided tester; consequently, if a function is ϵ-far from triangle-free and contains N_Δ triangles, then for the canonical tester to reject this function with constant probability, it must make $\Omega(\frac{2^{2n}}{N_\Delta})$ number of queries. Therefore, to prove lower bounds for the canonical tester, it suffices to construct Boolean functions that are ϵ-far from triangle-free but contain only a small number of triangles. On the other hand, our ultimate goal would be to understand the query complexity with respect to general testers, not just the canonical one. To this end, we show that if there is a one-sided, possibly adaptive tester for triangle-freeness with query complexity q, then one can transform that tester into a canonical one with query complexity at most $O(q^2)$. Combining with our results for canonical testers, this implies a query complexity lower bound of $\Omega\left((\frac{1}{\epsilon})^{2.423\cdots}\right)$ for testing triangle-freeness, with respect to one-sided testers. In

fact our result is a bit more general: we prove a polynomial relationship between the query complexity of the canonical tester and arbitrary one-sided testers, for any matroid-freeness property. This is analogous to a result in [2] for one-sided testers of subgraph-freeness in graphs[4]. Another related result is that of Ben-Sasson, Harsha and Raskhodnikova [11] who showed that there is no gap between the query complexities of adaptive testers and non-adaptive ones for testing linear properties.

1.2 Overview of Techniques

From a combinatorial point of view, proving a lower bound for the query complexity of the canonical tester for triangle-freeness amounts to constructing a Boolean function which is far from being triangle-free but contains only a small number of triangles. By an observation in [26][5], it suffices to construct a *function-triple* which is far from being triangle-free but contains a small number of triangles. A triangle in a function-triple $f_1, f_2, f_3 : \mathbb{F}_2^n \to \{0,1\}$ is a triple $(x, y, x+y) \in (\mathbb{F}_2^n)^3$ such that $f_1(x) = f_2(y) = f_3(x+y) = 1$. A triangle-free function-triple f_1, f_2, f_3 contains no triangles, and a function-triple f_1, f_2, f_3 is said to be ϵ-far from being triangle-free if for every triangle-free function-triple g_1, g_2, g_3, it is the case that:

$$\left(\mathop{\mathbf{Pr}}_{x}[f_1(x) \neq g_1(x)] \geq \epsilon\right) \vee \left(\mathop{\mathbf{Pr}}_{x}[f_2(x) \neq g_2(x)] \geq \epsilon\right) \vee \left(\mathop{\mathbf{Pr}}_{x}[f_3(x) \neq g_3(x)] \geq \epsilon\right).$$

The observation of [26] is that, if we define $f : \mathbb{F}_2^{n+2} \to \{0,1\}$ such that, for all $x \in \mathbb{F}_2^n$, $f(00,x) = 0$, $f(01,x) = f_1(x)$, $f(10,x) = f_2(x)$ and $f(11,x) = f_3(x)$, then there is a one-to-one correspondence between the triangles in f and the triangles in the function-triple (f_1, f_2, f_3). As the domain size blow-up is a only a constant, it follows that lower bounds for function-triples imply lower bounds for single functions.

Our lower bound for function-triples is based on constructing a *vertex-disjoint* function-triple, meaning that all the triangles in the triple are pairwise disjoint. The property of being vertex-disjoint makes it simple to calculate the function-triple's distance from triangle-freeness as well as counting the number of triangles within the function-triple. We start our construction of a vertex-disjoint function-triple from three sets, each of cardinality m, of k-bit binary vectors, $\{a_i\}_{i=1}^m$, $\{b_j\}_{j=1}^m$ and$\{c_\ell\}_{\ell=1}^m$, where k and m are fixed integers. We call such a collection of vectors (k,m)-PMF for reasons to be explained shortly. Next we define three sets, $\{A_I\}$, $\{B_J\}$ and$\{C_L\}$, of mk-bit vectors, each consisting of the vectors obtained by concatenating $\{a_i\}$, $\{b_j\}$ and$\{c_\ell\}$, respectively, in all possible orders. Finally we define our function-triple (f_A, f_B, f_C) to be the characteristic functions of the three sets $\{A_I\}$, $\{B_J\}$ and$\{C_L\}$. In order to make the triangles in this function-triple pairwise disjoint, we impose the constraint that $\{a_i\}$, $\{b_j\}$ and$\{c_\ell\}$ satisfy a certain 1-perfect-matching-free (1-PMF for short) property (see Section 3.2 for formal definition). To make this construction work for arbitrarily small ϵ, we concatenate with some $n' \geq 1$ copies of each $\{a_i\}$, $\{b_j\}$ and$\{c_\ell\}$ and require them to satisfy the n'-PMF property for any $n' \geq 1$. It turns out that $\{a_i\}$, $\{b_j\}$ and$\{c_\ell\}$ being PMF is equivalent to a (small) set of homogeneous Diophantine linear equations having no non-trivial solution, which in turn can be checked by linear programming. Numerical computation indicates the existence of PMF family of vectors for $k = 3, 4$, and 5. Our findings show that larger values of k give stronger lower bounds but unfortunately it was computationally infeasible to search for PMF families of vectors for $k \geq 6$. We conjecture that our approach may lead to super-polynomial query lower bounds for testing multi-function triangle-freeness.

[4] Goldreich and Trevisan in [22] prove a polynomial relationship between the query complexity of *two-sided* testers and canonical testers, for any graph property. For the purposes of this paper, our weaker result is sufficient.

[5] In the conference version of this article [13], we showed query lower bounds for single-function and function-triple cases separately, and the single-function lower bound is weaker than the function-triple one. Since, by the observation in [26], function-triple lower bound implies single-function lower bound, we omit the original (weaker) single-function lower bound of [13] in this version.

We remark that one may start with a function-triple obtained from 1-PMF and tensor itself multiple times to construct function-triples suitable for arbitrarily small ϵ's. However, since the parameters k and m are (small) finite numbers (in our case, the maximum values of k and m are 5 and 13 respectively), the query lower bound obtained this way would be $\Omega\left((\frac{1}{\epsilon})^{c_{m,k}}\right)$, where $c_{k,m}$ is some constant depending on k and m. Concatenating n' copies of $\{a_i\}$, $\{b_j\}$ and$\{c_\ell\}$ for larger values of n', on the other hand, allows one to obtain better bounds for smaller ϵ's, thus achieving the best asymptotic lower bound attainable employing functions constructed from (k, m)-PMF (see the proof of Theorem 3.4 for details). However, functions constructed this way are only in n_0 variables, where n_0 is a fixed constant depending on ϵ, k and m. In analogy to the blow-up operation on graphs [1], we tensor the function in n_0 variables with *bent functions* (see Section 2 for definition) in appropriate number of variables to construct functions on arbitrarily long bits that actually depend on all these bits.

Our result on canonical tester vs. general one-sided tester for triangle-freeness is an adaptation of the proof technique from [22] to the algebraic setting. The proof relies crucially on the facts that both the canonical and general testers are one-sided and the property of being triangle-free is invariant under non-singular linear transformations of the underlying domain $\mathbb{F}_2^n$. The latter is used to show that, under a random non-singular linear transformation, all linearly independent 2-tuples have essentially equal probability of witnessing a triangle. Therefore, in order to have guaranteed performance for *every* isomorphic copy of the input function, the best strategy for any one-sided tester (even an adaptive one) for triangle-freeness is to pick random points in the domain to query and check for triangles.

1.3 Subsequent work

In a recent work, Fu and Kleinberg [20] improved our query lower bound of general one-sided tester for triangle-freeness from $\Omega\left((1/\epsilon)^{2.423\cdots}\right)$ to $\Omega\left((1/\epsilon)^{6.619\cdots}\right)$. They observed a nice connection between PMFs and Uniquely Solvable Puzzles (USPs) (introduced in [16]) and then modified the (implicit) construction of USPs in [17] to get asymptotically better PMFs. Haviv and Xie [26] showed that the query complexity of testing triangle-freeness is super-polynomial if certain conjecture regarding sunflowers[6] is false.

1.4 Organization

After some necessary definitions in Section 2, the query complexity lower bound for canonically testing triangle-freeness is presented in Section 3. In Section 4, we study the relationship between the query complexities of the canonical tester and of a general one-sided tester for a broad class of algebraic properties. The proof of a well-known result on Diophantine linear system of equations may be found in the appendix.

2 Preliminaries

Let $n \geq 1$ be a natural number. We use $[n]$ to denote the set $\{1, \ldots, n\}$. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$. We view elements of $\mathbb{F}_2^n$ as n-bit strings, that is elements of $\{0,1\}^n$, alternatively. If x and y are two n-bit strings, then $x + y$ denotes bitwise addition (i.e. XOR) of x and y. We use (x, y) to denote the concatenation of two bit strings x and y.

Definition 2.1 (Tensor Product of Boolean Functions)**.** *Let $f_1 : \mathbb{F}_2^{n_1} \to \{0,1\}$ and $f_2 : \mathbb{F}_2^{n_2} \to \{0,1\}$ be two Boolean functions on n_1 and n_2 variables respectively. Then the tensor product of f_1 and f_2, denoted*

[6] A *sunflower* is a collection of sets such that the pairwise intersection of any two distinct member sets is equal to the mutual intersection of all member sets.

by $f_1 \otimes f_2$, is a Boolean function over $\mathbb{F}_2^{n_1+n_2}$ such that $f_1 \otimes f_2(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$ for all $x_1 \in \mathbb{F}_2^{n_1}$ and $x_2 \in \mathbb{F}_2^{n_2}$.

Note that if f_1 depends on all the n_1 variables and f_2 depends on all the n_2 variables, then $f_1 \otimes f_2$ depends on all the $n_1 + n_2$ input bits.

In order to define and study some properties of bent functions, first we recall the notion of Fourier transform.

