Electronic Colloquium on Computational Complexity - Reports Series 1997 - available via:

ECCC
TR97-010

FTP: ftp.eccc.uni-trier.de:/pub/eccc/
WWW: http://www.eccc.uni-trier.de/eccc/
Email: ftpmail@ftp.eccc.uni-trier.de with subject 'help eccc'

Testing and Weight Distributions of Dual Codes *

Marcos Kiwi†
U. of Chile

April 2, 1997

Abstract

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

We propose a framework in which to formulate and carry out the analyzes of several known tests. This framework establishes a connection between testing and the theory of weight distributions of dual 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 coding theoretic connection we establish naturally suggests a new way of testing for linearity over finite fields.

There are two important parameters associated to every test. The first one is the test's probability of rejecting the claim that the function to which it has oracle access satisfies a given property. The second one is the distance from the oracle function to any function that satisfies the property of interest. The goal when analyzing tests is to explain the relationship between these two parameters. There are several good reasons why good analyzes are worth striving for. For example, improved analyzes of a family of tests referred to as low-degree tests, typically translate into improved construction of probabilistically checkable proofs and better hardness of approximation results.

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. The analyzes 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. Other standard ways of addressing the testing problem do not capture this intuition. We discuss the apparent benefits and limitations of our approach to the testing problem and contrast it to the ones found in the literature.

Keywords — Dual codes, MacWilliams Theorems, program checking, self-testing, linearity testing, low-degree testing, probabilistically checkable proofs.

1 Introduction

Over the years, researchers and software developers have proposed strategies to deal with the problem of software reliability. The software reliability problem is that of efficiently and effectively identifying software bugs. It arises in every computer programming task and its hard to deal with. No perfect solution for it is likely to be found.

* Part of this work was done while the author was a graduate student in the Dept. of Mathematics at MIT. Some parts of this paper appeared in the author's Ph.D thesis [Kiw96].

†Dept. de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, U. de Chile, mkiwi@dim.uchile.cl. Partially supported by FONDECYT No. 1960849 and by Fundación Andes. This work was also supported by an AT&T PhD Scholarship and NSF Grant CCR-9503322.

One way of approaching the software reliability problem, *program verification*, consists in a formal proof of correctness of the program's code. This proof is done only once, before the program is ever used. As pointed out by Rubinfeld [Rub90], the programs that can be verified in this form are usually quite simple. When a program is proven correct, the proof is often very involved, and thus, even more susceptible to error. Furthermore, automatizing the verification process has not been a fruitful endeavor.

Another traditional method for insuring software reliability has been *program testing*. This method ascertains the program's correctness on some chosen inputs by contrasting the program's result with that of a solution independently obtained. Program testing does not guarantee that the program is correct on a particular input. Moreover, as observed in [Rub90], the choice of input distribution on which to check the program is usually debatable, and the "independently obtained solutions" usually turn out to be generated by a previous version of the same program that is being replaced, or one that is written by different people (who are likely to make the same types of errors) and thus is not independent at all.

The theory, pioneered by Blum et al. [BK89], on *self-checking programs*, and Blum, Luby, and Rubinfeld [BLR90], on *self-testing/correcting programs*, proposes an alternative approach to the problem of assuring software reliability. It is in this context that the testing problem arises. 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 sequence of the form $(\mathcal{F}_n, \mathcal{T}_n, \mathcal{D}_n)_{n\geq 1}$ where $\mathcal{F}_n$ is a collection of functions (with common domain and range), $\mathcal{T}_n$ is a collection of functionals mapping $\mathcal{F}_n$ to $\{accept, reject\}$, and $\mathcal{D}_n$ is a probability distribution over $\mathcal{T}_n$. We let $\mathcal{P}_n$ denote the collection of functions $f \in \mathcal{F}_n$ for which $T(f)$ equals *accept* for every T which is assigned a positive probability by the distribution $\mathcal{D}_n$. We assume from now on that $\mathcal{P}_n$ is nonempty.

Analyzing a test consists in explaining the relation between the following two quantities when the function f, with domain $\mathsf{Dom}(f)$, varies over the $\mathcal{F}_n$'s:

$\mathsf{Rej}(f) \stackrel{\text{def}}{=} \mathsf{Pr}_{T\sim\mathcal{D}_n}[T(f) = reject]$ — the probability that the test rejects f,

$\mathsf{Dist}(f) \stackrel{\text{def}}{=} \min\{\mathsf{Pr}_{u\in_R Dom(f)}[f(u) \neq g(u)] \mid g \in \mathcal{P}_n\}$ — minimum (relative) distance of f to its closest function in $\mathcal{P}_n$.

Of particular importance is to provide a *lower bound* for the test, i.e. to determine a function $\varphi\colon [0,1] \to [0,1]$ such that $\mathsf{Rej}(f) \geq \varphi(\mathsf{Dist}(f))$ holds (possibly assimptotically) for every n and $f \in \mathcal{F}_n$. The quality of the lower bound depends on the function φ. The reason being, that one is usually interested in determining the largest value $\mathsf{Dist}(f)$ can take given that $\mathsf{Rej}(f)$ is at most $\epsilon \in [0,1]$.

There always is an undercurrent in the testing problem. In it, we assume we have oracle access to a function $f \in \mathcal{F}_n$. (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 $\mathcal{P}_n$". We do not trust this claim. We verify it, probabilistically, by performing the test $(\mathcal{F}_n, \mathcal{T}_n, \mathcal{D}_n)_{n\geq 1}$ as follows: we sample according to $\mathcal{D}_n$ a functional T in $\mathcal{T}_n$; 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 the $\mathcal{P}_n$'s always pass the test. We think of the functions belonging to $\mathcal{P}_n$, for some n, as those that satisfy a property of interest. A natural question arises: *with what probability does the test reject the claim concerning f when* $\mathsf{Dist}(f)$ *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. One can easily envision using known tests as an aid in checking the correctness of libraries of mathematical functions, and of essential components of software like Matlab® and Mathematica®.

Research in self-testing/correcting programs also had unexpected theoretical consequences. Indeed, the theory of self-testing programs is part of the research that led to the PCP theorem of Arora, Lund, Motwani, Sudan, and Szegedy [ALM+92]. Testing is an essential component in the known constructions of probabilistically checkable proofs (PCPs) and the derivation of hardness of approximation results. Informally, a PCP is a proof that can be verified (probabilistically) to a significant degree of confidence with a small number of queries (spot-checks). A PCP has to be extremely redundant since every part of the proof has to witness the correctness or incorrectness of the whole. Otherwise a probabilistic verification that performs a small number of spot-checks would not detect incorrect proofs with a significant confidence. The error-detection properties that PCPs exhibit are inherited from the techniques used to construct them. To explain this, recall that since the work of Arora and Safra [AS92], PCPs have been built by recursion. Each level of the recursion uses a distinct form of error-correcting code. Correct encodings are viewed as representations of functions that satisfy a pre-specified property. Among the typical properties satisfied by correct encodings are linearity, multi-linearity, low-individual degree, low-total degree, etc. Thus, a central problem in the construction of PCPs is to probabilistically check (test) these properties with as few oracle queries as possible. Testing that a given function satisfies one of these properties is what is often referred to as *low-degree testing*.

Currently we have no general framework which allows us to naturally formulate tests for checking whether a 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 clear if many of the known analyzes are tight. Good analyzes are, for many reasons, worth endeavoring for. Improved analyzes usually yield better hardness of approximation results. Moreover, they typically enable better construction of PCPs.

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 Previous and Related Work

SELF-TESTING: This concept was introduced in [BLR90]. *Self-testing* is a way of using a program purported to compute a function to check that its computations are indeed correct, but without recomputing the function, nor checking the program's code. The concept of *property testing* was introduced in [GGR96]. It is related to that of self-testing. In property testing we are interested in determining whether a program satisfies a pre-specified property (equivalently, if a program belongs to a specific class of programs). Self-testing can be seen as a particular instance of property testing (one where there is only one function satisfying the property of interest). In this paper we do not make a distinction between self-testing and property testing. We simply refer to both tasks as testing.

There is a considerable body of literature dealing with self-testing issues. We refer the interested reader to the survey of Blum and Wasserman [BW94] and the thesis of Rubinfeld [Rub90]. For pointers to more recent work in the area see [Rub94, EKS96]. The work of Goldreich, Goldwasser, and Ron [GGR96] which introduces the concept of property testing also suggests exciting new directions of research. For recent developments concerning issues in property testing see [GR97].

SELF-CORRECTION: This concept was also introduced in [BLR90]. To explain it, assume we have a program that works correctly in a large fraction of its (finite) input space. *Self-correction* is a way

of using this same program to find the correct value everywhere (with high probability). Research addressing the self-testing problem usually also addresses self-correction issues. Thus, the reader interested in this topic is referred to [Rub90, BW94].

LOW-DEGREE TESTING: In the low-degree testing problem we are given an oracle for a function $f\colon F^n \to F$, where F is say a finite field. We are also given a positive integer d. Below we briefly describe some instances of this problem that have been studied in the literature. In the low *individual degree* testing problem we are asked to determine whether f is close to some polynomial of degree d in each of its n variables. When specialized to the case of $d = 1$, this task is referred to as *multi-linearity testing*. In the *linearity* testing problem we are asked to determine whether f is close to some homogeneous polynomial of total degree $d = 1$. In the *low total-degree* testing problem we are asked to determine whether f is close to some polynomial of total degree d in its n variables. Linearity tests were studied by [BLR90, BGLR93, BS94, BCH+95]. Multi-linearity tests were studied by [BFL90, FGL+91]. Low individual degree tests were studied by [BFLS91, AS92, FHS94, PS94]. Total degree tests were studied by [GLR+91, ALM+92, RS93, FS95, AS97].

HARDNESS OF APPROXIMATION RESULTS: Since the development of the theory of NP completeness it has been known that determining the optimum of many natural optimization problems is NP-hard. However, it was noted that good approximations would suffice for many applications. In fact, for several optimization problems good approximations were found. For many NP-optimization problems, however, no progress was made either on the approximation or the intractability front. This changed when Feige, Goldwasser, Lovász, Safra, and Szegedy [FGL+91] showed that, under certain complexity theoretic assumptions, reasonable approximations of the clique number of a graph cannot be obtained efficiently. Their argument, coupled with PCP techniques, has been exploited to show the intractability of approximating a wide range of optimization problems. In some cases tight results have been proved.

