Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Theoretical Computer Science 299 (2003) 81–106

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Algebraic testing and weight distributions of codes ☆

M. Kiwi[a,b,*]

[a] *Departmento de Ing. Matemática, Universidad de Chile, Casilla 170-3, Correo 3, Santiago, Chile*
[b] *Ctr. de Modelamiento Matemático, UMR2071 CNRS-UChile, Chile*

Received August 2000; received in revised form March 2001
Communicated by J. Díaz

Abstract

We study the testing problem, that is, the problem of determining (maybe probabilistically) if a function to which one has oracle access satisfies a given property.

We propose a framework in which to formulate and carry out the analysis of several known tests. This framework establishes a connection between testing and the theory of weight distributions of codes. We illustrate this connection by giving a coding theoretic interpretation of several tests that fall under the label of low-degree tests. We also show how the connection naturally suggests a new way of testing for linearity over finite fields.

We derive from the MacWilliams Theorems a general result, the Duality Testing Lemma, and use it to analyze the simpler tests that fall into our framework. In contrast to other analyses of tests, the ones we present elicit the fact that a test's probability of rejecting a function depends on how far away the function is from every function that satisfies the property of interest.

Keywords: Testing; MacWilliams Theorems; Program Checking; Probabilistically Checkable Proofs

1. Introduction

The theory of programs that check their work and of program testers was pioneered by Blum et al. [12, 13] as an alternative approach to the problem of assuring software reliability. Moreover, it had unexpected theoretical consequences. Indeed, it is part of the work that led to the PCP theorem of Arora et al. [3]. Testing techniques are

☆ Part of this work was done while the author was a graduate student in the Department of Mathematics at MIT and appeared in the author's Ph.D thesis [24].

* Corresponding author.
E-mail address: mkiwi@dim.uchile.cl (M. Kiwi).
URL: www.dim.uchile.cl/∼mkiwi. (M. Kiwi).

0304-3975/03/$ - see front matter
PII: S0304-3975(01)00816-2

also an essential component in the derivations of hardness of approximation results. Furthermore, Goldreich et al. [23, 22] have recently extended the testing paradigm and thus brought renewed attention to the testing problem.

Loosely stated, the testing problem consists in determining (maybe probabilistically) if a function satisfies a given property. In this work, we say that a *test* is a triple of the form $(\mathscr{F},\mathscr{T},\mathscr{D})$ where $\mathscr{F}$ and $\mathscr{T}$ are collections of functions and $\mathscr{D}$ is a probability distribution over $\mathscr{T}$. Specifically, all functions in $\mathscr{F}$ have the same domain and range. Functions in $\mathscr{T}$ map $\mathscr{F}$ to $\{accept, reject\}$. We let $\mathscr{P}$ denote the family of functions $f\in\mathscr{F}$ for which $T(f)$ equals *accept* for every T which is assigned a positive probability by the distribution $\mathscr{D}$. We assume from now on that $\mathscr{P}$ is nonempty. To state the testing problem nothing else, neither about the domain and/or range of the functions in $\mathscr{F}$ nor the feasibility of sampling from $\mathscr{D}$ needs to be assumed. Indeed, the testing problem consists in explaining the relation between the following two quantities when the function f, with domain $\mathrm{Dom}(f)$, varies over $\mathscr{F}$:

$\mathrm{Rej}(f)=\Pr_{T\sim\mathscr{D}}[T(f)=reject]$—the probability that the test rejects f,

$\mathrm{Dist}(f)=\min\{\Pr_{u\in_R\mathrm{Dom}(f)}[f(u)\neq g(u)]\mid g\in\mathscr{P}\}$—minimum (relative) distance of f to its closest function in $\mathscr{P}$ (here, $\Pr_{a\in_R A}[E_a]$ denotes the probability that the event E_a occurs when a is chosen at random according to the uniform distribution over A).

Of particular importance is to provide a *lower bound* for the test, i.e., to determine a function $\varphi:[0,1]\to[0,1]$ such that $\mathrm{Rej}(f)\geqslant\varphi(\mathrm{Dist}(f))$. The quality of the lower bound depends on the function φ. The reason being, that one is usually interested in determining the largest value $\mathrm{Dist}(f)$ can take given that $\mathrm{Rej}(f)$ is at most ε.

For the sake of concreteness, let us consider a typical (and actually the first) test addressed in the literature, i.e., the Blum–Luby–Rubinfeld (BLR) test [13]. Its goal is to verify whether a function f mapping a finite group G into another group H is such that $f(x+y)=f(x)+f(y)$ for all $x,y\in G$, i.e., it is a group homomorphism. Blum et al. proposed performing the following probabilistic verification procedure: First, independently and uniformly randomly choose x and y in G. Then, claim that f is a homomorphism if $f(x+y)=f(x)+f(y)$, otherwise say it is not a homomorphism. Here,

$\mathrm{Rej}(f)=\Pr_{x,y\in_R G}[f(x+y)\neq f(x)+f(y)]$—is the probability that the test rejects f,

$\mathrm{Dist}(f)=\min\{\Pr_{x\in_R G}[f(x)\neq g(x)]\mid l\in\mathscr{P}\}$—is the smallest fraction of values of f's domain that need to be modified in order to obtain a function in the family $\mathscr{P}$ of homomorphisms from G to H.

In our terminology, the BLR test is the triple $(\mathscr{F},\mathscr{T},\mathscr{D})$ where $\mathscr{F}$ is the collection of functions from G to H, the family $\mathscr{T}$ is the set of functionals $\{T_{x,y}\mid x,y\in G\}$ such that $T_{x,y}(f)=accept$ if $f(x+y)=f(x)+f(y)$ and *reject* otherwise, and $\mathscr{D}$ is the uniform distribution over $\mathscr{T}$.

There always is an undercurrent in the testing problem. In it, we assume we have oracle access to a function $f \in \mathscr{F}$. (That is, we can specify u in the domain of f and in one step are returned $f(u)$.) The function f is usually referred to as the *oracle function*. It is claimed that "f belongs to $\mathscr{P}$". We do not trust this claim. We verify it (probabilistically) by performing the test $(\mathscr{F}, \mathscr{T}, \mathscr{D})$ as follows: we sample according to $\mathscr{D}$ a functional T in $\mathscr{T}$; we accept the claim if $T(f) = accept$ and reject it otherwise. (Evaluating $T(f)$ requires performing queries to the oracle and carrying out some computation that depends on the outcome of the queries.) Note that the functions belonging to $\mathscr{P}$ always pass the test. These functions are thought of as those that satisfy a property of interest. A natural question arises: *with what probability does the test reject the claim concerning f when* $\mathrm{Dist}(f)$ *is at least* δ? A lower bound for the test in point provides a partial answer to this question.

Tests have been designed to verify many properties. Some of them are of practical relevance. We refer the reader interested in this subject to the survey of Blum and Wasserman [14] and the thesis of Rubinfeld [31]. For pointers to more recent work in the area see [32, 15]. For recent developments concerning issues in property testing see [22, 21].

As mentioned earlier, tests show up in the construction of PCPs. The reason being that a central problem in their construction is to probabilistically check/test algebraic properties of functions with as few oracle queries as possible. Among these properties, of prime importance are: linearity [13, 10, 11, 8], multi-linearity [7, 16], low-individual degree [6, 4, 18, 29], low-total degree [20, 3, 33, 19, 5]. Testing that a given function satisfies one of these properties is what is often referred to as *low-degree testing*. Typically, better analyses of the low-degree tests allow a more accurate determination of the characteristics of known PCP constructions. A more pragmatic reason for obtaining improved analysis of low-degree tests is that they usually translate, as first observed in [16], into improved hardness of approximation results. For a comprehensive discussion of hardness of approximation results and pointers to the relevant literature the reader is referred to [9, 2].

Currently we have no general framework which allows us to naturally formulate tests for checking whether an algebraic function satisfies a pre-specified property. Known tests might not be the most appropriate for performing the tasks which they are designed for. Furthermore, a characteristic of most tests is that they are hard to analyze. It is not even clear if many of the known analyses are tight. This work's main motivation is to contribute in the development of a unified framework in which to cast known tests, formulate new ones, and provide the tools to analyze them well.

1.1. Main contributions and new techniques

The crux of this work is the framework which we propose in order to formulate and carry out the analyses of tests. This framework establishes a connection between testing and the theory of weight distribution of codes. We illustrate this connection in two ways. First, we cast into the framework we develop several of the tests that

have been studied in the literature under the label of low-degree tests. We achieve this through an elegant abstraction of known low-degree tests. We call this abstraction the generic duality test. Second, we show how our framework naturally gives rise to some tests for verifying specific function properties. In particular, we formulate a new way of testing for linearity over finite fields, namely the *extended linearity test* (EL). When F_q is a prime field, being linear is equivalent to being a homomorphism. Hence, in this latter case, the BLR test is a test for verifying linearity. We literally *deduce* the following alternative way of checking whether the function $f : F_q^n \to F_q$ is linear:

> Randomly pick distinct $u, v, w \in F_q^n$ and nonzero scalars $\lambda_u, \lambda_v, \lambda_w \in F_q$ such that $\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. Then, reject if $\lambda_u f(u) + \lambda_v f(v) + \lambda_w f(w) \neq 0$ and accept otherwise.

Let $\mathrm{Rej}(f)$ denote the probability that the above test rejects f and $\mathrm{Dist}(f)$ the distance between f and its closest linear function. Our main result concerning the EL test shows that for every f,

$$\mathrm{Rej}(f) \geqslant \mathrm{Dist}(f) - \mathrm{O}\left(\frac{1}{q^n}\right). \tag{1}$$

Specifically,

$$\mathrm{Rej}(f) = \frac{\gamma}{q} - \left(\frac{q}{\gamma}\right)^2 \sum_l \left(\frac{\gamma}{q} - \mathrm{Dist}(f, l)\right)^3 - \mathrm{O}\left(\frac{1}{q^n}\right) \tag{2}$$

where the summation is over all linear functions $l : F_q^n \to F_q$.

Besides the EL test we focus attention on the low-degree tests.

Typically, the tests' analyses that we provide proceed in two steps. First, we derive an essentially tight characterization of the rejection probability of the test in terms of the distance between the oracle function and each function in the underlying space of interest (e.g., [2] in the case of the EL test). In the second step, we lower bound the alternative characterization of the test's rejection probability solely in terms of the distance between the oracle function and the underlying space of interest. This yields a lower bound for the test in point (e.g., [1] in the case of the EL test). All of our analyses use classical tools from coding theory. In particular, they heavily rely on the celebrated MacWilliams Theorems for linear codes. This is the first work in the testing and PCP literature where these theorems are explicitly used.

