Testing Reed Muller Codes *

Noga Alon† Tali Kaufman‡ Michael Krivelevich§ Simon Litsyn¶ Dana Ron‖

March 15, 2004

Abstract

A code is locally testable if there is a way to indicate with high probability that a vector is far enough from any codeword by accessing only a very small number of the vector's bits. We show that the Reed-Muller codes of constant order are locally testable. Specifically, we describe an efficient randomized algorithm to test if a given vector of length $n = 2^m$ is a word in the r-th order Reed-Muller code $\mathcal{R}(r,m)$ of length $n = 2^m$. For given integer $r \geq 1$, and real $\epsilon > 0$, the algorithm queries the input vector $\mathbf{v}$ at $O(\frac{1}{\epsilon} + r2^{2r})$ positions. If $\mathbf{v}$ is at distance at least ϵn from the closest codeword, then the algorithm discovers it with probability at least $2/3$. On the other hand, if $\mathbf{v}$ is a codeword, then the algorithm never falsely establishes that it is not a codeword. Our result is essentially tight: any algorithm for testing $\mathcal{R}(r,m)$ must perform $\Omega(\frac{1}{\epsilon} + 2^r)$ queries.

*A preliminary version of this paper appeared in the Proceedings of the 7^{th} International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'2003), Lecture Notes in Computer Science 2764, 188–199.

†Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel and Institute for Advanced Study, Princeton, NJ 08540, USA. E-mail: `nogaa@post.tau.ac.il`. Research supported in part by a USA Israeli BSF grant and by a grant from the Israel Science Foundation

‡School of Computer Science, Tel Aviv University, Tel Aviv 69978 Israel. E-mail: `kaufmant@post.tau.ac.il`, This work is part of the author's Ph.D. thesis prepared at Tel Aviv University under the supervision of Prof. Noga Alon, and Prof. Michael Krivelevich.

§Department of Mathematics, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `krivelev@post.tau.ac.il`. Research supported in part by a USA Israeli BSF grant and by a grant from the Israel Science Foundation.

¶Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `litsyn@eng.tau.ac.il`. Research supported in part by a grant from the Israel Science Foundation.

‖Department of Electrical Engineering-Systems, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `danar@eng.tau.ac.il`. Research supported by the Israel Science Foundation (grant number 32/00-1).

1 Introduction

The problem of locally testing codes, first explicitly defined in [13, 19], has attracted a great deal of attention in the last decade due to its relation to the analysis of Probabilistically Checkable Proofs (PCP). Though the problem can be stated in purely coding terms, most of the publications on the topic employed definitions and notations non-standard for coding theory. In this paper we attempt to relate the testing of codes to the structure of the set of the minimum-weight vectors in the dual code. Using this approach we show how the Reed-Muller codes of constant order can be locally tested. Earlier, in the binary case, only testing of the first-order Reed-Muller codes was known.

Roughly speaking, the goal of locally testing a code is to have an efficient way to discover that an arbitrary vector does not belong to the code. Specifically, using a constant number of (random) accesses to the vector positions (queries), we want all non-codewords to be rejected by the algorithm with probability proportional to their distance from the code, while never rejecting a codeword. The general strategy is as follows. We pick a set of words from the dual code having constant (independent of the length of the code) weight. Each test will be a randomly chosen word from this set. To show that the code of length n is (locally) testable we have to prove that whatever binary vector having distance to the code more than ϵn, $\epsilon > 0$, is chosen, the probability that the randomly picked word from the chosen set in the dual code is orthogonal to it is sufficiently small.

We want to emphasize the inherent distinction of the considered problem from the standard problem of decoding. In decoding we always assume that the complexity is at least linear in the length of code, since we wish to use all bits of the received vector to make a decision about the transmitted codeword. In testing we treat reading a bit as an expensive operation, thus we want to minimize the number of bits which are accessed. This situation can be relevant in some applications where the information is stored, and our problem is to make a very fast assessment if the stored information has not been substantially corrupted.

Our results

In this work we consider the problem of (local) testing of r-th order Reed-Muller codes $\mathcal{R}(r,m)$ of length 2^m. To be more accurate, we in fact test the *shortened Reed-Muller code*, $\mathcal{R}(r,m)^*$, obtained from $\mathcal{R}(r,m)$ by choosing all codewords with the first bit equal to zero, and deleting this bit. The Reed-Muller code $\mathcal{R}(r,m)$ has minimum distance 2^{m-r}. The dual code of $\mathcal{R}(r,m)$ is the Reed-Muller code $\mathcal{R}(m-r-1,m)$. The dual code of the shortened Reed-Muller code $\mathcal{R}(r,m)^*$ is the *punctured* Reed-Muller code with parameters $m-r-1$ and m, obtained from $\mathcal{R}(m-r-1,m)$ by deleting the first bit of every codeword. The minimum distance of the punctured Reed-Muller code with parameters $m-r-1$ and m is $2^{r+1}-1$, and its minimum weight codewords are obtained from the minimum weight codewords of $\mathcal{R}(m-r-1,m)$, having the first bit equal to 1, by deleting this bit. The number of the minimum weight vectors is proportional to $2^{(r+1)m}$.