Definition 2.2 (Fourier Transform)**.** *Let $f : \mathbb{F}_2^n \to \mathbb{R}$. The* Fourier transform $\widehat{f} : \mathbb{F}_2^n \to \mathbb{R}$ *of f is defined to be $\widehat{f}(\alpha) = \mathbf{E}_x[f(x)\chi_\alpha(x)]$, where $\chi_\alpha(x) = (-1)^{\sum_{i\in[n]} \alpha_i x_i}$. $\widehat{f}(\alpha)$ is called the* Fourier coefficient *of f at α, and the $\{\chi_\alpha\}_\alpha$ are called characters.*

For $\alpha, \beta \in \mathbb{F}_2^n$, the inner product between α and β: $\langle \chi_\alpha, \chi_\beta \rangle \stackrel{\text{def}}{=} \mathbf{E}_{x\in\mathbb{F}_2^n}[\chi_\alpha(x)\chi_\beta(x)]$ is 1 if $\alpha = \beta$ and 0 otherwise. Therefore the characters form an orthonormal basis for $\mathbb{F}_2^n$, and we thus have the Fourier inversion formula $f(x) = \sum_{\alpha\in\mathbb{F}_2^n} \widehat{f}(\alpha)\chi_\alpha(x)$ and Parseval's equality $\sum_{\alpha\in\mathbb{F}_2^n} \widehat{f}(\alpha)^2 = \mathbf{E}_x[f(x)^2]$. For two functions $f, g : \mathbb{F}_2^n \to \mathbb{R}$, we define their *convolution* as $(f * g)(x) \stackrel{\text{def}}{=} \frac{1}{2^n} \sum_{y\in\mathbb{F}_2^n} f(y)g(x - y)$. By the convolution theorem, $\widehat{f \cdot g} = \widehat{f} * \widehat{g}$ and $\widehat{f * g} = \widehat{f} \cdot \widehat{g}$.

Definition 2.3 (Bent Functions)**.** *Let $\phi : \mathbb{F}_2^n \to \{0, 1\}$ be a Boolean function and let $\psi(x) = (-1)^{\phi(x)}$. ϕ is called a* bent function *if the Fourier coefficients of ψ satisfy that $|\widehat{\psi}(\alpha)| = \frac{1}{2^{n/2}}$ for every $\alpha \in \mathbb{F}_2^n$.*

Bent functions have many applications in cryptographic constructions. For more properties of bent functions, we refer interested readers to [32]. It is well known that bent functions exist when the number of variables is even. For example, the inner-product function $\phi(x) = x_1x_2 + x_3x_4 + \cdots + x_{n-1}x_n$ is a bent function in n variables for every even n.

Let $f_1, f_2, f_3 : \mathbb{F}_2^n \to \{0, 1\}$ be a function-triple. We say (f_1, f_2, f_3) is *triangle-free* if there is no x and y such that $f_1(x) = f_2(y) = f_3(x + y)$. We use T-FREE to denote the set of triangle-free function-triples. We say f is triangle-free if (f, f, f) is. When there is no risk of confusion, we write T-FREE for the set of triangle-free (single) functions as well.

Let $f, g : \mathbb{F}_2^n \to \{0, 1\}$. The (relative) distance between f and g is defined to be the fraction of points at which they disagree: $\text{dist}(f, g) \stackrel{\text{def}}{=} \mathbf{Pr}_{x\in\mathbb{F}_2^n}[f(x) \neq g(x)]$. The distance between (f_1, f_2, f_3) and T-FREE is:

$$\text{dist}((f_1, f_2, f_3), \text{T-FREE}) \stackrel{\text{def}}{=} \min_{(g_1,g_2,g_3)\in\text{T-FREE}} \max_{i=1,2,3} \text{dist}(f_i, g_i).$$

Let f_1, f_2, f_3 be a Boolean function-triple. The "number of triangles passing through f_1 at x" is $D_{f_1}(x) \stackrel{\text{def}}{=} |\{y \in \mathbb{F}_2^n : f_1(x) = f_2(y) = f_3(x + y) = 1\}|$. We define the *triangle degree* of f_1 at x, denoted by $d_{f_1}(x)$, to be $d_{f_1}(x) \stackrel{\text{def}}{=} D_{f_1}(x)/2^n$. Note that if $f_1(x) = 0$ then $d_{f_1}(x) = 0$, however the converse may not be true. Triangle degrees of f_2 and f_3 are defined analogously. The triangle degree of a single Boolean function f at point x is defined in a similar way: $d_f(x) \stackrel{\text{def}}{=} \frac{1}{2^n}|\{y \in \mathbb{F}_2^n : f(x) = f(y) = f(x + y) = 1\}|$. When the function f is clear from context, we drop the subscript f and simply write the triangle degree as $d(x)$.

3 Lower Bound for the Canonical Tester

Intuitively, our hard instance for the canonical tester is constructed by packing as many pairwise disjoint triangles as possible into a Boolean function. The distance between such a function and T-FREE is immediate:

the number of triangles divided by 2^n. We can then deduce a lower bound for the query complexity of the canonical tester.

This section is organized as follows. First we present a theorem from [26] which offers us more flexibility in the construction by considering function-triples with pairwise disjoint functions instead of single functions directly. Next, we give a systematic scheme for generating such function-triples. We then describe how to efficiently do a computer search to find function-triples with the desired parameters. The computer search yields a hard instance for a fixed number of variables, which we then extend using a tensoring process to an arbitrary number of variables.

3.1 From Function-Triples to Functions

The following theorem [26] is due to the second author and was independently observed by Eli Ben-Sasson (also mentioned in [20]). For completeness we include a proof here.

Theorem 3.1. *For any $c > 0$ and any integer $n > 0$, suppose $f_1, f_2, f_3 : \mathbb{F}_2^n \to \{0,1\}$ is a function-triple such that (f_1, f_2, f_3) is ϵ-far from triangle-free and the canonical tester for triangle-freeness in function-triple needs to make $q = \Omega\left((\frac{1}{\epsilon})^c\right)$ queries to (f_1, f_2, f_3). Then there exists a function $f : \mathbb{F}_2^{n+2}$ such that f is at least $\epsilon' \geq \epsilon/4$-far from triangle-free and the canonical tester needs to make $q' = \Omega\left((\frac{1}{\epsilon'})^c\right)$ queries to f. In other words, strong canonical tester lower bounds for function-triples imply strong canonical tester lower bounds for single functions.*

Proof. Given the function-triple (f_1, f_2, f_3), define $f : \mathbb{F}_2^{n+2} \to \{0,1\}$ as follows. For all $u \in \mathbb{F}_2^2$ and $x \in \mathbb{F}_2^n$, let

$$f(u,x) = \begin{cases} 0, & \text{if } u = 00, \\ f_1(x), & \text{if } u = 01, \\ f_2(x), & \text{if } u = 10, \\ f_3(x), & \text{if } u = 11. \end{cases}$$

By our construction, there is no triangle of f across different cosets of the subspace defined by $u = 00$. Hence the correspondence between triangles in (f_1, f_2, f_3) and triangles in f is immediate. By the definition of distance to triangle-free for function-triples, $\mathrm{dist}(f, \text{T-FREE}) \geq \mathrm{dist}((f_1, f_2, f_3), \text{T-FREE})/4 \geq \epsilon/4$[7]. Let N_Δ be the number of triangles in (f_1, f_2, f_3) and f. Since the query complexity of the canonical tester for a function-triple (resp. function) is proportional to the inverse of the number of triangles in the input function-triple (resp. function), so

$$q' = \Theta(2^{2n+4}/N_\Delta) = \Theta(2^{2n}/N_\Delta) = \Theta(q) = \Omega\left((\frac{1}{\epsilon})^c\right) = \Omega\left((\frac{1}{\epsilon'})^c\right).$$

□

3.2 Perfect-Matching-Free Families of Vectors

We first introduce the notion of *perfect-matching free* families of vectors, and then show how to use them to build function-triples with only pairwise disjoint triangles.

[7] Here we abuse notation and use T-FREE to denote both the set of triangle-free functions and the set of triangle-free function-triples.

Definition 3.2 (Perfect-Matching-Free Families of Vectors)**.** *Let k and m be integers such that $0 < k < m < 2^k$. Let $\{a_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$ be two families of vectors, with $a_i, b_i \in \{0,1\}^k$ for every $1 \le i \le m$. Let $c_i = a_i + b_i$.*

1. *Let $\{A_I\}_I$ be the set of (mk)-bit vectors formed by concatenating the m vectors in $\{a_i\}$ in all possible orders (there are $m!$ such vectors), where $I = (i_1, i_2, \ldots, i_m)$ is a permutation of $[m]$. Similarly define $\{B_J\}_J$ and $\{C_L\}_L$ as the concatenations of vectors in $\{b_i\}$ and $\{c_i\}$ with $J = (j_1, j_2, \ldots, j_m)$ and $L = (\ell_1, \ell_2, \ldots, \ell_m)$, respectively. We say the set of vectors $\{a_i, b_i, c_i\}$ is a (k, m)* 1-perfect-matching-free *(abbreviated as 1-PMF) family of vectors if $A_I + B_J = C_L$ necessarily implies that $I = J = L$ (i.e., $i_s = j_s = \ell_s$ for every $1 \le s \le m$).*

2. *Let $n' \ge 1$ be an integer and now let $\{A_I\}_I$, $\{B_J\}_J$ and $\{C_L\}_L$ be the sets of $n'mk$-bit vectors by concatenating n' copies of $\{a_i\}$, $\{b_i\}$ and $\{c_i\}$, respectively, in all possible orders (two concatenations are regarded the same if they give rise to two identical strings in $\{0,1\}^{n'mk}$). We say the set of vectors $\{a_i, b_i, c_i\}$ is a (k, m) n'-*PMF *family of vectors if $A_I + B_J = C_L$ necessarily implies that $I = J = L$.*

3. *Finally we say $\{a_i, b_i, c_i\}$ is a (k, m)-*PMF *family of vectors if it is n'-PMF for all $n' \ge 1$.*

In other words, suppose we color all the $3m$ vectors in $\{a_i, b_i, c_i\}$ with m different colors so that a_i, b_i and c_i are assigned the same color. Suppose further we are given equal number of copies of $\{a_1, b_1, c_1; \ldots; a_m, b_m, c_m\}$ and we wish to arrange them in three aligned rows such that all the a_i's are in the first row, all the b_i's are in the second row and all the c_i's are in the third row. Then the only way of making every column summing to 0^k is to take the trivial arrangement in which every column is monochromatic.

