Testing Low-Degree Polynomials over Prime Fields

Charanjit S. Jutla
IBM Thomas J. Watson Research Center,
Yorktown Heights, NY 10598
csjutla@watson.ibm.com

Anindya C. Patthak*
University of Texas at Austin
Austin, TX 78712
anindya@cs.utexas.edu

Atri Rudra†
Dept. of Computer Science & Engineering
University of Washington
Seattle, WA 98915
atri@cs.washington.edu

David Zuckerman‡
University of Texas at Austin
Austin, TX 78712
diz@cs.utexas.edu

Abstract

We present an efficient randomized algorithm to test if a given function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ (where p is a prime) is a low-degree polynomial. This gives a local test for Generalized Reed-Muller codes over prime fields. For a given integer t and a given real $\epsilon > 0$, the algorithm queries f at $\frac{1}{\epsilon} + t \cdot p^{\frac{2t}{p-1}+O(1)}$ points to determine whether f can be described by a polynomial of degree at most t. If f is indeed a polynomial of degree at most t, our algorithm always accepts, and if f has a relative distance at least ϵ from every degree t polynomial, then our algorithm rejects f with probability at least $\frac{1}{2}$. Our result is almost optimal since any such algorithm must query f on at least $\Omega(\frac{1}{\epsilon} + p^{\frac{t+1}{p-1}})$ points.

1. Introduction

We present a low degree test for multivariate polynomials over any prime field $\mathbb{F}_p$. This generalizes the result obtained recently by Alon et al [AKK+03], which gives a low degree test for the field $\mathbb{F}_2$. A test for the case when the degree to be tested is less than the size of the field has been known for some time [BLR93, BFL91, GLR+91, RS96, FS95, AS97]. It was an open problem to give an efficient low degree tester over fields of size between two and the degree being tested.

The study of low degree testing along with self-correction dates back to [BLR93], where an algorithm was given to test whether a given function is linear. The approach in [BLR93] later naturally extended to testers for low degree polynomials over large fields. Roughly, the idea is to project the given function on a random line and then to test the projected univariate function for low degree. Specifically, for a purported degree k function $f : \mathbb{F}_p^n \to \mathbb{F}_p$, the test works as follows. Pick vectors y and b from $\mathbb{F}_p^n$ randomly (uniformly), and distinct $t_1, \cdots, t_{k+1}$ from $\mathbb{F}_p$ arbitrarily. Query the oracle representing f at the $k+1$ points $b + t_i y$ and extrapolate to a degree k polynomial $P_{b,y}$ in one variable t. Now test for a random $t \in \mathbb{F}_p$ whether

$$P_{b,y}(t) = f(b + ty)$$

(for details see [RS96, FS95]). Similar ideas are also employed to test whether a given function is a low degree polynomial in each of its variable (see [FGL+91, BFLS91, AS92]).

Low degree testing forms the core in the proof of MIP = NEXPTIME in [BFL91]. The efficiency of the tester also has direct implications in the constructions of efficient PCP's and various inapproximability results ([FGL+91, ALM+92]). Therefore a lot of attention has been paid to this problem ([BFL91, BFLS91, FGL+91, AS92, RS96, FS95, AKK+03]).

However with the sole exception of [AKK+03], all the above mentioned tests (and their variants) require the degree to be less than the field size. This is because the degree to be tested has to be smaller than the number of points on a line. Hence this approach cannot be used when the degree is larger than the field size.

---

* Supported in part by NSF Grant CCR-0310960.
† This work was done while the author was at the University of Texas at Austin.
‡ Supported in part by NSF Grants CCR-9912428 and CCR-0310960 and a David and Lucile Packard Fellowship for Science and Engineering.

Alon et al. in [AKK$^+$03] give a tester for the field $\mathbb{F}_2$ without any restriction on the degree. Their results have a natural interpretation in terms of coding theory. Recall that the collection of polynomials in n variables of degree at most k over $\mathbb{F}_2$ is the `Reed-Muller code` $\mathcal{R}(k,n)$ with parameters k and n. Therefore, a given function has low degree iff (the vector of evaluations of) the function is a valid codeword in the corresponding Reed-Muller code. In other words, low degree testing can be considered as (locally) testing Reed-Muller codes. Their strategy is then to pick a random minimum-weight codeword from the dual code and to check if it is orthogonal to the tested vector. It is important to note that these minimum-weight code words generate the Reed-Muller code.

Specifically their test works as follows: given a function $f : \{0,1\}^n \to \{0,1\}$, to test if the given function f has degree at most k, pick $(k+1)$-vectors $y_1, \cdots, y_{k+1} \in \{0,1\}^n$ and test if

$$\sum_{\emptyset \neq S \subseteq [k+1]} f(\sum_{i \in S} y_i) = 0.$$

Our result can also be interpreted in terms of local testability of `Generalized Reed-Muller codes`, henceforth abbreviated as GRM. Recall that $\mathrm{GRM}_q(n,k)$ is the collection of vectors of (evaluations of) all polynomials in n variables of total degree at most k over $\mathbb{F}_q$ (see [DGM70, DK00, PH98] for more details). In this paper, we consider a new basis (over prime fields) of Generalized Reed-Muller codes that in general differs from the minimum weight basis. This allows us to present a novel exact characterization of the multivariate polynomial of degree k in n variables over prime fields. Our basis has a clean geometric structure in terms of flats [PH98], and unions of parallel flats (but with different weights assigned to different parallel flats)[1]. Equivalent polynomial and geometric representations allow us to provide an almost optimal test.

It is easier to define our tester over $\mathbb{F}_3$. To test if f has degree at most t, set $k = \lceil \frac{t+1}{2} \rceil$, and let $i = (t+1) \pmod 2$, pick k-vectors $y_1, \cdots, y_k$ and b from $\mathbb{F}_3^n$, and test if[2]

$$\sum_{c \in \mathbb{F}_3^k; c=(c_1,\cdots,c_k)} c_1^i f(b + \sum_{j=1}^{k} c_j y_j) = 0.$$

We remark here that a polynomial of degree at most t always passes the test, whereas a polynomial of degree greater than t gets caught with small probability. To obtain a constant rejection probability we repeat the test.

The analysis of our test follows a similar general structure developed in [RS96] and borrows techniques from [RS96, AKK$^+$03]. The presence of a doubly transitive group suffices for the analysis given in [RS96]. Essentially we show that the presence of doubly transitive group acting on the coordinates of the dual code does indeed allow us to randomize the test. However, this gives a weaker result. We used techniques developed in [AKK$^+$03] for better results. However, the adoption is not immediate. Particularly the interplay between the geometric objects described earlier and its polynomial representation plays a pivotal role in getting results which are only quadratic factor away from optimal query complexity.

Our results may be stated quantitatively as follows: For a given integer $t \geq (p-1)$ and a given real $\epsilon > 0$, our testing algorithm queries f at $\frac{1}{\epsilon} + t \cdot p^{\frac{2t}{p-1}+O(1)}$ points to determine whether f can be described by a polynomial of degree at most t. If f is indeed a polynomial of degree at most t, our algorithm always accepts, and if f has a relative distance at least ϵ from every degree t polynomial, then our algorithm rejects f with probability at least $\frac{1}{2}$. (In the case $t < (p-1)$, our tester still works but more efficient testers are known). Our result is almost optimal since any such testing algorithm must query f in at least $\Omega(\frac{1}{\epsilon} + p^{\frac{t+1}{p-1}})$ many points.