The testing algorithm is given a distance parameter ϵ, and an arbitrary vector from $\{0,1\}^{2^m-1}$. It is required to accept if the vector belongs to the code, and reject (with probability at least 2/3), if the vector is at Hamming distance at least $\epsilon \cdot 2^m$ from the closest codeword of $\mathcal{R}(r,m)^*$. To this end the algorithm can query the input vector on locations of its choice, where the goal is to minimize the query complexity of the algorithm (as a function of r, $1/\epsilon$, and m). The testing of the shortened Reed Muller code $\mathcal{R}(r,m)^*$ instead of the Reed Muller code $\mathcal{R}(r,m)$ is imposed mainly by historical reasons, to make our definition and result consistent with the previously treated case of testing first-order Reed-Muller (or Hadamard) codes. With minor changes our algorithm can be adapted to test the class of Reed Muller codes $\mathcal{R}(r,m)$. Our strategy is to pick a random minimum weight vector from the punctured code $\mathcal{R}(m-r-1,m)$, and to check if it is orthogonal to the tested vector. Clearly, this will always confirm orthogonality if the considered vector is from the code. However, we prove that if the tested vector is far enough from the code, with high probability the test will detect it, and give a lower bound on this probability.

More precisely, we describe and analyze an algorithm that tests whether an arbitrary vector from $\{0,1\}^{2^m-1}$ belongs to the *shortened Reed-Muller code* $\mathcal{R}(r,m)^*$, or is at distance at least $\epsilon \cdot 2^m$ from the closest codeword of $\mathcal{R}(r,m)^*$. The algorithm uses $O(1/\epsilon + r \cdot 2^{2r})$ queries (independent of the length of the codeword). As we show, an exponential dependency on r is unavoidable. This is in contrast to the case of testing the *shortened Generalized Reed-Muller code* $\mathcal{R}(r,m)^*$ over $GF(q)$ when q is sufficiently larger than r, where the sample complexity is polynomial in r. Our testing algorithm repeats the following check $\Theta(\frac{1}{2^r\epsilon} + r2^r)$ times: select, uniformly and at random, a minimum weight vector from the punctured code $\mathcal{R}(m-r-1,m)$, and check if it is orthogonal to the tested vector. If all checks succeed then it accepts, otherwise it rejects. Our choice of a minimum weight vector from the punctured $\mathcal{R}(m-r-1,m)$ corresponds to a random selection of an $(r+1)$-dimensional subspace in the affine geometry $\mathrm{AG}(m,2)$ (see for example [16, Chap. 12] for relevant definitions and terminology). In the case $r=1$ we deal with lines of the affine geometry $\mathrm{PG}(m,2)$.

Relation to low degree polynomials

The shortened Reed-Muller code $\mathcal{R}(r,m)^*$ has interpretation in terms of low degree polynomials. It can be defined as the set of evaluations of all polynomials $f:\{0,1\}^m \to \{0,1\}$ of degree at most r satisfying $f(0,\ldots,0)=0$.

Thus, the question of testing is equivalent to the following one: decide whether a binary function $f:\{0,1\}^m \to \{0,1\}$ is a polynomial of degree at most r satisfying $f(0,\ldots,0)=0$, or it should be modified on more than an ϵ-fraction of its domain to become a degree-r polynomial satisfying $f(0,\ldots,0)=0$. A function $f:\{0,1\}^m \to \{0,1\}$ that is a polynomial of degree at most r satisfying $f(0,\ldots,0)=0$, is simply a sum (modulo 2) of monomials each of them being a product of at most r variables, with the free term equal to zero.

Throughout this paper, we use the equivalence between the above two formulations. Namely, we actually derive a testing algorithm and evaluation of its performance for the second representation.

Related Prior Work

In earlier publications special attention was given to (what is usually called in the Property Testing literarure) Hadamard codes that coincide with the shortened first-order Reed-Muller codes. This corresponds to testing of multivariate linear polynomials. Blum, Luby and Rubinfeld [11] proved that the Hadamard code is locally testable. Their test is also known as a linearity test. In fact, our test can be viewed as an extension of the algorithm in [11]. Linearity testing has also been studied in subsequent papers [6, 4, 12, 7, 8].

The problem of testing multivariate low-degree polynomials has been studied quite extensively [4, 3, 14, 12, 19, 13, 2], and has important applications in the context of PCP. However, with the exception of the case $r=1$, that is, of linear functions, all results apply only to testing polynomials over fields that are of size larger than r (the degree bound). When the field F is sufficiently large, it is possible to reduce the problem of testing whether a function $f: F^m \to F$ is a multivariate degree-r polynomial to testing whether a function is a degree-r *univariate* polynomial, where the latter task is just based on interpolation. Namely, the test for f selects random *lines* in F^m (more precisely, in the finite projective geometry $\mathrm{PG}(m-1,|F|)$), and verifies that the restriction of f to each of these lines is a (univariate) polynomial of degree at most r. This reduction does not hold for small fields, and in particular for $GF(2)$, which is our focus.

Some related works deal with existence of general locally testable linear codes. In particular, in [15] a question is raised whether there exist good locally testable codes (that is, codes having non-vanishing rate and relative minimum distance). In [15] it is shown (by probabilistic arguments) that there exists a locally testable (linear) $[n,k]$ code that has almost constant rate (i.e. $n = k^{1+o(1)}$) and linear minimum distance (i.e., the minimum distance is $\Omega(n)$). This result holds even for the binary alphabet. In [10] the construction of [15] is derandomized. In [9] it is shown that local testing of random linear LDPC codes, which have linear minimum distance and constant rate, requires $\Omega(n)$ queries. In [5] it is proved that there are no locally testable cyclic codes that have constant rate and linear minimum distance.

2 Preliminaries

For any integer ℓ, we denote by $[\ell]$ the set $\{1,\ldots,\ell\}$. For any $r \in [m]$, let $\mathcal{P}_r$ denote the family of all Boolean functions over $\{0,1\}^m$ which are polynomials of degree at most r without a free term. That is, $f \in \mathcal{P}_r$ if and only if there exist coefficients $a_S \in \{0,1\}$, for every $S \subseteq [m], 1 \leq |S| \leq r$, such that

$$f = \sum_{S\subseteq[m],|S|\in[r]} a_S \cdot \prod_{i\in S} x_i \ , \tag{1}$$

where the addition is in $GF(2)$. In particular, $\mathcal{P}_1$ is the family of all linear functions over $\{0,1\}^m$, that is, all functions of the form $\sum_{i\in S} x_i$, where $S \subseteq [m]$.