3.2.1 Construction Based on PMF Families of Vectors

Let $\{a_i, b_i, c_i\}$ be a (k, m)-PMF family of vectors. Let n be an integer such that $mk | n$ and let $n' = \frac{n}{mk}$. let $\{A_I\}_I$, $\{B_J\}_J$ and $\{C_L\}_L$ be the sets of n-bit vectors by concatenating n' copies of $\{a_i\}$, $\{b_i\}$ and $\{c_i\}$ respectively. Note that $|\{A_I\}| = |\{B_J\}| = |\{C_L\}| = \frac{(n'm)!}{(n'!)^m}$. Now let $f_A, f_B, f_C : \mathbb{F}_2^n \to \{0,1\}$ be three Boolean functions which are the characteristic functions of sets $\{A_I\}_I$, $\{B_J\}_J$ and $\{C_L\}_L$ respectively. That is, $f_A(x) = 1$ iff $x \in \{A_I\}$, $f_B(x) = 1$ iff $x \in \{B_J\}$ and $f_C(x) = 1$ iff $x \in \{C_L\}$.

Proposition 3.3. *All the triangles in the function-triple (f_A, f_B, f_C) are pairwise disjoint.*

Proof. This follows directly from the definition that $\{a_i, b_i, c_i\}$ is a PMF family of vectors. □

Theorem 3.4. *If (k, m)-PMF family of vectors exists, then there exists $\epsilon_0 = \epsilon_0(k, m)$ such that for all $\epsilon < \epsilon_0$, there is a $n_0 = n_0(\epsilon)$ and functions $f_A, f_B, f_C : \mathbb{F}_2^{n_0} \to \{0,1\}$ such that (f_A, f_B, f_C) is ϵ-far from being triangle-free and testing triangle-freeness in (f_A, f_B, f_C) requires the canonical tester to query the functions $\Omega\left((\frac{1}{\epsilon})^{\alpha - o(1)}\right)$ times, where*[8] *$\alpha = \frac{2 - \frac{\log m}{k}}{1 - \frac{\log m}{k}}$.*

Proof. Given a small enough $\epsilon > 0$, let n' be the largest integer such that $\epsilon \le \frac{\frac{(n'm)!}{(n'!)^m}}{2^{n'mk}}$. Let f_A, f_B and f_C be the characteristic functions of $\{A_I\}_I$, $\{B_J\}_J$ and $\{C_L\}_L$ respectively defined above. Set $n_0 = n'mk$ and then f_A, f_B and f_C are Boolean functions on n_0 variables. Let N_Δ be the number of triangles in

[8] All logarithms in this paper are base 2.

(f_A, f_B, f_C). Then by Stirling's formula, for all small enough ϵ (therefore large enough n' since we assume that m and k are fixed constants),

$$N_\Delta = \frac{(n'm)!}{(n'!)^m} = \frac{\sqrt{2\pi mn'}(\frac{mn'}{e})^{mn'}(1+O(\frac{1}{n'}))}{\left(\sqrt{2\pi n'}(\frac{n'}{e})^{n'}(1+O(\frac{1}{n'}))\right)^m}$$
$$= \Theta\left(\frac{m^{mn'}}{n'^{\frac{m-1}{2}}}\right) = 2^{(m\log m)n' - \frac{m-1}{2}\log n' - o(1)} = 2^{(\beta - o(1))n_0},$$

where $\beta = \frac{\log m}{k}$.

By Proposition 3.3, all the triangles in (f_A, f_B, f_C) are pairwise disjoint, therefore modifying the function-triple at one point in the domain can remove at most one triangle. Hence $\mathrm{dist}((f_A, f_B, f_C), \text{T-FREE}) \geq \frac{N_\Delta}{2^{n_0}} \geq \epsilon$. Consequently, the query complexity of the canonical tester is at least $\Omega\left(\frac{2^{2n_0}}{N_\Delta}\right) = \Omega\left(2^{(2-\beta+o(1))n_0}\right) = \Omega\left((\frac{1}{\epsilon})^{\alpha - o(1)}\right)$. □

One can construct f_A, f_B, f_C to be Boolean functions on $\mathbb{F}_2^n$ for any $n \geq n_0$, by simply making the functions ignore the last $n - n_0$ bits and behave as defined above on the first n_0 bits. In Theorem 3.15, we give a construction by tensoring with bent functions so that the resulting functions depend on all n bits.

We conjecture the following to be true.

Conjecture 3.5. *There are infinitely many (k, m)-PMF families of vectors with $m \geq 2^{k(1-o_k(1))}$ as k (and hence m as well) tends to infinity.*

By Theorem 3.4 and Theorem 3.1, Conjecture 3.5 would imply a super-polynomial query lower bound for testing triangle-freeness in any Boolean function using the canonical tester. To be more specific, if there exists a (k, m)-PMF family of vectors with $m \geq 2^{k(1-o_k(1))}$, then query complexity is at least $\Omega\left((\frac{1}{\epsilon})^{\frac{1}{o_k(1)}}\right)$. Moreover, when composed with Theorem 4.4 it would also give a super-polynomial lower bound for *any* one-sided triangle-freeness tester.

3.3 Existence of PMF Families of Vectors

In this section we present an efficient algorithm which, given a family of vectors $\{a_i, b_i, c_i\}_{i=1}^m$, checks if it is PMF. Let $\{a_i, b_i, c_i\}_{i=1}^m$ be a family of vectors such that $a_i, b_i, c_i \in \mathbb{F}_2^k$ and $c_i = a_i + b_i$ for every $1 \leq i \leq m$. First we observe that if $\{a_i, b_i, c_i\}$ is PMF, then all the vectors in $\{a_i\}$ must be distinct. The same distinctness condition holds for vectors in $\{b_i\}$ and $\{c_i\}$. From now on, we assume these to be true. Next we define a set of "collision blocks".

Definition 3.6 (Collision Blocks)**.** *Let $\{a_i, b_i, c_i\}_{i=1}^m$ be a family of vectors satisfying the distinctness condition. We say (i, j, ℓ) is a* collision block *if $a_i + b_j = c_\ell$, and for simplicity will just call it a block. We denote the set of all blocks by $\mathcal{B}$. We will call a block* trivial *if $i = j = \ell$ and* non-trivial *otherwise.*

Since $\{a_i, b_i, c_i\}$ satisfies the distinctness condition, clearly $|\mathcal{B}| < m^2$. Let r be the number of non-trivial blocks, and let $\{\mathsf{bl}_1, \ldots, \mathsf{bl}_r\}$ be the set of non-trivial blocks. For a collision block bl_s, we use bl_s^a, bl_s^b and bl_s^c to denote the three indices of the colliding vectors. That is, if $\mathsf{bl}_s = (i, j, \ell)$ is a block, then $\mathsf{bl}_s^a = i$, $\mathsf{bl}_s^b = j$ and $\mathsf{bl}_s^c = \ell$.

Now suppose $\{a_i, b_i, c_i\}_{i=1}^m$ is not PMF. Then by the definition of PMF, there exists an integer n' such that $A_I, B_J, C_L \in \{0,1\}^{n'mk}$, $A_I + B_J = C_L$ and I, J, and L are not the same sequence of indices. We

consider the equation $A_I + B_J = C_L$ as a tiling of $3 \times (n'm)$ k-bit vectors: the first row consists of the $n'm$ vectors from $\{a_i\}$ with each a_i appearing exactly n' times and the ordering is consistent with that of A_I. Similarly we arrange the second row with vectors from $\{b_i\}$ according to B_J and the third row with vectors from $\{c_i\}$ according to C_L. Observe that when we look at the columns of the tiling, each column corresponds to a block in $\mathcal{B}$. Now we remove all the trivial blocks, then because I, J, and L are not identical sequences of indices, there are some non-trivial blocks left in the tiling. Since all the blocks removed are trivial blocks, the remaining tiling still has equal number of a_i, b_i and c_i for every $1 \le i \le m$. We denote these numbers by $y_1, \ldots, y_m$. Note that y_i's are non-negative integers and not all of them are zero. Let the number of blocks bl_i left in the tiling be x_i, $1 \le i \le r$. Again x_i's are non-negative integers and not all zero. Moreover, we have the following constraints when counting the number of a_i, b_i and c_i vectors, respectively, left in the tiling:

$$\begin{cases} \sum_{j \in [r]: \mathsf{bl}_j^a = i} x_j - y_i = 0 \\ \sum_{j \in [r]: \mathsf{bl}_j^b = i} x_j - y_i = 0 \\ \sum_{j \in [r]: \mathsf{bl}_j^c = i} x_j - y_i = 0 \end{cases} \qquad \text{(for every } 1 \le i \le m\text{)} \tag{1}$$

$x_j =$ number of type j blocks left after removing trivial blocks

$y_i =$ number of vectors a_i (equiv. b_i or c_i) left after removing trivial blocks