Our analysis also enables us to obtain a *self-corrector* (as defined in [BLR93]) for f, in case the function f is reasonably close to a degree t polynomial. Specifically, we show that the value of the function f at any given point $x \in \mathbb{F}_p^n$ may be obtained with good probability by querying f on $\Theta(p^{t/p})$ random points. Using pairwise-independence we can even achieve even higher probability by querying f on $p^{O(t/p)}$ random points and using majority logic decoding.

Related Work: Independently, Kaufman and Ron have given a tester to test low degree polynomials over general fields (see [KR04] in this volume). Briefly, they have shown that a given polynomial is of degree at most d *if and only if* the restriction of the polynomial to every affine subspace of suitable dimension is of degree at most d. Following this idea, their tester chooses a random affine subspace of a suitable dimension and computes the polynomial restricted to this subspace and verifies that the coefficients of the higher degree terms are identically zero. To obtain constant soundness, the test is repeated many times. An advantage of our approach is that in one round of the test we test only one linear constraint, whereas their approach needs to test multiple linear constraints.

A basis consisting of minimum-weight codewords was considered in [DGM70, DK00]. We extend their result to obtain a different exact characterization for low-degree polynomials. Furthermore, an analysis similar to ours seems to go through, though we have not worked out the details. However, our basis is cleaner and yields a

1 The natural basis given in [DGM70, DK00] assigns the same weight to each parallel flat

2 For notational convenience we use $0^0 = 1$.

simpler analysis. We emphasize that we have not explicitly used dual-theoretic argument in our proof, and therefore our work also gives a direct elementary proof of the duality of the Generalized Reed-Muller code (over prime fields). We point out that for degree smaller than the field size, the exact characterization obtained from [DGM70, DK00] coincides with [BLR93, RS96, FS95]. (This provides an alternate proof to the exact characterization of [FS95]. For more details, see Remark 3.14 later and [FS95]).

Organization of the paper: The rest of the paper is organized as follows. In Section 2 we introduce few notations and mention few preliminary facts. Section 3 contains the exact characterization of the low degree polynomials over prime fields. In Section 4 we formally describe the tester and prove its correctness. In Section 5 we sketch a lower bound that implies that the query complexity of our tester is almost optimal, and suggest how to self-correct a function which is correct on most of its input. Section 6 contains some concluding remarks.

2. Preliminaries

For any integer l, we denote the set $\{1, \cdots, l\}$ by $[l]$. Throughout we will use p to denote a prime and $\mathbb{F}_p$ to denote a prime field of size p. For any $t \in [n(p-1)]$, let $\mathcal{P}_t$ denote the family of all functions over $\mathbb{F}_p^n$ which are polynomials of degree at most t in n variables. In particular $f \in \mathcal{P}_t$ if and only if there exists coefficients $a_{(e_1,\cdots,e_n)}$, for every $(e_1, \cdots, e_n) \in \mathbb{F}_p^n$, $\sum_{i=1}^n e_i \leq t$, such that

$$f = \sum_{(e_1,\cdots,e_n)\in\mathbb{F}_p^n, 0\leq\sum_{i=1}^n e_i\leq t} a_{(e_1,\cdots,e_n)} \prod_{i=1}^n x_i^{e_i}, \quad (1)$$

where the addition is over $\mathbb{F}_p$.

The relative distance between two given function $f, g : \mathbb{F}_p^n \to \mathbb{F}_p^n$ is defined as $\delta(f,g) \stackrel{def}{=} \frac{|\{y\in\mathbb{F}_p^n | f(y)\neq g(y)\}|}{p^n}$. For a function g and a family of functions F (defined over the same domain and range), we say g is ϵ-`close` to F, for some $0 < \epsilon < 1$, if, there exists an $f \in F$, $\delta(f,g) \leq \epsilon$. Otherwise it is ϵ-`far` from F.

A one-sided testing algorithm (`one-sided tester`) for $\mathcal{P}_t$ is a probabilistic algorithm, that is given query access to a function f, and a distance parameter ϵ, $0 < \epsilon < 1$. If $f \in \mathcal{P}_t$, then the tester should always accept f (perfect completeness), and if f is ϵ-`far` from $\mathcal{P}_t$, then with probability at least $\frac{1}{2}$ the tester should reject f. (A two-sided tester may be defined analogously.)

To motivate the next notation which we will use frequently, we give a definition.

Definition 2.1 *A k-flat ($k \geq 0$)[3] in $\mathbb{F}_p^n$ is a k-dimensional affine subspace. Let $y_1, \cdots, y_k \in \mathbb{F}_p^n$ be linearly independent vectors and $b \in \mathbb{F}_p^n$ be a point. Then the subset $L = \{\sum_{i=1}^k c_i y_i + b | \forall i \in [k]\ c_i \in \mathbb{F}_p\}$ is a k-dimensional flat. We will say that L is generated by $y_1, \cdots, y_k$ at b. The incidence vector of the points in a given k-flat will be referred to as the codeword corresponding to the given k-flat.*

Given a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$, for $y_1, \cdots, y_l, b \in \mathbb{F}_p^n$ we define

$$T_f(y_1, \cdots, y_l, b) \stackrel{def}{=} \sum_{c=(c_1,\cdots,c_l)\in\mathbb{F}_p^l} f(b + \sum_{i\in[l]} c_i y_i), \quad (2)$$

which is the sum of the evaluations of function f over an l-flat generated by $y_1, \cdots, y_l$, at b. We also let for all $i \in [p-2]$,

$$T_f^i(y_1, \cdots, y_l, b) \stackrel{def}{=} \sum_{c=(c_1,\cdots,c_l)\in\mathbb{F}_p^l} c_1^i \cdot f(b + \sum_{j\in[l]} c_j y_j). \quad (3)$$

The above can also be interpreted similarly by a suitably weighted sum over a similar geometric object which we will define a little later. With a slight abuse of notation[4] we will use $T_f^0(y_1, \cdots, y_l, b)$ to denote $T_f(y_1, \cdots, y_l, b)$.

2.1. Some $\mathbb{F}_p$ facts

In this subsection we spell out some facts which hold over prime fields and will be used later. We denote the multiplicative group of $\mathbb{F}_p$ by $\mathbb{F}_p^*$. The following lemma is straightforward.

Lemma 2.2 *For any $t \in [p-1]$, $\sum_{a\in\mathbb{F}_p} a^t \neq 0$ if and only if $t = p-1$.*

This immediately implies the following lemma:

Lemma 2.3 *Let $q_1, \cdots, q_l \in [p-1]$. Then*

$$\sum_{(c_1,\cdots,c_l)\in(\mathbb{F}_p)^l} c_1^{q_1} c_2^{q_2} \cdots c_l^{q_l} \neq 0 \quad (4)$$

if and only if $q_1 = q_2 = \cdots = q_l = p-1$.

Proof: Note that the left hand side can be rewritten as $\prod_{i\in[l]} \left(\sum_{c_i\in\mathbb{F}_p} c_i^{q_i}\right)$. ■

We will need to transform products of variables to powers of linear functions in these variables. With this motivation, we present the following identity :

Fact 2.4 *For each k, s.t. $0 < k \leq (p-1)$ there exists $c_k \in \mathbb{F}_p^*$ such that*

$$c_k \prod_{i=1}^k x_i = \sum_{i=1}^k (-1)^{k-i} S_i \quad (5)$$

3 A zero-dimensional flat is just a point.

4 We set $0^0 = 1$, for notational convenience.

where $S_i = \sum_{\emptyset \neq I \subseteq [k]; |I|=i} (\sum_{j\in I} x_j)^k$.