We would like to stress that in our opinion this paper's main contribution is of a non-technical nature. Its worth resides in the unifying framework it introduces for formulating and analyzing many of the low-degree tests that have been discussed in the literature. Besides the elegance of the proposed framework we think it provides new insights and elicits facts related to low-degree testing that other methods of analysis do not capture, e.g., the intuition that the probability with which a function fails a test for a given property depends on how far away that function is from *each* function that satisfies the property.

Finally, this article widens the fruitful connection that has developed over the last decade between coding theory and complexity theory (for a discussion of these mutually enriching interconnections see the survey articles of Sudan [35, 36] and Feigenbaum [17]).

The structure of this work is as follows: In Section 2, we first present some basic notions from coding theory. We then describe the coding theoretic framework that we propose for addressing the testing problem and state a generic testing paradigm which we call generic duality test.

In Section 3, we state several results from coding theory. In particular, the MacWilliams Theorem for Hamming weight enumerators of linear codes. We then derive our main technical contribution: the Duality Testing Lemma. Through this result we can analyze some of the tests that can be stated as an instance of the generic duality test. Two examples of this type of analysis are given in Section 4.

Finally, in Section 5, we state some conclusions and point out possible extensions of our work that would be of interest.

1.2. Conventions

Henceforth q denotes a prime power, F_q denotes GF(q), and $\gamma = q - 1$. Thus, the number of nonzero elements of F_q is γ. For ease of reading, the zero element of F_q^n will appear in boldface type (e.g., $\mathbf{0}$) in order to distinguish it from the zero element of F_q (i.e., 0).

2. A coding theoretic framework for testing

For the sake of some readers convenience, in Section 2.1 we define all the coding theoretic terms that we will use. In Section 2.2, we (informally) establish a relationship between testing and coding theory (in particular the theory of weight distributions of a code versus the one of its dual).

2.1. Coding theory: the basics

For the sake of brevity the following exposition will be terse. For an in-depth discussion of issues in coding theory, the reader is referred to [27, 28, 38].

A *code* of *block length* N over the alphabet F is a subset $\mathscr{C}$ of F^N. We will only consider codes whose alphabet F is a finite field. The elements $x = (x_1, \ldots, x_N) \in \mathscr{C}$ are called *codewords* or *words*. Their *length* is usually denoted by N. The support of a word x, denoted as $\mathrm{supp}(x)$, is the set of coordinates where x is nonzero. Formally, $\mathrm{supp}(x) = |\{i \mid x_i \neq 0\}|$. If x and y are two words, then the *Hamming distance* between x and y, denoted as $\mathrm{d}(x, y)$, is equal to the number of components where x and y are distinct. Formally, $\mathrm{d}(x, y) = |\{i \mid x_i \neq y_i\}|$. The *Hamming weight* of word x is the number of nonzero components of x and is denoted as $\mathrm{wt}(x)$. Formally, $\mathrm{wt}(x) = |\mathrm{supp}(x)|$. The *minimum distance* of a code $\mathscr{C}$ is $\min\{\mathrm{d}(x, y) \mid x \in \mathscr{C}, y \in \mathscr{C}, x \neq y\}$. The *minimum*

weight of a code $\mathscr{C}$, denoted as $\mathrm{wt}(\mathscr{C})$, is $\min\{\mathrm{wt}(x) \mid x \in \mathscr{C}, x \neq 0\}$. We adopt the convention that the empty code has minimum distance and minimum weight 0. For a code $\mathscr{C}$ of block length N we let $A_i(\mathscr{C})$ be the number of codewords in $\mathscr{C}$ of weight i and say that $(A_0(\mathscr{C}), \ldots, A_N(\mathscr{C}))$ is its *weight distribution*. A code $\mathscr{C}$ of block length N is *linear* if its a linear subspace of F^N. If the linear code $\mathscr{C}$ has dimension k then $\mathscr{C}$ is called an $[N,k]$ code. For a linear code, the minimum distance is equal to the minimum weight. If $\mathscr{C}$ is a code of block length N over the alphabet F, then its *dual code* $\mathscr{C}^\perp$ is the collection of y's in F^N such that for all $x \in \mathscr{C}$, $\langle x, y\rangle = \sum_{i=1}^{N} x_i \cdot y_i = 0$ (arithmetic over F). Typically, one considers dual codes $\mathscr{C}^\perp$ only when $\mathscr{C}$ is a linear code. It is only for these codes that $(\mathscr{C}^\perp)^\perp = \mathscr{C}$. Finally, note that $\mathscr{C}$ is an $[N,k]$ code if and only if $\mathscr{C}^\perp$ is an $[N, N-k]$ code.

Example 1. Since $\mathscr{H}_2 = \{0000, 0101, 0011, 0110\}$ is a linear subspace of F_2^4 of dimension 2, it is a $[4,2]$ code over F_2. Its minimum distance is 2 and $\mathscr{H}_2^\perp = \{0000, 1000, 0111, 1111\}$.

2.2. Tests and dual codes

Suppose we have oracle access to a function $f : D \to R$, where D and R are finite objects. We can also view f as a vector in $R^{|D|}$. Conversely, a vector in $R^{|D|}$ can be thought of as a function from D to R. We henceforth view functions and vectors in both of these ways (the viewpoint being clear from context). Also, from now on, we restrict our discussion to the case where $R = F$ is a finite field, $D \subseteq F^n$, and the size of D is N.

We want to determine if the function f has a specific property, say it represents a linear function or a degree d polynomial. Equivalently, we want to verify that f belongs to a specific subset $\mathscr{C}$ of F^N. For simplicity's sake, we focus on function properties that are preserved under addition of functions. In other words, we only consider subsets $\mathscr{C} \subseteq F^N$ which are subspaces, i.e., linear codes of block length N over the alphabet F. We associate with f the smallest linear code of block length N containing both f and every codeword in $\mathscr{C}$, i.e.,

$$\mathscr{C}_f = \{\phi f + \theta g \mid \phi, \theta \in F, g \in \mathscr{C}\}.$$

Note that $\mathscr{C} \subseteq \mathscr{C}_f$, hence $\mathscr{C}_f^\perp \subseteq \mathscr{C}^\perp$. Equality holds if and only if f belongs to $\mathscr{C}$. Thus, $\mathscr{C}^\perp = \mathscr{C}_f^\perp$ if f exhibits the function property of interest. To test the claim "f belongs to $\mathscr{C}$", we perform the following:

Definition 2 (*Generic Duality Test*). Let $\mathscr{C}$ be a linear code of block length $|D|$. Given oracle access to a function $f : D \subseteq F^n \to F$, randomly choose (according to some probability distribution) a codeword $g \in \mathscr{C}^\perp$. Then, accept if $g \in \mathscr{C}_f^\perp$ and reject otherwise.

Observe that the test always accepts when f belongs to $\mathscr{C}$. Particular choices of the code $\mathscr{C}$ and the distribution by which dual codewords are sampled will give rise to specific tests.

In the PCP context we are usually interested in tests that can be implemented with very few queries to the oracle function. In the context of self-testing programs we are not so much concerned with restricting the number of queries. But, we require that the running time of the overall procedure be faster than that of any correct program for computing the function represented by the oracle. Furthermore, it is typically required that the sampling procedure be efficient, i.e., that it should be possible to sample in time which is polynomial in the size of the input to the oracle function. Thus, the more interesting instances of the generic duality test are those in which the sampling distribution assigns positive probability only to codewords of small support/weight. For these sampling distributions the duality test's decision of whether to accept or reject requires making a small number of queries to the oracle function and performing a small amount of computation.

We will see that the weight distribution of a code is fixed once we know the weight distribution of its dual. Thus, by performing the duality test we gain information on how much the weight distribution of $\mathscr{C}_f$ deviates from that of $\mathscr{C}$. In particular, we gain some information about the minimum weight of $\mathscr{C}_f \backslash \mathscr{C}$. This minimum weight, normalized by the block length of $\mathscr{C}$, is the distance from f to the nearest function represented by a codeword in $\mathscr{C}$. The generic duality test will be a "good" test if the larger the fraction of values of f that have to be modified in order to obtain a codeword in $\mathscr{C}$, the smaller the probability of sampling a codeword from $\mathscr{C}^{\perp}$ that also belongs to $\mathscr{C}_f^{\perp}$.

3. Analysis of the generic duality test

In this section, we show that casting the testing problem in a coding theory setting also provides us with tools to study some of the tests that fit into our framework. These tools arise from the theory of weight distributions of codes. This theory describes the relation between the weight distribution of a code and the one of its dual through the celebrated MacWilliams Theorems [26]. The results that we will derive require some knowledge about a family of polynomials called *Krawtchouk polynomials*. For the sake of the reader's convenience Section 3.1 discusses theses polynomials. We state a version of the MacWilliams Theorems in Section 3.2. In Section 3.3, we prove some general useful results, in particular the Duality Testing Lemma. These results will enable us to fully analyze some of the tests that fit into our framework.

3.1. Krawtchouk polynomials

In the sequel, we describe the properties of Krawtchouk polynomials that we use in the remaining part of this work. For additional facts about these polynomials we refer the reader to [27, Chapter 5, Section 7] and [38, Chapter 1, Section 2].

Definition 3. For any integer $N>0$, the Krawtchouk polynomial $\mathrm{K}_k(x;N,q)$ is such that for any real number x,

$$\mathrm{K}_k(x;N,q)=\mathrm{K}_k(x)=\sum_{j=0}^{k}(-1)^j\gamma^{k-j}\binom{x}{j}\binom{N-x}{k-j},\quad k=0,1,\ldots,N,$$

where the binomial coefficient $\binom{x}{j}$ denotes the degree j polynomial $x(x-1)\cdots(x-j+1)/j!$.

Clearly, $\mathrm{K}_k(x)$ is a polynomial of degree k in x. Its leading coefficient is $(-q)^k/k!$. Also, $\mathrm{K}_0(x)=1$, $\mathrm{K}_1(x)=\gamma N-qx$, and $2\mathrm{K}_2(x)=q^2(x-\gamma N/q)^2+q(\gamma-1)(x-\gamma N/q)-\gamma N$.

The generating function for the sequence $\mathrm{K}_0(x),\mathrm{K}_1(x),\ldots$ is

$$\sum_{k=0}^{\infty}\mathrm{K}_k(x)z^k=(1+\gamma z)^{N-x}(1-z)^x. \tag{3}$$

If x is an integer with $0\leqslant x\leqslant N$ the upper limit of summation can be replaced by N.