For any two functions $f,g : \{0,1\}^m \to \{0,1\}$, the symmetric difference between f and g is $\Delta(f,g) \stackrel{\mathrm{def}}{=} \{y \in \{0,1\}^m : \ f(y) \neq g(y)\}$. The relative distance $\mathrm{dist}(f,g) \in [0,1]$ between f and g is: $\mathrm{dist}(f,g) \stackrel{\mathrm{def}}{=} |\Delta(f,g)|/2^m$. For a function g and a family of functions F, we say that g is ϵ-far from F, for some $0 < \epsilon < 1$, if, for every $f \in F$, $\mathrm{dist}(g,f) > \epsilon$. Otherwise it is ϵ-close to F.

A testing algorithm (tester) for $\mathcal{P}_r$ is a probabilistic algorithm, that is given query access to a function f, and a distance parameter ϵ, $0 < \epsilon < 1$. If f belongs to $\mathcal{P}_r$ then with probability at least $\frac{2}{3}$, the tester should accept f, and if f is ϵ-far from $\mathcal{P}_r$, then with probability at least $\frac{2}{3}$ the tester should reject it. If the tester accepts every f in $\mathcal{P}_r$ with probability 1, then it is a one-sided tester.

The following notation will be used extensively in this paper. Given a function $f : \{0,1\}^m \to \{0,1\}$, for $y_1,...,y_\ell \in \{0,1\}^m$ let

$$T_f(y_1,\ldots,y_\ell) \stackrel{\mathrm{def}}{=} \sum_{\emptyset\neq S\subseteq[\ell]} f\left(\sum_{i\in S} y_i\right) \ , \tag{2}$$

where the first sum is over $GF(2)$ and the second one is over $(GF(2))^m$, and let

$$T_f^{y_1}(y_2,\ldots,y_\ell) \stackrel{\mathrm{def}}{=} T_f(y_1,\ldots,y_\ell) + f(y_1) \ . \tag{3}$$

3 Characterization of Low Degree Polynomials over $\{0,1\}^n$

Claim 1 *A function f belongs to $\mathcal{P}_r$ (i.e., it is a polynomial of total degree at most r satisfying $f(0,0,\ldots,0)=0$), if and only if* for every $y_1,\ldots,y_{r+1}\in\{0,1\}^m$ *we have*

$$T_f(y_1,\ldots,y_{r+1})=0\ . \tag{4}$$

Proof: Since a polynomial from $\mathcal{P}_r$ can be viewed as a code word in the appropriate Reed-Muller code, the above characterization can be proved using known facts about its dual. For completeness we provide a direct, simple proof.

We first prove that if a function f belongs to $\mathcal{P}_r$ then $T_f(y_1,\ldots,y_{r+1})=0$ for every $y_1,\ldots,y_{r+1}\in\{0,1\}^m$.
As f is a sum of monomials of total degree at most r it suffices to show that for every monomial $M=\prod_{i\in I}x_i$, where $1\leq|I|\leq r$, $T_M(y_1,\ldots,y_{r+1})=0$ for every $y_1,\ldots,y_{r+1}\in\{0,1\}^m$. The number of linear combinations $\sum_{j=1}^{r+1}b_jy_j$, where $b_j\in\{0,1\}$, for which $M(\sum_{j=1}^{r+1}b_jy_j)=1$ is clearly the number of solutions of a linear system of $|I|$ equations in the $r+1$ variables b_j, and the trivial combination $b_j=0$ for all j is not one of the solutions. Therefore, this number of solutions (which is possibly zero) is divisible by $2^{r+1-|I|}$, showing that there is an even number of sets S satisfying $\emptyset\neq S\subset[r+1]$ such that $M(\sum_{i\in S}y_i)=1$. This implies that $T_M(y_1,\ldots,y_{r+1})=0$, as needed.
We next show that if $f=f(x_1,x_2,\ldots,x_m):\{0,1\}^m\mapsto\{0,1\}$ satisfies Equation (4) for every $y_1,y_2,\ldots,y_{r+1}\in\{0,1\}^m$, then $f\in\mathcal{P}_r$. Every function from $\{0,1\}^m$ to $\{0,1\}$ can be written uniquely as a polynomial over $GF(2)$:

$$f=\sum_{I\subset[m]}a_I\prod_{i\in I}x_i.$$

Our objective is to show that $a_\emptyset=0$ and that $a_I=0$ for all $|I|>r$. Taking $y_j=(0,0,\ldots,0)$ for every j we conclude, by (4), that $a_\emptyset=0$. Suppose, now, that there is a nonzero a_I with $|I|>r$. Take such an I of minimum cardinality, and assume, without loss of generality, that $I=[s]$ with $s\geq r+1$.
Let e_i denote the i-th unit vector in $\{0,1\}^m$, and define $y_1=e_1,y_2=e_2,\ldots,y_r=e_r$ and $y_{r+1}=e_{r+1}+\ldots+e_s$. Then the monomial $M=a_I\prod_{i\in I}x_i$ does not vanish on $\sum_{i=1}^{r+1}y_i$ and does vanish on $\sum_{i\in S}y_i$ for every $\emptyset\neq S\neq[r+1]$. Thus $T_M(y_1,\ldots,y_{r+1})\neq 0$. On the other hand, for any other monomial, say, $M'=\prod_{i\in I'}x_i$ with a nonzero coefficient in the representation of f, $T_{M'}(y_1,\ldots,y_{r+1})=0$. Indeed, if $|I'|\leq r$ this holds by the first part of the proof. Otherwise, by the minimality of I, $M'(\sum_{i\in S}y_i)=0$ for all $S\subset[r+1]$. Altogether this implies that $T_f(y_1,y_2,\ldots,y_{r+1})=1$, contradicting the assumption.
This completes the proof of Claim 1. ■

4 A Tester for Low Degree Polynomials over $\{0,1\}^m$

In this section we present and analyze a one-sided tester for $\mathcal{P}_r$.