Proof: Consider the right hand side of the Equation 5. Note that all the monomials are of degree exactly k. Also note that $\prod_{i=1}^{k} x_i$ appears only in the S_k and nowhere else. Now consider any other monomial of degree k that has a support of size j, where $0 < j < k$. Further note that the coefficient of any such monomial in the expansion of $(\sum_{j\in I} x_j)^k$ is the same and non-zero. Therefore, summing up the number of times it appears (along with the $(-1)^{k-i}$ factor) in each S_i is enough which is just

$$1 - \binom{k-j}{k-j-1} + \binom{k-j}{k-j-2} + \cdots +$$

$$(-1)^{k-j}\binom{k-j}{k-j-(k-j)} = (1-1)^{k-j} = 0.$$

Moreover, it is clear that $c_k = k!$ (mod p) and $c_k \neq 0$ for the choice of k. ■

3. Characterization of Low Degree Polynomials over $\mathbb{F}_p$

In this section we will show how to characterize low degree polynomials. We first recall the definition of the Generalized (Primitive) Reed-Muller code as described in [PH98, DK00].

Definition 3.1 *Let $V = \mathbb{F}_q^n$ be the vector space of n-tuples, for $n \geq 1$, over the field $\mathbb{F}_q$. For any k such that $0 \leq k \leq n(q-1)$, the k^{th} order Generalized Reed-Muller code* $\mathrm{GRM}_q(k,n)$ *is the subspace of $\mathbb{F}_q^{|V|}$ (with the basis as the characteristic functions of vectors in V) of all reduced n-variable polynomial functions (over $\mathbb{F}_q$, reduced modulo $x_i^q - x_i$) of degree at most k.*

The following fact can be found in [PH98].

Lemma 3.2 $\mathrm{GRM}_p(k,n)$ *is a linear code with block length p^n and minimum distance $(R+1)p^Q$ where R is the remainder and Q the quotient resulting from dividing $(p-1)\cdot n-k$ by $(p-1)$. Denote the dual of a code $\mathcal{C}$ by $\mathcal{C}^\perp$. Then* $\mathrm{GRM}_p(k,n)^\perp = \mathrm{GRM}_p((p-1)\cdot n-k-1,n)$.

We begin with a few simple observations about flats. Note that an l-flat L is the intersection[5] of $(n-l)$ hyperplanes. Equivalently it consists of all points v which satisfy $(n-l)$ linear equations over $\mathbb{F}_p$ (i.e. one equation for each hyperplane): $\forall i \in [n-l]$ $\sum_{j=1}^{n} c_{ij}x_j = b_i$ where c_{ij}, b_i defines the i^{th} hyperplane (i.e., v satisfies $\sum_{j=1}^{n} c_{ij}v_j = b_i$). We mention here that we assume that the matrix c_{ij} has rank $(n-l)$. Note that then the incidence vector of L can be written as

$$\prod_{i=1}^{n-l}(1-(\sum_{j=1}^{n} c_{ij}x_j - b_i)^{p-1}) = \begin{cases} 1 & \text{if } (v_1,\cdots,v_l)\in L \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We record a lemma here that will be used later in this section. We leave the proof as a straightforward exercise.

Lemma 3.3 *For $l \geq k$, the incidence vector of any l flat is a linear sum of the incidence vectors of k-flats.*

We give an explicit basis for $\mathrm{GRM}_p(r,n)$. For the special case of $p=3$, our basis coincides[6] with the min-weight basis given in [DK00]. However, in general, our basis differs from the min-weight basis provided in [DK00].

The following lemma shows that the incidence vectors of flats are equivalent to that of Generalized Reed-Muller code of order multiples of $(p-1)$.

Proposition 3.4 $\mathrm{GRM}_p((p-1)(n-l),n)$ *is generated by the incidence vectors of the l-flats.*

Proof: We first show that the incidence vectors of the l-flats are in $\mathrm{GRM}_p((p-1)(n-l),n)$. Recall that L is the intersection of $(n-l)$ independent hyperplanes. Therefore using Equation 6, L can be represented by a polynomial of degree at most $(n-l)(p-1)$ in $x_1,\cdots,x_n$. Therefore the incidence vectors of l-flats are in $\mathrm{GRM}_p((p-1)(n-l),n)$.

We prove that $\mathrm{GRM}_p((p-1)(n-l),n)$ is generated by l-flats by induction on $n-l$. When $n-l=0$, the code consists of constants, which is clearly generated by n-flats i.e., the whole space.

To prove for an arbitrary $(n-l) > 0$, we show that any monomial of total degree $d \leq (p-1)(n-l)$ can be written as a linear sum of the incidence vectors of l-flats. Let the monomial be $x_1^{e_1}\cdots x_t^{e_t}$. Rewrite the monomials as $\underbrace{x_1\cdots x_1}_{e_1 \text{ times}}\cdots\underbrace{x_t\cdots x_t}_{e_t \text{ times}}$. Group into products of $(p-1)$ (not necessarily distinct) variable as much as possible. Rewrite each group using Equation 5 setting $k=(p-1)$. For any incomplete group of size d', use the same equation by setting the last $(p-1-d')$ variables to the constant 1. After expansion, the monomial can be seen to be a sum of product of at most (n-l) degree $(p-1)^{\text{th}}$ powered linear terms. We can add to it a polynomial of degree at most $(p-1)(n-l-1)$ so as to represent the resulting polynomial as a sum of polynomials, each polynomial as in Equation 6. Each such non-zero polynomial is generated by a t flat, $t \geq l$. By induction, the polynomial we added is generated by $(l+1)$ flats. Thus, by Lemma 3.3 our given monomial is generated by l-flats. ■

5 We here disregard degenerate cases.

6 The equations of the hyperplanes are slightly different in our case; nonetheless, both of them define the same basis generated by the min-weight codewords.

This leads to the following observation.

Observation 3.5 *Consider an l-flat generated by $y_1, \cdots, y_l$ at b. Denote the incidence vector of this flat by I. Then the right hand side of Equation 2 may be identified as $I \cdot f$, where I and f denote the vector corresponding to respective codewords and $\cdot$ is the scalar product.*

To generate Generalized Reed-Muller code of any arbitrary order, we need another important geometric object, namely pseudoflat, which we define next. A k-pseudoflat is a union of $(p-1)$ parallel $(k-1)$-flats. Also, a k-pseudoflat can have different exponents ranging from 1 to[7] $(p-2)$. We stress that the point set of a k-pseudoflat remains the same irrespective of its exponent. It is the value assigned to a point that changes with the exponents.