Lemma 3.7. *$\{a_i, b_i, c_i\}_{i=1}^m$ is not PMF if and only there is a non-zero integral solution to the system of linear equations (1).*

Proof. We only need to show that if there is a non-zero solution to (1), then $\{a_i, b_i, c_i\}_{i=1}^m$ is not PMF. Let $\{x_i, y_j\}$ be a set of non-zero integer solution. Note that the solution corresponds to a partial tiling with equal number of a_i, b_i and c_i for every $1 \le i \le m$. Set $n' = \max_i y_i$. Since the solution is non-trivial, $n' \ge 1$. Now for each $1 \le i \le m$, add $(n' - y_i)$ number of trivial blocks (i, i, i) to the tiling. Then the resulting tiling gives $A_I, B_J, C_L \in \{0,1\}^{n'mk}$ and $A_I + B_J = C_L$ such that I, J and L are not identical. □

Writing equations (1) in matrix form, we have

$$\mathbf{M}\vec{Z} = \vec{0},$$

where

$$\mathbf{M} = \begin{bmatrix} 1 & & \cdots & 1 & -1 & & \\ & 1 & \cdots & & & \ddots & \\ & & \cdots & & & & -1 \\ & & \cdots & 1 & -1 & & \\ & 1 & \cdots & & & \ddots & \\ 1 & & \cdots & & & & -1 \\ & 1 & \cdots & & -1 & & \\ & & \cdots & & & \ddots & \\ 1 & & \cdots & 1 & & & -1 \end{bmatrix}$$

is a $3m \times (r + m)$ integer-valued matrix (actually all entries are in the set $\{-1, 0, 1\}$) and

$$\vec{Z} = [x_1, \ldots, x_r, y_1, \ldots, y_m]^T$$

is an $(r+m) \times 1$ non-negative integer-valued column vector. Note that each of first r columns of $\mathbf{M}$ has exactly three 1s and all other entries are zero, and the last m columns of $\mathbf{M}$ consist of three $-I_{m\times m}$ matrices.

The following observation of Domenjoud [18], which essentially follows from Carathéodory's theorem, gives an exact characterization of when the system of equations (1) has a non-zero integral solution. We include a proof in Appendix A for completeness.

Theorem 3.8 ([18]). *Let $\mathbf{M}$ be an $s \times t$ integer matrix, then the Diophantine linear system of equations $\mathbf{M}\vec{Z} = \vec{0}$ with $\vec{Z} \in \mathbb{N}^t$ has a non-zero solution if and only if $\vec{0} \in$ Conv$(M_1, \ldots, M_t)$[9], where M_i's are the column vectors of $\mathbf{M}$ and Conv$(M_1, \ldots, M_t)$ denotes the convex hull of vectors $M_1, \ldots, M_t$.*

It is well known that checking point-inclusion in a convex hull can be solved by Linear Programming, see e.g. [10]. In particular, following the definition of convex hulls, $\vec{0} \in \text{Conv}(M_1, \ldots, M_t)$ if and only if there exist real numbers $\theta_1 \geq 0, \ldots, \theta_t \geq 0$ such that

$$\sum_{i=1}^{t} \theta_i M_i = \vec{0}$$

and

$$\sum_{i=1}^{t} \theta_i = 1.$$

After introducing additional slack variables and plugging in our collision matrix $\mathbf{M}$ into the formalism, we finally arrive at the following characterization of a family of vectors being PMF.

Lemma 3.9. *The family of vectors $\{a_i, b_i, c_i\}_{i=1}^m$ is PMF if and only if the following LP*

$$\begin{gathered} \textit{Maximize } W = \vec{c} \cdot \vec{\theta} \\ \textit{Subject to } \mathbf{M}'\vec{\theta} = \vec{b} \\ \vec{\theta} \geq \vec{0} \end{gathered}$$

has no feasible solution with $W \geq 0$.

Here

$$\mathbf{M}' = \left[\begin{array}{ccc|c} & \mathbf{M} & & \\ 1 & \cdots & 1 & \mathbf{I}_{(3m+1)} \end{array}\right]$$

is a $(3m+1) \times (4m+r+1)$ integer matrix with $\mathbf{M}$ being the collision matrix of the family of vectors $\{a_i, b_i, c_i\}_{i=1}^m$,

$$\vec{b} = [0, \ldots, 0, 1]^T$$

is a $3m+1$-dimensional integer vector and

$$\vec{c} = [\underbrace{0, \ldots, 0}_{r+m}, \underbrace{-1, \ldots, -1}_{3m+1}]^T$$

is the objective function vector of dimension $4m+r+1$.

Using this procedure for checking if a family of vectors $\{a_i, b_i, c_i\}_{i=1}^m$ is PMF or not, we find the following (k, m)-PMF families of vectors.

[9] Here and after $\mathbb{N} = \{0, 1, \ldots\}$ denotes the set of natural numbers.

Theorem 3.10. *There are* $(3,4)$*-PMF,* $(4,7)$*-PMF and* $(5,13)$*-PMF families of vectors.*

Proof. By numerical calculation, the following set of vectors is $(3,4)$-PMF:

$$\begin{aligned} a_1 &= 110 & b_1 &= 001 \\ a_2 &= 010 & b_2 &= 100 \\ a_3 &= 101 & b_3 &= 111 \\ a_4 &= 011 & b_4 &= 011. \end{aligned}$$

The following set of vectors is $(4,7)$-PMF:

$$\begin{aligned} a_1 &= 1101 & b_1 &= 0011 \\ a_2 &= 0001 & b_2 &= 1011 \\ a_3 &= 0010 & b_3 &= 0111 \\ a_4 &= 0110 & b_4 &= 1001 \\ a_5 &= 0000 & b_5 &= 0000 \\ a_6 &= 0111 & b_6 &= 0100 \\ a_7 &= 1001 & b_7 &= 0101. \end{aligned}$$

The following set of vectors is $(5,13)$-PMF:

$$\begin{aligned} a_1 &= 11101 & b_1 &= 01101 \\ a_2 &= 11001 & b_2 &= 11101 \\ a_3 &= 11000 & b_3 &= 10011 \\ a_4 &= 00101 & b_4 &= 10001 \\ a_5 &= 10010 & b_5 &= 00101 \\ a_6 &= 11110 & b_6 &= 10100 \\ a_7 &= 10000 & b_7 &= 10000 \\ a_8 &= 01000 & b_8 &= 01111 \\ a_9 &= 00011 & b_9 &= 01010 \\ a_{10} &= 11100 & b_{10} &= 00111 \\ a_{11} &= 00010 & b_{11} &= 11010 \\ a_{12} &= 01100 & b_{12} &= 10010 \\ a_{13} &= 01010 & b_{13} &= 11111. \end{aligned}$$

□

We were unable to check the cases $k \geq 6$ since they are too large to do numerical calculations. However, our best findings for $k = 3, 4, 5$ seem to suggest that the exponent α defined in Theorem 3.4 increases as k increases, which we view as a supporting evidence for Conjecture 3.5.

Now using the $(5,13)$-PMF family of vectors as the building block, Theorem 3.4 combined with Lemma 3.1 implies the following.

Theorem 3.11. *For all small enough ϵ, there is an $n_0 = n_0(\epsilon)$ and a Boolean functions $f : \mathbb{F}_2^{n_0} \to \{0,1\}$ such that f is ϵ-far from being triangle-free and testing triangle-freeness of f requires the canonical tester to query the function f $\Omega\left((\frac{1}{\epsilon})^{4.847\cdots}\right)$ times.*

3.4 Extending the Hard-to-test Functions

Theorem 3.11 asserts the existence of only one value n_0 such that there is a Boolean function f in n_0 variables for which the canonical tester requires $\Omega\left((\frac{1}{\epsilon})^{4.847\cdots}\right)$ queries. However, this hard instance is interesting only if it can be extended to infinitely many values of n. We can overcome this objection in a very direct way. For any $n \geq n_0(\epsilon)$, let $g_n : \mathbb{F}_2^n \to \{0,1\}$ be the function that equals f from Theorem 3.11 evaluated on the first $n_0(\epsilon)$ bits. A straightforward argument (that we omit) shows that the distance of g_n to T-FREE is at least ϵ, so that g_n also requires $\Omega\left((\frac{1}{\epsilon})^{4.847\cdots}\right)$ queries for the canonical tester to test.

However, this example is somewhat unsatisfactory as g_n depends only on n_0 of its n input variables. We construct below a stronger example that satisfies the lower bound of Theorem 3.11 and also depends on all its input variables. The idea behind this construction is to tensor f with an appropriate function such that the triangle-degree is not affected too much.

Let us introduce some notation to analyze the tensoring process. We define the *density* of f to be $\rho_f \stackrel{\text{def}}{=} \mathbf{Pr}_x[f(x) = 1]$. We say f is (ρ, d)-*regular* if $\rho_f = \rho$ and $d_f(x) = d$ for all x with $f(x) = 1$ (where d_f denotes the triangle-degree defined in Section 2). Observe that tensor product preserves the triangle-degree regularity of Boolean functions.

Lemma 3.12. *Let $f_1 : \{0,1\}^{n_1} \to \{0,1\}$ and $f_2 : \{0,1\}^{n_2} \to \{0,1\}$ such that f_1 is (ρ_1, d_1)-regular and f_2 is (ρ_2, d_2)-regular. Then $f_1 \otimes f_2$ is $(\rho_1 \cdot \rho_2, d_1 \cdot d_2)$-regular.*

Proof. The density of $f_1 \otimes f_2$ is straightforward from definition. For the degree part, notice that for any $x = (x_1, x_2)$ and $y = (y_1, y_2)$, where $x_1, y_1 \in \{0,1\}^{n_1}$ and $x_2, y_2 \in \{0,1\}^{n_2}$, $(x, y, x + y)$ is a triangle of $f_1 \otimes f_2$ if and only if both $(x_1, y_1, x_1 + y_1)$ is a triangle of f_1 and $(x_2, y_2, x_2 + y_2)$ is a triangle of f_2. □

The reason of studying (ρ, d)-regular functions is that it is extremely simple to analyze the query complexity of the canonical tester of triangle-freeness for such functions.