One of the more pragmatic reasons for obtaining improved analysis of some of the low-degree tests is that they typically translate into improved hardness of approximation results. For example, usage of the linearity test in the the derivation of hardness of approximability results for MaxSNP problems, begins in [ALM+92] and continues in [BGLR93, BS94, BGS95]. The NP-hardness of approximating Set Cover to within $\Omega(\log n)$ was deduced in [AS97] through an improved analysis of a low-degree test. For a comprehensive discussion of hardness of approximation results and pointers to the literature the reader is referred to [BGS95].

1.2 Main Contributions and New Techniques

The crux of this work is the framework which we propose in order to formulate and carry out the analyzes of tests. This framework establishes a connection between testing and the theory of dual 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. 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). To be precise, we literally deduce the following very natural way of checking whether the function $f\colon F_q^n \to F_q$ is linear (F_q prime field of size q)

> 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 $\mathsf{Rej}(f)$ denote the probability that the above test rejects f and $\mathsf{Dist}(f)$ the distance between f and its closest linear function. Our main results concerning the EL test establish that for every $f\colon F_q^n \to F_q$,

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

and,

$$\mathsf{Rej}(f) \quad \geq \quad \mathsf{Dist}(f) - O\left(\frac{1}{q^n}\right), \tag{2}$$

where both summations are over all linear functions $l\colon F_q^n \to F_q$.

The first of the above expressions is essentially an equality. Thus, it captures the intuition that the probability with which a function fails the EL test depends on how far away that function is from *each* linear function. This intuition is not elicited by the standard arguments that have been used to analyze known tests.[1]

Besides the EL test we focus attention on the low-degree tests. To analyze these tests we derive a few general results. Typically, the tests' analyzes we provide proceed in two steps. In the first step 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. (1) 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. (2) in the case of the EL test). All of our analyzes use classical tools from coding theory. In particular, our analyzes 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.

Connections between testing, PCPs, and coding theory have been pointed out before (see for example [Sud94]). 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 analyzes we provide are meaningful even when the rejection probability is very high. (For example, the lower bound in (2) for the EL test shows that when the rejection probability is $1-\epsilon$ then the distance from the oracle function to the space of linear functions is at most $1-\epsilon$.) As discussed below (in Section 1.3), the previous situation rarely occurs in the low-degree testing literature.

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 a test, the punctured Hadamard code test, whose purpose is to verify whether a function $f\colon S \subseteq F_2^n \to F_2$ is linear when restricted to S.

[1] In fact, the discrete Fourier analysis based study of the BLR linearity test over GF(2), due to Bellare et al. [BCH+95], elicits such intuition. The arguments put forth in this work reduce to discrete Fourier analysis arguments when the underlying group considered is GF(2). Thus, the techniques used in [BCH+95] are particular versions of, not different from, the techniques used in this work.

1.3 Discussion

The techniques discussed in this work seem to be especially well suited for achieving nontrivial lower bounds on the rejection probability of a test even when this probability is high. Other techniques that achieve similar results are due to Arora [Aro95] and Tardos [Tar95] 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 [Sud92] shows that such lower bounds can be obtained for the basic univariate test. Recently, in a significant breakthrough, Arora and Sudan [AS97] 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. Since the low-total degree test is a crucial component in the derivation of the NP = PCP$(\log n, 1)$ result of [ALM+92], Arora and Sudan derive from their result several complexity theoretic consequences (see also [RS97] for an alternative derivation of similar complexity theoretic results).

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 for this work's arguments to have a wider applicability. It would be specially interesting to derive an alternative proof of the results in [AS97] concerning the low-total degree test based on our coding theoretic approach to the testing problem.

1.4 Organization

In Section 2 we first present some basic notions from coding theory. We then describe the coding theoretic framework we propose for addressing the testing problem. Finally, we illustrate how to interpret known tests and formulate new ones in the mentioned framework.

In Section 3 we state several results from coding theory. In particular, we state the MacWilliams Theorem for Hamming weight enumerators of linear codes. We then derive our main technical contribution, that is the Duality Testing Lemma.

In Section 4 we use the Duality Testing Lemma to analyze several of the tests discussed in Section 2.

CONVENTIONS: For the remaining part of this work q will denote a prime number, F_q denotes GF(q), and $\gamma \stackrel{\text{def}}{=} q-1$. 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

As stated before, the main contribution of this work is a framework in which to interpret, formulate, and analyze tests. Given some familiarity with elementary terms from coding theory this framework can be easily explained. 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 develop our framework and establish a relationship between testing and coding theory, in particular the theory of weight distributions of dual codes. In Section 2.3 we cast known tests into our framework. We also formulate new tests for verifying particular function properties, most notably linearity over finite fields.

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 [MS77, PW72, vL92].

A *code* of *block length* N over the alphabet F is a subset $\mathcal{C}$ of F^N. We will only consider codes whose alphabet F is a finite field. The elements $x = (x_1, \ldots, x_N) \in \mathcal{C}$ are called *codewords*, and their *length* is said to be N. The support of a codeword x, denoted $\mathsf{supp}(x)$, is the set of coordinates where x is nonzero. Formally, $\mathsf{supp}(x) \stackrel{\text{def}}{=} |\{\, i \mid x_i \neq 0 \,\}|$.

If x and y are codewords in $\mathcal{C}$, then the *Hamming distance*, $\mathsf{d}(x, y)$, between x and y is equal to the number of components where x and y are distinct. Formally, $\mathsf{d}(x, y) \stackrel{\text{def}}{=} |\{\, i \mid x_i \neq y_i \,\}|$. The *Hamming weight* of codeword x is the number of nonzero components of x and is denoted $\mathsf{wt}(x)$. Formally, $\mathsf{wt}(x) \stackrel{\text{def}}{=} |\mathsf{supp}(x)|$.

Definition 2.1 The minimum distance of a code $\mathcal{C}$ is $\min\{\, \mathsf{d}(x, y) \mid x \in \mathcal{C}, y \in \mathcal{C}, x \neq y \,\}$. The minimum weight of a code $\mathcal{C}$, denoted $\mathsf{wt}(\mathcal{C})$, is $\min\{\, \mathsf{wt}(x) \mid x \in \mathcal{C}, x \neq 0 \,\}$. We adopt the convention that the empty code has minimum distance and minimum weight 0.

Definition 2.2 (Weight distribution) For a code $\mathcal{C}$ of block length N we let $A_i(\mathcal{C})$ be the number of codewords in $\mathcal{C}$ of weight i, and say that $(A_0(\mathcal{C}), \ldots, A_N(\mathcal{C}))$ is its weight distribution.

Definition 2.3 (Linear codes) A code $\mathcal{C}$ of block length N is linear if its a subspace of F^N. If $\mathcal{C}$ has dimension k then $\mathcal{C}$ is called an $[N, k]$ code.

For a linear code, the minimum distance is equal to the minimum weight.

Definition 2.4 (Dual code) If $\mathcal{C}$ is a code of block length N over the alphabet F, then its dual code $\mathcal{C}^\perp$ is the collection of y's in F^N such that for all $x \in \mathcal{C}$, $\langle x, y \rangle \stackrel{\text{def}}{=} \sum_{i=1}^{N} x_i \cdot y_i = 0$ (arithmetic over F).

Clearly, if $\mathcal{C}$ is an $[N, k]$ code, then $\mathcal{C}^\perp$ is an $[N, N-k]$ code.

Example 2.5 Let $\mathcal{H}_2 = \{\, 0000, 0101, 0011, 0110 \,\}$. Thus $\mathcal{H}_2$ has minimum distance 2 and is a $[4, 2]$ code. Also, observe that $\mathcal{H}_2^\perp = \{\, 0000, 1000, 0111, 1111 \,\}$.

2.2 Tests and Dual Codes

In this section we show that there is a strong relationship between testing and the theory of weight distributions of dual codes.

We begin by describing the scenario we will be considering. Suppose we have oracle access to a function $f\colon D \subseteq F^n \to F$, where F is a finite field. When we write f we sometimes mean the function $f\colon D \subseteq F^n \to F$ and other times the table for f, i.e. the tuple $(f(u) : u \in D)$ (the meaning being clear from context). Henceforth, let N denote the size of the domain of f.

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 $\mathcal{C}$ 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 $\mathcal{C} \subseteq F^N$ which are subspaces, i.e. linear codes of block length N over the alphabet F. We stress that this restriction is not essential, since the results of coding theory that we will use have been generalized to the case of nonlinear codes.

We associate to f the smallest linear code of block length N containing both f and every codeword in $\mathcal{C}$, i.e.

$$\mathcal{C}_f \stackrel{\text{def}}{=} \{\, \phi f + \theta g \,|\, g \in \mathcal{C} \text{ and } \phi, \theta \in F \,\}.$$

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

> GENERIC DUALITY TEST: *Assume we have oracle access to a function f: $D \subseteq F^n \to F$. Let $\mathcal{C}$ be a linear code of block length $|D|$. Randomly choose (according to some probability distribution) a codeword $g \in \mathcal{C}^{\perp}$. Then, accept if $g \in \mathcal{C}_f^{\perp}$, and reject otherwise.*

Observe that the test always accepts when f belongs to $\mathcal{C}$. Particular choices of the code $\mathcal{C}$ and the distribution by which dual codewords are sampled will give specific tests. Particular instances of the generic duality test will be illustrated in Section 2.3. Indeed, we will show that several tests studied in the literature are specific examples of the above described test. Furthermore, we will derive from the generic duality test a new way of testing for linearity over large fields.

There is an issue regarding the duality test that we need to address, namely its efficiency. 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 in 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. 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 theory of weight distributions of dual codes tells us 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 $\mathcal{C}_f$ deviates from that of $\mathcal{C}$. In particular, we gain some information about the minimum weight of $\mathcal{C}_f \setminus \mathcal{C}$. This minimum weight, normalized by the block length of $\mathcal{C}$, is the distance from f to the nearest function represented by a codeword in $\mathcal{C}$. 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 $\mathcal{C}$, the smaller the probability of sampling a codeword from $\mathcal{C}^{\perp}$ that is in $\mathcal{C}_f^{\perp}$.

2.3 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. It does not seem that this test can be cast into the framework discussed so far. But, we will show that the coding theoretic setting proposed in Section 2.2 for formulating and studying tests can be extended in order to capture the low total-degree test.