Algorithm Test-$\mathcal{P}_r$

1. Uniformly and independently select $\Theta(\frac{1}{2^r\epsilon}+r2^r)$ groups of vectors. Each group contains $r+1$ uniformly selected random vectors $y_1,\ldots,y_{r+1}\in\{0,1\}^m$.

2. If for some group of vectors $y_1, \ldots, y_{r+1}$ it holds that $T_f(y_1, \ldots, y_{r+1}) \neq 0$, then reject, otherwise, accept.

We note that for the special case of $k = 1$, we obtain the linearity test of [11] which uniformly selects $O(1/\epsilon)$ pairs $y_1, y_2 \in \{0,1\}^n$, and verifies for each pair that $f(y_1) + f(y_2) = f(y_1 + y_2)$.

Theorem 1 *The algorithm* **Test-$\mathcal{P}_r$** *is a one-sided tester for $\mathcal{P}_r$ with query complexity $\Theta(\frac{1}{\epsilon} + r2^{2r})$.*

From the test definition and from Claim 1 it is obvious that if $f \in \mathcal{P}_r$, then the tester accepts. Thus, the crux of the proof is to show that if f is ϵ-far from $\mathcal{P}_r$, then the tester rejects with probability at least 2/3. Our proof is similar in structure to known derivations of the linearity test in [11], but requires some additional ideas. In particular, if f is the function tested, we can define a function g as follows. For any $y \in \{0,1\}^m$:

$$g(y) = 1 \text{ if } \Pr_{y_2,\ldots,y_{r+1} \in \{0,1\}^m}[T_f^y(y_2, \ldots, y_{r+1}) = 1] \geq 1/2 \text{ and } g(y) = 0 \text{ otherwise.} \tag{5}$$

Thus g is a kind of *majority* function. That is, for every vector $y \in \{0,1\}^m$, $g(y)$ is chosen to satisfy most of the equations $T_f^y(y_2, \ldots, y_{r+1}) = g(y)$. We also define

$$\begin{aligned} \eta &\stackrel{\text{def}}{=} \Pr_{y_1,\ldots,y_{r+1} \in \{0,1\}^m}[T_f(y_1, \ldots, y_{r+1}) \neq 0] \\ &= \Pr_{y_1,\ldots,y_{r+1} \in \{0,1\}^m}[T_f^{y_1}(y_2, \ldots, y_{r+1}) \neq f(y_1)] \; . \end{aligned} \tag{6}$$

Note that η is simply the probability that a single group of vectors $y_1, \ldots, y_{r+1}$ selected by the algorithm provides evidence that $f \notin \mathcal{P}_r$. We shall prove two claims. The first, and simpler claim (in Lemma 2), is that if η is small, then g is close to f. The second and more involved claim (in Lemma 5) is that if η is small, then g must belong to $\mathcal{P}_r$. This would suffice for proving the correctness of a slight variation on our algorithm that uses a larger sample size. In order to attain the sample complexity claimed in Theorem 1, we shall need to prove one more claim that deals with the case in which η is very small (see Lemma 6).

Lemma 2 *For a fixed function f, let g and η be as defined in Equations (5) and (6), respectively. Then,* $\text{dist}(f, g) \leq 2\eta$.

Proof: Recall that for every $y \in \{0,1\}^m$, $\Pr_{y_2,\ldots,y_{r+1} \in \{0,1\}^n}[T_f^y(y_2, \ldots, y_{r+1}) = g(y)] \geq 1/2$. Hence

$$\begin{aligned} \eta &= \Pr_{y, y_2,\ldots,y_{r+1} \in \{0,1\}^m}[T_f^y(y_2, \ldots, y_{r+1}) \neq f(y)] \\ &= \frac{1}{2^m} \sum_{y \in \{0,1\}^n} \Pr_{y_2,\ldots,y_{r+1} \in \{0,1\}^m}[T_f^y(y_2, \ldots, y_{r+1}) \neq f(y)] \\ &\geq \frac{1}{2^m} \sum_{y \in \Delta(f,g)} \Pr_{y_2,\ldots,y_{r+1} \in \{0,1\}^m}[T_f^y(y_2, \ldots, y_{r+1}) = g(y)] \\ &\geq \frac{1}{2^m} \cdot |\Delta(f,g)| \cdot \frac{1}{2} \end{aligned}$$

Thus, $\text{dist}(f, g) = \frac{|\Delta(f,g)|}{2^m} \leq 2\eta$. ■

Recall that by the definition of g as a majority function, for every y, we have that for at least one half of the r-tuples of vectors $y_2, \ldots, y_{r+1}$, $T_f^y(y_2, \ldots, y_{r+1}) = g(y)$. In the next lemma we show that this equality actually holds for a vast majority of the r-tuples $y_2, \ldots, y_{r+1}$ (assuming η is sufficiently small).

Lemma 3 *For every* $y \in \{0,1\}^n$: $\Pr_{y_2,\ldots,y_{r+1}\in\{0,1\}^m}[g(y) = T_f^y(y_2,\ldots,y_{r+1})] \geq 1-2r\eta$.

In order to prove Lemma 3 we shall first establish the following claim.