Definition 3.6 *Let $L_1, L_2, \cdots, L_{p-1}$ be parallel $(k-1)$ flats ($k \geq 1$), such that for some $y \in \mathbb{F}_p^n$ and all $t \in [p-2]$, $L_{t+1} = y + L_t$. We define the* `points` *of k-pseudoflat L with any exponent r ($1 \leq r \leq p-2$) to be the union of the set of points L_1 to L_{p-1}. Also, let I_j be the incidence vector of L_j for $j \in [p-1]$. Then the* `evaluation vector` *of this k*`-pseudoflat with exponent` *r is defined to be $\sum_{j=1}^{p-1} j^r I_j$.*

Let L be a k-pseudoflat. Also, for $j \in [p-1]$, let L_j be the $(k-1)$-flat generated by $y_1, \cdots, y_{k-1}$ at $b + j \cdot y$, where $y, y_1, \cdots, y_{k-1}$ are linearly independent. Then we say that L, a k-pseudoflat with exponent r, is generated by $y, y_1, \cdots, y_{k-1}$ at b exponentiated along y. The points in a k-pseudoflat may alternatively be viewed as the space given by the union of intersections of $(n-k-1)$ hyperplanes, where the union is parameterized by another hyperplane and which does not take one particular value. Concretely, it is the set of points v which satisfy the following constraints over $\mathbb{F}_p$:

$$\forall i \in [n-k-1] \quad \sum_{j=1}^{n} c_{ij} x_j = b_i; \text{ and}$$

$$\sum_{j=1}^{n} c_{n-k,j} x_j \neq b_{n-k}.$$

Thus the values taken by the points of a k-pseudoflat with exponent r is given by the polynomial

$$\prod_{i=1}^{n-k-1} (1 - (\sum_{j=1}^{n} c_{ij} x_j - b_i)^{(p-1)}) \cdot (\sum_{j=1}^{n} c_{n-k,j} x_j - b_{n-k})^r \tag{7}$$

7 With slight abuse, a k-pseudoflat with exponent zero corresponds to a flat.

Remark 3.7 *Note the difference between Equation 7 and the basis polynomial in [DK00] which (along with the action of the affine general linear group) yields the min-weight codewords:*

$$h(x_1, \cdots, x_m) = \prod_{i=1}^{k-1} (1 - (x_i - w_i)^{(p-1)}) \prod_{j=1}^{r} (x_k - u_j).$$

The next lemma shows that the code generated by the evaluation vectors of l-pseudoflats with exponent r is a subcode of that of l-pseudoflats with exponent $(r+1)$. Intuitively, a decrease in exponent refines the code.

Claim 3.8 *The evaluation vectors of l-pseudoflats with exponent $(r+1)$ ($1 \leq r \leq p-3$, $p > 3$) generate a code containing the evaluation vectors of l-pseudoflats with exponent r. Moreover, the evaluation vectors of l-pseudoflats with exponent one generate a code containing the incidence vectors of l-flats.*

Proof: For the first part, let W be the evaluation vector of the l-pseudoflat with exponent r generated by $y_1, \cdots, y_l$, exponentiated along y_1 at b. Clearly $W = \langle 0, 1^r, \cdots, (p-1)^r \rangle$ are the values taken by subflats generated by $y_1, \cdots, y_l$ at $b, b+y_1, \cdots, b-y_1$. Let this denote the standard basis. Let L_t be the be the pseudoflat with exponent $(r+1)$ generated by $y_1, \cdots, y_l$ exponentiated at y_1 at $b + t \cdot y_1$, for each $t \in [p-1] \cup \{0\}$, and let V_t be the corresponding evaluation vector. Rewriting them in the standard basis yields that $V_t = \langle ((p-1)-t+1)^{r+1}, ((p-1)-t+2)^{r+1}, \cdots, ((p-1)-t+(p-1)+1)^{r+1} \rangle \in \mathbb{F}_p^p$. Let λ_t denote p variables for $t = 0, 1, \cdots, (p-1)$, each taking values in $\mathbb{F}_p$. Then a solution to the following system of equations

$$\forall j \in [p-1] \cup \{0\} \quad j^r = \sum_{i=0}^{(p-1)} \lambda_i (j-i)^{r+1} \tag{8}$$

implies that $W = \sum_{i=0}^{(p-1)} \lambda_i V_i$, which suffices to establish the claim. Consider the identity ($0 \leq r < p-1$)

$$j^r = \frac{-1}{r+1} \sum_{i=0}^{p-1} (j+i)^{r+1} i^{p-2}$$

which may be verified by expanding and applying Lemma 2.2. This establishes the first part of the claim.

For the second part, observe that $\langle 1, \cdots, 1 \rangle = \langle 0, 1, 2, \cdots, -1 \rangle - \langle -1, 0, 1, \cdots, -2 \rangle$. This completes the proof. ∎

The next Proposition complements Proposition 3.4. Together they say that by choosing dimension and exponent appropriately, Generalized Reed-Muller code of any given order can be generated. This gives an equivalent representation of Generalized Reed-Muller code. An exact characterization then comes out from this alternate representation.

Proposition 3.9 *For every $r \in [p-2]$, the linear code generated by the evaluation vectors of l-pseudoflats with exponent r is equivalent to* $\mathrm{GRM}_p((p-1)(n-l)+r, n)$.

Proof: For the forward direction, consider an l-pseudoflat L with exponent r. Its evaluation vector is given by an equation similar to Equation 7. Thus the codeword corresponding to the evaluation vector of this flat can be represented by a polynomial of degree at most $(p-1)(n-l)+r$. This completes the forward direction.

To prove the other direction, we induct on r. In the base case, $r = 1$, following Proposition 3.4 and Claim 3.8 we restrict our attention to monomials of degree exactly $((p-1)(n-l)+1)$. We show that these monomials are generated by l-pseudoflats with exponent one. Now consider any such monomial. Rewrite it as in Proposition 3.4. Group $(p-1)$ terms and rewrite each group using Equation 5 setting $k = (p-1)$. Since the degree of the monomial is $(p-1)(n-l)+1$, we will be left with a linear term. We can add to this a polynomial of degree at most $(p-1)(n-l-1)+1 \leq (p-1)(n-l)$ to express it as in Equation 7 with $r = 1$. By Proposition 3.4 and Claim 3.8, this additional polynomial is generated by l-pseudoflats of exponent 1 as well, and we are done.

For the inductive step, we again restrict our attention to monomials of degree exactly $((p-1)(n-l)+r)$ by induction hypothesis. We show that these monomials are generated by l-pseudoflats with exponent r. Consider any such monomial. Rewrite it as in Proposition 3.4. Since the degree of the monomial is $(p-1)(n-l)+r$, we will be left with an incomplete group of degree r. We can add to this a polynomial of degree at most $(p-1)(n-l-1)+r \leq (p-1)(n-l)+(r-1)$ to express it as in Equation 7. By the inductive hypothesis and Claim 3.8, this additional polynomial is generated by l-pseudoflats of exponent r as well. This completes the proof. ∎

The following is analogous to Observation 3.5.

Observation 3.10 *Consider an l-pseudoflat with exponent r, generated by $y_1, \cdots, y_l$ at b exponentiated along y_1. Let E be the evaluation vector of this pseudoflat with exponent r. Then the right hand side of Equation 3 may be interpreted as $E \cdot f$.*

We are now ready to present the exact characterization.