Let us begin by illustrating how our framework captures the simplest tests known.

2.3.1 Testing Hadamard-like Codes

One of the most basic questions that arises in testing is that of testing for linearity. In fact, the seminal paper of Blum et al. [BLR90] proposed a linearity test and provided an analysis of it. (For subsequent work on this problem see [BGLR93, BS94, BCH+95].) We now precisely describe the problem. Let G and H be finite groups, and recall that a function $f\colon G \to H$ is *linear* if $f(u) + f(v) = f(u+v)$ for all $u, v \in G$. (That is, f is a group homomorphism.) As usual, we are given oracle access to a function f mapping G to H. We want to test that f is close (in relative distance) to a linear function. We are charged for each oracle call. Blum, Luby, and Rubinfeld [BLR90] proposed 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 three problems related to linearity testing.

Testing Hadamard Codes: The case of linearity testing where $G = F_2^n$ and $H = F_2$ is of particular relevance in the construction of PCPs and the proof of hardness of approximation results (see [BCH+95] for a thorough discussion). For these specific groups, the framework of Section 2.2 suggests performing the following

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

We now give an equivalent interpretation of the preceding test. Let $\lambda = (\, \lambda_u : u \in F_2^n \,)$ be such that its support is $\{\, u, v, w \,\}$. Then, by definition of $\mathcal{C}^\perp$, we have that $\lambda \in \mathcal{C}^\perp$ if and only if $l(u) + l(v) + l(w) = 0$ for every linear function $l\colon F_2^n \to F_2$. Equivalently, $\lambda \in \mathcal{C}^\perp$ if and only if $u + v + w = \mathbf{0}$. Similarly, it can be shown that $\lambda \in \mathcal{C}_f^\perp$ if and only if $\lambda \in \mathcal{C}$ and $f(u) + f(v) + f(w) = 0$. Thus, the preceding test amounts to choosing distinct $u, v, w \in F_2^n$ such that $u + v + w = \mathbf{0}$ and verifying whether $f(u) + f(v) + f(w) = 0$. This is almost equivalent to performing the BLR test when the underlying groups are $G = F_2^n$ and $H = F_2$.[2]

We now discuss why in the Hadamard code test the sampling distribution only picks codewords of weight three. Note that both $\mathcal{C}^\perp$ and $\mathcal{C}_f^\perp$ always have one codeword of weight zero and no codeword of weight two. Since independent of f, both $\mathcal{C}^\perp$ and $\mathcal{C}_f^\perp$ have the same number of weight zero and two codewords, there is no point in sampling from those codewords. Also, note that $\mathcal{C}^\perp$ has only 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 $\mathcal{C}_f^\perp$ 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 $\mathbf{0}$, than bother to check whether this holds by choosing the weight one codeword in $\mathcal{C}^\perp$. We are left with the option of sampling codewords in $\mathcal{C}^\perp$ 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 $\mathcal{C}^\perp$ that have weight three.

The name of the test, namely Hadamard code test, is due to the fact that if we drop the first coordinate of each codeword of $\mathcal{C}$ we get the well known Hadamard code [MS77, Ch. 2.§3].

[2] 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 performs such a check for values of $u, v, u+v$ not necessarily distinct. But, this occurs 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.

TESTING PUNCTURED HADAMARD CODES: Suppose again that we are given oracle access to a function $f\colon S \subseteq F_2^n \to F_2$. Moreover, we want to test whether f is linear. The framework proposed in Section 2.2 suggests a natural way of addressing this problem. Indeed, perform the following test:

> PUNCTURED HADAMARD CODE TEST: *Let $\mathcal{C}$ be the code whose elements are of the form $(\, l(u) : u \in S \,)$ where $l\colon F_2^n \to F_2$ is linear. Randomly choose a dual codeword $\lambda \in \mathcal{C}^\perp$ of weight three. Then, accept if $\lambda \in \mathcal{C}_f^\perp$, and reject otherwise.*

Note that the Hadamard code test corresponds to the special case of the punctured Hadamard code test where $S = F_2^n$. Again, if we let $\lambda = (\, \lambda_u : u \in S \,)$ be such that its support is $\{\, u, v, w \,\} \subseteq S$. Then, by definition of $\mathcal{C}^\perp$, we have that $\lambda \in \mathcal{C}^\perp$ if and only if $l(u) + l(v) + l(w) = 0$ for every linear function $l\colon F_2^n \to F_2$. Equivalently, $\lambda \in \mathcal{C}^\perp$ if and only if $u + v + w = \mathbf{0}$. Similarly, it can be shown that $\lambda \in \mathcal{C}^\perp$ if and only if $\lambda \in \mathcal{C}$ 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 $f(u) + f(v) + f(w) = 0$.

The reasons why the sampling procedure's support is restricted to weight three codewords are similar to those given to justify the choice of sampling procedure in the Hadamard code test.

The name of the test is due to the fact that the code $\mathcal{C}$ is obtained from a Hadamard code by a process called *puncturing*, i.e. deleting one or more coordinates from each codeword of $\mathcal{C}$.

LINEARITY TESTING OVER FINITE FIELDS: Assume now we have oracle access to a function $f\colon F_q^n \to F_q$. Again, we are interested in determining whether f is linear. Since F_q^n and F_q are finite groups, we could use the BLR test. The framework presented in Section 2.2 naturally suggests performing the following alternative test:

> EXTENDED LINEARITY TEST: *Let $\mathcal{C}$ be the collection of words of the form $(\, l(u) : u \in F_q^n \,)$ where $l\colon F_q^n \to F_q$ is linear. Randomly choose a dual codeword $\lambda \in \mathcal{C}^\perp$ of weight three. Then, accept if $\lambda \in \mathcal{C}_f^\perp$, 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 $\mathcal{C}^\perp$, we have that $\lambda \in \mathcal{C}^\perp$ 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, $\lambda \in \mathcal{C}^\perp$ if and only if $\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. Similarly, $\lambda \in \mathcal{C}_f^\perp$ if and only if $\lambda \in \mathcal{C}^\perp$ 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 $\lambda_u u + \lambda_v v + \lambda_w w = \mathbf{0}$. 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 almost equivalent to the BLR test when the underlying groups are $G = \mathrm{GF}(2)^n$ and $H = \mathrm{GF}(2)$ (see discussion of the Hadamard code test). For the general case, we show in Section 4.1 that the EL test is provably more efficient than the BLR test at the cost of using a negligible amount of additional randomness.[3] Intuitively, the gain in efficiency comes from the fact that the EL test looks at the same constraints that the BLR test does, and more (by a factor of $\Theta(q^2)$).

[3] Nevertheless, it should be noted that the BLR test only requires that the function to be tested take values from one finite group into another finite group. In contrast, the EL test requires that the function to be tested take values from F_q^n into F_q, where F_q is a finite field. The finite group assumption is used in the context of program checking. However, the finite field assumption is the standard assumption in the PCP context.

2.3.2 Low-degree Testing

The Hadamard code test suffices to construct PCPs for proofs of membership in NP languages. The PCPs thus built are far from being the best known constructions. For efficiency reasons, researchers use low-degree tests to build better PCPs. The most important of these tests is the low total-degree test. The low total-degree test showed up for the first time in [ALM+92]. 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.

THE BASIC UNIVARIATE TEST: Here the scenario is such that we have oracle access to a function $f\colon F_q \to F_q$. We want to test whether f is a polynomial of total degree d. We now describe a very simple test that achieves this goal. Let $\mathcal{P}_d$ denote the set of polynomials from F_q to F_q of total degree d.

> BASIC UNIVARIATE TEST [SUD92]: *Randomly pick $d+2$ distinct points $x_0, \ldots, x_{d+1}$ in F_q. Then, accept if there exists a polynomial in $\mathcal{P}_d$ 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 $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q \,)$ where p ranges over $\mathcal{P}_d$. Randomly choose a dual codeword $\lambda \in \mathcal{C}^\perp$ of weight $d+2$. Then, accept if $\lambda \in \mathcal{C}_f^\perp$, and reject otherwise.*

Clearly, both tests always accept polynomials of degree d. We will show that both tests are equally likely to accept when they have access to the same oracle function. We start by proving some intermediate results.

Claim 2.6 Let $q > d+1$, $1 \leq j \leq d+2$, and $x_0, \ldots, x_{j-1}$ be distinct elements of F_q. The rank of the matrix

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_{j-1} \\ x_0^2 & x_1^2 & \ldots & x_{j-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^d & x_1^d & \ldots & x_{j-1}^d \end{bmatrix},$$

is $\min\{\, j, d+1 \,\}$.

Proof: Denote by M the matrix in the statement of the claim.

Assume $j \leq d+1$. Observe that the first j 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 j. 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 is j.

Assume now that $j = d+2$. In this case, the first $d+1$ columns of M form a Vandermonde matrix. Hence, the column rank of M is at least $d+1$. It follows that the rank of M is at least $d+1$. Moreover, it is at most $d+1$ since M has $d+1$ rows. The claim follows. ∎

The following three lemmas are consequences of the previous claim.

Lemma 2.7 Let $q > d+1$. Let $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q \,)$ where p ranges over $\mathcal{P}_d$. The code $\mathcal{C}^\perp$ does not have codewords of weight $j \in \{\, 1, \ldots, d+1 \,\}$.

Proof: Assume $\lambda = (\lambda_x : x \in F_q) \in \mathcal{C}^\perp$ is of weight $j \in \{1, \ldots, d+1\}$. Let $\{x_0, \ldots, x_{j-1}\}$ be the support of λ. By definition of $\mathcal{C}^\perp$, we have that $\lambda \in \mathcal{C}^\perp$ if and only if for every polynomial $p \in \mathcal{P}_d$ it holds that $\sum_{i=0}^{j-1} \lambda_{x_i} p(x_i) = 0$. But, every such p is a linear combination of the polynomials $1, x, \ldots, x^d$. It follows that $\lambda \in \mathcal{C}^\perp$ if and only if

$$\begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_{j-1} \\ x_0^2 & x_1^2 & \ldots & x_{j-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^d & x_1^d & \ldots & x_{j-1}^d \end{bmatrix} \begin{bmatrix} \lambda_{x_0} \\ \lambda_{x_1} \\ \vdots \\ \lambda_{x_{j-1}} \end{bmatrix} = \mathbf{0}. \tag{3}$$

Denote by M the matrix in the previous expression. By Claim 2.6 we know that the rank of M is j. Hence, all the columns of M are linearly independent. Thus, (3) does not have a nontrivial solution. It follows that λ has zero weight, a contradiction. ∎

Lemma 2.8 Let $q > d+1$. Let $\mathcal{C}$ be the code whose elements are of the form $(p(x) : x \in F_q)$ where p ranges over $\mathcal{P}_d$. Let $x_0, \ldots, x_{d+1}$ be distinct elements of F_q. There are exactly $\gamma = q - 1$ codewords in $\mathcal{C}^\perp$ with support $\{x_0, \ldots, x_{d+1}\}$.