Claim 4 *For every* $y, z, w, y_2, \ldots, y_r \in \{0,1\}^m$,

$$\begin{aligned} & T_f(y,y_2,\ldots,y_r,w) + T_f(y,y_2,\ldots,y_r,z) \\ = \;& T_f(y+w,y_2,\ldots,y_r,y+w+z) + T_f(y+z,y_2,\ldots,y_r,y+w+z) \end{aligned} \tag{7}$$

Proof: Let $Y = \{y_2,\ldots,y_r\}$, and consider any set $I \subseteq \{2,\ldots,r\}$, which may be the empty set. For a vector $x \in \{0,1\}^m$ denote $f_{Y,I}(x) \stackrel{\text{def}}{=} f(\sum_{i\in I} y_i + x)$.
For every set $I \subseteq \{2,\ldots,r\}$, each element of type $f(\sum_{i\in I} y_i)$ appears twice in both sides of Equation (7) and thus cancels out. Now for every set $I \subset \{2,\ldots,r\}$ (including the empty set), we get in the left hand side of Equation (7):

$$f_{Y,I}(y) + f_{Y,I}(w) + f_{Y,I}(y+w) + f_{Y,I}(y) + f_{Y_I}(z) + f_{Y,I}(y+z)\ .$$

In the right hand side of Equation (7) we get:

$$f_{Y,I}(y+w) + f_{Y,I}(y+z+w) + f_{Y,I}(z) + f_{Y,I}(y+z) + f_{Y,I}(y+w+z) + f_{Y,I}(w)\ .$$

This implies equality over $GF(2)$. ■

We now turn to prove Lemma 3.

Proof of Lemma 3: We fix $y \in \{0,1\}^m$ and let $\gamma \stackrel{\text{def}}{=} \Pr_{y_2,\ldots,y_{r+1}\in\{0,1\}^m}[g(y) = T_f^y(y_2,\ldots,y_{r+1})]$. Recall that we are interested in proving that $\gamma \geq 1-2r\eta$. To this end, we shall bound a slightly different, but related probability. Let

$$\delta \stackrel{\text{def}}{=} \Pr_{y_2,\ldots,y_{r+1},z_2,\ldots,z_{r+1}\in\{0,1\}^m}[T_f^y(y_2,\ldots,y_{r+1}) = T_f^y(z_2,\ldots,z_{r+1})]\ . \tag{8}$$

Then, by the definitions of γ and δ,

$$\begin{aligned} \delta \;&=\; \Pr[T_f^y(y_2,\ldots,y_{r+1}) = g(y) \text{ and } T_f^y(z_2,\ldots,z_{r+1}) = g(y)] \\ &\qquad + \Pr[T_f^y(y_2,\ldots,y_{r+1}) \neq g(y) \text{ and } T_f^y(z_2,\ldots,z_{r+1}) \neq g(y)] \\ &=\; \gamma^2 + (1-\gamma)^2 \end{aligned} \tag{9}$$

where the probabilities are over the choice of $y_2,\ldots,y_{r+1},z_2,\ldots,z_{r+1} \in \{0,1\}^m$. Since we are working over $GF(2)$,

$$\delta = \Pr_{y_2,\ldots,y_{r+1},z_2,\ldots,z_{r+1}\in\{0,1\}^m}[T_f(y,y_2,\ldots,y_{r+1}) + T_f(y,z_2,\ldots,z_{r+1}) = 0]\ .$$

Now, for any choice of $y_2,\ldots,y_{r+1}$ and $z_2,\ldots,z_{r+1}$:

$$\begin{array}{lcll} T_f(y,y_2,\ldots,y_{r+1}) & + & T_f(y,z_2,\ldots,z_{r+1}) & = \\ T_f(y,y_2,\ldots,y_{r+1}) & + & T_f(y,y_2,\ldots,y_r,z_{r+1}) & + \\ T_f(y,y_2,\ldots,y_r,z_{r+1}) & + & T_f(y,y_2,\ldots,y_{r-1},z_r,z_{r+1}) & + \\ T_f(y,y_2,\ldots,y_{r-1},z_r,z_{r+1}) & + & T_f(y,y_2,\ldots,y_{r-2},z_{r-1},z_r,z_{r+1}) & + \\ . & & & \\ . & & & \\ . & + & & \\ T_f(y,y_2,z_3,\ldots,z_{r+1}) & + & T_f(y,z_2,\ldots,z_{r+1}). & \end{array}$$

Consider any pair $T_f(y, y_2, \ldots, y_\ell, z_{\ell+1}, \ldots, z_{r+1}) + T_f(y, y_2, \ldots, y_{\ell-1}, z_\ell, \ldots, z_{r+1})$ that appears in the above sum. Note that $T_f(y, y_2, \ldots, y_\ell, z_{\ell+1}, \ldots, z_{r+1})$ and $T_f(y, y_2, \ldots, y_{\ell-1}, z_\ell, \ldots, z_{r+1})$ differ only in a single parameter. Since $T_f(\cdot)$ is a symmetric function we can apply Claim 4 and obtain that

$$\begin{aligned} & T_f(y, y_2, \ldots, y_\ell, z_{\ell+1}, \ldots, z_{r+1}) + T_f(y, y_2, \ldots, y_{\ell-1}, z_\ell, \ldots, z_{r+1}) \\ = \;& T_f(y + y_\ell, y_2, \ldots, y_{\ell-1}, z_{\ell+1}, \ldots, z_{r+1}, y + y_\ell + z_\ell) \\ & + \; T_f(y + z_\ell, y_2, \ldots, y_{\ell-1}, z_{\ell+1}, \ldots, z_{r+1}, y + y_\ell + z_\ell) \end{aligned} \tag{10}$$