Lemma 3.13. *Let f be a (ρ, d)-regular function on n variables. Then there are $N_\Delta = \frac{\rho d 2^{2n}}{6}$ triangles of f and f is $\rho/3$-far from being triangle-free. Consequently, testing triangle-freeness requires the canonical tester to query f $\Omega(1/\rho d) = \Omega\left((\frac{1}{\epsilon})^{1+\frac{\log 1/d}{\log 1/\epsilon}}\right)$ times, where $\epsilon \geq \rho/3$ is the distance between f and* T-FREE.

Proof. Since f is (ρ, d)-regular, there are $\rho 2^n$ x's with $f(x) = 1$ and for every such x there are $d \cdot 2^n/2$ triangles passing through it (since every triangle is counted twice in the definition of triangle degree). It follows that there are in total $\frac{\rho d 2^{2n}}{6}$ triangles, as each triangle is counted once by each of its three vertices. Since triangle-freeness is a monotone property, one can only change the function values from 1 to 0 to possibly remove triangles. Now changing the function value at one point can remove at most $d \cdot 2^n/2$ triangles, so one needs to change the function value of f (from 1 to 0) on at least $\frac{\rho d 2^{2n}/6}{d \cdot 2^n/2} = \rho 2^n/3$ points in the domain. That is, $\mathrm{dist}(f, \text{T-FREE}) \geq \rho/3$. Finally combining the query lower bound of the canonical tester $q = \Omega\left(2^{2n}/N_\Delta\right)$ with the lower bound $\epsilon \geq \rho/3$ on f's distance from T-FREE gives the desired bound. □

In order to construct Boolean functions on arbitrarily large Boolean domains, we utilize bent functions to "stretch" the input bits. We show next that any bent function which evaluates to 0 at 0 is regular and

satisfies $\rho \approx 1/2$ and $d \approx 1/4$. Such bent functions on $\mathbb{F}_2^m$ are well-known to exist for every even number $m \geq 2$.

Lemma 3.14. *For every even number $m \geq 2$, if $\phi : \mathbb{F}_2^m \to \{0,1\}$ is a bent function with $\phi(0) = 0$, then ϕ is (ρ_m, d_m)-regular, where $\rho_m = \frac{1}{2} \pm O(2^{-m/2})$ and $d_m = \frac{1}{4} \pm O(2^{-m/2})$ are two constants depending only on m.*

Proof. Let $\phi(x)$ be a bent function on m variables and let $\psi(x) = (-1)^{\phi(x)}$. Note that $\psi(x) = 1 - 2\phi(x)$ and $\phi(x) = \frac{1}{2}(1 - \psi(x))$. Then by linearity of Fourier coefficients,

$$\widehat{\phi}(\alpha) = \begin{cases} \frac{1}{2} - \frac{1}{2}\widehat{\psi}(0), & \text{if } \alpha = 0, \\ -\frac{1}{2}\widehat{\psi}(\alpha), & \text{otherwise.} \end{cases}$$

It follows that $\rho_\phi = \widehat{\phi}(0) = \frac{1}{2} \pm \frac{1}{2\sqrt{2^m}}$.

Without loss of generality, we can assume that $\phi(0) = 0$. This is because, if otherwise, we can do a shift without changing the magnitudes of the Fourier coefficients of ψ. By definition of d_ϕ, for all x with $\phi(x) = 1$,

$$d_\phi(x) = \frac{1}{2^m} \sum_y \phi(y)\phi(x+y) = (\phi * \phi)(x),$$

therefore $\widehat{d_\phi}(\alpha) = \widehat{\phi}^2(\alpha)$.

Now we have, for all x such that $\phi(x) = 1$ (note that $x \neq 0$),

$$\begin{aligned} d_\phi(x) &= \sum_\alpha \widehat{d_\phi}(\alpha)\chi_\alpha(x) \\ &= \sum_\alpha \widehat{\phi}^2(\alpha)\chi_\alpha(x) \\ &= (\frac{1}{2} - \frac{1}{2}\widehat{\psi}(0))^2\chi_0(x) + \sum_{\alpha \neq 0} \frac{1}{4}\widehat{\psi}^2(\alpha)\chi_\alpha(x) \\ &= (\frac{1}{2} - \frac{1}{2}\widehat{\psi}(0))^2 + \frac{1}{2^{m+2}} \sum_{\alpha \neq 0} \chi_\alpha(x). \end{aligned}$$

Since $x \neq 0$, $\sum_\alpha \chi_\alpha(x) = 0$, so $\sum_{\alpha \neq 0} \chi_\alpha(x) = -\chi_0(x) = -1$. Plugging this into $d_\phi(x)$, we conclude that, for all x with $\phi(x) = 1$, $d_\phi(x) = \frac{1}{4} - \frac{1}{2}\widehat{\psi}(0) = \frac{1}{4} \pm \frac{1}{2\sqrt{2^m}}$. □

Tensoring regular bent functions on appropriate number of bits with the function constructed in Theorem 3.11 yields the following Theorem.

Theorem 3.15. *For all small enough ϵ there is an integer $n_0(\epsilon)$ such that the following holds. For every integer $n \geq n_0$, there is a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ such that f is ϵ-far from being triangle-free and testing triangle-freeness of f requires the canonical tester to query the function $\Omega\left((\frac{1}{\epsilon})^{4.847\cdots}\right)$ times. Moreover, f depends on all n input variables.*

Proof. Let H be the function on three variables: $H(000) = H(111) = 0$ and $H(x) = 1$ otherwise. By direct calculation, H is $(3/4, 1/2)$-regular and depends on all three input bits.

Note that the function f constructed in Theorem 3.11 is (ρ, d)-regular, where $\rho = \frac{N_\Delta}{2^{n'mk}} = \Theta(\epsilon)$ and $d = \frac{2}{2^{n'mk}}$ satisfying $\frac{1}{\rho d} = \Omega\left((\frac{1}{\rho})^{4.847\cdots}\right)$. Now we can tensor f with ϕ_{n-n_0} (or $\phi_{n-n_0-1} \otimes H$, depending

on the parity of n) to get $f_n : \mathbb{F}_2^n \to \{0,1\}$. The density and degree of f_n satisfies the condition that $\frac{1}{\rho d} = \Omega\left(\left(\frac{1}{\rho}\right)^{4.847\cdots}\right)$. Finally, applying Lemma 3.13 to f_n completes the proof of the theorem. □

4 Query Complexities of the Canonical Tester and General One-sided Testers

In this section, we prove a general result between the query complexities of an arbitrary one-sided tester and the canonical tester, for a large class of algebraic properties. A property in our class is specified[10] by k vectors $v_1, \ldots, v_k$ in the vector space $\mathbb{F}_2^r$. Following the notation in [12], we call this set of vectors a rank-r *matroid* $\mathcal{M}$. An alternative, equivalent notation based on solutions of systems of linear equations is adopted in [36].

Definition 4.1 ($\mathcal{M}^*$-free)**.** *Given a rank-r matroid $\mathcal{M} = (v_1, \ldots, v_k)$ with each $v_i \in \mathbb{F}_2^r$, a k-tuple of Boolean functions $f_1, \ldots, f_k : \mathbb{F}_2^n \to \{0,1\}$ is said to be* $\mathcal{M}^*$-free *if there is no full-rank linear transformation $L : \mathbb{F}_2^r \to \mathbb{F}_2^n$ such that $f_i(L(v_i)) = 1$ for every $i \in [k]$. Otherwise, if such an L exists, $f_1, \ldots, f_k$ is said to* contain $\mathcal{M}$ at L*, or equivalently, L is called a* violating linear transformation *of $\mathcal{M}$.*

Remark Let $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r)$ be a set of basis vectors in $\mathbb{F}_2^r$. Each linear map L in the above definition is then specified by r vectors $z_1, \ldots, z_r$ in $\mathbb{F}_2^n$ such that $L(\boldsymbol{e}_i) = z_i$ for every $1 \le i \le r$. The linear map L is *full rank* if $(z_1, \ldots, z_r)$ are linearly independent.

To see that this generalizes the triangle-freeness property, let $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ be the two unit vectors in $\mathbb{F}_2^2$ and consider the matroid $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_1 + \boldsymbol{e}_2)$. Then the three elements of the matroid will be mapped to all triples of the form $(x, y, x+y)$ by the set of full-rank linear transformations, where x and y are two distinct non-zero elements in $\mathbb{F}_2^n$. Also note that in this case, $r = 2$ and $k = 3$.

The property of being $\mathcal{M}^*$-free is not linear-invariant. The original notion of $\mathcal{M}$-freeness, as defined in [12], allows L in the above definition to be arbitrary linear transformations, not restricted to full-rank ones, and is hence truly linear-invariant. However, from a conceptual level, for a fixed matroid $\mathcal{M}$, the property of being $\mathcal{M}$-free and being $\mathcal{M}^*$-free are very similar. It is analogous to the distinction between a graph being free of H as a subgraph and being free of homomorphic images of H, for a fixed graph H.