Rearranging the binomial coefficients in Definition 3 shows that if i and j are non-negative integers,

$$\gamma^i\binom{N}{i}\mathrm{K}_j(i)=\gamma^j\binom{N}{j}\mathrm{K}_i(j). \tag{4}$$

Special properties of Krawtchouk polynomials let us express a polynomial $P(x)$ of degree k as a linear combination of $\mathrm{K}_0(x),\mathrm{K}_1(x),\ldots,\mathrm{K}_k(x)$. Formally:

Theorem 4 (see [27, Chapter 5, Section 7]). *If $P(x)$ is a degree k polynomial in x, then*

$$P(x)=\sum_{j=0}^{k}c_j(P)\mathrm{K}_j(x),\quad \textit{where } c_j(P)=q^{-N}\sum_{i=0}^{N}P(i)\mathrm{K}_i(j).$$

The coefficient $c_j(P)$ is called the jth coefficient in the Krawtchouk polynomial expansion of $P(x)$.

Remark 5. Let the random variable X be distributed as a *Binomial* $(N,\gamma/q)$ (the distribution of the sum of N independent identically distributed $\{0,1\}$-random variables with expectation γ/q). By (4), we have the following identities for the coefficients in the Krawtchouk expansion of the polynomial P:

$$\binom{N}{j}\gamma^j c_j(P)=\sum_{i=0}^{N}\binom{N}{i}\left(\frac{\gamma}{q}\right)^i\left(\frac{1}{q}\right)^{N-i}P(i)\mathrm{K}_j(i)=\mathrm{E}[P(X)\mathrm{K}_j(X)].$$

For the sake of future reference, recall that the kth moment about the mean of the random variable X is $v_k=\mathrm{E}[(X-\gamma N/q)^k]$. Hence, in our case $v_1=0$, $v_2=N\gamma/q^2$, $v_3=-N\gamma(\gamma-1)/q^3$, $v_4=3N^2\gamma^2/q^4+N\gamma(1-6\gamma/q^2)/q^2$, and $v_5=-10N^2\gamma^2(\gamma-1)/q^5-N\gamma(\gamma-1)(1+12\gamma/q^2)/q^3$.

3.2. MacWilliams Theorems

We saw that the Hamming weight, or simply the weight, of a codeword equals the number of its nonzero coordinates. We denoted by $A_i(\mathscr{C})$, the number of codewords of weight i in $\mathscr{C}$. The *Hamming weight enumerator* of the code $\mathscr{C}$ of block length N is

$$\mathrm{W}_{\mathscr{C}}(x,y)=\sum_{i=0}^{N} A_i(\mathscr{C})x^{N-i}y^i.$$

Note that the Hamming weight enumerator of a code tells us everything about its weight distribution, i.e., it completely determines $(A_0(\mathscr{C}),\ldots,A_N(\mathscr{C}))$. The surprising fact is that if $\mathscr{C}$ is a linear code, then the weight enumerator of the dual code $\mathscr{C}$ is completely determined by the weight enumerator of $\mathscr{C}$ itself. In fact, it is given by a linear transformation of the weight enumerator of $\mathscr{C}$.

Theorem 6 (MacWilliams Theorem for linear codes, see [27, Chapter 5, Theorem 13])**.** *If $\mathscr{C}$ is a linear code over F_q, then*

$$\mathrm{W}_{\mathscr{C}^\perp}(x,y)=\frac{1}{|\mathscr{C}|}\mathrm{W}_{\mathscr{C}}(x+\gamma y,x-y).$$

Corollary 7. *If $\mathscr{C}$ is a linear code over F_q of block length N, then for $k\in\{0,\ldots,N\}$*

$$A_k(\mathscr{C}^\perp)=\frac{1}{|\mathscr{C}|}\sum_{i=0}^{N} A_i(\mathscr{C})\mathrm{K}_k(i).$$

Proof. Setting $x=1$ in the MacWilliams theorem for linear codes implies that

$$\sum_{k=0}^{N} A_k(\mathscr{C}^\perp)y^k=\frac{1}{|\mathscr{C}|}\sum_{i=0}^{N} A_i(\mathscr{C})(1+\gamma y)^{N-i}(1-y)^i.$$

Since (3) implies that $(1+\gamma y)^{N-i}(1-y)^i=\sum_{k=0}^{N}\mathrm{K}_k(i)y^k$, the corollary follows by equating the RHS and LHS coefficients of y^k in the expression above. □

3.3. The Duality Testing Lemma

The theory discussed in the preceding sections can be used to analyze tests. To show this, we first develop some general useful results.

Proposition 8. *Let $P(x)$ be a degree k polynomial and let $c_j(P)$ be the jth coefficient in the Krawtchouk polynomial expansion of $P(x)$. Then, for every linear code $\mathscr{C}$ of block length N*

$$\sum_{j=0}^{k} c_j(P)A_j(\mathscr{C}^\perp)=\frac{1}{|\mathscr{C}|}\sum_{i=0}^{N} A_i(\mathscr{C})P(i)=\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(g))].$$

Proof. The second equality is obvious. To prove the first identity recall that by Theorem 4 we know that $P(i)=\sum_{j=0}^{k} c_j(P)\mathrm{K}_j(i)$. Multiplying both sides of the previous expression by $A_i(\mathscr{C})$, summing over $i\in\{0,\ldots,N\}$, and dividing by $|\mathscr{C}|$ we get

$$\frac{1}{|\mathscr{C}|}\sum_{i=0}^{N} A_i(\mathscr{C})P(i)=\sum_{j=0}^{k} c_j(P)\left(\frac{1}{|\mathscr{C}|}\sum_{i=0}^{N} A_i(\mathscr{C})\mathrm{K}_j(i)\right).$$

Corollary 7 implies that the inner summation on the RHS is $A_j(\mathscr{C}^{\perp})$. □

Recall that $\mathscr{C}_f$ denotes the smallest linear code containing both $\mathscr{C}$ and the word representing the function $f:F_q^n\to F_q$. The following lemma gives expressions for the number of codewords in $\mathscr{C}^{\perp}$ that are not in $\mathscr{C}_f^{\perp}$.

Lemma 9 (Duality Testing Lemma). *Let $P(x)$ be a polynomial of degree k and let $c_j(P)$ be the jth coefficient in the Krawtchouk polynomial expansion of $P(x)$. Then, for any linear code $\mathscr{C}$ of block length N and every $f\in F_q^N$*

$$\sum_{j=0}^{k} c_j(P)(A_j(\mathscr{C}^{\perp})-A_j(\mathscr{C}_f^{\perp}))=\frac{\gamma}{q}\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(g))-P(\mathrm{wt}(f-g))]. \tag{5}$$

Proof. Two applications of Proposition 8 yield

$$\sum_{j=0}^{k} c_j(P)(A_j(\mathscr{C}^{\perp})-A_j(\mathscr{C}_f^{\perp}))=\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(g))]-\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}_f}[P(\mathrm{wt}(g))].$$

From $\mathscr{C}_f$'s definition and since $\mathscr{C}$ is a linear code,

$$\mathscr{C}_f=\{\phi f-\theta g\,|\,g\in\mathscr{C}\text{ and }\phi,\theta\in F,\phi\neq 0\}\cup\mathscr{C}.$$

Moreover, if $\phi\neq 0$, then $\mathrm{wt}(\phi f-\theta g)=\mathrm{wt}(f-(\theta/\phi)g)$. Thus,

$$\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}_f}[P(\mathrm{wt}(g))]=(\gamma/q)\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(f-g))]+(1/q)\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(g))].$$

The lemma follows. □

Inclusion and closure under linear combinations of f in $\mathscr{C}$ has an effect on $\mathscr{C}^{\perp}$ which is captured by the LHS of (5) while the effect on the weight distribution of $\mathscr{C}$ is captured by the RHS of the same equation. The usefulness of the Duality Testing Lemma in the context of testing is that one might be able to, for some fixed polynomial P, estimate the LHS of (5) and thus gain some information about the RHS of (5). The LHS is estimated (if possible) by designing a test whose rejection probability is an approximation of the LHS term. The RHS gives information about how far away f is from the underlying code $\mathscr{C}$.

Corollary 10. *Let $P(x)$ be a polynomial of degree k and $\mathscr{C}$ be a linear code of block length N. If $\mathscr{C}^{\perp}$ does not have codewords of weight $i \in \{1,\ldots,k\}$ then, for every $f \in F_q^N$*

$$\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(f-g))] = \mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[P(\mathrm{wt}(g))].$$

Proof. Observe that in this case $A_i(\mathscr{C}^{\perp}) = A_i(\mathscr{C}_f^{\perp})$ for all $i \in \{0,\ldots,k\}$ and apply the Duality Testing Lemma. □

4. New and old tests: A coding theory viewpoint

In this section, we illustrate how to interpret known tests and formulate new ones in the setting put forth in Section 2.2. We then discuss a test that has received a considerable amount of attention in the PCP literature; the low total-degree test. We will show that the coding theoretic setting we have proposed can be extended in order to capture the low total-degree test. We begin by illustrating how our framework captures the simplest tests known.

4.1. Testing Hadamard-like codes

One of the most basic questions that arises in testing is that of testing for linearity. As mentioned in Section 1, the seminal paper of Blum et al. [13] proposed a linearity test, the BLR test, and analyzed it. (Actually, the BLR test checks whether a function is a group homomorphism. In some cases, this is equivalent to verifying linearity.)

Recall that in the BLR test one is given oracle access to a function f mapping a finite group G into another group H. One is charged for each oracle call and wants to test that f is close (in relative distance) to a group homomorphism. To do so, one performs the following test:

> BLR Test: Pick $u, v \in G$ at random, query the oracle to obtain $f(u), f(v)$, $f(u+v)$, and accept if and only if $f(u)+f(v) = f(u+v)$.

In the remaining part of this section, we discuss a couple of problems related to the BLR test.

Testing punctured Hadamard codes: The particular case of the BLR test where $G = F_2^n$ and $H = F_2$ is of relevance in the construction of PCPs and the proof of hardness of approximation results (see [8] for a thorough discussion). In this case, being a group homomorphism is equivalent to being linear. Consider the more general problem of testing whether a function $f : S \subseteq F_2^n \to F_2$ to which we have oracle access is linear. The framework proposed in Section 2.2 suggests a way of addressing this problem. Indeed, perform the following test:

> Punctured Hadamard code test: Let $S \subseteq F_2^n$ and $\mathscr{C}$ be the code whose elements are of the form $(l(u) : u \in S)$ where $l : S \to F_2$ is linear. Randomly choose a dual codeword $\lambda \in \mathscr{C}^{\perp}$ of weight three. Then, accept if $\lambda \in \mathscr{C}_f^{\perp}$ and reject otherwise.