Recall that y is fixed and $y_2, \ldots, y_{r+1}, z_2, \ldots, z_{r+1} \in \{0,1\}^m$ are uniformly selected, and so all parameters on the right hand side in the above equation are uniformly distributed. Also recall that by the definition of η, for $T_f(p_1, \ldots, p_{r+1})$, where p_i are uniformly selected at random, $\Pr_{p_1,\ldots,p_{r+1} \in \{0,1\}^n}[T_f(p_1, \ldots, p_{r+1}) \neq 0] = \eta$. Hence, by the union bound:

$$\begin{aligned} \delta &= \Pr_{y_2,\ldots,y_{r+1},z_2,\ldots,z_{r+1} \in \{0,1\}^m}[T_f(y, y_2, \ldots, y_{r+1}) + T_f(y, z_2, \ldots, z_{r+1}) = 0] \\ &\geq 1 - 2r\eta. \end{aligned} \tag{11}$$

By combining Equations (9) and (11) we get that $\gamma^2 + (1-\gamma)^2 \geq 1 - 2r\eta$. Since $\gamma \geq 1/2$ it follows that $\gamma = \gamma^2 + \gamma(1-\gamma) \geq \gamma^2 + (1-\gamma)^2 \geq 1 - 2r\eta$. ■

Lemma 5 *If $\eta < \frac{1}{(4r+2)2^r}$, then the function g belongs to $\mathcal{P}_r$.*

Proof: By Claim 1 it suffices to prove that if $\eta < \frac{1}{(4r+2)2^r}$, then $T_g(y_1, \ldots, y_{r+1}) = 0$, for every $y_1, \ldots, y_{r+1} \in \{0,1\}^m$. Let us fix the choice of $y_1, \ldots, y_{r+1}$, and recall that as defined in Equation (2), $T_g(y_1, \ldots, y_{r+1}) = \sum_{\emptyset \neq I \subseteq [r+1]} g(\sum_{i \in I} y_i)$. Suppose we uniformly select $r \cdot (r+1)$ random vectors $z_{i,j} \in \{0,1\}^m$, $1 \leq i \leq r+1$, $1 \leq j \leq r$. Then by Lemma 3, for every I, $\emptyset \neq I \subseteq [r+1]$, with probability at least $1 - 2r\eta$ over the choice of the $z_{i,j}$'s,

$$g\left(\sum_{i \in I} y_i\right) = T_f\left(\sum_{i \in I} y_i, \sum_{i \in I} z_{i,1}, \sum_{i \in I} z_{i,2}, \ldots, \sum_{i \in I} z_{i,r}\right) + f\left(\sum_{i \in I} y_i\right). \tag{12}$$

Let E_1 be the event that Equation (12) holds for all $\emptyset \neq I \subseteq [r+1]$. By the union bound:

$$\Pr[E_1] \geq 1 - (2^{r+1} - 1) \cdot 2r\eta \tag{13}$$

Assume that E_1 holds. Then

$$\begin{aligned} & T_g(y_1, \ldots, y_{r+1}) \\ &= \sum_{\emptyset \neq I \subseteq [r+1]} \left[T_f\left(\sum_{i \in I} y_i, \sum_{i \in I} z_{i,1}, \sum_{i \in I} z_{i,2}, \ldots, \sum_{i \in I} z_{i,r}\right) + f\left(\sum_{i \in I} y_i\right)\right] \\ &= \sum_{\emptyset \neq I \subseteq [r+1]} \sum_{\emptyset \neq J \subseteq [r]} \left[f\left(\sum_{i \in I} \sum_{j \in J} z_{i,j}\right) + f\left(\sum_{i \in I} y_i + \sum_{i \in I} \sum_{j \in J} z_{i,j}\right)\right] \\ &= \sum_{\emptyset \neq J \subseteq [r]} \sum_{\emptyset \neq I \subseteq [r+1]} f\left(\sum_{i \in I} \sum_{j \in J} z_{i,j}\right) \end{aligned}$$

$$+ \sum_{\emptyset \neq J \subseteq [r]} \sum_{\emptyset \neq I \subseteq [r+1]} f\left(\sum_{i \in I} y_i + \sum_{i \in I} \sum_{j \in J} z_{i,j}\right)$$

$$= \sum_{\emptyset \neq J \subseteq [r]} T_f\left(\sum_{j \in J} z_{1,j}, \ldots, \sum_{j \in J} z_{r+1,j}\right)$$

$$+ \sum_{\emptyset \neq J \subseteq [r]} T_f\left(y_1 + \sum_{j \in J} z_{1,j}, \ldots, y_{r+1} + \sum_{j \in J} z_{r+1,j}\right). \tag{14}$$

Let E_2 be the event that for every $\emptyset \neq J \subseteq [r]$, $T_f\left(\sum_{j \in J} z_{1,j}, \ldots, \sum_{j \in J} z_{r+1,j}\right) = 0$ and $T_f\left(y_1 + \sum_{j \in J} z_{1,j}, \ldots, y_{r+1} + \sum_{j \in J} z_{r+1,j}\right) = 0$. By the definition of η:

$$\Pr[E_2] \geq 1 - 2(2^r - 1)\eta \tag{15}$$

Suppose that $\eta < \frac{1}{(4r+2)2^r}$. Then, by Equations (13) and (15), the probability that both E_1 and E_2 hold, is strictly positive. In other words, there exists a choice of the $z_{i,j}$'s for which all summands in Equation (14) are 0. But this implies that $T_g(y_1, \ldots, y_{r+1}) = 0$. We conclude that if $\eta < \frac{1}{(4r+2)2^r}$, then g belongs to $\mathcal{P}_r$, and this completes the lemma's proof. ■

By combining Lemmas 2 and 5 we obtain that if f is $\Omega(1/(r2^r))$-far from $\mathcal{P}_r$, then $\eta = \Omega(1/(r2^r))$, and so the algorithm rejects f with sufficiently high constant probability (since it selects $\Omega(r2^r)$ groups of vectors $y_1, \ldots, y_{r+1}$). We next deal with the case in which η is small. By Lemma 2, in this case the distance $d = \mathrm{dist}(f, g)$ between f and g is small, and we show that the test rejects f with probability that is close to $(2^{r+1} - 1)d$. This follows from the fact that in this case, the probability over the selection of $y_1, \ldots, y_{r+1}$, that among the $(2^{r+1} - 1)$ points $\sum_{\emptyset \neq I \subseteq [r+1]} y_i$, the functions f and g differ in precisely one point, is close to $(2^{r+1} - 1)d$. This is formally proved in the following lemma.