In terms of testability, we have some evidence that the distinction is unimportant, although we are unable to prove a formal statement at this time. For the case when $\mathcal{M} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_1 + \boldsymbol{e}_2)$, we can show that a tester for triangle-freeness can be converted to one for triangle*-freeness. Consider a function-triple (f_1, f_2, f_3) that is promised to be either triangle*-free or ϵ-far from being triangle*-free, where the distance parameter ϵ is a constant. Define a new function-triple (f_1', f_2', f_3') by setting, for $i = 1, 2, 3$, $f_i'(0) = 0$ and $f_i'(x) = f_i(x)$ for all $x \neq 0$. Observe that if (f_1, f_2, f_3) is triangle*-free, then (f_1', f_2', f_3') is triangle-free because setting $f_i'(0) = 0$ removes all degenerate triangles. On the other hand, if (f_1, f_2, f_3) is ϵ-far from triangle*-free, then (f_1', f_2', f_3') is still $\epsilon' \ge \epsilon - 3/2^n$ far from triangle*-free and, hence, also from triangle-free. Since ϵ' approaches ϵ as n goes to infinity, assuming the continuity of the query complexity as a function of the distance parameter, the query complexity of triangle-freeness is therefore lower-bounded[11] by the query-complexity of triangle*-freeness.

For general binary matroids $\mathcal{M} = (v_1, \ldots, v_k)$ with each $v_i \in \mathbb{F}_2^r$, observe that if a function tuple is far from being $\mathcal{M}$-free, then almost all the linear maps where $\mathcal{M}$ is contained are full-rank. This is because the

[10] We assume that r is the minimal dimension of the vector space which preserves the linear dependencies between $v_1, \ldots, v_k$. That is, r is the rank of the matrix with $v_1, \ldots, v_k$ as its columns.

[11] The other direction is easy to show in general: for any binary matroid $\mathcal{M}$ and constant ϵ, an ϵ-tester for $\mathcal{M}^*$-freeness can be used to ϵ-test $\mathcal{M}$-freeness (again assuming continuity of the query complexity function).

main theorems of [36] and [31] show that if a function tuple is $\Omega(1)$-far from $\mathcal{M}$-free, then $\mathcal{M}$ is contained at $\Omega(2^{nr})$ many linear maps, while there are only $o(2^{nr})$ many linear maps $L : \mathbb{F}_2^r \to \mathbb{F}_2^n$ of rank less than r. Therefore, in fact, any $\mathcal{M}^*$-free function tuple is $o(1)$-close to $\mathcal{M}$-free. If there were a more query efficient one-sided tester for $\mathcal{M}$-freeness than for $\mathcal{M}^*$-freeness, it must be the case that the few linear maps with rank less than r where $\mathcal{M}$ is contained can somehow be discovered more efficiently than the full-rank maps. But on the other hand, we know of a large class of matroids $\mathcal{M}$ for which there exist functions that are far from $\mathcal{M}$-free but do not contain $\mathcal{M}$ at *any* non-full-rank linear map. More precisely, letting $C_k = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{k-1}, \boldsymbol{e}_1 + \cdots + \boldsymbol{e}_{k-1})$ be the graphic matroid of the k-cycle, Theorem 1.3 in [12] proves that for any odd $k \geq 5$, there exist functions which are far from C_k-free but contain C_k only at full-rank linear maps (by showing a separation between the classes C_k-free and C_{k-2}-free). So, for these reasons, it seems unlikely that the query complexities of testing $\mathcal{M}^*$-freeness properties are very different from those of testing $\mathcal{M}$-freeness properties. We conjecture that the query complexities of testing $\mathcal{M}$-freeness and $\mathcal{M}^*$-freeness properties are the same[12] and leave this as an open problem.

We first observe a simple fact about the behavior of any *one-sided* tester for $\mathcal{M}^*$-freeness.

Lemma 4.2. *Let $\mathcal{M}$ be a matroid of k vectors. Then any one-sided tester T for $\mathcal{M}^*$-freeness rejects if and only if it detects a violating full-rank linear transformation L of $\mathcal{M}$.*

Proof. Let $f_1, \ldots, f_k : \mathbb{F}_2^n \to \{0,1\}$ be the input k-tuple of Boolean functions. If T finds a violating full-rank linear transformation L, clearly it should reject. For the other direction, suppose that T rejects $(f_1, \ldots, f_k)$ without seeing any violating linear maps from the points it queried. Since $\mathcal{M}^*$-freeness is a monotone property, we can set all the points of the function-tuple that have not been queried by T to 0, thus making $(f_1, \ldots, f_k)$ $\mathcal{M}^*$-free. Therefore T errs on this function-tuple. But this contradicts our assumption that T is a one-sided tester for $\mathcal{M}^*$-freeness. □

Next, we define the canonical tester for $\mathcal{M}^*$-freeness, which naturally extends the previously described canonical tester for triangle-freeness.

Definition 4.3 (Canonical Tester)**.** *Let $\mathcal{M} = (v_1, \ldots, v_k)$, with each $v_i \in \mathbb{F}_2^r$, be a rank-r matroid of k vectors. A tester $\mathcal{T}$ for $\mathcal{M}^*$-freeness is* canonical *if $\mathcal{T}$ operates as follows. Given as input a distance parameter ϵ and oracle access to k-tuple of Boolean functions $f_1, \ldots, f_k : \mathbb{F}_2^n \to \{0,1\}$, the tester $\mathcal{T}$ repeats the following process independently $\ell(\epsilon)$ times: select uniformly at random a rank-r linear transformation $L : \mathbb{F}_2^r \to \mathbb{F}_2^n$ and check if f contains $\mathcal{M}$ at L. If so, $\mathcal{T}$ rejects and halts. If $\mathcal{T}$ does not reject after $\ell(\epsilon)$ iterations, then $\mathcal{T}$ accepts. The query complexity of the canonical tester is therefore at most $\ell(\epsilon) \cdot k$.*

Our main theorem in this section is the following.

Theorem 4.4. *For a given rank-r matroid $\mathcal{M} = (v_1, \ldots, v_k)$ with each $v_i \in \mathbb{F}_2^r$, suppose there is a one-sided tester for $\mathcal{M}^*$-freeness with query complexity $q(\mathcal{M}, \epsilon)$. Then the canonical tester for $\mathcal{M}^*$-freeness has query complexity at most $O(k \cdot q(\mathcal{M}, \epsilon)^r)$.*

Proof. Since the rank of $\mathcal{M}$ is r, without loss of generality, we assume that $v_1, \ldots, v_r$ are the r basis vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_r$. Thus, any linear transformation $L : \mathbb{F}_2^r \to \mathbb{F}_2^n$ is uniquely determined by $L(v_1), \ldots, L(v_r)$.

Suppose we have a one-sided, possibly adaptive, tester T for $\mathcal{M}$-freeness with query complexity $q(\mathcal{M}, \epsilon)$. We say T operates in *steps*, where at each step $i \in [q(\mathcal{M}, \epsilon)]$, T selects an element y_i from $\mathbb{F}_2^n$ (based on

[12] It seems possible that for any two given tests for $\mathcal{M}$-freeness and $\mathcal{M}^*$-freeness, there is a function that is ϵ-far from both properties but for which the two tests behave quite differently in terms of number of queries made when the function is presented as input. However, the query complexities in our conjecture are measured as (non-increasing) functions of the distance parameter ϵ, which are *worst-case* query complexities among all input functions that are ϵ-far from the corresponding properties.

a distribution that depends arbitrarily on internal coin tosses and oracle answers in previous steps) and then queries the oracle for the value of $f_j(y_i)$, for some $1 \leq j \leq k$.

We convert the tester T into another tester T' that operates as follows. Given oracle access to a function tuple $f_1, \ldots, f_k : \mathbb{F}_2^n \to \{0,1\}$, T' first selects, uniformly at random, a *non-singular* linear map $\Pi : \mathbb{F}_2^n \to \mathbb{F}_2^n$, and then invokes the tester T, providing it with $f_j(\Pi(y))$ whenever it queries for $f_j(y)$. For convenience the linear map may be generated on-the-fly in the following sense. Suppose in the first $i-1$ queries, T queries $(y_1, \ldots, y_{i-1})$ and T' queries $(x_1, \ldots, x_{i-1})$. Now if T chooses a new point y_i to query, tester T' picks a Π uniformly at random from all non-singular maps that are consistent with all the points queried previously, that is, maps satisfying $\Pi(y_1) = x_1, \ldots, \Pi(y_{i-1}) = x_{i-1}$, and feeds the query result at $\Pi(y_i)$ to the original tester T.

Claim 4.5. *T' is also a tester of $(f_1, \ldots, f_k)$ for $\mathcal{M}^*$-freeness with the same query complexity as T.*

Proof. If $(f_1, \ldots, f_k)$ is $\mathcal{M}^*$-free, then $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$ is also $\mathcal{M}^*$-free because $\mathcal{M}^*$-freeness is closed under composition with non-singular linear transformations. Therefore T accepts $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$ with probability 1 and so does T' to $(f_1, \ldots, f_k)$.