Proof: Let $\lambda = (\lambda_x : x \in F_q) \in \mathcal{C}^\perp$ be such that it has support $\{x_0, \ldots, x_{d+1}\}$. An argument similar to the one used to prove Lemma 2.7 shows that $\lambda \in \mathcal{C}^\perp$ if and only if

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

Denote by M the matrix in the previous expression. By Claim 2.6 we know that the rank of M is $d+1$. It follows that the nucleus of M has dimension 1. Hence, the nucleus of M contains q elements of which γ are nonzero. Also by Claim 2.6 we know that every $d+1$ columns of M are linearly independent. Hence, every coordinate of a nonzero element in the nucleus of M is nonzero. This proves the claim and the lemma. ∎

Lemma 2.9 Let $q > d+1$. Let $f\colon F_q \to F_q$ and let $\mathcal{C}$ be the code whose elements are of the form $(p(x) : x \in F_q)$ where p ranges over $\mathcal{P}_d$. Let $x_0, \ldots, x_{d+1}$ be distinct elements of F_q. Then, there exists a polynomial in $\mathcal{P}_d$ that agrees with f on $x_0, \ldots, x_{d+1}$, if and only if, every dual codeword in $\mathcal{C}^\perp$ with support $\{x_0, \ldots, x_{d+1}\}$ belongs to $\mathcal{C}_f^\perp$.

Proof: Let $\lambda = (\lambda_x : x \in F_q) \in \mathcal{C}^\perp$ be such that it has support $\{x_0, \ldots, x_{d+1}\}$. An argument similar to the one used to prove Lemma 2.7 shows that $\lambda \in \mathcal{C}_f^\perp$ 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 Claim 2.6, 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 the latter 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 $\mathcal{P}_d$. The second claim says that $\lambda \in \mathcal{C}_f^{\perp}$ if and only if M_f is not full rank. The lemma follows. ∎

Assuming both versions of the basic univariate test access the same oracle function, Lemma 2.8 and Lemma 2.9 imply that they both accept with equal probability.

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 $\mathcal{C}^{\perp}$ and $\mathcal{C}_f^{\perp}$ always have one codeword of weight zero and no codewords of weight $1, \ldots, d+1$. Thus, independent of f, both $\mathcal{C}^{\perp}$ and $\mathcal{C}_f^{\perp}$ have the same number of weight $0, \ldots, d+1$ codewords. We are left with the option of sampling codewords in $\mathcal{C}^{\perp}$ 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 $\mathcal{C}^{\perp}$ are sampled.

It was observed in [Sud92] 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 described below.

THE $(d+2)$-POINT TEST: Here the scenario is such that we have oracle access to a function $f\colon F_q^n \to F_q$. 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 $\mathcal{P}_{d,n}$ 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 $\mathcal{P}_{d,1}$. 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 $\mathcal{L}_n$.

> $(d+2)$-POINT TEST [RS96]: *Randomly pick a line $L \in \mathcal{L}_n$ and $d+2$ distinct points $x_0, \ldots, x_{d+1} \in L$. Then, accept if there exists a univariate polynomial in $\mathcal{P}_d$ that agrees with f on $x_0, \ldots, x_{d+1}$, and reject otherwise.*

Remark 2.10 Observe that when $n=1$ the $(d+2)$-point test is simply the basic univariate test. Moreover, if we denote by $f|_L$ the restriction of f to the line $L \in \mathcal{L}_n$, then 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 $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q^n \,)$ where p ranges over $\mathcal{P}_{d,n}$. Randomly choose a dual codeword $\lambda \in \mathcal{C}^{\perp}$ of weight $d+2$. Then, accept if $\lambda \in \mathcal{C}_f^{\perp}$, 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. they are equally likely to accept the same oracle function.

First we need a generalization of Claim 2.6. In order to state it let $p_0(\cdot), \ldots, p_{m-}(\cdot)$, where $m = \binom{n+d}{d}$, 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 $\psi(x) \stackrel{\text{def}}{=} (\, p_0(x), \ldots, p_{m-1}(x) \,)^T$.

Claim 2.11 Let $q > d+1$, $1 \leq j \leq d+2$, and let $x_0, \ldots, x_{j-1}$ be distinct elements of F_q. The rank of the $\binom{n+d}{d} \times j$ matrix $[\psi(x_0) \ \ldots \ \psi(x_{j-1})]$ is $d+1$ if $j = d+2$ and $x_0, \ldots, x_{j-1}$ are collinear and j otherwise.

Proof: The proof is rather tedious and long. It is included in the appendix but can be omitted without loss. ∎

The following three lemmas follow from Claim 2.11 in the same way that Lemmas 2.7, 2.8 and 2.9 follow from Claim 2.6. Thus, their proofs are omitted.

Lemma 2.12 Let $q > d+1$. Let $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q \,)$ where p ranges over $\mathcal{P}_{d,n}$. The code $\mathcal{C}^{\perp}$ does not have codewords of weight $j \in \{\, 1, \ldots, d+1 \,\}$.

Lemma 2.13 Let $q > d+1$. Let $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q \,)$ where p ranges over $\mathcal{P}_{d,n}$. Let L be a line in $\mathcal{L}_n$. Let $x_0, \ldots, x_{d+1}$ be distinct elements of L. There are exactly $\gamma = q-1$ codewords in $\mathcal{C}^{\perp}$ with support $\{\, x_0, \ldots, x_{d+1} \,\}$.

Lemma 2.14 Let $q > d+1$. Let $f\colon F_q^n \to F_q$ and let $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q^n \,)$ where p ranges over $\mathcal{P}_{d,n}$. Let L be a line in $\mathcal{L}_n$. Let $x_0, \ldots, x_{d+1}$ be distinct elements of L. There exists a univariate polynomial in $\mathcal{P}_d$ that agrees with f on $x_0, \ldots, x_{d+1}$, if and only if, every dual codeword in $\mathcal{C}^{\perp}$ with support $\{\, x_0, \ldots, x_{d+1} \,\}$ belongs to $\mathcal{C}_f^{\perp}$.

Assuming that the two stated versions of the $(d+2)$-point test access the same oracle function, from Lemma 2.13 and Lemma 2.14 it follows that both versions 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 $f\colon F_q^n \to F_q$. 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 $L \in \mathcal{L}_n$ we let $P_{f,L}\colon L \subseteq F_q^n \to F_q$ 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 $L \in \mathcal{L}_n$. (That is, we can specify $L \in \mathcal{L}_n$ and in one step are returned and encoding of $P_{f,L}$.)

> LOW TOTAL-DEGREE TEST [ALM+92]: *Randomly pick a line $L \in \mathcal{L}_n$ 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 $P_{f,L}$, $L \in \mathcal{L}_n$. 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 $\mathcal{C} \subseteq F_q^N$ be a code whose codeword's coordinates are indexed by the elements of F_q^n. Let $\mathcal{C}_L$ denote the collection of codewords of $\mathcal{C}$ whose support is contained in L and where in addition the codeword's coordinates indexed by elements of $F_q^n \setminus L$ have been omitted (punctured). Formally, if $\varphi_L\colon F_q^N \to F_q^q$ is such that $\varphi_L((\, x_u : u \in F_q^n \,)) \stackrel{\text{def}}{=} (\, x_u : u \in L \,)$ then

$$\mathcal{C}_L \stackrel{\text{def}}{=} \{\, \varphi_L(x) \mid x \in \mathcal{C} \text{ s.t. } \mathsf{supp}(x) \subseteq L \,\}.$$

Note that $\mathcal{C}_L$ is a code of block length $|L| = q$. Instead of $(\mathcal{C}_f)_L$ or $(\mathcal{C}_f^{\perp})_L$ we write $\mathcal{C}_{f,L}$ and $\mathcal{C}_{f,L}^{\perp}$ respectively.

Recall that the weight of a code $\mathcal{C}$ is the minimum weight codeword in $\mathcal{C}$, and is denoted $\mathsf{wt}(\mathcal{C})$. We define the relative minimum weight of a code $\mathcal{C}$, denoted $\rho\mathsf{wt}(\mathcal{C})$, as the weight of the code $\mathcal{C}$ divided by its block length.

The following is a simple but key observation.

Lemma 2.15 Let $\mathcal{C}$ be the code whose elements are of the form $(\, p(x) : x \in F_q^n \,)$ where p ranges over $\mathcal{P}_{d,n}$. For every function $f\colon F_q^n \to F_q$

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

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

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

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 $\mathcal{C}_{f,L} \setminus \mathcal{C}$. ∎

Towards the end of this paper we discuss more in length the previous lemma.

3 Analyzes of Tests Through Coding Theoretic Arguments

We have seen that several known tests can be cast into the framework developed in Section 2.2. We have also shown that the same framework suggests a natural way of designing tests to verify a given function property.

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. Indeed, these tools arise form the well developed theory of weight distributions of dual codes. This theory describes the relation between the weight distribution of a code and its dual. This relationship is captured by the celebrated MacWilliams Theorems [Mac62]. We state a version of the MacWilliams Theorems in Section 3.2. 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. In Section 3.3 we prove some general useful results, in particular the Duality Testing Lemma. They will enable us to fully analyze some of the tests that fit into our framework.

3.1 Krawtchouk Polynomials

This section provides a handy reference for the basic notions concerning a family of polynomials called Krawtchouk polynomials. These polynomials play an important role in several areas of coding theory.

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 [MS77, Ch. 5.§7] and [vL92, Ch. 1.§2]. Krawtchouk polynomials are orthogonal polynomials. There is a well developed theory concerning orthogonal polynomials, see [Sze75].