The name of the test is due to the fact that the code $\mathscr{C}$ is obtained from a Hadamard code [27, Chapter 2, Section 3] by a process called *puncturing*, i.e., deleting one or more coordinates from each codeword of $\mathscr{C}$. The Hadamard code test corresponds to the special case of the punctured Hadamard code test where $S = F_2^n$.

We now give an equivalent interpretation of the punctured Hadamard code test. If $\lambda = (\lambda_u : u \in S)$ has support $\{u, v, w\} \subseteq S$, by definition of $\mathscr{C}^\perp$, we have that $\lambda \in \mathscr{C}^\perp$ if and only if $l(u) + l(v) + l(w) = 0$ for every linear function $l : F_2^n \to F_2$. Equivalently, $\lambda \in \mathscr{C}^\perp$ if and only if $u + v + w = \mathbf{0}$. Similarly, it can be shown that $\lambda \in \mathscr{C}_f^\perp$ if and only if $\lambda \in \mathscr{C}^\perp$ and $f(u) + f(v) + f(w) = 0$. In other words, the punctured Hadamard code test randomly selects distinct $u, v, w \in S$ such that $u + v + w = \mathbf{0}$ and verifies whether $f(u) + f(v) + f(w) = 0$. When $S = F_2^n$, this is almost equivalent to performing the BLR test when the underlying groups are $G = F_2^n$ and $H = F_2$.[1]

We now discuss why in the punctured Hadamard code test the sampling distribution only picks codewords of weight three. Note that both $\mathscr{C}^\perp$ and $\mathscr{C}_f^\perp$ always have one codeword of weight zero and no codeword of weight two. Since independent of f, both $\mathscr{C}^\perp$ and $\mathscr{C}_f^\perp$ have the same number of weight zero and two codewords, there is no point in sampling from those codewords. Also, note that $\mathscr{C}^\perp$ has at most one codeword of weight one, namely the codeword whose coordinate indexed by $\mathbf{0} \in \mathrm{GF}(2)^n$ is nonzero. The same codeword is in $\mathscr{C}_f^\perp$ if and only if $f(\mathbf{0}) = 0$. Every linear function is zero at $\mathbf{0}$. Thus we might as well assume that the oracle function takes the value zero at $\mathbf{0}$ rather than bother to check whether this holds by choosing the weight one codeword in $\mathscr{C}^\perp$. We are left with the option of sampling codewords in $\mathscr{C}^\perp$ of weight at least three. In order to minimize the number of queries and the time required to perform the test, the Hadamard code test samples only those codewords of $\mathscr{C}^\perp$ that have weight three.

Linearity testing over finite fields: Assume now we have oracle access to a function $f : F_q^n \to F_q$. We are interested in determining whether f is linear, i.e., whether $f(u + v) = f(u) + f(v)$ for all $u, v \in F_q^n$ and $f(\lambda u) = \lambda f(u)$ for all $\lambda \in F_q$ and $u \in F_q^n$. The framework presented in Section 2.2 naturally suggests performing the following test:

> Extended linearity test: Let $\mathscr{C}$ be the collection of words of the form $(l(u) : u \in F_q^n)$ where $l : F_q^n \to F_q$ is linear. Randomly choose a dual codeword $\lambda \in \mathscr{C}^\perp$ of weight three. Then, accept if $\lambda \in \mathscr{C}_f^\perp$ and reject otherwise.

Let us have a closer look at what the extended linearity (EL) test is doing. First observe that if the support of $\lambda = (\lambda_u : u \in F_q^n)$ is $\{u, v, w\}$, then by definition of $\mathscr{C}^\perp$, we have that $\lambda \in \mathscr{C}^\perp$ if and only if $\lambda_u l(u) + \lambda_v l(v) + \lambda_w l(w) = 0$ for every F_q-valued linear function over F_q^n. Equivalently, $\lambda \in \mathscr{C}^\perp$ if and only if $\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. Similarly, $\lambda \in \mathscr{C}_f^\perp$ if and only if $\lambda \in \mathscr{C}^\perp$ and $\lambda_u f(u) + \lambda_v f(v) + \lambda_w f(w) = 0$. Hence, the test randomly picks distinct $u, v, w \in F_q^n$ and nonzero scalars $\lambda_u, \lambda_v, \lambda_w \in F_q$ such that

[1] It is not exactly the BLR test since it never checks whether $f(u) + f(v) = f(u + v)$ when $u, v, u + v$ are not all distinct. In contrast, the BLR test might perform such checks. But, with a negligible probability. To be precise, with a probability of $O(1/2^n)$. Hence, a given oracle function has approximately the same rejection probability under the Hadamard code test and the BLR test.

$\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. The test rejects if $\lambda_u f(u) + \lambda_v f(v) + \lambda_w f(w) \neq 0$ and accepts otherwise. Clearly, the test always accepts a linear function.

We leave to the reader to justify the choice of sampling procedure in the EL test.

Note that for the particular case where $q = 2$ the EL test and the Hadamard code test are the same. Hence, in the case that $q = 2$, the EL test is essentially equivalent to the BLR test when the underlying groups are $G = \mathrm{GF}(2)^n$ and $H = \mathrm{GF}(2)$.

4.1.1. Analyses of Hadamard-like code tests

First, we need to introduce a notion of distance, denoted as $\mathrm{Dist}(f, g)$, between two functions f and g defined over the same domain. We define it as the fraction of places in which f and g differ. Formally, $\mathrm{Dist}(f, g) = \Pr_u[f(u) \neq g(u)]$, where the probability is taken over the choices of elements in the domain of f.

Hadamard code & punctured Hadamard code test: As pointed out before, implementing the Hadamard code test is essentially equivalent to performing the BLR test when the oracle function's domain and range are F_2^n and F_2, respectively. This case was exhaustively analyzed in [8] using discrete Fourier analysis techniques. Similarly, given access to the function $f : S \subseteq F_2^n \to F_2$ the punctured Hadamard code test consists (almost) in randomly choosing $u, v \in S$ and verifying whether $f(u) + f(v) = f(u + v)$ if $u + v \in S$.[2] This latter test was analyzed in [24, Chapter 2] using arguments similar to those of [8] also based in discrete Fourier analysis techniques. The coding theoretic approach taken here generalizes the discrete Fourier analysis technique introduced by Bellare et al. [8]. Rewriting the discrete Fourier based analyses in the language used in this work yields the following

Lemma 11. *Let* $S \subseteq F_2^n$ *and define*

$$\varphi(S) = \frac{1}{|F_2|^{2n}} \{(u, v, w) \in S \times S \times S | u + v + w = \mathbf{0}\}.$$

For every $f : S \subseteq F_2^n \to F_2$ *the probability that the punctured Hadamard code test rejects* f *is*

$$\mathrm{Rej}(f) = \frac{1}{2}\left(1 - \frac{(|S|/|F_2^n|)^3}{\varphi(S)} \sum_l (1 - 2\mathrm{Dist}_S(f, l))^3\right) - \mathrm{O}\left(\frac{1}{|S|}\right)$$

$$\geqslant \frac{(|S|/|F_2^n|)^2}{\varphi(S)}\left(\mathrm{Dist}_S(f) - \frac{1}{2}\right) + \frac{1}{2} - \mathrm{O}\left(\frac{1}{|S|}\right),$$

where the summation is over the linear functions $l : F_2^n \to F_2$, $\mathrm{Dist}_S(f, l) = \Pr_{u \in_R S}[f(u) \neq l(u)]$, *and* $\mathrm{Dist}_S(f)$ *is the distance from* f *to the nearest linear function* $l : S \subseteq F_2^n \to F_2$.

[2] The "almost" is due to the fact that a given oracle function is accepted with a probability similar to within an $\mathrm{O}(1/|S|)$ additive term of the acceptance probability of the punctured Hadamard code.

A consequence of the Summation Lemma [8, 25] is that among all subsets $S \subseteq F_2^n$ of cardinality m, the quantity $\varphi(S)$ is maximized when S is the set of the lexicographically smallest m elements in F_2^n. In some sense, $\varphi(S)$ measures how close S is to being a subspace. Thus, the closer the set S is to a subspace, the better the lower bound of Lemma 11. This validates the intuition that the more the number of constraints of the form $(u, v, u+v) \in S \times S \times S$, the better the punctured Hadamard test should perform.

Recalling that the Hadamard code test is a particular version of the punctured Hadamard code test we get the following:

Corollary 12. *For every $f : F_2^n \to F_2$ the probability that the Hadamard code test rejects f is*

$$\begin{aligned}\mathrm{Rej}(f) &= \frac{1}{2}\left(1 - \sum_l (1-2\mathrm{Dist}(f,l))^3\right) - \mathrm{O}\left(\frac{1}{|F_2|^n}\right)\\ &\geqslant \mathrm{Dist}(f) - \mathrm{O}\left(\frac{1}{|F_2|^n}\right),\end{aligned}$$

where the summation ranges over all the linear functions $l : F_2^n \to F_2$ and $\mathrm{Dist}(f)$ *is the distance from f to the nearest linear function $l : F_2^n \to F_2$.*

Proof. Take $S = F_2^n$ in Lemma 11 and observe that $\varphi(F_2^n) = 1$. □

Extended linearity test: Recall that in this test we assume we are given oracle access to a function $f : F_q^n \to F_q$. We let $\mathscr{C}$ be the collection of codewords of the form $(l(u) : u \in F_q^n)$ where $l : F_q^n \to F_q$ is linear. Note that the collection of functions $\{l_\alpha : F_q^n \to F_q \mid \alpha \in F_q^n, l_\alpha(x) = \sum_{i=1}^n \alpha_i x_i\}$ is the set of all linear functions over F_q^n. In particular it follows that the cardinality of $\mathscr{C}$ is q^n. The test consists in randomly choosing a dual codeword $\lambda \in \mathscr{C}^\perp$ of weight three, accepting if $\lambda \in \mathscr{C}_f^\perp$ and rejecting otherwise. As usual, we denote by $\mathrm{Rej}(f)$ the probability that the test rejects and by $\mathrm{Dist}(f)$ the distance from f to its nearest linear function.

Lemma 13. *For every $f : F_q^n \to F_q$,*

$$\mathrm{Rej}(f) = \frac{\gamma}{q} - \left(\frac{q}{\gamma}\right)^2 \sum_l \left(\frac{\gamma}{q} - \mathrm{Dist}(f,l)\right)^3 - \mathrm{O}\left(\frac{1}{q^n}\right),$$

where the summation is over all linear functions $l : F_q^n \to F_q$.

Proof. We restrict our discussion to the case where $f(\mathbf{0}) = 0$. The general case follows immediately since modification of any one value taken by f can produce an $\mathrm{O}(1/q^n)$ change in $\mathrm{Rej}(f)$.