On the other hand, if $(f_1, \ldots, f_k)$ is ϵ-far from $\mathcal{M}^*$-free, then $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$ is also ϵ-far from $\mathcal{M}^*$-free since Π preserves the distance between $(f_1, \ldots, f_k)$ and $\mathcal{M}^*$-free functions. To see this, suppose $\mathrm{dist}((f_1 \circ \Pi, \ldots, f_k \circ \Pi), \mathcal{M}^*\text{-free}) = \epsilon'$ and moreover, some $(g_1, \ldots, g_k) \in \mathcal{M}^*$-free achieves this distance from $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$. Since Π is invertible and hence a permutation of the elements in $\mathbb{F}_2^n$, we have

$$\begin{aligned}
\mathrm{dist}((f_1, \ldots, f_k), \mathcal{M}^*\text{-free}) &\leq \mathrm{dist}((f_1, \ldots, f_k), (g_1 \circ \Pi^{-1}, \ldots, g_k \circ \Pi^{-1})) \\
&= \mathrm{dist}\left((f_1 \circ \Pi, \ldots, f_k \circ \Pi), (g_1, \ldots, g_k)\right) \\
&= \mathrm{dist}((f_1 \circ \Pi, \ldots, f_k \circ \Pi), \mathcal{M}^*\text{-free}),
\end{aligned}$$

because if $(g_1, \ldots, g_k)$ is in $\mathcal{M}^*$-free, so is $(g_1 \circ \Pi^{-1}, \ldots, g_k \circ \Pi^{-1})$. By the same argument, $\mathrm{dist}((f_1 \circ \Pi, \ldots, f_k \circ \Pi), \mathcal{M}^*\text{-free}) \leq \mathrm{dist}((f_1, \ldots, f_k), \mathcal{M}^*\text{-free})$ and consequently $\mathrm{dist}((f_1 \circ \Pi, \ldots, f_k \circ \Pi), \mathcal{M}^*\text{-free}) = \mathrm{dist}((f_1, \ldots, f_k), \mathcal{M}^*\text{-free})$. Finally we have, since $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$ is ϵ-far from $\mathcal{M}^*$-free, T rejects $(f_1 \circ \Pi, \ldots, f_k \circ \Pi)$ with probability at least $2/3$ and so does T' to $(f_1, \ldots, f_k)$. □

For convenience, let us fix the following notation. At a step $i \in [q(\mathcal{M}, \epsilon)]$, the element whose value is requested by T is denoted y_i, and the element of $\mathbb{F}_2^n$ queried by T' (and whose value is supplied to T) is denoted x_i. Both x_i and y_i are of course random variables, and also $x_i = \Pi(y_i)$. We now make the simple observation that at each step, no matter how cleverly T selects the y_i's, each x_i is either uniformly distributed outside or lies inside the span of elements selected at previous steps. More precisely:

Lemma 4.6. *Fix an integer $i \in [q(\mathcal{M}, \epsilon)]$. Let $y_1, \ldots, y_i$ be the elements in $\mathbb{F}_2^n$ requested by T in the first i stages, and elements $x_1, \ldots, x_{i-1}$ be the points queried by T' in the first $i-1$ steps. Then, x_i, the element queried by T' at the i^{th} step is either an element in $\mathrm{span}(x_1, \ldots, x_{i-1})$ or is uniformly distributed in $\mathbb{F}_2^n - \mathrm{span}(x_1, \ldots, x_{i-1})$.*

Proof. Recall that we may pick the random non-singular linear transformation in an on-the-fly fashion: after T queries y_i, Π is chosen uniformly among all non-singular linear transformations that satisfy $\Pi(y_1) = x_1, \ldots, \Pi(y_{i-1}) = x_{i-1}$. If $y_i \in \mathrm{span}(y_1, \ldots, y_{i-1})$, then clearly $x_i \in \mathrm{span}(x_1, \ldots, x_{i-1})$. Otherwise, Π maps y_i to a uniformly chosen element $x_i \in \mathbb{F}_2^n - \mathrm{span}(x_1, \ldots, x_{i-1})$. □

Due to Lemma 4.6, we may divide the queries of T into two types: *staying query* if the newly queried point is in the span of the previously queried points, and *expanding query* if the newly queried point is a

random point outside the span of previously queried points. Let the number of expanding queries of T' be t, $t \leq q(\mathcal{M}, \epsilon)$. Note that t is a random variable depending on T's queries. Let the subspace spanned by $(x_1, \ldots, x_{q(\mathcal{M},\epsilon)})$ be $V_{T'}$, then clearly $\dim(V_{T'}) = t$ and the expanding query points generate $V_{T'}$ (i.e., the set of expanding queries $(x_{i_1}, \ldots, x_{i_t})$ form a basis for $V_{T'}$). Therefore, as a corollary to Lemma 4.6, we have the following property of $V_{T'}$.

Corollary 4.7. *The subspace $V_{T'}$ spanned by the query points of tester T' is a random subspace of dimension t in $\mathbb{F}_2^n$.*

The next Lemma shows, when we look at any fixed linearly independent r-tuple inspected by T, the corresponding r-tuple queried by T' after a random non-singular transformation of the space $\mathbb{F}_2^n$, distributes uniformly over all linearly independent r-tuples.

Lemma 4.8. *Let $V_{T'}$ be a random subspace in $\mathbb{F}_2^n$ of dimension $t < n$ generated by picking uniformly at random a set of t linearly independent vectors $(b_1, \ldots, b_t)$[13] in $\mathbb{F}_2^n$ as basis. Let $x = (x_1, \ldots, x_r)$ be any* fixed *linearly independent r-tuple, $r \leq t$, given by a set of linear combinations of the basis vectors $(b_1, \ldots, b_t)$. Then x is uniformly distributed over all linearly independent r-tuples in $(\mathbb{F}_2^n)^r$.*

Proof. Let $\vec{b}$ and $\vec{x}$ be the column vectors representing $(b_1, \ldots, b_t)$ and $(x_1, \ldots, x_r)$, respectively. Let $A \in \mathbb{F}_2^{r \times t}$ be the matrix representation of the linear combinations of $(x_1, \ldots, x_r)$ in terms of $(b_1, \ldots, b_t)$, that is, $\vec{x} = A\vec{b}$. Since $(x_1, \ldots, x_r)$ are linearly independent, it follows that $\mathrm{rank}(A) = r$. Therefore we can append $t - r$ linearly independent rows to the bottom of A and form an invertible t-by-t matrix A'. Now we employ an SVD-like (singular-value decomposition) decomposition and write matrix A as $A = I_r \Sigma A'$, where $\Sigma \in \mathbb{F}_2^{r \times t}$ consists of r 1's in the diagonals and 0's otherwise. Since $(b_1, \ldots, b_t)$ is distributed uniformly over all t linearly independent vectors and A' is invertible, $\vec{b'} \stackrel{\text{def}}{=} A'\vec{b}$ is also a set of t random linearly independent vectors. Therefore, the first r vectors in $\vec{b'}$,

$$\vec{b'}_{[r]} \stackrel{\text{def}}{=} \Sigma A' \vec{b} = \Sigma \vec{b'} = \begin{bmatrix} b'_1 \\ \vdots \\ b'_r \end{bmatrix},$$

is a random linearly independent r-tuple. Finally because I_r is the identity map, $\vec{x} = I_r \vec{b'}_{[r]}$ is a random linearly independent r-tuple. □

By Lemma 4.2, T' rejects if and only if it detects a violating full-rank linear transformation. Notice that each full-rank linear transformation $L : \mathbb{F}_2^r \to \mathbb{F}_2^n$ corresponds to a linearly independent r-tuple $(z_1, \ldots, z_r) \in (\mathbb{F}_2^n)^r$, where the corresponding linear transformation is given by $L_{z_1,\ldots,z_r}(u_1, \ldots, u_r) = \sum_{i=1}^r u_i z_i$. Thus, T' rejects if and only if it finds a linearly independent r-tuple $(z_1, \ldots, z_r)$ such that the corresponding linear transformation is violating. Furthermore, because $v_1 = \boldsymbol{e}_1, \ldots, v_r = \boldsymbol{e}_r$, the elements $z_1, \ldots, z_r$ must lie in the set of queries made by T'. Let δ be the fraction of violating linearly independent r-tuples $z = (z_1, \ldots, z_r) \in (\mathbb{F}_2^n)^r$.

Lemma 4.9. *The probability that T' rejects $(f_1, \ldots, f_k)$ after $q(\mathcal{M}, \epsilon)$ queries is at most $\delta q(\mathcal{M}, \epsilon)^r$.*

Proof. Let $x_1, \ldots, x_{q(\mathcal{M},\epsilon)}$ be the queries made by T'. Note that these vectors are random variables. For each $S \subseteq [q(\mathcal{M}, \epsilon)]$ with $|S| = r$, let A_S be the event that all the vectors in the r-tuple $(x_i : i \in S)$ are

[13] One may think of the basis of $V_{T'}$ as the set of expanding query points $(x_{i_1}, \ldots, x_{i_t})$ of tester T'.

linearly independent *and* they form a violating tuple. By our definition of δ, for every S, $\mathbf{Pr}[A_S] \leq r! \cdot \delta$. Since the total number of such r-subsets is $\binom{q(\mathcal{M},\epsilon)}{r}$, by the union bound,

$$\begin{aligned}
&\mathbf{Pr}[T' \text{ rejects after } q(\mathcal{M},\epsilon) \text{ queries}] \\
=&\mathbf{Pr}[\text{There exists an } S \subset [q(\mathcal{M},\epsilon)] \text{ such that } A_S \text{ happens}] \\
\leq&\binom{q(\mathcal{M},\epsilon)}{r} \cdot r! \cdot \delta < \delta \cdot q(\mathcal{M},\epsilon)^r.
\end{aligned}$$

□

By Lemma 4.9, in order for T' to reject with probability at least $2/3$, the query complexity of T' is at least $q(\mathcal{M},\epsilon) \geq (\frac{2}{3\delta})^{1/r}$. Now consider the canonical tester T'' that runs in ℓ independent stages which, at each stage, selects uniformly at random a linearly independent r-tuple $(z_1,\ldots,z_r)$ and checks for violation of $\mathcal{M}^*$-freeness. How many queries does T'' need to make to achieve the same rejection probability on $(f_1,\ldots,f_k)$ as T' does after $q(\mathcal{M},\epsilon)$ queries? Clearly the probability that T'' rejects $(f_1,\ldots,f_k)$ after ℓ stages is $1-(1-\delta)^\ell \geq 2/3$, for all $\ell \geq \ell_0 = \frac{2}{\delta} = O(q(\mathcal{M},\epsilon)^r)$. Since T'' makes k queries in each stage, the total number of queries T'' makes is at most $k\ell_0 = O(k \cdot q(\mathcal{M},\epsilon)^r)$. □

Combining Theorem 3.15 and Theorem 4.4, and using that for the triangle-free property $k=3$ and it corresponds to the matroid $\mathcal{M} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_1+\boldsymbol{e}_2)$ with rank $r=2$, we finally have the following query lower bound on *all* one-sided testers for triangle-freeness:

Theorem 4.10. *For all small enough ϵ there is an integer $n_0(\epsilon)$ such that the following holds. For every integer $n \geq n_0$, there is a Boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ such that f depends on all n input variables, f is ϵ-far from being triangle-free, and testing triangle-freeness of f requires any one-sided tester to query the function $\Omega\left((\frac{1}{\epsilon})^{2.423\cdots}\right)$ times.*

5 Concluding Remarks and Open Problems

We have given polynomial lower bounds on the query complexity of one-sided testers for triangle-freeness. We strongly believe that there exist a super-polynomial lower bound. One possible approach is try to prove Conjecture 3.5. It seems that one of the main difficulties in understanding triangle-freeness lower bound is that there is no good characterization of the distance between a Boolean function and the set of triangle-free functions (as opposed to the linearity case, where the distance is exactly characterized by the Fourier coefficients of the function). It is also interesting to study the query complexities of (cycle) C_r-freeness for $r \geq 5$.