Definition 3.1 For any positive integer N, the Krawtchouk polynomial $\mathsf{K}_k(x; N, q)$ is[4]

$$\mathsf{K}_k(x; N, q) \stackrel{\text{def}}{=} \mathsf{K}_k(x) = \sum_{j=0}^{k} (-1)^j \gamma^{k-j} \binom{x}{j} \binom{N-x}{k-j}, \qquad k = 0, 1, \ldots, N.$$

Clearly, $\mathsf{K}_k(x)$ is a polynomial of degree k in x.

Multiplying the Taylor series expansion of $(1+\gamma z)^{N-x}$ and $(1-z)^x$ we get the generating function for the sequence $\mathsf{K}_0(x), \mathsf{K}_1(x), \ldots$

$$\sum_{k=0}^{\infty} \mathsf{K}_k(x) z^k = (1+\gamma z)^{N-x}(1-z)^x. \tag{4}$$

If x is an integer with $0 \le x \le N$ the upper limit of summation can be replaced by N. An alternative expression for this generating function is

$$\sum_{k=0}^{\infty} \mathsf{K}_k(x) z^k = (1+\gamma z)^N \left(1 - \frac{qz}{1+\gamma z}\right)^x.$$

This expression shows that the constant coefficient of $\mathsf{K}_k(x)$ is $\binom{N}{k}\gamma^k$. From Definition 3.1 it follows that the leading coefficient of $\mathsf{K}_k(x)$ is $(-q)^k/k!$.

Differentiating (4) with respect to z, multiplying by $(1+\gamma z)(1-z)$, and equating coefficients of z^k shows that Krawtchouk polynomials satisfy the following three term recurrence:

$$(k+1)\mathsf{K}_{k+1}(x) = [(\gamma N - qx) - k(\gamma - 1)]\mathsf{K}_k(x) - \gamma(N-k+1)\mathsf{K}_{k-1}(x),$$

for every positive integer k, with initial values $\mathsf{K}_0(x) = 1$, $\mathsf{K}_1(x) = \gamma N - qx$.

Krawtchouk polynomials are called orthogonal polynomials due to the following *orthogonality relations*: for every nonnegative integers r, s,

$$\sum_{i=0}^{N} \binom{N}{i} \gamma^i \mathsf{K}_r(i) \mathsf{K}_s(i) = q^N \gamma^s \binom{N}{s} \delta_{r,s},$$

where $\delta_{r,s}$ is the Kronecker symbol, i.e. $\delta_{r,s} = 1$ if $r = s$, and 0 otherwise.

Rearranging the binomial coefficients in Definition 3.1 shows that if i and j are nonnegative integers, then

$$\gamma^i \binom{N}{i} \mathsf{K}_j(i) = \gamma^j \binom{N}{j} \mathsf{K}_i(j). \tag{5}$$

Thus, we can rewrite the orthogonality relations in the following useful way: for every nonnegative integer r, s,

$$\sum_{i=0}^{N} \mathsf{K}_r(i) \mathsf{K}_i(s) = q^N \delta_{r,s}. \tag{6}$$

These orthogonality relations let us express a polynomial $P(x)$ of degree k as a linear combination of $\mathsf{K}_0(x), \mathsf{K}_1(x), \ldots, \mathsf{K}_k(x)$. Formally

[4] The binomial coefficient $\binom{x}{j}$ denotes $x(x-1)\cdots(x-j+1)/j!$.

Theorem 3.2 (see [MS77, Ch. 5.§7]) If $P(x)$ is a polynomial of degree k in x then its Krawtchouk expansion is

$$P(x) \;=\; \sum_{j=0}^{k} c_j(P)\mathsf{K}_j(x)\;, \qquad \text{where } \; c_j(P) \;=\; q^{-N}\sum_{i=0}^{N} P(i)\mathsf{K}_i(j)\;.$$

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

3.2 MacWilliams Theorems

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

$$\mathsf{W}_{\mathcal{C}}(x,y) \;\stackrel{\text{def}}{=}\; \sum_{i=0}^{N} A_i(\mathcal{C})x^{N-i}y^i\;.$$

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

Theorem 3.3 (MacWilliams Theorem for linear codes, see [MS77, Ch. 5.§6]) If $\mathcal{C}$ is a linear code over F_q

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

Corollary 3.4 If $\mathcal{C}$ is a linear code over F_q of block length N, then for $k \in \{0,\ldots,N\}$

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

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

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

Since (4) implies that $(1+\gamma y)^{N-i}(1-y)^i = \sum_{k=0}^{N}\mathsf{K}_k(i)y^k$, 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. We then use them, in Section 4, to derive good lower bounds for several of the tests described in Section 2.3.

Proposition 3.5 Let $P(x)$ be a polynomial of degree k and let $c_j(P)$ be the j-th coefficient in the Krawtchouk polynomial expansion of $P(x)$. Then, for every linear code $\mathcal{C}$ of block length N

$$\sum_{j=0}^{k} c_j(P)A_j(\mathcal{C}^\perp) \;=\; \frac{1}{|\mathcal{C}|}\sum_{i=0}^{N} A_i(\mathcal{C})P(i) \;=\; \mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(g))\right].$$

Proof: The second equality is obvious. To prove the first identity recall that by Theorem 3.2 we know that $P(i) = \sum_{j=0}^{k} c_j(P)\mathsf{K}_j(i)$. Multiplying both sides of the previous expression by $A_i(\mathcal{C})$, summing over $i \in \{0, \ldots, N\}$, and dividing by $|\mathcal{C}|$ we get

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

Corollary 3.4 implies that the inner summation on the RHS is equal to $A_j(\mathcal{C}^\perp)$. ∎

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

Lemma 3.6 (Duality Testing Lemma) Let $P(x)$ be a polynomial of degree k and let $c_j(P)$ be the j-th coefficient in the Krawtchouk polynomial expansion of $P(x)$. Then, for any linear code $\mathcal{C}$ of block length N and every $f \in F_q^N$

$$\sum_{j=0}^{k} c_j(P)\left(A_j(\mathcal{C}^\perp) - A_j(\mathcal{C}_f^\perp)\right) \quad = \quad \frac{\gamma}{q}\left(\mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(g)) - P(\mathsf{wt}(f-g))\right]\right).$$

Proof: Two applications of Proposition 3.5 yield

$$\sum_{j=0}^{k} c_j(P)\left(A_j(\mathcal{C}^\perp) - A_j(\mathcal{C}_f^\perp)\right) \quad = \quad \mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(g))\right] - \mathsf{E}_{g\in_R\mathcal{C}_f}\left[P(\mathsf{wt}(g))\right].$$

From the definition of $\mathcal{C}_f$ and since $\mathcal{C}$ is a linear code, $\{\phi f + \theta g \mid g \in \mathcal{C} \text{ and } \phi, \theta \in F\}$. Equivalently, $\mathcal{C}_f = \{\phi f - \theta g \mid g \in \mathcal{C} \text{ and } \phi, \theta \in F,\ \phi \neq 0\} \cup \mathcal{C}$. Moreover, if $\phi \neq 0$ then $\mathsf{wt}(\phi f - \theta g) = \mathsf{wt}(f - \frac{\theta}{\phi}g)$. Thus, $\mathsf{E}_{g\in_R\mathcal{C}_f}\left[P(\mathsf{wt}(g))\right] = \frac{\gamma}{q}\mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(f-g))\right] + \frac{1}{q}\mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(g))\right]$. The lemma follows. ∎

Corollary 3.7 Let $P(x)$ be a polynomial of degree k and $\mathcal{C}$ be a linear code of block length N. If $\mathcal{C}^\perp$ does not have codewords of weight $i \in \{1, \ldots, k\}$ then, for every $f \in F_q^N$

$$\mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(f-g))\right] \quad = \quad \mathsf{E}_{g\in_R\mathcal{C}}\left[P(\mathsf{wt}(g))\right].$$

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

Example 3.8 Let $f\colon F_q^n \to F_q$ be arbitrary. Let $\mathcal{C}$ be the code whose elements are of the form $(p(x) : x \in F_q^n)$ where p ranges over the n-variate polynomials over F_q^n of total degree d. From Lemma 2.12 we know that $\mathcal{C}^\perp$ does not have codewords of weight $1, \ldots, d+1$. Hence, by Corollary 3.7, for every polynomial $Q(x)$ of degree at most $d+1$

$$\mathsf{E}_{g\in_R\mathcal{C}}\left[Q\left(\mathsf{wt}(f-g)\right)\right] \quad = \quad \mathsf{E}_{g\in_R\mathcal{C}}\left[Q\left(\mathsf{wt}(g)\right)\right].$$

Note that $\mathsf{wt}(f-g)$ and $\mathsf{wt}(g)$ divided by the block length of $\mathcal{C}$ is $\mathsf{Pr}_{u\in_R F_q^n}\left[f(u) \neq g(u)\right]$ and $\mathsf{Pr}_{u\in_R F_q^n}\left[g(u) \neq 0\right]$ respectively. Thus, for every polynomial $Q(x)$ of degree at most d,

$$\sum_{p} Q\left(\mathsf{Pr}_{u\in_R F_q^n}\left[f(u) \neq p(u)\right]\right) \quad = \quad \sum_{p} Q\left(\mathsf{Pr}_{u\in_R F_q^n}\left[p(u) \neq 0\right]\right),$$

where the summation is over all n-variate polynomials over F_q^n of total degree at most d. (Observe that the RHS of the latter equality does not depend on f.)

Remark 3.9 Let the random variable X be distributed according to a *Binomial* $(N, \gamma/q)$.[5] Observe that by (5), we have the following identities for the coefficients in the Krawtchouk expansion of the polynomial P:

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

Because of this and other reasons that will later become apparent we recall some facts about the moments about the mean of the binomial distribution. The moment generating function of $X-\mu$, where $\mu = \gamma N/q$, is given by

$$\Phi(t) \;\stackrel{\text{def}}{=}\; \mathsf{E}\left[\,e^{t(X-\mu)}\,\right] \;=\; e^{-t\mu}\left(pe^t + 1 - p\right)^N.$$

If k is a nonnegative integer, then the k-th moment about the mean of random variable X is $\nu_k \stackrel{\text{def}}{=} \mathsf{E}\left[\,(X-\gamma N/q)^k\,\right]$. Moreover, ν_k equals the k-th derivative of $\Phi(t)$ evaluated at $t=0$. Thus, $\nu_0 = 1$, $\nu_1 = 0$, $\nu_2 = N\gamma/q^2$, $\nu_3 = -N\gamma(\gamma-1)/q^3$, $\nu_4 = 3N^2\gamma^2/q^4 + N\gamma\left(1-6\gamma/q^2\right)/q^2$, and $\nu_5 = -10N^2\gamma^2(\gamma-1)/q^5 - N\gamma(\gamma-1)\left(1+12\gamma/q^2\right)/q^3$.