Let $N = q^n$. Note that $A_0(\mathscr{C}_f^\perp) = A_0(\mathscr{C}^\perp)$, $A_1(\mathscr{C}_f^\perp) = A_1(\mathscr{C}^\perp)$. Now, let $P_k(x) = c_k(x - \gamma N/q)^k$, where $c_k = (-q)^k/k!$. Since $P_k(x)$ and $\mathrm{K}_k(x)$ have the same leading term, the

kth coefficient in the Krawtchouk expansion of $P_k(x)$, i.e., $c_k(P_k)$, is equal to 1. Thus, by the Duality Testing Lemma

$$c_2(P_3)(A_2(\mathscr{C}^\perp) - A_2(\mathscr{C}_f^\perp)) + (A_3(\mathscr{C}^\perp) - A_3(\mathscr{C}_f^\perp)) = \frac{\gamma}{q}\mathrm{E}_{g\in_R\mathscr{C}}[P_3(\mathrm{wt}(g)) - P_3(\mathrm{wt}(f-g))].$$

Since $\mathrm{Rej}(f) = 1 - A_3(\mathscr{C}_f^\perp)/A_3(\mathscr{C}^\perp)$, we conclude that

$$\mathrm{Rej}(f) = \frac{\gamma}{q}\frac{1}{A_3(\mathscr{C}^\perp)}\mathrm{E}_{g\in_R\mathscr{C}}[P_3(\mathrm{wt}(g)) - P_3(\mathrm{wt}(f-g))] - c_2(P_3)\frac{1}{A_3(\mathscr{C}^\perp)}(A_2(\mathscr{C}^\perp) - A_2(\mathscr{C}_f^\perp)). \tag{6}$$

Since, $2\mathrm{K}_2(x) = q^2(x-\gamma N/q)^2 + q(\gamma-1)(x-\gamma N/q) - \gamma N$, Remark 5 implies that

$$c_2(P_3) = \frac{c_3}{2\binom{N}{2}\gamma^2}(q^2 v_5 + q(\gamma-1)v_4 - \gamma N v_3),$$

where v_i is the ith moment about the mean of a $Binomial(N,\gamma/q)$. Again, from Remark 5 we know that $v_3 = \mathrm{O}(N/q)$ and $v_4, v_5 = \mathrm{O}(N^2/q^2)$, it follows that $|c_2(P_3)| = \mathrm{O}(q)$. Furthermore, $A_2(\mathscr{C}_f^\perp) \leqslant A_2(\mathscr{C}^\perp) \leqslant N\gamma^2$ and $A_3(\mathscr{C}_f^\perp) \leqslant A_3(\mathscr{C}^\perp) = N^2\gamma^3/3! - \mathrm{O}(Nq^3)$. Thus, the second term in the RHS of (6) is $\mathrm{O}(1/N) = \mathrm{O}(1/q^n)$.

Note that in $\mathscr{C}$ there is one codeword of weight 0 and $N-1$ codewords of weight $N\gamma/q$. Hence, $\mathrm{E}_{g\in_R\mathscr{C}}[P_3(\mathrm{wt}(g))] = -c_3N^2\gamma^3/q^3$. Moreover, since $\mathrm{wt}(f-g) = N\mathrm{Dist}(f,g)$ and $|\mathscr{C}| = N$,

$$\mathrm{E}_{g\in_R\mathscr{C}}[P_3(\mathrm{wt}(f-g))] = c_3N^3\mathrm{E}_{g\in_R\mathscr{C}}[(\mathrm{Dist}(f,g) - \gamma/q)^3] = -c_3N^2\sum_{l\in\mathscr{C}}\left(\frac{\gamma}{q} - \mathrm{Dist}(f,l)\right)^3.$$

Putting everything together we conclude that

$$\mathrm{Rej}(f) = \frac{-c_3N^2}{A_3(\mathscr{C}^\perp)}\frac{\gamma}{q}\left(\left(\frac{\gamma}{q}\right)^3 - \sum_{l\in\mathscr{C}}\left(\frac{\gamma}{q} - \mathrm{Dist}(f,l)\right)^3\right) - \mathrm{O}\left(\frac{1}{q^n}\right).$$

Recalling that $c_3 = (-q)^3/3!$ and $A_3(\mathscr{C}^\perp) = N^2\gamma^3/3! - \mathrm{O}(Nq^3)$, a simple algebraic manipulation yields the desired result. □

Lemma 14. *For every* $f: F_q^n \to F_q$, $\mathrm{Rej}(f) \geqslant \mathrm{Dist}(f) - \mathrm{O}(1/q^n)$.

Proof. Again it suffices to prove the result for the case in which $f(\mathbf{0}) = 0$. The general case follows immediately. As usual let $N = q^n$. The expected fraction of places where a randomly chosen linear function agrees with f is at least $(1-1/N)/q$ points. There must

be a linear function that agrees with f in such a fraction of points. Thus, $\mathrm{Dist}(f) \leqslant \gamma/q$. Hence, from Lemma 13 we get that

$$\mathrm{Rej}(f) \geqslant \frac{\gamma}{q} - \left(\frac{q}{\gamma}\right)^2 \left(\frac{\gamma}{q} - \mathrm{Dist}(f)\right) \sum_{l \in \mathscr{C}} \left(\frac{\gamma}{q} - \mathrm{Dist}(f,l)\right)^2 - \mathrm{O}\left(\frac{1}{q^n}\right).$$

To conclude, it suffices to show that $\sum_{l \in \mathscr{C}} (\gamma/q - \mathrm{Dist}(f,l))^2 \leqslant (\gamma/q)^2$.

As in the proof of Lemma 13 let $c_k = (-q)^k/k!$ and $P_k(x) = c_k(x - \gamma N/q)^k$. Recall that $c_k(P_k) = 1$ and observe that

$$\sum_{l \in \mathscr{C}} \left(\frac{\gamma}{q} - \mathrm{Dist}(f,l)\right)^2 = \frac{1}{c_2 N} \mathrm{E}_{g \in_{\mathrm{R}} \mathscr{C}}[P_2(\mathrm{wt}(f-g))].$$

Since $c_2(P_2) = 1$, $A_0(\mathscr{C}_f^{\perp}) = A_0(\mathscr{C}^{\perp})$, $A_1(\mathscr{C}_f^{\perp}) = A_1(\mathscr{C}^{\perp})$, and $A_2(\mathscr{C}_f^{\perp}) \leqslant A_2(\mathscr{C}^{\perp})$, by the Duality Testing Lemma, $\mathrm{E}_{g \in_{\mathrm{R}} \mathscr{C}}[P_2(\mathrm{wt}(f-g))] \leqslant \mathrm{E}_{g \in_{\mathrm{R}} \mathscr{C}}[P_2(\mathrm{wt}(g))]$. In $\mathscr{C}$ there is one codeword of weight 0 and $N-1$ codewords of weight $N\gamma/q$. Hence, $\mathrm{E}_{g \in_{\mathrm{R}} \mathscr{C}}[P_2(\mathrm{wt}(g))] = c_2 N\gamma^2/q^2$. Putting everything together proves the claimed inequality. □

The following lower bound complements Lemma 14. Its proof is based on an argument used in [10] in obtaining an improved lower bound for the BLR test when the underlying field is GF(2).

Proposition 15. *For every* $f : F_q^n \to F_q$,

$$\mathrm{Rej}(f) \geqslant 3\mathrm{Dist}(f)\left(1 - \frac{q}{\gamma}\mathrm{Dist}(f)\right) - \mathrm{O}\left(\frac{1}{q^n}\right).$$

Proof. Let l be the closest linear function to f (ties broken arbitrarily). Observe that the zero function is the closest linear function to $f - l$ and that $\mathrm{Rej}(f) = \mathrm{Rej}(f - l)$. Thus, without loss of generality we may assume that l is the zero function. Note that $\mathrm{Rej}(f) = \Pr[\lambda_u f(u) + \lambda_v f(v) + \lambda_w f(w) \neq 0] - \mathrm{O}(1/q^n)$, where the probability is taken over the choices of the nonzero scalars $\lambda_u, \lambda_v, \lambda_w \in F_q$ and the points $u, v \in F_q^n$, and where w is the unique solution of $\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. Thus, partitioning according to how many of the values $f(u), f(v), f(w)$ are nonzero we get

$$\begin{aligned}\mathrm{Rej}(f) \geqslant\ & 3\Pr[\lambda_u f(u) + \lambda_v f(v) \neq 0, f(u) \neq 0, f(v) \neq 0, f(w) = 0] \\ & + 3\Pr[f(u) \neq 0, f(v) = 0, f(w) = 0] - \mathrm{O}(1/q^n).\end{aligned}$$

It follows that,

$$\begin{aligned}\mathrm{Rej}(f) \geqslant\ & -3\Pr[\lambda_u f(u) + \lambda_v f(v) = 0, f(u) \neq 0, f(v) \neq 0, f(w) = 0] \\ & + 3\Pr[f(u) \neq 0, f(v) = 0] - \mathrm{O}(1/q^n).\end{aligned}$$

The second term in the latter inequality's RHS equals $3\mathrm{Dist}(f)(1 - \mathrm{Dist}(f))$ and the first term is at least $-3\Pr[\lambda_u f(u) + \lambda_v f(v) = 0, f(u) \neq 0, f(v) \neq 0] = -3\mathrm{Dist}(f)^2/\gamma$. The claim follows. □

4.2. Low-degree tests

The Hadamard code test was used in [3] to show that NP languages have probabilistic time verification procedures that use a polynomial number of random bits and require constant number of oracle queries. This result is far from optimal. Indeed, also in [3], it is shown that a logarithmic number of random bits suffices. To establish this latter result, the low total-degree test was introduced. In this section, we give a new interpretation of this test. This interpretation is achieved through an extension of the framework discussed in Section 2. Let us begin by describing some of the low-degree tests that led to the total-degree test.

The basic univariate test: Here the scenario is such that we have oracle access to a function $f : F_q \to F_q$. We want to test whether f belongs to the set $\mathscr{P}_d$ of polynomials from F_q to F_q of total degree d. We now describe a very simple test that achieves this goal.

Basic univariate test [34]: Randomly pick $d+2$ distinct $x_0,\ldots,x_{d+1}$ in F_q. Then, accept if there exists a polynomial in $\mathscr{P}_d$ that agrees with f on $x_0,\ldots,x_{d+1}$ and reject otherwise.