Another interesting problem is whether the tower of 2's type query upper bound of testing triangle-freeness [23, 19] can be improved. Is it possible that some two-sided testers can achieve much better upper bounds?

Acknowledgments

We thank Victor Chen and Madhu Sudan for collaboration during the early stages of this research as well as enlightening discussions. We are indebted to Ilan Newman for asking a question that initiated the work presented in Section 4. We thank Avinatan Hassidim, Ronitt Rubinfeld and Andy Yao for helpful discussions and Alex Samorodnitsky for valuable comments. Finally, we are grateful to the reviewers whose suggestions greatly improved the presentation of the paper.

References

[1] Noga Alon. Testing subgraphs in large graphs. *Random Structures and Algorithms*, 21(3-4):359–370, 2002.

[2] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. Efficient testing of large graphs. *Combinatorica*, 20(6):451–476, 2000.

[3] Noga Alon, Eldar Fischer, Ilan Newman, and Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 251–260, 2006.

[4] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing low-degree polynomials over GF(2). In *Random'03: Proceedings of 7th International Workshop on Randomization and Computation*, pages 188–199, 2003.

[5] Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy. Regular languages are testable with a constant number of queries. *SIAM Journal on Computing*, 30(6):1842–1862, 2000.

[6] Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer and System Sciences*, 69(3):354–382, 2004.

[7] Noga Alon and Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. In *FOCS'05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 429–438. IEEE Computer Society, 2005.

[8] Noga Alon and Asaf Shapira. Every monotone graph property is testable. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 128–137. ACM, 2005.

[9] Tim Austin and Terence Tao. On the testability and repair of hereditary hypergraph properties, 2008. http://arxiv.org/abs/0801.2179.

[10] Thomas Bailey and John Cowles. A convex hull inclusion test. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2):312–316, 1987.

[11] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005. Early version in STOC'03.

[12] Arnab Bhattacharyya, Victor Chen, Madhu Sudan, and Ning Xie. Testing linear-invariant non-linear properties. In *STACS'09: Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science*, pages 135–146, 2009.

[13] Arnab Bhattacharyya and Ning Xie. Lower bounds on testing triangle-freeness in Boolean functions. In *SODA'10: Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 87–98, 2010.

[14] Christian Borgs, Jennifer T. Chayes, László Lovász, Vera T. Sós, Balázs Szegedy, and Katalin Vesztergombi. Graph limits and parameter testing. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 261–270, 2006.

[15] Henri Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 2000.

[16] Henry Cohn, Robert Kleinbergy, Balázs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *FOCS'05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 379–388, 2005.

[17] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251–280, 1990. Earlier version in STOC'87.

[18] Eric Domenjoud. Solving systems of linear diophantine equations: an algebraic approach. In *In Proc. 16th Mathematical Foundations of Computer Science, Warsaw, LNCS 520*, pages 141–150. Springer-Verlag, 1991.

[19] Jacob Fox. A new proof of the graph removal lemma. *Annals of Mathematics*, 174(1):561–579, 2011.

[20] Hu Fu and Robert Kleinberg. Improved lower bounds for testing triangle-freeness in boolean functions via fast matrix multiplication, 2013. http://arxiv.org/abs/1308.1643.

[21] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

[22] Oded Goldreich and Luca Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, 23(1):23–57, 2003.

[23] Ben Green. A Szemerédi-type regularity lemma in abelian groups, with applications. *Geom. Funct. Anal.*, 15(2):340–376, 2005.

[24] Peter Gruber. *Convex and Discrete Geometry*. Springer, New York, 2007.

[25] Pooya Hatami, Sushant Sachdeva, and Madhur Tulsiani. An arithmetic analogue of Fox's triangle removal argument, 2013. http://arxiv.org/abs/1304.4921.

[26] Ishay Haviv and Ning Xie. Sunflowers and testing triangle-freeness of functions. Manuscript, 2013.

[27] Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. In *FOCS'04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 423–432, 2004.

[28] Tali Kaufman and Dana Ron. Testing polynomials over general fields. In *FOCS'04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 413–422, 2004.

[29] Tali Kaufman and Madhu Sudan. Algebraic property testing: The role of invariance. In *STOC'08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 403–412, 2008.

[30] Daniel Král', Oriol Serra, and Lluís Vena. A combinatorial proof of the removal lemma for groups. *Journal of Combinatorial Theory*, 116(4):971–978, May 2009.

[31] Daniel Král', Oriol Serra, and Lluís Vena. A removal lemma for systems of linear equations over finite fields. *Israel Journal of Mathematics*, 187:193–207, 2012.

[32] Florence J. MacWilliams and Neil J.A. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[33] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic Boolean formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2003.

[34] Vojtěch Rödl and Mathias Schacht. Generalizations of the removal lemma. *Combinatorica*, 29(4):467–501, 2009. Earlier version in STOC'07.

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

[36] Asaf Shapira. Green's conjecture and testing linear-invariant properties. In *STOC'09: Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 159–166, 2009.

A Proof of Theorem 3.8

We will need the following well-known theorem of Carathéodory in convex geometry (see, e.g., [24]).

Theorem A.1 (Carathéodory's Theorem)**.** *Suppose V is a subset of $\mathbb{R}^n$ that contains a point $X \in \mathbb{R}^n$ in its convex hull. Then there exists a set $V' \subseteq V$ such that $|V'| \leq n+1$ and X is contained in the convex hull of V'. An implication is that if V contains $\vec{0}$ in its convex hull and there is no strict subset V' containing $\vec{0}$ in its convex hull, then* $\mathrm{rank}(V) = |V| - 1$.

Proof of Theorem 3.8. If there exists a non-zero vector $\vec{Z} \in \mathbb{N}^t$ such that $\mathbf{M}\vec{Z} = \vec{0}$, the vector $\vec{z} = \frac{\vec{Z}}{\|\vec{Z}\|_1}$ also satisfies $M\vec{z} = \vec{0}$. But then, $\vec{0} \in \mathrm{Conv}(M_1, \ldots, M_t)$ because $\sum_i z_i M_i = \vec{0}$ and each $z_i \geq 0$ with $\sum_i z_i = 1$.

In the other direction, suppose $\vec{0} \in \mathrm{Conv}(M_1, \ldots, M_t)$. Let $\{M_{i_1}, \ldots, M_{i_k}\}$ be a minimal subset of $\{M_1, \ldots, M_t\}$ which contains $\vec{0}$ in its convex hull. Carathéodory's theorem (Theorem A.1) implies that the rank of $\{M_{i_1}, \ldots, M_{i_k}\}$ is $k - 1 \leq s$. Let $\mathbf{M}'$ be the s-by-k matrix with columns $\{M_{i_1}, \ldots, M_{i_k}\}$. Then there exists a unimodular (that is, the determinant of the matrix is either 1 or -1) s-by-s matrix $\mathbf{U}$ such that

$$\mathbf{U}\mathbf{M}' = \begin{bmatrix} \mathbf{N} \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $\mathbf{N}$ is a $(k-1)$-by-k integer matrix of rank $(k-1)$ in row-echelon form[14]. It follows that the nullspace of $\mathbf{N}$ is spanned by a single non-zero vector in $\mathbb{R}^k$. Since $\vec{0}$ is in the convex hull of $\{M_{i_1}, \ldots, M_{i_k}\}$, there exists a non-zero vector $\vec{X} \in (\mathbb{R}^{\geq 0})^k$ such that $\mathbf{N}\vec{X} = \vec{0}$. It follows that all the vectors in the nullspace of $\mathbf{N}$ have the same sign at each coordinate. But the vector consists of the cofactors of $\mathbf{N}$, namely, $\vec{Y} = (\left|N_2 \cdots N_k\right|, \ldots, (-1)^{k-1}\left|N_1 \cdots N_{k-1}\right|)$ is a solution to $\mathbf{N}\vec{X} = \vec{0}$. Furthermore, all the entries in $\vec{Y}$ are non-zero since the rank of $\mathbf{N}$ is $k-1$. Hence either $\vec{Y}$ or $-\vec{Y}$ is a positive integer solution to $\mathbf{N}\vec{X} = \vec{0}$, and because $\mathbf{U}$ is invertible, the same positive integer vector satisfies $\mathbf{M}'\vec{X} = \vec{0}$. Appending 0 entries to $\vec{X}$ at all the remaining $(t-k)$ coordinates gives a non-negative integer solution to $\mathbf{M}\vec{Z} = \vec{0}$. □

[14] See, for example, Theorem 2.4.3 in [15].