Theorem 3.11 *Let $t = (p-1) \cdot k + R$. (Note $0 \leq R \leq p-2$.) Let $r = p-2-R$. Then a function f belongs to $\mathcal{P}_t$, if and only if* `for every` $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$, *we have*

$$T_f(y_1, \cdots, y_{k+1}, b) = 0 \quad \textit{if } R = p-2; \tag{9}$$

$$T_f^r(y_1, \cdots, y_{k+1}, b) = 0 \quad \textit{otherwise.} \tag{10}$$

Proof: Denote $T_f(y_1, \cdots, y_{k+1}, b) = T_f^0(y_1, \cdots, y_{k+1}, b)$. We first show that if $f \in \mathcal{P}_t$ then $T_f^r(y_1, \cdots, y_{k+1}, b) = 0$ `for every` $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$. Fix $y_1, \cdots, y_{k+1}, b$. We show that

$$T_f^r(y_1, \cdots, y_{k+1}, b) = \sum_{c=(c_1,\cdots,c_{k+1}) \in \mathbb{F}_p^{k+1}} c_1^r f(b + \sum_{i=1}^{k+1} c_i y_i)$$
$$= 0.$$

Consider the polynomial $h(c_1, \cdots, c_{k+1}) = c_1^r f(b + \sum_{i \in [k+1]} c_i y_i)$. Every monomial m of h is of the form $m = a_m c_1^{r+q_1} c_2^{q_2} \cdots c_{k+1}^{q_{k+1}}$ for some $a_m \in \mathbb{F}_p^*$, where $\sum_{i=1}^{k+1} q_i + r \leq t + r < (p-1)(k+1)$. Note for any $i > 1$, $q_i = 0$ implies that the sum is zero (by summing over c_i). Similarly if $q_1 + r = 0$, the sum evaluates to zero. Therefore consider the case when each exponent is non-zero. But then there exists one j such that $q_j < (p-1)$. Thus by Lemma 2.3, $\sum_{(c_1,\cdots,c_{k+1}) \in \mathbb{F}_p^{k+1}} m = 0$. This implies that $T_f^r(y_1, \cdots, y_{k+1}, b) = \sum_{(c_1,\cdots,c_{k+1}) \in \mathbb{F}_p^{k+1}} h(c_1, \cdots, c_{k+1}) = 0$.

We next show that if $f \notin \mathcal{P}_t$ then there exist a group of vectors $y_1', \cdots, y_{k+1}'$ and b' such that $T_f^r(y_1', \cdots, y_{k+1}', b') \neq 0$. Let f be a polynomial in $x_1, \cdots, x_n$ and m be an arbitrary monomial in f of the maximum total degree (which is at least $t+1$). Then define the polynomial $\ell(x_1, \cdots, x_n)$ by $\ell(x_1, \cdots, x_n) = \prod_{i=1}^n x_i^{(p-1)}/m$. Clearly, $\ell(x_1, \cdots, x_n)$ has degree at most $(p-1)(n-k-1)+r$. Therefore, ℓ belongs to $\mathrm{GRM}_p((p-1)(n-k-1)+r, n)$.

Consider the case $r = 0$. By Proposition 3.4, ℓ is generated by $(k+1)$-flats. Moreover, by Lemma 2.3, $\sum_{x_1,\cdots,x_n} \ell(x_1, \cdots, x_n) m(x_1, \cdots, x_n) \neq 0$. Also, the sum with m replaced by any other monomial m' in f, yields zero[8]. Therefore, by linearity, we obtain $f \cdot \ell \neq 0$, where the function has been identified with corresponding codeword. Since ℓ is a linear sum of $(k+1)$-flats, there exists a $(k+1)$-flat (generated by $y_1', \cdots, y_{k+1}'$ at b') such that when f is summed up over the flat, it yields a non-zero value.

The case $r \neq 0$ is similar with pseudoflats replacing flats. ∎

Remark 3.12 *The above claim may directly be obtained from Lemma 3.2, Proposition 3.4, Proposition 3.9, Observation 3.5 and Observation 3.10. The above proof of Theorem 3.11 has been given as an alternate self contained proof.*

Remark 3.13 *One can describe an alternate characterization from Remark 3.7 which we state here without proof.*

---

8 By Lemma 2.3 a non-zero sum implies $m \cdot \ell = x_1^{(p-1)} \cdots x_n^{(p-1)} = m' \cdot \ell$, which implies $m = m'$ or $m' = m \prod_{i \in I; \emptyset \neq I \subseteq [n]} x_i^{(p-1)}$, a contradiction on the choice of m.

Let $t = (p-1)\cdot k + R$. (Note $0 < R \le (p-2)$.) Let $r = (p-1) - R - 1$. Let $W \subseteq \mathbb{F}_p$ with $|W| = r$. Define the polynomial $g(x) \stackrel{def}{=} \prod_{\alpha \in W}(x-\alpha)$ if W is non-empty; and $g(x) = 1$ otherwise. Then a function belong to $\mathcal{P}_t$ if and only if for every $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$, we have

$$\sum_{c_1 \in \mathbb{F}_p \setminus W} g(c_1) \sum_{(c_2,\cdots,c_{k+1}) \in \mathbb{F}_p^k} f(b + c_1 \cdot y_1 + \sum_{i=2}^{k+1} c_i \cdot y_i) = 0.$$

Moreover, this characterization can also be extended to certain degrees for more general fields, i.e., $\mathbb{F}_{p^t}$ (see the next remark).

Remark 3.14 *The exact characterization of low degree polynomials as claimed in [FS95] may be proved using duality. Note that their proof works as long as the dual code has a min-weight basis (see [DK00]). Suppose that the polynomial has degree $d \le q - q/p - 1$, then the dual of $\mathrm{GRM}_q(d,n)$ is $\mathrm{GRM}_q((q-1)n - d - 1, n)$ and therefore has a min-weight basis. Note that then the dual code has min-weight $(d+1)$. Therefore, assuming the minimum weight codewords constitute a basis, any $d+1$ evaluations of the original polynomial are dependent and vice-versa. We leave the details as an exercise for interested readers.*

4. A Tester for Low Degree Polynomials over $\mathbb{F}_p^n$

In this section we present and analyze a one-sided tester for $\mathcal{P}_t$. The analysis of the algorithm roughly follows the proof structure given in [RS96, AKK+03] and most details are omitted due to lack of space. We emphasize that the generalization from [AKK+03] to our case is not straightforward. As in [RS96, AKK+03] we define a self-corrector version of the (possibly corrupted) function being tested. The straightforward adoption of the analysis given in [RS96] gives a reasonable bound. However, a better bound is achieved by following the techniques developed in [AKK+03]. In there, they show that the self-corrector function can be interpolated with overwhelming probability. However their approach appears to use special properties of $\mathbb{F}_2$ and hence is not clear how to generalize. We give a clean formulation which relies on the flats being represented through polynomials as described earlier. In particular, Claims 4.7, 4.8 and their generalization appear to require our new polynomial based view.

4.1. Tester in $\mathbb{F}_p$

In this subsection we describe the algorithm when the underlying field is $\mathbb{F}_p$.

Algorithm Test-$\mathcal{P}_t$ in $\mathbb{F}_p$

0. Let $t = (p-1)\cdot k + R$, $0 \le R < (p-1)$. Denote $r = p - R - 2$.
1. Uniformly and independently select $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$.
2. If $T_f^r(y_1, \cdots, y_{k+1}, b) \ne 0$, then `reject`, else `accept`.

Theorem 4.1 *The algorithm* **Test-$\mathcal{P}_t$ in** *$\mathbb{F}_p$ is a one-sided tester for $\mathcal{P}_t$ with a success probability at least min$(\Omega(p^{k+1}\epsilon), \frac{1}{(\frac{t}{p}+6)p^{k+3}})$.*

Corollary 4.2 *Repeating the algorithm* **Test-$\mathcal{P}_t$ in** *$\mathbb{F}_p$ $\Theta(\frac{1}{p^{k+1}\epsilon} + kp^k)$ times, the probability of error can be reduced to less than $1/2$.*

We will provide a general proof framework. However, we content ourselves by proving main technical lemmas for the case of $\mathbb{F}_3$. The proof idea in the general case is similar and the details are omitted. Therefore we will essentially prove the following.