In contrast, our coding theoretic approach suggests performing the following test (we give the test the same name for reasons that will shortly be explained):

Basic univariate test: Let $\mathscr{C}$ be the code whose elements are of the form $(p(x) : x \in F_q)$ where p ranges over $\mathscr{P}_d$. Randomly choose a dual codeword $\lambda \in \mathscr{C}^{\perp}$ of weight $d+2$. Then, accept if $\lambda \in \mathscr{C}_f^{\perp}$ and reject otherwise.

Clearly, both tests always accept polynomials of degree d.

Theorem 16. *Both versions of the* basic univariate test *are equally likely to reject when they have access to the same oracle function.*

We start by proving some intermediate results. First, for $x_0,\ldots,x_j \in F_q$, let

$$M(x_0,\ldots,x_j) = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_j \\ x_0^2 & x_1^2 & \ldots & x_j^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^d & x_1^d & \ldots & x_j^d \end{bmatrix}.$$

Lemma 17. *Let $q > d+1$. Let $\mathscr{C}$ be the code whose elements are of the form $(p(x) : x \in F_q)$ where p ranges over $\mathscr{P}_d$. Let $x_0,\ldots,x_j$ be distinct elements of F_q. If $0 < j < d+1$ there are no codewords in $\mathscr{C}^{\perp}$ with support $\{x_0,\ldots,x_j\}$. If $j = d+1$, there are exactly $\gamma = q-1$ codewords in $\mathscr{C}^{\perp}$ with support $\{x_0,\ldots,x_{d+1}\}$.*

Proof. Let $M = M(x_0,\ldots,x_j)$. Observe that the first $\min\{j+1, d+1\}$ rows of M form a Vandermonde matrix. It is well known that Vandermonde matrices are full rank. It follows that the row rank of M is at least $\min\{j+1, d+1\}$. Since the column and row rank of a matrix are equal and at most the minimum of its number of rows and columns, the rank of M equals $\min\{j+1, d+1\}$. It follows that the kernel of M has dimension 0 and 1 if $j < d+1$ and $j = d+1$, respectively. In the former case, M's kernel contains no elements, while in the latter case it has q elements γ of which are nonzero (and none of them can have a 0 in any of their coordinates). Hence, there are no codewords $\lambda = (\lambda_x : x \in F_q) \in \mathscr{C}^{\perp}$ with support $\{x_0,\ldots,x_{d+1}\}$, since otherwise $(\lambda_x : x \in \{x_0,\ldots,x_{d+1}\})$ would be in M's kernel. Moreover, when $j < d+1$ there are γ codewords $\lambda = (\lambda_x : x \in F_q) \in \mathscr{C}^{\perp}$ with support $\{x_0,\ldots,x_j\}$, each one corresponding to one of the γ nonzero elements $(\lambda_x : x \in \{x_0,\ldots,x_j\})$ of M's kernel. □

Lemma 18. *Let $q > d+1$. Let $f : F_q \to F_q$ and let $\mathscr{C}$ be the code whose elements are of the form $(p(x) : x \in F_q)$ where p ranges over $\mathscr{P}_d$. Let $x_0,\ldots,x_{d+1}$ be distinct elements of F_q. Then, there exists a polynomial in $\mathscr{P}_d$ that agrees with f on $x_0,\ldots,x_{d+1}$ if and only if every dual codeword in $\mathscr{C}^{\perp}$ with support $\{x_0,\ldots,x_{d+1}\}$ belongs to $\mathscr{C}_f^{\perp}$.*

Proof. Let $\lambda = (\lambda_x : x \in F_q) \in \mathscr{C}^{\perp}$ be a word with support $\{x_0,\ldots,x_{d+1}\}$. An argument similar to the one used to prove Lemma 17 shows that $\lambda \in \mathscr{C}_f^{\perp}$ if and only if

$$\begin{bmatrix} f(x_0) & f(x_1) & \ldots & f(x_{d+1}) \\ 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_{d+1} \\ x_0^2 & x_1^2 & \ldots & x_{d+1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^d & x_1^d & \ldots & x_{d+1}^d \end{bmatrix} \begin{bmatrix} \lambda_{x_0} \\ \lambda_{x_1} \\ \vdots \\ \lambda_{x_{d+1}} \end{bmatrix} = \mathbf{0}.$$

Denote by M_f the matrix in the previous expression. Let M be the matrix obtained by removing the first row from M_f. In the proof of Lemma 17, we showed that the rank of M is $d+1$. It follows that M_f is full rank if and only if its first row is not a linear combination of its last $d+1$ rows. From this statement we can derive two claims. The first one states that M_f is full rank if and only if f does not agree on $x_0,\ldots,x_{d+1}$ with a polynomial in $\mathscr{P}_d$. The second claim says that $\lambda \in \mathscr{C}_f^{\perp}$ if and only if M_f is not full rank. The lemma follows. □

Lemmas 17 and 18 immediately imply Theorem 16.

We now discuss why in the second version of the basic univariate test the sampling procedure only picks codewords of weight $d+2$. Note that both $\mathscr{C}^{\perp}$ and $\mathscr{C}_f^{\perp}$ always have one codeword of weight zero and no codewords of weight $1,\ldots,d+1$. Thus, independent of f, both $\mathscr{C}^{\perp}$ and $\mathscr{C}_f^{\perp}$ have the same number of weight $0,\ldots,d+1$ codewords. We are left with the option of sampling codewords in $\mathscr{C}^{\perp}$ of weight

at least $d+2$. In order to minimize the number of queries and the time required to perform the test, only weight $d+2$ codewords of $\mathscr{C}^{\perp}$ are sampled.

It was observed in [34] that from the point of view of testing, the basic univariate test is not very useful. The reason being that performing it is essentially equivalent to computing a polynomial of degree d. Nevertheless, understanding of the basic univariate test makes it easier to describe and study the test for multivariate polynomials discussed below.

The $(d+2)$*-Point Test*: Here the scenario is such that we have oracle access to a function $f: F_q^n \to F_q$. We want to test whether f is a multivariate polynomial of total degree d. We now describe a natural extension of the basic univariate test which works for multivariate polynomials. First, we introduce some more notation. We let $\mathscr{P}_{d,n}$ denote the set of polynomials from F_q^n to F_q of total degree at most d. When $n=1$ we drop the second subindex from $\mathscr{P}_{d,1}$. We say that the set L is a *line* in F_q^n if it is a collection of points of the form $\{u+tv \in F_q^n \mid t \in F_q\}$ where $u, v \in F_q^n$ and $v \neq \mathbf{0}$. The family of all lines in F_q^n will be denoted by $\mathscr{L}_n$.

$(d+2)$-point test [33]: Randomly pick a line $L \in \mathscr{L}_n$ and $d+2$ distinct points $x_0, \ldots, x_{d+1} \in L$. Then, accept if there exists a univariate polynomial in $\mathscr{P}_d$ that agrees with f on $x_0, \ldots, x_{d+1}$ and reject otherwise.

Remark 19. Observe that when $n=1$ the $(d+2)$-point test is simply the basic univariate test. Moreover, denoting by $f|_L$ the restriction of f to the line $L \in \mathscr{L}_n$, the $(d+2)$-point test can be understood as randomly selecting a line L in F_q^n and performing a basic univariate test on the oracle function $f|_L$.

In contrast, our coding theoretic approach to the testing problem suggests performing the following test to determine whether f is a multivariate polynomial of total degree d (we give the test the same name for reasons that will shortly be explained, but the attentive reader might easily guess):

$(d+2)$-point test: Let $\mathscr{C}$ be the code whose elements are of the form $(p(x): x \in F_q^n)$ where p ranges over $\mathscr{P}_{d,n}$. Randomly choose a dual codeword $\lambda \in \mathscr{C}^{\perp}$ of weight $d+2$.

Then, accept if $\lambda \in \mathscr{C}_f^{\perp}$ and reject otherwise.

Clearly, both tests always accept polynomials in n variables of total degree d. We now show that both tests are equivalent, i.e.,

Theorem 20. *Both versions of the* $(d+2)$*-point test are equally likely to accept the same oracle function.*

In order to formally state our result, let $p_0(\cdot), \ldots, p_{m-1}(\cdot)$, be all the monic monomials over F_q^n of total degree at most d. Say that $x_0, \ldots, x_{j-1} \in F_q^n$ are *collinear* if there exists $a, b \in F_q^n$, $b \neq 0$, and scalars $t_0, \ldots, t_{j-1} \in F_q$ such that $x_i = a + t_i b$ for every $i \in \{0, \ldots, j-1\}$. Furthermore, for $x \in F_q^n$ let $\psi(x) = (p_0(x), \ldots, p_{m-1}(x))^{\mathrm{T}}$.

The following two lemmas as well as their proofs are generalizations of Lemmas 17 and 18. Their derivation are tedious exercises, thus we omit them.

Lemma 21. *Let $q>d+1$. Let $\mathscr{C}$ be the code whose elements are of the form $(p(x):x\in F_q)$ where p ranges over $\mathscr{P}_{d,n}$. Let $x_0,\ldots,x_j$ be distinct elements of F_q^n. If $0<j<d+1$ there are no codewords in $\mathscr{C}^\perp$ with support $\{x_0,\ldots,x_j\}$. If $j=d+1$ and $\{x_0,\ldots,x_{d+1}\}$ are contained in a line L of $\mathscr{L}_n$, then there are exactly $\gamma=q-1$ codewords in $\mathscr{C}^\perp$ with support $\{x_0,\ldots,x_{d+1}\}$.*

Lemma 22. *Let $q>d+1$. Let $f:F_q^n\to F_q$ and let $\mathscr{C}$ be the code whose elements are of the form $(p(x):x\in F_q^n)$ where p ranges over $\mathscr{P}_{d,n}$. Let L be a line in $\mathscr{L}_n$ and $x_0,\ldots,x_{d+1}$ be distinct elements of L. There exists a univariate polynomial in $\mathscr{P}_d$ that agrees with f on $x_0,\ldots,x_{d+1}$ if and only if every dual codeword in $\mathscr{C}^\perp$ with support $\{x_0,\ldots,x_{d+1}\}$ belongs to $\mathscr{C}_f^\perp$.*

Assuming that the two stated versions of the $(d+2)$-point test access the same oracle function, from Lemmas 21 and 22, it follows that both accept with equal probability.

The low total-degree test: The scenario we consider is similar to the one we had in the $(d+2)$-point test. Indeed, we again are given oracle access to a function $f:F_q^n\to F_q$. We want to test whether f is a multivariate polynomial of total degree d. We use the same notation introduced when we described the $(d+2)$-point test. Furthermore, for every $L\in\mathscr{L}_n$ we let $P_{f,L}:L\subseteq F_q^n\to F_q$ denote the total degree d polynomial which most agrees with the restriction of f to L (ties are broken arbitrarily). The difference with the scenario we encountered in the $(d+2)$-point test is that we also have oracle access to a table containing an encoding of the polynomials $P_{f,L}$ for every $L\in\mathscr{L}_n$. (That is, we can specify $L\in\mathscr{L}_n$ and in one step are returned and encoding of $P_{f,L}$.)