4 Applications

In this section we analyze several of the instances of the generic duality test that where presented in Section 2.3. First, we need a notion of distance between two functions f and g defined over the same domain. We let the distance between f and g, denoted $\mathsf{Dist}(f,g)$, be the fraction of places in which f and g differ. Formally, $\mathsf{Dist}(f,g) \stackrel{\text{def}}{=} \mathsf{Pr}_u\left[\,f(u) \neq g(u)\,\right]$, where the probability is taken over the elements in the domain of f. A key feature of all the analyzes we present is that 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. This feature of our analyzes is shared with that given in [BCH+95] for the Hadamard code test.

4.1 Hadamard-like code test

Hadamard Code & Punctured Hadamard Code Test: As pointed out before, implementing the Hadamard code test is almost equivalent to performing the BLR test when the oracle function's domain and range are F_2^n and F_2 respectively. This latter test was exhaustively analyzed in [BCH+95] using discrete Fourier analysis techniques. Similarly, given access to the function $f\colon S \subseteq F_2^n \to F_2$ 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 $u+v \in S$.[6] This latter test was analyzed in [Kiw96, Ch. 2] using arguments similar to those of [BCH+95] also based in discrete Fourier analysis techniques. Both of the discrete Fourier analysis based analyzes are very elegant. The coding theoretic approach taken in this work generalizes the discrete Fourier analysis technique introduced by Bellare et al. [BCH+95]. In fact, the realization that extending the arguments in [BCH+95] required results from coding theory was the starting point for this work.

[5] The distribution *Binomial* (m,p) corresponds to the distribution of the sum of m independent identically distributed $\{0,1\}$-random variables with expectation p.

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

We refer the interested reader to [BCH+95] and [Kiw96, Ch. 2] for the discrete Fourier analysis based studies of the the two mentioned tests which are essentially equivalent to the Hadamard code and the punctured Hadamard code test. Both studies can be rewritten in the language used in this work. In the case of the punctured Hadamard code this yields the following

Lemma 4.1 Let $S \subseteq F_2^n$ and define

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

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

$$\mathsf{Rej}(f) \;=\; \frac{1}{2}\left(1 - \frac{1}{\varphi(S)}\sum_l (1 - 2\,\mathsf{Dist}_S(f,l))\right) - O\left(\frac{1}{|S|}\right) \;\geq\; \frac{1}{\varphi(S)}\left(\mathsf{Dist}_S(f) - \frac{1}{2}\right) + \frac{1}{2} - O\left(\frac{1}{|S|}\right),$$

where the summations are over the linear functions $l\colon F_2^n \to F_2$, $\mathsf{Dist}_S(f,l) = \mathsf{Pr}_{u \in_R S}[\,f(u) \neq l(u)\,]$, and $\mathsf{Dist}(f)$ is the distance from f to the nearest linear function $l\colon S \subseteq F_2^n \to F_2$.

A consequence of the Summation Lemma [BCH+95, Kle94] 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 of being a subspace. Thus, the closer the set S is to a subspace, the better the lower bound of Lemma 4.1. 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.

Since the Hadamard code test is a particular version of the punctured Hadamard code test we have the following:

Corollary 4.2 For every $f\colon F_2^n \to F_2$ the probability that the Hadamard code test rejects f is

$$\mathsf{Rej}(f) \;=\; \frac{1}{2}\left(1 - \sum_l (1 - 2\,\mathsf{Dist}(f,l))\right) - O\left(\frac{1}{|F_2|^n}\right) \;\geq\; \mathsf{Dist}(f) - O\left(\frac{1}{|F_2|^n}\right),$$

where the summations range over all the linear functions $l\colon F_2^n \to F_2$, and $\mathsf{Dist}(f)$ is the distance from f to the nearest linear function $l\colon F_2^n \to F_2$.

Proof: Take $S = F_2^n$ in Lemma 4.1 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 $f\colon F_q^n \to F_q$. We let $\mathcal{C}$ be the collection of codewords of the form $(\,l(u) : u \in F_q^n\,)$ where $l\colon F_q^n \to F_q$ is linear. Note that there is a bijection between the linear functions from F_q^n to F_q and the elements of F_q^n. Indeed, the collection of functions $\{\,l_\alpha\colon F_q^n \to F_q \mid \alpha \in F_q^n, l_\alpha(x) = \sum_{i=1}^n \alpha_i x_i\,\}$ is the set of all F_q-valued linear functions over F_q^n. In particular it follows that the cardinality of $\mathcal{C}$ is q^n.

Recall that the test consists in randomly choosing a dual codeword $\lambda \in \mathcal{C}^\perp$ of weight three, accepting if $\lambda \in \mathcal{C}_f^\perp$, and rejecting otherwise. As usual, we denote by $\mathsf{Rej}(f)$ the probability that the test rejects and by $\mathsf{Dist}(f)$ the distance from f to its nearest linear function.

Lemma 4.3 For every $f\colon F_q^n \to F_q$,

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

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

Proof: We restrict our discussion to the case where $f(\mathbf{0}) = 0$. The general case follows immediately.

Let $N = q^n$. Note that $A_0(\mathcal{C}_f^\perp) = A_0(\mathcal{C}^\perp)$, $A_1(\mathcal{C}_f^\perp) = A_1(\mathcal{C}^\perp)$. Now, let $P_k(x) = c_k\,(x - \gamma N/q)^k$, where $c_k = (-q)^k/k!$. Since $P_k(x)$ and $\mathsf{K}_k(x)$ have the same leading term, the k-th 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 and observing that $\mathsf{Rej}(f) = 1 - A_3(\mathcal{C}_f^\perp)/A_3(\mathcal{C}^\perp)$,

$$\begin{aligned}\mathsf{Rej}(f) &= \frac{\gamma}{q}\frac{1}{A_3(\mathcal{C}^\perp)}\mathsf{E}_{g\in_R\mathcal{C}}\left[P_3\left(\mathsf{wt}(g)\right) - P_3\left(\mathsf{wt}(f-g)\right)\right] \\ &\quad - c_2(P_3)\frac{1}{A_3(\mathcal{C}^\perp)}\left(A_2(\mathcal{C}^\perp) - A_2(\mathcal{C}_f^\perp)\right).\end{aligned} \tag{7}$$

From the recurrence relation for Krawtchouk polynomials

$$\mathsf{K}_2(x) = \frac{1}{2}[q^2(x-\gamma N/q)^2 + q(\gamma-1)(x-\gamma N/q) - \gamma N].$$

Hence, Remark 3.9 implies that $c_2(P_3) = \frac{c_3}{2\binom{N}{2}\gamma^2}\left(q^2\nu_5 + q(\gamma-1)\nu_4 - \gamma N\nu_3\right)$, where ν_i is the i-th moment about the mean of a *Binomial* $(N, \gamma/q)$. It follows that $|c_2(P_3)| = O(q)$. Furthermore, $A_2(\mathcal{C}_f^\perp) \le A_2(\mathcal{C}^\perp) \le N\gamma^2$ and $A_3(\mathcal{C}_f^\perp) \le A_3(\mathcal{C}^\perp) = N^2\gamma^3/3! - O(Nq^3)$. Thus, the second term in the RHS of (7) is $O(1/N) = O(1/q^n)$.

Note that, in $\mathcal{C}$ there is one codeword of weight 0 and $N-1$ codewords of weight $N\gamma/q$. Hence, $\mathsf{E}_{g\in_R\mathcal{C}}\left[P_3\left(\mathsf{wt}(g)\right)\right] = -c_3N^2\gamma^3/q^3$. Moreover, since $\mathsf{wt}(f-g) = N\mathsf{Dist}(f,g)$, we get that $\mathsf{E}_{g\in_R\mathcal{C}}\left[P_3\left(\mathsf{wt}(f-g)\right)\right] = c_3N^3\mathsf{E}_{g\in_R\mathcal{C}}\left[\left(\mathsf{Dist}(f,g) - \gamma/q\right)^3\right] = -c_3N^2\sum_{l\in\mathcal{C}}\left(\frac{\gamma}{q} - \mathsf{Dist}(f,l)\right)^3$. Putting everything together we conclude that

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

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

Lemma 4.4 For every $f\colon F_q^n \to F_q$, $\mathsf{Rej}(f) \ge \mathsf{Dist}(f) - O(1/q^n)$.

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 $N = q^n$. A randomly chosen linear function agrees with f in a fraction of at least $(1 - 1/N)/q$ points. Thus, $\mathsf{Dist}(f) \le \gamma/q$. Hence from Lemma 4.3 we get that

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

To conclude, it suffices to show that the following inequality holds: $\sum_{l\in\mathcal{C}}\left(\frac{\gamma}{q} - \mathsf{Dist}(f,l)\right)^2 \le (\gamma/q)^2$.

As in the proof of Lemma 4.3 let $P_k(x) = c_k\,(x - \gamma N/q)^k$, where $c_k = (-q)^k/k!$. Recall that the k-th coefficient in the Krawtchouk expansion of $P_k(x)$, i.e. $c_k(P_k)$, is equal to 1. Observe that $\sum_{l\in\mathcal{C}}\left(\frac{\gamma}{q} - \mathsf{Dist}(f,l)\right)^2 = \frac{1}{c_2N}\mathsf{E}_{g\in_R\mathcal{C}}\left[P_2(\mathsf{wt}(f-g))\right]$. Since $c_2(P_2) = 1$, $A_0(\mathcal{C}_f^\perp) = A_0(\mathcal{C}^\perp)$, $A_1(\mathcal{C}_f^\perp) = A_1(\mathcal{C}^\perp)$, and $A_2(\mathcal{C}_f^\perp) \le A_2(\mathcal{C}^\perp)$, by the Duality Testing Lemma, $\mathsf{E}_{g\in_R\mathcal{C}}\left[P_2(\mathsf{wt}(f-g))\right] \le \mathsf{E}_{g\in_R\mathcal{C}}\left[P_2(\mathsf{wt}(g))\right]$. In $\mathcal{C}$ there is one codeword of weight 0 and $N-1$ codewords of weight $N\gamma/q$.