Lemma 6 *Suppose* $0 < \eta < \frac{1}{(4r+2)2^r}$. *Let* $d = \mathrm{dist}(f, g)$ *denote the distance between* f *and* g, *and let*

$$p \stackrel{\mathrm{def}}{=} \frac{1 - (2^{r+1} - 1)d}{1 + (2^{r+1} - 1)d} \cdot (2^{r+1} - 1)d.$$

Then, when $y_1, y_2, \ldots, y_{r+1}$ *are chosen randomly, the probability that for exactly one point* v *among the* $(2^{r+1} - 1)$ *points* $\sum_{i \in S} y_i$, $(\emptyset \neq S \subseteq [r+1])$, $f(v) \neq g(v)$, *is at least* p.

By definition of η and the above lemma, $\eta \geq p$ (under the premise of the lemma). In particular, since (by Lemma 2) $d \leq 2\eta \leq \frac{1}{(2r+1)2^r}$ and $r \geq 1$, $\eta \geq \frac{1}{3}(2^{r+1} - 1)d$, and, for fixed r, as d tends to zero, $\eta \geq (2^{r+1} - 1)d - O(d^2)$.

Proof: For each subset S, $\emptyset \neq S \subseteq [r+1]$, let X_S be the indicator random variable whose value is 1 if and only if $f(\sum_{i \in S} y_i) \neq g(\sum_{i \in S} y_i)$. Obviously, $\Pr[X_S = 1] = d$ for every S. It is not difficult to check that the random variables X_S are pairwise independent, since for any two distinct nonempty S_1, S_2, the sums $\sum_{i \in S_1} y_i$ and $\sum_{i \in S_2} y_i$ attain each pair of distinct values in $\{0,1\}^n$ with equal probability when the vectors y_i are chosen randomly and independently. It follows that the random variable $X = \sum_S X_S$ which counts the number of points v of the required form in which $f(v) \neq g(v)$ has expectation $\mathrm{E}[X] = (2^{r+1} - 1)d$ and variance $\mathrm{Var}[X] = (2^{r+1} - 1)d(1 - d) \leq \mathrm{E}[X]$.

Our objective is to lower bound the probability that $X = 1$. We need the well known, simple fact that for a random variable X that attains nonnegative, integer values,

$$\Pr[X > 0] \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X^2]}.$$

Indeed, if X attains the value i with probability p_i for $i > 0$, then, by Cauchy-Schwartz,

$$(\mathrm{E}[X])^2 = (\sum_{i>0} ip_i)^2 = (\sum_{i>0} i\sqrt{p_i}\sqrt{p_i})^2 \leq (\sum_{i>0} i^2 p_i)(\sum_{i>0} p_i) = \mathrm{E}[X^2]\Pr[X > 0].$$

In our case, this implies

$$\Pr[X > 0] \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X^2]} \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X] + (\mathrm{E}[X])^2} = \frac{\mathrm{E}[X]}{1 + \mathrm{E}[X]}.$$

Therefore

$$\mathrm{E}[X] \geq \Pr[X = 1] + \left(\frac{\mathrm{E}[X]}{1 + \mathrm{E}[X]} - \Pr[X = 1]\right) \cdot 2 = \frac{2\mathrm{E}[X]}{1 + \mathrm{E}[X]} - \Pr[X = 1],$$

implying that

$$\Pr[X = 1] \geq \frac{\mathrm{E}[X] - (\mathrm{E}[X])^2}{1 + \mathrm{E}[X]}.$$

Substituting the value of $\mathrm{E}[X]$, the desired result follows. ■

We are now ready to wrap up the proof of Theorem 1.

Proof of Theorem 1: As we have noted previously, if f is in $\mathcal{P}_r$, then by Claim 1 the tester accepts (with probability 1). We next show that if f is ϵ-far from $\mathcal{P}_r$, then the tester rejects with probability at least $\frac{2}{3}$.

Suppose that $dist(f, \mathcal{P}_r) > \epsilon$. Denote $d = dist(f, g)$. If $\eta < \frac{1}{(4r+2)2^r}$ then by Lemma 5 $g \in \mathcal{P}_r$ and, by Lemma 6, $\eta \geq \Omega(2^r d) \geq \Omega(2^r \epsilon)$. Hence, $\eta \geq min\left(\Omega(2^r \epsilon), \frac{1}{(4r+2)2^r}\right)$. Clearly it is enough to perform $O(\frac{1}{\eta})$ rounds of the algorithm in order to detect a violation with probability at least $\frac{2}{3}$. This completes the proof of the theorem. ■

4.1 A Lower Bound

The following is a general lower bound on testing linear codes.

Theorem 2 *Let $\mathcal{C}$ be a a linear code of length n. Let d denote the minimum distance of the code $\mathcal{C}$ and let $\bar{d}$ denote the minimum distance of the dual code of $\mathcal{C}$.*

If a random binary word of length n is ϵ-far from $\mathcal{C}$, then every testing algorithm for $\mathcal{C}$ must perform $\Omega(\bar{d})$ queries. In addition, if the distance parameter ϵ is at most $d/(2n)$, then $\Omega(1/\epsilon)$ is also a lower bound for the necessary number of queries.