Low total-degree test [3]: Randomly pick a line $L\in\mathscr{L}_n$ and a point $x\in L$. Then, accept if $f(x)=P_{f,L}(x)$ and reject otherwise.

Clearly, the low total-degree test always accepts if f is a polynomial of total degree d.

There is something that distinguishes the low total-degree test from all the tests we have encountered so far. That is, in the low total-degree test we are given oracle access to *two* tables. One for the function f and another one for the list of polynomials $P_{f,L}$ for $L\in\mathscr{L}_n$. This does not seem to fit the framework developed in Section 2. Nevertheless, below we describe a reformulation of the low total-degree test in purely coding theoretic terms.

First we need to introduce some additional notation. Let $N=q^n$. Let L be a line in F_q^n and $\mathscr{C}\subseteq F_q^N$ be a code whose codewords coordinates are indexed by the elements of F_q^n. Let $\mathscr{C}_L$ denote the collection of codewords of $\mathscr{C}$ whose support is contained in L and where in addition the codewords coordinates indexed by elements of $F_q^n\backslash L$ have been omitted (punctured). Formally, if $\varphi_L:F_q^N\to F_q^q$ is such that $\varphi_L((x_u:u\in F_q^n))=(x_u:u\in L)$ then

$$\mathscr{C}_L=\{\varphi_L(x)\,|\,x\in\mathscr{C},\mathrm{supp}(x)\subseteq L\}.$$

Note that $\mathscr{C}_L$ is a code of block length $|L|=q$.

Henceforth, instead of $(\mathscr{C}_f)_L$ or $(\mathscr{C}_f^{\perp})_L$ we write $\mathscr{C}_{f,L}$ and $\mathscr{C}_{f,L}^{\perp}$, respectively. Recall that the weight of a code $\mathscr{C}$ is the minimum weight codeword in $\mathscr{C}$, and is denoted as $\mathrm{wt}(\mathscr{C})$. We define the relative minimum weight of a code $\mathscr{C}$, denoted as $\rho\,\mathrm{wt}(\mathscr{C})$, as the weight of the code $\mathscr{C}$ divided by its block length.

The following is a simple but key observation.

Lemma 23. *Let $\mathscr{C}$ be the code whose elements are of the form $(p(x) : x \in F_q^n)$ where p ranges over $\mathscr{P}_{d,n}$. For every function $f : F_q^n \to F_q$:*

- *The minimum (relative) distance of f to its closest n-variate total degree d polynomial is* $\mathrm{Dist}(f) = \rho\,\mathrm{wt}(\mathscr{C}_f \backslash \mathscr{C})$.
- *The probability that the low-total degree test rejects the claim "f is an n-variate total degree d polynomial" is* $\mathrm{Rej}(f) = \mathrm{E}_{L \in_R \mathscr{L}_n}[\rho\,\mathrm{wt}(\mathscr{C}_{f,L} \backslash \mathscr{C}_L)]$.

Proof. If f is an n-variate total degree d polynomial the stated lemma is trivial since in this case $\mathscr{C}_f = \mathscr{C}$ and $\mathscr{C}_{f,L} = \mathscr{C}_L$ (recall that by the convention we adopted the weight of the empty code is zero).

Assume f is not an n-variate total degree d polynomial. Let $p \in \mathscr{P}_{n,d}$ be the closest polynomial to f. Observe that there is a codeword $g \in \mathscr{C}_f \backslash \mathscr{C}$ that represents the function $f - p$. But, the weight of g divided by $\mathscr{C}$'s block length equals $\mathrm{Dist}(f)$. Hence, $\mathrm{Dist}(f) \geqslant \rho\,\mathrm{wt}(\mathscr{C}_f \backslash \mathscr{C})$. Let g be a function represented by one of the codewords in $\mathscr{C}_f \backslash \mathscr{C}$. Hence, $g = \phi f + \theta p$ for some $p \in \mathscr{P}_{n,d}$ and $\phi, \theta \in F_q$ where $\phi \neq 0$. Let $p' = -(\theta/\phi)p$ and observe that $p' \in \mathscr{P}_{n,d}$ and $\mathrm{wt}(g) = \mathrm{wt}(f - p')$. Thus, the weight of g divided by the block length of $\mathscr{C}$ is equal to the fraction of places where f and p' differ. It follows that $\mathrm{Dist}(f) \leqslant \rho\,\mathrm{wt}(\mathscr{C}_f \backslash \mathscr{C})$.

To prove the second equality, observe that the fraction of elements of a line L in which $f|_L$ and $P_{f,L}$ differ is exactly the relative minimum weight of the code $\mathscr{C}_{f,L} \backslash \mathscr{C}_L$. □

4.2.1. Analysis of the basic univariate test

Let $f : F_q \to F_q$ denote the function to which we have oracle access. We want to determine whether f belongs to $\mathscr{P}_d$. Recall that this test randomly picks $d+2$ distinct points in F_q. Then, it accepts if there exists a univariate polynomial in $\mathscr{P}_d$ that agrees with f on the $d+2$ points, and rejects otherwise. As usual, we denote by $\mathrm{Rej}(f)$ the probability that the test rejects f and by $\mathrm{Dist}(f)$ the distance from f to its closest polynomial in $\mathscr{P}_d$.

Lemma 24. *Let $q > d+1$ and $C_d = q^{d+2}/((d+2)!\binom{q}{d+2})$. Let $Q(x)$ be the polynomial over the reals that at x takes the value $x\prod_{i=0}^{d}(x-(q-i)/q)$. For every $f : F_q \to F_q$ the probability that the $(d+2)$-point test rejects f is*

$$\mathrm{Rej}(f) = (-1)^{d+1} C_d \sum_{p \in \mathscr{P}_d} Q(\mathrm{Dist}(f, p)).$$

Proof. Let $\mathscr{C}$ be the collection of codewords of the form $(p(x): x \in F_q)$ where p varies over $\mathscr{P}_d$. From Lemma 17 we know that $\mathscr{C}^{\perp}$ does not have codewords of weight $i \in \{1,\ldots,d+1\}$. It follows that the same holds for $\mathscr{C}_f^{\perp}$. Hence, $A_i(\mathscr{C}^{\perp}) = A_i(\mathscr{C}_f^{\perp})$ for every $i \in \{0,\ldots,d+1\}$.

Let $R(x)$ denote the $d+2$ degree polynomial defined over the reals which at x takes the value $cx\prod_{i=0}^{d}(x-q+i)$, where $c=(-q)^{d+2}/(d+2)!$. Since $R(x)$ and $\mathrm{K}_{d+2}(x)$ have the same leading term, the $(d+2)$th coefficient in the Krawtchouk expansion of $R(x)$, i.e., $c_{d+2}(R)$, is equal to 1. Hence, by the Duality Testing Lemma

$$A_{d+2}(\mathscr{C}^{\perp}) - A_{d+2}(\mathscr{C}_f^{\perp}) = \frac{\gamma}{q}\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[R(\mathrm{wt}(g)) - R(\mathrm{wt}(f-g))].$$

A polynomial in $\mathscr{P}_d$ vanishes in either all of F_q or in at most d elements of F_q. Thus, the weight of a codeword in $\mathscr{C}$ belongs to $\{0, q-d,\ldots,q\}$. Hence, for every $g \in \mathscr{C}$ we have that $R(\mathrm{wt}(g)) = 0$. Since $\mathrm{Rej}(f) = 1 - A_{d+2}(\mathscr{C}_f^{\perp})/A_{d+2}(\mathscr{C}^{\perp})$

$$\mathrm{Rej}(f) = -\frac{\gamma}{q}\frac{1}{A_{d+2}(\mathscr{C}^{\perp})}\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[R(\mathrm{wt}(f-g))].$$

But, $\mathrm{wt}(f-g) = q\mathrm{Dist}(f,g)$, so $R(\mathrm{wt}(f-g)) = cq^{d+2}Q(\mathrm{Dist}(f,g))$. Thus,

$$\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[R(\mathrm{wt}(f-g))] = cq^{d+2}\mathrm{E}_{g\in_{\mathrm{R}}\mathscr{C}}[Q(\mathrm{Dist}(f,g))] = cq\sum_{p\in\mathscr{P}_d} Q(\mathrm{Dist}(f,p)),$$

where the last equality follows since $|\mathscr{P}_d| = q^{d+1}$. The claim regarding the basic univariate test is obtained by recalling that from Lemma 17 we know that $A_{d+2}(\mathscr{C}) = \binom{q}{d+2}\gamma$. □

We can use Lemma 24 to rederive a sharper version of a result due to Sudan [34], albeit in a much more involved way. For completeness we include this rederivation below. Although the result is not new, its proof is different than the ones known.

Lemma 25. *Let $q > d+1$. For every $f: F_q \to F_q$, the probability that the basic univariate test rejects the claim "f is a polynomial of degree d" is*

$$\mathrm{Rej}(f) \geqslant \frac{q}{q-(d+1)}\mathrm{Dist}(f).$$

Proof. Let $C_{j-2} = q^j/\binom{q}{j}j!$ and let $Q'(x)$ be the polynomial over the reals that at x takes the value $x\prod_{i=0}^{d}((q-i)/q-x)$. By Lemma 24 we know that $\mathrm{Rej}(f) = C_d \sum_{p\in\mathscr{P}_d} Q'(\mathrm{Dist}(f,p))$. Let $p \in \mathscr{P}_d$ and let $Q''(x)$ be the polynomial over the reals that at x takes the value $\prod_{i=0}^{d}((q-i)/q-x)$. Note that there exists a $j \in \{0,\ldots,q\}$ such that $\mathrm{Dist}(f,p) = j/q$. Hence, $Q''(\mathrm{Dist}(f,p))$ is nonnegative. Moreover, since $\mathrm{Dist}(f,p) \geqslant \mathrm{Dist}(f)$, we conclude that $Q'(\mathrm{Dist}(f,p)) \geqslant \mathrm{Dist}(f)Q''(\mathrm{Dist}(f,p))$. Hence,

$$\mathrm{Rej}(f) \geqslant C_d\mathrm{Dist}(f)\sum_{p\in\mathscr{P}_d} Q''(\mathrm{Dist}(f,p)).$$

Since $|\mathscr{P}_d| = q^{d+1}$, we get that $\sum_{p\in\mathscr{P}_d} Q''(\mathrm{Dist}(f,p)) = \mathrm{E}_{p\in_{\mathrm{R}}\mathscr{P}_d}(Q''(\mathrm{wt}(f-p)))$. But, the degree of Q'' is $d+1$ and, by Lemma 17, $\mathscr{C}^{\perp}$ does not have codewords of weight $\{1,\ldots,d+1\}$. Hence, Corollary 10 implies that the summation in the previous inequality does not depend on f. Thus, assuming without loss of generality that $f\equiv 0$, $\mathrm{Rej}(f)\geqslant C_d\mathrm{Dist}(f)\sum_{p\in\mathscr{P}_d} Q''(\mathrm{Dist}(0,p))$. Since two distinct polynomials agree in at most d places, every term in the latter summation is zero unless p is the zero polynomial. Hence,

$$\mathrm{Rej}(f) \geqslant C_d\mathrm{Dist}(f)Q''(0) = \frac{C_d}{C_{d-1}}\mathrm{Dist}(f) = \frac{q}{q-(d+1)}\mathrm{Dist}(f). \qquad \square$$

Remark 26. Sudan [34] established the above stated lemma but with a 1 instead of the $q/(q-(d+1))$ factor, which is strictly bigger than 1. For large fields, the improvement is irrelevant.