Hence, $\mathsf{E}_{g\in_R\mathcal{C}}[P_2(\mathsf{wt}(g))] = c_2N\gamma^2/q^2$. Putting everything together proves the claimed inequality. ∎

The following lower bound complements Lemma 4.4. Its proof is an extension of an argument used in [BGLR93] in the obtaining of an improved lower bound for the BLR test when the underlying field is GF(2).

Proposition 4.5 For every $f\colon F_q^n \to F_q$, $\mathsf{Rej}(f) \geq 3\,\mathsf{Dist}(f)\left(1 - \frac{q}{\gamma}\mathsf{Dist}(f)\right) - O\left(\frac{1}{q^n}\right)$.

Proof: Let l be the closes linear function to f. Note that the zero function is the closest linear function to $f-l$ and that $\mathsf{Rej}(f) = \mathsf{Rej}(f-l)$. Thus, without loss of generality we may assume that l is the zero function.

Observe that $\mathsf{Rej}(f) = \mathsf{Pr}[\lambda_u f(u) + \lambda_v f(v) + \lambda_w f(w) \neq 0] - O(1/q^n)$, 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

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

It follows that,

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

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

4.2 Low-degree Tests

THE BASIC UNIVARIATE TEST: As pointed out in Remark 2.10, this test is a particular version of the $(d+2)$-point test. We omit the analysis of the basic univariate test and present below the analysis for the more general test.

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

In Remark 2.10 we observed that if $f|_L$ denotes the restriction of f to the line L, then the $(d+2)$-point test can be interpreted as randomly selecting a line $L \in \mathcal{L}_n$ and performing a basic univariate test on the oracle function $f|_L$. Hence, $\mathsf{Rej}(f) = \mathsf{E}_{L\in_R\mathcal{L}_n}[\mathsf{Rej}(f|_L)]$, where $\mathsf{Rej}(f|_L)$ denotes the probability that the basic univariate test rejects the claim "$f|_L$ is a univariate degree d polynomial".

Lemma 4.6 Let $q > d+1$. Let $Q(x)$ be the polynomial over the reals that at x takes the value $x\prod_{i=0}^{d}(x - \frac{q-i}{q})$. For every $f\colon F_q^n \to F_q$ the probability that the $(d+2)$-point test rejects f is

$$\mathsf{Rej}(f) \;=\; (-1)^{d+1}C_d \sum_{p\in\mathcal{P}_d} \mathsf{E}_{L\in_R\mathcal{L}_n}[Q(\mathsf{Dist}(f|_L, p))]\,,$$

where $C_d = q^{d+2}/\left((d+2)!\binom{q}{d+2}\right)$.

Proof: We know that $\mathsf{Rej}(f) = \mathsf{E}_{L \in_R \mathcal{L}_n}[\,\mathsf{Rej}(f|_L)\,]$. Hence, it suffices to prove the lemma for the case $n = 1$. In other words, we need to show that for the basic univariate test

$$\mathsf{Rej}(h) \;=\; (-1)^{d+1} C_d \sum_{p \in \mathcal{P}_d} Q(\mathsf{Dist}(h,p))\,, \qquad \text{for every } h\colon F_q \to F_q\,.$$

Let $\mathcal{C}$ be the collection of codewords of the form $(\,p(x) : x \in F_q\,)$ where p varies over $\mathcal{P}_d$. From Lemma 2.7 we know that $\mathcal{C}^\perp$ does not have codewords of weight $i \in \{\,1,\ldots,d+1\,\}$. It immediately follows that the same holds for $\mathcal{C}_h^\perp$. Hence, $A_i(\mathcal{C}^\perp) = A_i(\mathcal{C}_h^\perp)$ 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 $c\,x\prod_{i=0}^{d}(x-q+i)$, where $c = (-q)^{d+2}/(d+2)!$. Since $R(x)$ and $\mathsf{K}_{d+2}(x)$ 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

$$A_{d+2}(\mathcal{C}^\perp) - A_{d+2}(\mathcal{C}_h^\perp) \;=\; \frac{\gamma}{q}\mathsf{E}_{g \in_R \mathcal{C}}\,[\,R(\mathsf{wt}(g)) - R(\mathsf{wt}(h-g))\,]\,.$$

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

$$\mathsf{Rej}(h) \;=\; -\frac{\gamma}{q}\frac{1}{A_{d+2}(\mathcal{C}^\perp)}\mathsf{E}_{g \in_R \mathcal{C}}\,[\,R(\mathsf{wt}(h-g))\,]\,.$$

Since $\mathsf{wt}(h-g) = q\,\mathsf{Dist}(h,g)$, then $R(\mathsf{wt}(h-g)) = cq^{d+2}Q(\mathsf{Dist}(h,g))$. Thus, $\mathsf{E}_{g \in_R \mathcal{C}}\,[\,R(\mathsf{wt}(h-g))\,] = cq^{d+2}\mathsf{E}_{g \in_R \mathcal{C}}\,[\,Q(\mathsf{Dist}(h,g))\,] = cq\sum_{p \in \mathcal{P}_d} Q(\mathsf{Dist}(h,p))$, where the last equality follows since $|\mathcal{P}_d| = q^{d+1}$. The claim regarding the basic univariate test follows by recalling that from Lemma 2.7 we know that $A_{d+2}(\mathcal{C}) = \binom{q}{d+2}\gamma$. ∎

We currently are not able to use Lemma 4.6 to give a significant lower bound for the $(d+2)$-point test. In the case of the basic univariate test we can use Lemma 4.6 to re-derive a slightly improved version of a result due to Sudan [Sud92], albeit in a much more involved way. For completeness we include this result below.

Lemma 4.7 Let $q > d+1$. For every $f\colon F_q \to F_q$, the probability that the basic univariate test rejects the claim "f is a polynomial of degree d" is

$$\mathsf{Rej}(f) \;\geq\; \frac{q}{q-(d+1)}\,\mathsf{Dist}(f)\,.$$

Proof: Let $C_{j-2} = \frac{q^j}{\binom{q}{j}j!}$. From Lemma 4.6 we know that

$$\mathsf{Rej}(f) \;=\; C_d \sum_{p \in \mathcal{P}_d} Q'(\mathsf{Dist}(f,p))\,,$$

where $Q'(x)$ is the polynomial over the reals that at x takes the value $x\prod_{i=0}^{d}(\frac{q-i}{q}-x)$. Let $p \in \mathcal{P}_d$ and let $Q''(x)$ be the polynomial over the reals that at x takes the value $\prod_{i=0}^{d}(\frac{q-i}{q}-x)$. Note that there exists a $j \in \{\,1,\ldots,q\,\}$ such that $\mathsf{Dist}(f,p) = j/q$. Hence, $Q''(\mathsf{Dist}(f,p))$ is nonnegative. Moreover, since $\mathsf{Dist}(f,p) \geq \mathsf{Dist}(f)$, we conclude that $Q'(\mathsf{Dist}(f,p)) \geq \mathsf{Dist}(f)\,Q''(\mathsf{Dist}(f,p))$. Hence,

$$\mathsf{Rej}(f) \;\geq\; C_d\,\mathsf{Dist}(f) \sum_{p \in \mathcal{P}_d} Q''(\mathsf{Dist}(f,p))\,.$$

Since the degree of Q'' is $d+1$, by Example 3.8 we know that the summation in the previous inequality does not depend on f. Thus, assuming without loss of generality that $f \equiv 0$, $\mathsf{Rej}(f) \geq C_d \, \mathsf{Dist}(f) \sum_{p \in \mathcal{P}_d} Q''(\mathsf{Dist}(p,0))$. 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,

$$\mathsf{Rej}(f) \;\geq\; C_d \, \mathsf{Dist}(f) \, Q''(0) \;=\; \frac{C_d}{C_{d-1}} \, \mathsf{Dist}(f) \, .$$

The lemma follows. ∎

Remark 4.8 Lower bounds for the low total-degree test have more significant consequences (for a discussion of these consequences see [AS97]) than lower bounds for the $(d+2)$-point test. Thus, research has concentrated in analyzing the low total-degree test. The best known lower bounds for the $(d+2)$-point test follow from the fact, well known and easy to prove, that the probability that the $(d+2)$-point test rejects the claim "f belongs to $\mathcal{P}_{d,n}$" is always at least as large as the probability that the low total-degree test rejects the same claim.

THE LOW TOTAL-DEGREE TEST: We currently are not able to use the framework and tools developed in this paper to analyze this test. An alternative analysis of this test would be interesting. In particular one that would be simple and tight.

Our approach might yield such an alternative analysis. Indeed, recall that in Lemma 2.15 we established that if $\mathcal{C} = \{ \, (\, p(x) \, : \, x \in F_q^n \,) \, | \, p \in \mathcal{P}_{d,n} \, \}$, then for every function $f \colon F_q^n \to F_q$

- The minimum (relative) distance of f to its closest n-variate total degree d polynomial equals the relative minimum weight of the code $\mathcal{C}_f \setminus \mathcal{C}$. Formally, $\mathsf{Dist}(f) = \rho\mathsf{wt}(\mathcal{C}_f \setminus \mathcal{C})$.
- The probability that the low total-degree test rejects the claim "f is a total degree d polynomial" equals the expected (when L is randomly chosen in $\mathcal{L}_n$) relative minimum weight of the code $\mathcal{C}_{f,L} \setminus \mathcal{C}_L$. Formally, $\mathsf{Rej}(f) = \mathsf{E}_{L \in_R \mathcal{L}_n} [\, \rho\mathsf{wt}(\mathcal{C}_{f,L} \setminus \mathcal{C}_L) \,]$.

Moreover, observe that the weight distributions of $\mathcal{C}$ and $\mathcal{C}_L$ for every $L \in \mathcal{L}_n$ are known. We now show that the weight distribution of $\mathcal{C}_f$ and $\mathcal{C}_{f,L}$, $L \in \mathcal{L}_n$, are related. Indeed, $\mathcal{C}_f$ completely determines $\mathcal{C}_f^{\perp}$. Similarly, for every $L \in \mathcal{L}_n$, $\mathcal{C}_{f,L}$ completely determines $\mathcal{C}_{f,L}^{\perp}$. But, the codewords of $\mathcal{C}_f^{\perp}$ whose support is completely contained in a line in L can be placed in 1 to 1 correspondence with those codewords in $\mathcal{C}_{f,L}^{\perp}$, where L varies over $\mathcal{L}_n$. Hence, there is a relationship between the weight distributions of $\mathcal{C}_f$ and $\mathcal{C}_{f,L}$, $L \in \mathcal{L}_n$. It follows that the mathematical relationship between $\mathsf{Dist}(f)$ and $\mathsf{Rej}(f)$ cannot be arbitrary.