Theorem 4.3 *The algorithm* **Test-$\mathcal{P}_t$ in** *$\mathbb{F}_3$ is a one-sided tester for $\mathcal{P}_t$ with success probability at least min$(\Omega(3^{k+1}\epsilon), \frac{1}{(t+7)3^{t/2+2}})$.*

4.2. Analysis of Algorithm *Test*-$\mathcal{P}_t$

In this subsection we analyze the algorithm described in Section 4.1. From Theorem 3.11 it is clear that if $f \in \mathcal{P}_t$, then the tester accepts. Thus, the bulk of the proof is to show that if f is ϵ-`far` from $\mathcal{P}_t$, then the tester (repeated sufficient number of times as in Corollary 4.2) rejects with probability at least $\frac{1}{2}$. Our proof structure follows that of the analysis of the test in [AKK+03]. In what follows, we will denote $T_f(y_1, \cdots, y_l, b)$ by $T_f^0(y_1, \cdots, y_l, b)$ for ease of exposition. In particular, let f be a function to be tested for membership in $\mathcal{P}_t$. Assume we calculate T_f^i for an appropriate i as required by the algorithm described in Section 4.1. For such an i, we define $g_i : \mathbb{F}_p^n \to \mathbb{F}_p$ as follows: For $y \in F_p^n, \alpha \in \mathbb{F}_p$, denote $p_{y,\alpha} = \Pr_{y_1,\cdots,y_{k+1}}[f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1) = \alpha]$. Define $g_i(y) = \alpha$ such that $\forall \beta \ne \alpha \in \mathbb{F}_p, p_{y,\alpha} \ge p_{y,\beta}$ with ties broken arbitrarily. With this meaning of plurality, for all $i \in [p-2] \cup \{0\}$, g_i can be written as:

$$g_i(y) = \text{plurality}_Y \left[f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1) \right] \tag{11}$$

where $Y = \langle y_1, \cdots, y_{k+1} \rangle$.

Further we define

$$\eta_i \stackrel{def}{=} Pr_{y_1,\cdots,y_{k+1},b}[T_f^i(y_1, \cdots, y_{k+1}, b) \ne 0] \tag{12}$$

The next lemma follows from a Markov-type argument.

Lemma 4.4 *[RS96] For a fixed $f : \mathbb{F}_p^n \to \mathbb{F}_p$, let g_i, η_i be defined as above. Then, $\delta(f, g_i) \le 2\eta_i$.*

Proof: Consider the set of elements y such that $\Pr_{y_1,\cdots,y_{k+1}}[f(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] < 1/2$. If the fraction of such elements is more than $2\eta_i$ then that contradicts the condition that

$$\begin{aligned}\eta_i &= \Pr_{y_1,\cdots,y_{k+1},b}[T_f^i(y_1,\cdots,y_{k+1},b) \neq 0] \\ &= \Pr_{y_1,y_2,\cdots,y_{k+1},b}[T_f^i(y_1 - b, y_2, \cdots, y_{k+1}, b) \neq 0] \\ &= \Pr_{y,y_1,\cdots,y_{k+1}}[f(y) \neq f(y) - \\ &\qquad T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)].\end{aligned}$$

Therefore, we obtain $\delta(f, g_i) \le 2\eta_i$. ∎

Note that $\Pr_{y_1,\cdots,y_{k+1}}[g_i(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] \ge \frac{1}{p}$. We now show that this probability is actually much higher. The next lemma gives a weak bound in that direction following the analysis in [RS96].

Lemma 4.5 $\forall y \in \mathbb{F}_p^n$, $\Pr_{y_1,\cdots,y_{k+1} \in \mathbb{F}_p^n}[g_i(y) = f(y) - T_f^i(y - y_1, \cdots, y_{k+1}, y_1)] \ge 1 - 2p^{k+1}\eta_i$.

However, when the degree being tested is larger than the field size, we can improve the above lemma considerably. The following lemma strengthens Lemma 4.5 when $t \ge (p-1)$ or equivalently $k \ge 1$.

Lemma 4.6 $\forall y \in \mathbb{F}_3^n$, $\Pr_{y_1,\cdots,y_{k+1} \in \mathbb{F}_3^n}[g_i(y) = f(y) - T_f^i(y - y_1, \cdots, y_{k+1}, y_1)] \ge 1 - \frac{3}{2}(4k + 14)\eta_i$.

In order to prove Lemma 4.6, we will need the following claims. They can easily be verified by expanding the terms on both sides like the proof of Claim 4 in [AKK+03]. However, this does not give much insight into the general case i.e. for $\mathbb{F}_p$. We provide an alternate proof which can be generalized to get similar claims and has a much cleaner structure based on the underlying geometric structure, i.e. flats or pseudoflats.

Claim 4.7 *For every $i \in \{2, \cdots, k+1\}$, for every $y(= y_1), z, w, b, y_2, \cdots, y_{i-1}, y_{i+1}, \cdots, y_{k+1} \in \mathbb{F}_3^n$,* $T_f(y, y_2, \cdots, y_{i-1}, w, y_{i+1}, \cdots, y_{k+1}, b) - T_f(y, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b) =$
$T_f(y + w, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b) +$
$T_f(y - w, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b) -$
$T_f(y + z, y_2, \cdots, y_{i-1}, w, y_{i+1}, \cdots, y_{k+1}, b) -$
$T_f(y - z, y_2, \cdots, y_{i-1}, w, y_{i+1}, \cdots, y_{k+1}, b)$.

Proof: Assume y, z, w are independent. Observe that it is enough to prove the result for $k = 1$ and $b = \mathbf{0}$. Consider the space $\mathcal{H}$ generated by y, z and w at $\mathbf{0}$. Note that $T_f(y, w, \mathbf{0})$ is just $f \cdot 1_L$, where 1_L is the incidence vector of the flat given by the equation $z = 0$. Therefore 1_L is equivalent to the polynomial $(1 - z^2)$. Similarly $T_f(y, z) = f \cdot 1_{L'}$ where L' is given by the equation $(1 - w^2)$. We use the following polynomial identity (in $\mathbb{F}_3$) $w^2 - z^2 = [1 - (y - w)^2 + 1 - (y + w)^2] - [1 - (y + z)^2 + 1 - (y - z)^2]$. Now observe that the equation $(1 - (y - w)^2)$ is the incidence vector of the flat generated by $y + w$ and z. Similar observations hold for other terms. Therefore, interpreting the above equation in terms of incidence vectors of flats, we complete the proof with Observation 3.5. ∎

Proof sketch of Lemma 4.6: We prove the lemma for $g_0(y)$. We fix $y \in \mathbb{F}_3^n$ and let $\gamma \stackrel{def}{=} \Pr_{y_1,\cdots,y_{k+1} \in \mathbb{F}_3^n}[g_0(y) = T_f^y(y_1, \cdots, y_{k+1})]$. As in [AKK+03, RS96] we bound the following collision probability.

$$\begin{aligned}\mu &\stackrel{def}{=} \Pr_{Y,Z \in \mathbb{F}_3^n}[T_f(y - y_1, \cdots, y_{k+1}, y_1) \\ &\qquad -T_f(y - z_1, \cdots, z_{k+1}, z_1) = 0] \quad (13)\end{aligned}$$

where $W = \langle w_1, \cdots, w_{k+1} \rangle$. It can be shown that $\mu \le \gamma^2 + (1-\gamma)^2$.

We rewrite $T_f(y - y_1, \cdots, y_{k+1}, y_1) - T_f(y - z_1, \cdots, z_{k+1}, z_1)$ as a telescopic sum as in [AKK+03].