5. Conclusion

There is a key feature in all the analyses we present. Specifically, they capture the intuition that the probability with which a function fails a test for a given property depends on how far away that function is from *each* function that satisfies the property (examples of this are Lemmas 11, 13, and 24). This intuition is not elicited by the standard arguments that have been used to analyze known tests.[3]

The self-testing and PCP literature has, for the most part, focused attention on the testing problem in the case where the oracle function's domain is either a finite group or a finite dimensional vector space over a finite field. Another appealing aspect of our approach is that to study a test we do not impose restrictions on the domain of the oracle function to be tested (other than it should be a subset of a finite dimensional space over a finite field). This is illustrated by the analysis we provide of the punctured Hadamard code test.

Connections between testing, PCPs, and coding theory have been pointed out before (see for example [35]). Our main contribution is to further clarify these connections. Our approach consists in associating to a test and an oracle function a combinatorial object (a code). Then, we show that codes with some specific combinatorial characteristic cannot exist. This allows us to conclude that functions at a given distance from those that satisfy the property of interest and with some particular probability of failing the test in point cannot exist either. Our argument does not need to make any assumption on the magnitude of the rejection probability of the test. This explains why the analyses we provide are meaningful even when the rejection probability is very high. (For example, the lower bound in Lemma 14 for the EL test shows that

[3] In fact, the discrete Fourier analysis based study of the BLR linearity test over GF(2), due to Bellare et al. [8], elicits such intuition. The arguments put forth in this work reduce to discrete Fourier analysis arguments when the underlying field considered is GF(2). Thus, the techniques used in [8] are particular versions of, not different from, the techniques used in this work.

when the rejection probability is $1-\varepsilon$ the distance from the oracle function to the space of linear functions is at most $1-\varepsilon$.) This type of bounds rarely occur in the low-degree testing literature. Other techniques that achieve similar bounds are due to Arora [1] and Tardos [37] who use algebraic arguments to prove a lower bound for a low total-degree test, but under the assumption that the underlying field is of size *exponential* in the degree. For the case of testing *univariate* polynomials, Sudan [34] shows that such lower bounds can be obtained for the basic univariate test. Recently, in a significant breakthrough, Arora and Sudan [5] showed that, under some simple conditions, if the low-total degree test accepts the claim "f is a polynomial of total degree d" with a very small probability δ, then f must agree in approximately a δ fraction of its values with a polynomial of degree d. A result with similar characteristics and consequences was independently obtained by Raz and Safra [30]. It concerns a consistency test implying a new low degree-test.

The techniques we develop in this work have, so far, failed to give nontrivial lower bounds for tests concerning *multivariate* polynomials whose total degree is not bounded by one. This issue needs to be addressed in order to widen the applicability of this work's arguments. It would be specially interesting to derive an alternative proof of the results in [5] concerning the low-total degree test based on our coding theoretic approach to the testing problem.

Acknowledgements

We are very grateful to Madhu Sudan for helpful discussions and encouragement. We also wish to thank Mike Sipser for useful remarks. This work was written while the author was visiting the International Computer Science Institute (ICSI), Berkeley, CA. The author gratefully acknowledges the support of Conicyt via Fondecyt No. 1960849, Fundación Andes, AT&T via a Ph.D. Scholarship, and NSF via Grant CCR-9503322.

References

[1] S. Arora, Private communication, 1995.
[2] S. Arora, The approximability of NP-hard problems, Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, May 1998, ACM, New York, pp. 337–348.
[3] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and hardness of approximation problems, J. ACM 45 (3) (1998) 501–555.
[4] S. Arora, S. Safra, Probabilistic checking of proofs: A new characterization of NP, J. ACM 45 (1) (1998) 70–122.
[5] S. Arora, M. Sudan, Improved low-degree testing and its applications, Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, October 1997, ACM, New York, pp. 485–495.
[6] L. Babai, L. Fortnow, L. Levin, M. Szegedy, Checking computations in polylogarithmic time, Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, New Orleans, LA, May 1991, ACM, New York, pp. 21–31.
[7] L. Babai, L. Fortnow, C. Lund, Non-deterministic exponential time has two-prover interactive protocols, Comput. Complex. 1 (1991) 3–40.

[8] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, M. Sudan, Linearity testing in characteristic two, IEEE Trans. Inform. Theory 42 (6) (1996) 1781–1795.
[9] M. Bellare, O. Goldreich, M. Sudan, Free bits, PCP and non-approximability—towards tight results, SIAM J. Comput. 27 (3) (1998) 804–915.
[10] M. Bellare, S. Goldwasser, C. Lund, A. Russell, Efficient probabilistically checkable proofs and applications to approximation, Proceedings of the 25th Annual ACM Symposium on Theory of Computing, San Diego, CA, May 1993, ACM, New York, pp. 294–304. (See also errata sheet in Proceedings of the 25th Annual ACM Symposium on Theory of Computing, 1994, ACM, New York, p. 820.)
[11] M. Bellare, M. Sudan, Improved non-approximability results, Proceedings of the 26th Annual ACM Symposium on Theory of Computing, Montréal, Qué., Canada, May 1994, ACM, New York, pp. 184–193.
[12] M. Blum, S. Kannan, Designing programs that check their work, Proceedings of the 21st Annual ACM Symposium on Theory of Computing, Seattle, WA, May 1989, ACM, New York, pp. 86–97.
[13] M. Blum, M. Luby, R. Rubinfeld, Selt-testing/correcting with applications to numerical problems, J. Comput. System Sci. 47 (3) (1993) 549–595.
[14] M. Blum, H. Wasserman, Program result-checking: A theory of testing meets a test of theory, Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, November 1994, IEEE Press, New York, pp. 382–392.
[15] F. Ergün, S. Ravi Kumar, D. Sivakumar, Self-testing without the generator bottleneck, SIAM J. Comput. 29 (5) (2000) 1630–1651.
[16] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, Approximating clique is almost NP-complete, J. ACM 43 (2) (1996) 268–292.
[17] J. Feigenbaum, The use of coding theory in computational complexity, in: R. Calderbank (Ed.), Proceedings of Symposia in Applied Mathematics, AMS, Providence, RI, 1995, pp. 207–233.
[18] K. Friedl, Z. Hátsági, A. Shen, Low-degree tests, Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, Arlington, VA, January 1994, ACM, New York, pp. 57–64.
[19] K. Friedl, M. Sudan, Some improvements to total degree testing, Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems, Tel Aviv, Israel, January 1995, IEEE, New York, pp. 190–198.
[20] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, A. Wigderson, Self-testing/correcting for polynomials and for approximate functions, Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, New Orleans, LA, May 1991, ACM, New York, pp. 32–42.
[21] O. Goldreich, Combinatorial property testing—A survey, in: P. Pardalos, S. Rajasekaran (Eds.), Randomization Methods in Algorithm Design, Vol. 43, Dimacs Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, RI, 1999, pp. 45–59.
[22] O. Goldreich, D. Ron, Property testing in bounded degree graphs, Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TA, October 1997, ACM, New York, pp. 406–415.
[23] S. Goldwasser, O. Goldreich, D. Ron, Property testing and its connection to learning and approximation, J. ACM 48 (4) (1998) 653–750.
[24] M. Kiwi, Probabilistic checkable proofs and the testing of Hadamard-like codes, Ph.D. Thesis, Massachusetts Institute of Technology, February 1996.
[25] D.J. Kleitman, Private communication, 1994.
[26] F.J. MacWilliams, Combinatorial problems of elementary group theory, Ph.D. Thesis, Harvard University, May 1962.
[27] F.J. MacWilliams, N.J. A Sloane, The theory of error-correcting codes, 1st Edition, Elsevier Science Publisher B.V., Amsterdam, 1977. (Eight impression: 1993).
[28] W.W. Peterson, E.J. Weldon, Jr, Error-correcting Codes, 2nd Edition, MIT Press, Cambridge, MA, 1972. (Tenth printing: 1990).
[29] A. Polishchuk, D. Spielman, Nearly-linear size holographic proofs, Proceedings of the 26th Annual ACM Symposium on Theory of Computing, Montréal, Qué., Canada, May 1994, ACM, New York, pp. 194–203.
[30] R. Raz, S. Safra, A sub-constant error-probability PCP characterization of NP, Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, October 1997, ACM, New York, pp. 475–484.

[31] R. Rubinfeld, A mathematical theory of self-checking, self-testing and self-correcting programs, Ph.D. Thesis, University of California, Berkeley, 1990.
[32] R. Rubinfeld, On the robustness of functional equations, SIAM J. Comput. 28 (6) (1999) 1972–1997.
[33] R. Rubinfeld, M. Sudan, Robust characterizations of polynomials with applications to program testing, SIAM J. Comput. 25 (1) (1996) 252–271.
[34] M. Sudan, Efficient checking of polynomials and proofs and the hardness of approximation problems, Ph.D. Thesis, University of California, Berkeley, May 1992.
[35] M. Sudan, On the role of algebra in the efficient verification of proofs, December 1994, Presented in Workshop on Algebraic Methods in Complexity Theory (AMCOT).
[36] M. Sudan, List decoding: Algorithms and applications, SIGACT News 31 (1) (2000) 16–27.
[37] G. Tardos, Manuscript, 1995.
[38] J.H. van Lint, Introduction to Coding Theory, Vol. 86, Graduate Texts in Mathematics, 2nd Edition, Springer, Berlin, 1992.