We have failed to exploit the above facts in order to derive an alternative analysis of the low total-degree test. Can it be done? If not, what additional results are needed to do so?

5 Acknowledgments

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.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 14–23, Pittsburgh, Pennsylvania, October 1992. IEEE.

[Aro95] S. Arora. Private communication, 1995.

[AS92] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 2–13, Pittsburgh, Pennsylvania, October 1992. IEEE.

[AS97] S. Arora and M. Sudan. Improved low-degree testing and its applications. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, Texas, October 1997. ACM. (To appear).

[BCH+95] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in chacaracteristic two. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 432–441, Milwaukee, Wisconsin, October 1995. IEEE.

[BFL90] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 16–25, St. Louis, Missouri, October 1990. IEEE.

[BFLS91] L. Babai, L. Fortnow, L. A. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 21–31, New Orleans, Louisiana, May 1991. ACM.

[BGLR93] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 294–304, San Diego, California, May 1993. ACM.

[BGS95] M. Bellare, O. Goldreich, and M. Sudan. Free bits and non-approximability. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pages 422–431, Milwaukee, Wisconsin, October 1995. IEEE.

[BK89] M. Blum and S. Kannan. Designing programs that check their work. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 86–97, Seattle, Washington, May 1989. ACM.

[BLR90] M. Blum, M. Luby, and R. Rubinfeld. Selt-testing/correcting with applications to numerical problems. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 73–83, Baltimore, Maryland, May 1990. ACM.

[BS94] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 184–193, Montréal, Québec, Canada, May 1994. ACM.

[BW94] M. Blum and H. Wasserman. Program result-checking: A theory of testing meets a test of theory. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 382–392, Santa Fe, New Mexico, November 1994. IEEE.

[EKS96] F. Ergun, S. R. Kumar, and D. Sivakumar. On testing without the generator bottleneck, 1996. Submitted to SIAM Journal on Computing.

[FGL+91] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pages 2–12, San Juan, Puerto Rico, October 1991. IEEE.

[FHS94] K. Friedl, Z. Hátsági, and A. Shen. Low-degree tests. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 57–64, Arlington, Virginia, January 1994. ACM.

[FS95] K. Friedl and M. Sudan. Some improvements to total degree testing. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems*, pages 190–198, Tel Aviv, Israel, January 1995. IEEE.

[GGR96] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 339–348, Burlington, Vermont, October 1996. IEEE.

[GLR+91] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 32–42, New Orleans, Louisiana, May 1991. ACM.

[GR97] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, El Paso, Texas, October 1997. ACM. (To appear).

[Kiw96] M. Kiwi. *Probabilistic checkable proofs and the testing of Hadamard-like codes.* PhD thesis, Massachusetts Institute of Technology, February 1996.

[Kle94] D. J. Kleitman. Private communication, 1994.

[Mac62] F. J. MacWilliams. *Combinatorial problems of elementary group theory.* PhD thesis, Harvard University, May 1962.

[MS77] F. J. MacWilliams and N. J. A Sloane. *The theory of error-correcting codes.* Elsevier Science Publisher B.V., first edition, 1977. (Eight impression: 1993).

[PS94] A. Polishchuk and D. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 194–203, Montréal, Québec, Canada, May 1994. ACM.

[PW72] W. W. Peterson and E. J. Weldon, Jr. *Error-correcting codes.* MIT Press, second edition, 1972. (Tenth printing: 1990).

[RS93] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials and their applications to program testing. Technical Report RC 19156, IBM, 1993.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal of Computing*, 25(1):252–271, April 1996.

[RS97] R. Raz and S. Safra. A sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29rd Annual ACM Symposium on Theory of Computing*, El Paso, Texas, October 1997. ACM. (To appear).

[Rub90] R. Rubinfeld. *A mathematical theory of self-checking, self-testing and self-correcting programs.* PhD thesis, University of California, Berkeley, 1990.

[Rub94] R. Rubinfeld. On the robustness of functional equations. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 288–299, Santa Fe, New Mexico, November 1994. IEEE.

[Sud92] M. Sudan. *Efficient checking of polynomials and proofs and the hardness of approximation problems.* PhD thesis, University of California, Berkeley, May 1992.

[Sud94] 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).

[Sze75] G. Szegö. *Orthogonal polynomials*, volume 23 of *Colloquium Publications.* American Mathematical Society, fourth edition, 1975. (Reprint: 1991).

[Tar95] G. Tardos. Manuscript, 1995.

[vL92] J. H. van Lint. *Introduction to coding theory*, volume 86 of *Graduate Texts in Mathematics.* Springer–Verlag, second edition, 1992.

A Appendix: Proof of Claim 2.11

Here, we let $p_1(\cdot), \ldots, p_{m-1}(\cdot)$, where $m = \binom{n+d}{d}$, denote all the nonzero monic monomials over F_q^n of total degree at most d. For $x \in F_q^n$ let $\psi(x) = (\, p_1(x), \ldots, p_{m-1}(x)\,)^T$. Given $x_0, \ldots, x_{j-1}$ distinct elements of F_q^n, define the following $(m-1) \times j$ and $m \times j$ matrix:

$$M^*(\{x_i\}_{i=0}^{j-1}) \stackrel{\text{def}}{=} \left[\begin{array}{ccc} \psi(x_0) & \ldots & \psi(x_{j-1}) \end{array} \right],$$

$$M(\{x_i\}_{i=0}^{j-1}) \stackrel{\text{def}}{=} \left[\begin{array}{ccc} 1 & \ldots & 1 \\ \psi(x_0) & \ldots & \psi(x_{j-1}) \end{array} \right].$$

We 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 $i \in \{\, 0, \ldots, j-1 \,\}$.

We want to prove that if $q > d+1$ and $x_0, \ldots, x_{j-1}$ are distinct elements of F_q^n, then

$$\mathsf{Rank}\ M(\{x_i\}_{i=0}^{j-1}) \;=\; \begin{cases} d+1 & \text{if } j = d+2 \text{ and } x_0, \ldots, x_{j-1} \text{ are collinear,} \\ j & \text{otherwise.} \end{cases}$$

First we need a technical claim.

Technical Claim A.1 [Rank Invariance] For every $k, l \in \{1, \ldots, n\}$, such that $k \neq l$, and for $\alpha \in F_q$ let $R_{k,l,\alpha}$ be the $n \times n$ matrix such that

$$(R_{k,l,\alpha})_{i,j} = \begin{cases} 1 & \text{if } i = j \\ \alpha & \text{if } i = k,\ j = l \\ 0 & \text{otherwise.} \end{cases}$$

Let $\{x_i\}_i$ be a collection of elements of F_q^n. The following holds:

- $\mathsf{Rank}\ M^*(\{R_{k,l,\alpha} x_i\}_i) = \mathsf{Rank}\ M^*(\{x_i\}_i)$.
- $\mathsf{Rank}\ M(\{x_i\}_i) = 1 + \mathsf{Rank}\ M^*(\{x_i - x_0\}_{i \neq 0})$.

Proof: Both identities follow by repeated application of the fact that adding to a row of a matrix the multiple of another row does not change the rank of the matrix. ∎

Proof: [Of Claim 2.11]

Let $x_i' = x_i - x_0$. From Technical Claim A.1 we get that

$$\mathsf{Rank}\ M(\{x_i\}_{i=0}^{j-1}) \;=\; 1 + \mathsf{Rank}\ M^*(\{x_i'\}_{i=1}^{j-1})\,.$$

For x in F_q^n let $\mathsf{Span}\{x\}$ denote the subspace of F_q^n generated by x. We can assume, without loss of generality, that there exists $1 = k_0 < k_1 < \ldots < k_r < k_{r+1} = j$ such that if $i \in \{k_p, \ldots, k_{p+1} - 1\}$ then $x_i' \in \mathsf{Span}\{x_{k_p}'\}$, and if $s \neq p$ then $x_i' \notin \mathsf{Span}\{x_{k_s}'\}$. Thus there exists a matrix R which is equal to a product of matrices of the form $R_{k,l,\alpha}$, and scalars t_i's such that if $i \in \{\, k_p, \ldots, k_{p+1} - 1 \,\}$

$$R x_i' \;=\; t_i e_p\,,$$

where e_p is the p-th element of of the canonical basis of F_q^n. Since the x_i''s are distinct and R is invertible (since it is a product of invertible matrices), we have that the t_i's are nonzero and $t_i \neq t_j$

for every $i \neq j$, $i, j \in \{ k_p, \ldots, k_{p+1} - 1 \}$. Hence,

$$\begin{aligned}
\mathsf{Rank}\ M^*(\{x_i'\}_{i=1}^{j-1}) &= \sum_{p=0}^{r} \mathsf{Rank} \begin{bmatrix} t_{k_p} & t_{k_p+1} & \cdots & t_{k_{p+1}-1} \\ t_{k_p}^2 & t_{k_p+1}^2 & \cdots & t_{k_{p+1}-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ t_{k_p}^d & t_{k_p+1}^d & \cdots & t_{k_{p+1}-1}^d \end{bmatrix} \\
&= \sum_{p=0}^{r} \mathsf{Rank} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ t_{k_p} & t_{k_p+1} & \cdots & t_{k_{p+1}-1} \\ t_{k_p}^2 & t_{k_p+1}^2 & \cdots & t_{k_{p+1}-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ t_{k_p}^{d-1} & t_{k_p+1}^{d-1} & \cdots & t_{k_{p+1}-1}^{d-1} \end{bmatrix} \\
&= \sum_{p=0}^{r} \min\{k_{s+1} - k_s, d\} \\
&= \begin{cases} d & \text{if } r = 0 \text{ and } j = d+2, \\ j-1 & \text{otherwise.} \end{cases}
\end{aligned}$$

Since $r = 0$ if and only if $x_0, \ldots, x_{j-1}$ are collinear, we are done. ∎