Any pair that appears in the telescopic sum can further be rewritten using Claim 4.7 so that all the input parameters of $T_f(\cdot)$ on the right hand side are independent and uniformly distributed. By union bound it follows then that $(1 - \mu) \le (4k + 10)\eta_0 \le (4k + 14)\eta_0$. Furthermore following the definition of μ, it can be shown that $\mu \le \gamma^2 + (1-\gamma)^2$. Rearranging gives us $2\gamma(1-\gamma) \le 1 - \mu$. Since $\gamma \ge \frac{1}{3}$, we get $(1 - \mu) \ge 2\gamma(1-\gamma) \ge \frac{2}{3}(1-\gamma)$. Therefore $\gamma \ge 1 - \frac{3}{2}(1-\mu) \ge 1 - \frac{3}{2}(4k + 14)\eta_0$.

A similar argument along with Claim 4.8 proves the lemma for $i = 1$. We remark that the the above proof uses the fact that $T_f(\cdot)$ is symmetric in all but the last input which is not true for T_f^1. We solve this problem with another identity similar to Claim 4.8. We leave the details. ∎

Claim 4.8 *For every $i \in \{2, \cdots, k+1\}$, for every $y(= y_1), z, w, b, y_2, \cdots, y_{i-1}, y_{i+1}, \cdots, y_{k+1} \in \mathbb{F}_3^n$,* $T_f^1(y, y_2, \cdots, y_{i-1}, w, y_{i+1}, \cdots, y_{k+1}, b) - T_f^1(y, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b) =$
$T_f^1(y + z, y_2, \cdots, y_{i-1}, w, y_{l+1}, \cdots, y_{k+1}, b) +$
$T_f^1(y - z, y_2, \cdots, y_{i-1}, w, y_{i+1}, \cdots, y_{k+1}, b) -$
$T_f^1(y + w, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b) -$
$T_f^1(y - w, y_2, \cdots, y_{i-1}, z, y_{i+1}, \cdots, y_{k+1}, b)$.

Proof: Note here that the defining equation of $T_f^1(y, z)$ is $y(1 - w^2)$. Now consider the following identity in $\mathbb{F}_3$:

$$\begin{aligned}y(z^2 - w^2) &= (y+w)[1-(y-w)^2] + (y-w)[1-(y+w)^2] \\ &\quad -(y+z)[1-(y-z)^2] - (y-z)[1-(y+z)^2]\end{aligned}$$

for variables $y, z, w \in \mathbb{F}_3$. Rest of the proof is similar to the proof of Claim 4.7 (the proof replaces flats by pseudoflats) and is omitted. ∎

The next lemma shows that sufficiently small η_i implies that g_i self-corrects the function f.

Lemma 4.9 *Over $\mathbb{F}_3$, if $\eta_i < \frac{1}{(2k+7)3^{k+2}}$, then the function g_i belongs to $\mathcal{P}_t$ (assuming $k \geq 1$).*

The proof idea follows from [RS96, AKK$^+$03]. We omit the details.

Remark 4.10 *Over $\mathbb{F}_p$ we have: if $\eta_i < \frac{p^{-(k+2)}}{((p-1)k+6(p-1)+1)}$, then g_i belongs to $\mathcal{P}_t$ (if $k \geq 1$).*

By combining Lemma 4.4 and Lemma 4.9 we obtain that if f is $\Omega(1/(k3^k))$-far from $\mathcal{P}_t$ then $\eta_i = \Omega(1/(k3^k))$. We next consider the case in which η_i is small. By Lemma 4.4, in this case, the distance $\delta = \delta(f, g)$ is small. The next lemma shows that in this case the test rejects f with probability that is close to $3^{k+1}\delta$. This follows from the fact that in this case, the probability over the selection of $y_1, \cdots, y_{k+1}, b$, that among the 3^{k+1} points $\sum_i c_i y_i + b$, the functions f and g differ in precisely one point, is close to $3^{k+1} \cdot \delta$.

Lemma 4.11 *Suppose $0 \leq \eta_i \leq \frac{1}{(2k+7)3^{k+2}}$. Let δ denote the relative distance between f and g, $\ell = 3^{k+1}$, and $Q \stackrel{def}{=} (\frac{1-\ell\delta}{1+\ell\delta}) \cdot \ell\delta$. Then, when $y_1, \cdots, y_{k+1}, b$ are chosen randomly, the probability that for exactly one point v among the ℓ points $\{\sum_{i=1}^{k+1} c_i y_i + b\}_{c \in \mathbb{F}_p^{k+1}}$, $f(v) \neq g(v)$ is at least Q.*

Note that the points $\{\sum_{i=1}^{k+1} c_i y_i + b\}_{c \in \mathbb{F}_p^{k+1}}$ are pairwise independent. We use an inequality from [AKK$^+$03] to complete the proof. We omit the details.

Observe that $\eta_i = \Omega(Q) = \Omega(3^{k+1}\delta)$.

Proof of Theorem 4.3: Clearly if f belongs to $\mathcal{P}_t$, then by Theorem 3.11 the tester accepts f with probability 1.

Therefore let $\delta(f, \mathcal{P}_t) \geq \epsilon$. Let $d = \delta(f, g)$. If $\eta < \frac{1}{(2k+7)3^{k+2}}$ then by Lemma 4.9 $g \in \mathcal{P}_t$ and, by Lemma 4.11, $\eta_i = \Omega(3^{k+1} \cdot d) = \Omega(3^{k+1}\epsilon)$. Hence $\eta_i \geq min\left(\Omega(3^{k+1}\epsilon), \frac{1}{(2k+7)3^{k+2}}\right)$. ■

5. Self-Correcting and a Lower Bound

From Lemmas 4.4, 4.6 and 4.9 the following corollary is immediate:

Corollary 5.1 *Consider a function $f : \mathbb{F}_3^n \to \mathbb{F}_3$ that is ϵ-close to a degree-t polynomial $g : \mathbb{F}_3^n \to \mathbb{F}_3$ where $\epsilon < \frac{1}{(2k+7)3^{k+2}}$ (assume $k \geq 1$). Then the function f can be self-corrected. That is, for any $x \in \mathbb{F}_3^n$, the value of $g(x)$ can be obtained with probability at least $1 - 3^{k+1}\epsilon$ by querying f on 3^{k+1} points on $\mathbb{F}_3^n$.*

An analogous result may be obtained for the general case. We, however, improve the above corollary slightly. The above corrector does not allow any errors in the 3^{k+1} points it queries. We obtain a stronger result by querying on a slight larger flat H, but allowing some errors. Errors are handled by decoding the induced Generalized Reed-Muller code code on H.

Proposition 5.2 *Consider a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ that is ϵ-close to a degree-t polynomial $g : \mathbb{F}_p^n \to \mathbb{F}_p$. Then the function f can be self-corrected. That is, assume $K > (k+1)$, then for any given $x \in \mathbb{F}_p^n$, the value of $g(x)$ can be obtained with probability at least $1 - \frac{\epsilon}{(1-\epsilon \cdot p^{k+1})^2} \cdot p^{-(K-2k-3)}$ from queries to f.*

Proof: Our goal is to correct the $GRM_p(t, n)$ at the point x. Assume $t = (p-1) \cdot k + R$, where $0 \leq R \leq (p-2)$. Then the relative distance of the code δ is $(1 - R/p)p^{-k}$. Note that $2p^{-k-1} \leq \delta \leq p^{-k}$. Recall that the local testability test requires a $(k+1)$-flat, i.e., it tests $\sum_{c_1, \cdots, c_{k+1} \in \mathbb{F}_p} c_1^{p-2-R} f(y_0 + \sum_{i=1}^{k+1} c_i y_i) = 0$, where $y_i \in \mathbb{F}_p^n$.