The first assumption is a trivial one in most cases and is satisfied, for example, by any linear code $\mathcal{C}$ of rate at most $1 - \delta$, for a constant $\delta > 0$, whenever the distance parameter ϵ is sufficiently small compared to δ. As noted in the introduction, the family $\mathcal{P}_r$ corresponds to the shortened Reed-Muller code $\mathcal{R}(r, m)^*$. It is well known (see [18, Chap. 13]) that the distance of $\mathcal{R}(r, m)^*$ is 2^{m-r} and the distance of the dual code (which is a punctured Reed-Muller code) is $2^{r+1} - 1$. Hence we obtain the following corollary.

Corollary 7 *Every algorithm for testing $\mathcal{P}_r$ with distance parameter ϵ must perform $\Omega\left(\max(\frac{1}{\epsilon}, 2^{r+1})\right)$ queries.*

Proof of Theorem 2: We start with showing that $\Omega(\bar{d})$ queries are necessary. A well known fact from coding theory (see [18, Chap. 5]) states the following: for every linear code $\mathcal{C}$ whose dual code has distance $\bar{d}$, if we examine any sub-word having length d', $d' < \bar{d}$, of a uniformly selected codeword in $\mathcal{C}$, then the resulting sub-word is uniformly distributed in $\{0,1\}^{d'}$. Hence it is not possible to distinguish between a random codeword in $\mathcal{C}$ and a random binary word of length n using less than $\bar{d}$ queries.

We now turn to the case $\epsilon < d/2^{m+1}$. To prove the lower bound here, we apply, as usual, the Yao duality principle ([20]) by defining two distributions, one on positive instances, and the other on negative ones, and then by showing that in order to distinguish between those distributions any algorithm must perform $\Omega(1/\epsilon)$ queries. The positive distribution has all its mass at the zero vector $\bar{0} = (0, \ldots, 0)$. To define the negative distribution, partition the set of all coordinates randomly into $t = 1/\epsilon$ nearly equal parts $I_1, \ldots, I_t$ and give weight $1/t$ to each of the characteristic vectors w_i of I_i, $i = 1, \ldots, t$. (Observe that indeed $\bar{0} \in \mathcal{C}$ by linearity, and $dist(w_i, \mathcal{C}) = \epsilon$ by the assumption on the minimum distance of $\mathcal{C}$). Finally, a random instance is generated by first choosing one of the distributions with probability 1/2, and then generating a vector according to the chosen distribution. It is easy to check (see, e.g., [1] for details) that in order to give a correct answer with probability at least 2/3, the algorithm has to query $\Omega(1/\epsilon)$ bits of the input.

■

5 Concluding remarks

We first note that in view of the above lower bound, our upper bound is almost tight.

It will be interesting to study analogous questions for other linear binary codes. As noted in the introduction, several recent papers, including [15], [9], [10], deal with related questions. As shown above, a code is not testable with a constant number of queries if its dual distance is not a constant, and it seems plausible to conjecture that if the dual distance is a constant, and there is a doubly transitive permutation group acting on the coordinates that maps the dual code to itself, then the code can be testable with a constant number of queries. The automorphism group of punctured Reed-Muller codes contains the general linear group $\mathrm{GL}(n, 2)$, and thus those codes supply an example with these properties. Another interesting example is the set of duals of BCH codes (this class also contains linear functions as a particular case).

References

[1] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy. Regular languages are testable with a constant number of queries. In *Proceedings of the Fortieth Annual Symposium on Foundations of Computer Science*, pages 645–655, 1999.

[2] S. Arora and S. Safra. Improved low-degree testing and its applications. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 485–495, 1997.

[3] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 21–31, 1991.

[4] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[5] L. Babai, A. Shpilka, and D. Stefankovic. Locally testable cyclic codes. In *Proceedings of the Forty-forth Annual Symposium on Foundations of Computer Science*, pages 116–125, 2003.

[6] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan. Linearity testing in characteristic two. *IEEE Trans. Inform. Theory*, 42:1781–1795, 1996.

[7] M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 294–304, 1993.

[8] M. Bellare and M. Sudan. Improved non-approximability results. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 184–193, 1994.

[9] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF properties are hard to test. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 345–354, 2003.

[10] E. Ben-Sasson, M. Sudan, S. Vadhan, and A. Wigderson. Randomness-efficient low degree tests and short pcps via epsilon-biased sets. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 612–621, 2003.

[11] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993.

[12] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating clique is almost NP-complete. *Journal of the Association for Computing Machinery*, pages 268–292, 1996.

[13] K. Friedl and M. Sudan. Some improvements to total degree tests. In *Proceedings of the 3rd Annual Israel Symposium on Theory of Computing and Systems*, pages 190–198, 1995. Corrected version available online at http://theory.lcs.mit.edu/˜madhu/papers/friedl.ps.

[14] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, pages 32–42, 1991.

[15] O. Goldreich and M. Sudan. Locally testable codes and pcps of almost-linear length. In *Proceedings of the Forty-Third Annual Symposium on Foundations of Computer Science*, pages 13–22, 2002.

[16] M. Hall. *Combinatorial Theory.* John Wiley & Sons, 1967.

[17] T. Kasami, S. Lin, and W.W. Peterson. New generalizations of the reed-muller codes, part i: Primitive codes. *IEEE Transactions on Information Theory*, pages 189–199, 1968.

[18] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes.* North Holland, 1977.

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

[20] A. C. Yao, Probabilistic computation, towards a unified measure of complexity. Proceedings of the 18^{th} IEEE FOCS (1977), 222–227.