We choose a slightly larger flat, i.e., a K-flat with $K > (k+1)$ to be chosen later. We consider the code restricted to this K-flat with point x being the origin. We query f on this K-flat. It is known that a majority logic decoding algorithm exists that can decode Generalized Reed-Muller code up to half the minimum distance for any choice of parameters (see [Sud01]). Thus if the number of error is small we can recover $g(x)$.

Let the relative distance of f from the code be ϵ and let S be the set of points where they disagree. Let the random K-flat be $H = \{x + \sum_{i=1}^{K} t_i u_i | t_i \in \mathbb{F}, u_i \in_R \mathbb{F}_p^n\}$.
Let the random variable $Y_{\langle t_1, \cdots, t_K \rangle}$ take the value 1 if $x + \sum_{i=1}^{K} u_i t_i \in S$ and 0 otherwise. Let $D = \mathbb{F}^K \setminus \{0\}$ and $U = \langle u_1, \cdots, u_K \rangle$. Define $Y = \sum_{\langle t_1, \cdots, t_K \rangle \in D} Y_{\langle t_1, \cdots, t_K \rangle}$ and $\ell = (p^K - 1)$. We would like to bound the probability

$$\Pr_U[|Y - \epsilon\ell| \geq (\delta/2 - \epsilon)\ell].$$

Since $\Pr_U[Y_{t_1, \cdots, t_K} = 1] = \epsilon$, by linearity we get $\mathrm{E}_U[Y] = \epsilon\ell$. Let $T = \langle t_1, \cdots, t_K \rangle$. Now

$$\begin{aligned} Var[Y] &= \sum_{T \in \mathbb{F}^K - \{0\}} Var[Y_T] + \sum_{T \neq T'} Cov[Y_T, Y_{T'}] \\ &= \ell(\epsilon - \epsilon^2) + \sum_{T \neq \lambda T'} Cov[Y_T, Y_{T'}] \\ &\quad + \sum_{T = \lambda T'; 1 \neq \lambda \in \mathbb{F}^*} Cov[Y_T, Y_{T'}] \\ &\leq \ell(\epsilon - \epsilon^2) + \ell \cdot (p-2)(\epsilon - \epsilon^2) \\ &= \ell(\epsilon - \epsilon^2)(p-1) \end{aligned}$$

The above follows from the fact that when $T \neq \lambda T'$ then they are independent and therefore $Cov[Y_T, Y_{T'}] = 0$. Also, when Y_T and $Y_{T'}$ are dependent then $Cov[Y_T, Y_{T'}] = \mathrm{E}_U[Y_T Y_{T'}] - \mathrm{E}_U[Y_T]\mathrm{E}_U[Y_{T'}] \leq \epsilon - \epsilon^2$.
Therefore, by Chebyshev's inequality we have (assuming

$\epsilon < p^{-(k+1)}$)

$$\Pr_U[|Y - \epsilon\ell| \geq (\delta/2 - \epsilon)\ell] \leq \frac{\ell\epsilon(1-\epsilon)(p-1)}{(\delta/2-\epsilon)^2\ell^2}$$

Now note $(\delta/2 - \epsilon) \geq (p^{-k-1} - \epsilon) = (1 - \epsilon \cdot p^{k+1})p^{-k-1}$. We thus have

$$\Pr_U[|Y - \epsilon\ell| \geq (\delta/2 - \epsilon)\ell] \leq \frac{\epsilon(1-\epsilon)(p-1)}{(1-\epsilon \cdot p^{k+1})^2 p^{-2k-2}\ell}$$

$$\leq \frac{\epsilon p}{(1-\epsilon \cdot p^{k+1})^2 p^{-2k-2}(\ell+1)}$$

$$= \frac{\epsilon}{(1-\epsilon \cdot p^{k+1})^2} \cdot p^{-(K-2k-3)}. \blacksquare$$

The next theorem is a simple modification of a theorem in [AKK+03] and essentially implies that our result is almost optimal.

Proposition 5.3 *Let $\mathcal{F}$ be any family of functions $f : \mathbb{F}_p^n \to \mathbb{F}_p$ that corresponds to a linear code $\mathcal{C}$. Let d denote the minimum distance of the code $\mathcal{C}$ and let $\bar{d}$ denote the minimum distance of the dual code of $\mathcal{C}$.*
Every one-sided testing algorithm for the family $\mathcal{F}$ must perform $\Omega(\bar{d})$ queries, and if the distance parameter ϵ is at most d/p^{n+1}, then $\Omega(1/\epsilon)$ is also a lower bound for the necessary number of queries.

Lemma 3.2 and Proposition 5.3 gives us the following corollary.

Corollary 5.4 *Every one-sided tester for testing $\mathcal{P}_t$ with distance parameter ϵ must perform $\Omega(max(\frac{1}{\epsilon}, (1 + ((t+1) \bmod (p-1)))p^{\frac{t+1}{p-1}}))$ queries.*

6. Conclusions

The lower bound in Corollary 5.4 implies that our upper bound is almost tight. We resolved the question posed in [AKK+03] for all prime fields. Although our method fails for general fields, [KR04] have resolved the problem.

References

[AKK+03] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing low-degree ploynomials over $\mathrm{GF}(2)$. In *Proc. of RANDOM 03*, 2003.

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the intractability of approximation problems. In *Proc. of IEEE Symposium of the Foundation of Computer Science*, pages 14–23, 1992.

[AS92] S. Arora and S. Safra. Probabilistic chekcing of proofs: A new characterization of NP. In *Proc. of IEEE Symposium of the Foundation of Computer Science*, pages 2–13, 1992.

[AS97] S. Arora and M. Sudan. Improved low-degree testing and its application. In *Proc. of Symposium on the Theory of Computing*, 1997.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two prover interactive protocols. In *Computational Complexity*, pages 3–40, 1991.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. of Symposium on the Theory of Computing*, pages 21–31, 1991.

[BLR93] 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.

[DGM70] P. Delsarte, J. M. Goethals, and F. J. MacWillams. On generalized reed-muller codes and their relatives. *Information and Control*, 16:403–442, 1970.

[DK00] P. Ding and J. D. Key. Minimum-weight codewords as generators of generalized reed-muller codes. *IEEE Trans. on Information Theory.*, 46:2152–2158, 2000.

[FGL+91] U. Fiege, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy. Approximating clique is almost np-complete. In *Proc. of IEEE Symposium of the Foundation of Computer Science*, pages 2–12, 1991.

[FS95] 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 at http://theory.lcs.mit.edu/∼madhu/papers/friedl.ps.

[GLR+91] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approxiamte functions. In *Proc. of Symposium on the Theory of Computing*, 1991.

[KR04] T. Kaufman and D. Ron. Testing polynomials over general fields. In *Proc. of IEEE Symposium of the Foundation of Computer Science*, 2004.

[PH98] V. S. Pless, Jr. and W. C. Huffman, editors. *Handbook of Coding Theory, Vol II*, chapter 16. Elsevier, 1998.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[Sud01] M. Sudan. Lecture notes on algorithmic introduction to coding theory, Fall 2001. Lecture 15.