LOCAL TESTING OF LATTICES *

KARTHEKEYAN CHANDRASEKARAN †, MAHDI CHERAGHCHI ‡, VENKATA GANDIKOTA§, AND ELENA GRIGORESCU ¶

Abstract.
Testing membership in lattices is of practical relevance, with applications to integer programming, error detection in lattice-based communication and cryptography. In this work, we initiate a systematic study of *local testing* for membership in lattices, complementing and building upon the extensive body of work on locally testable codes. In particular, we formally define the notion of local tests for lattices and present the following:

1. We show that in order to achieve low query complexity, it is sufficient to design one-sided non-adaptive *canonical* tests. This result is akin to, and based on an analogous result for error-correcting codes due to Ben-Sasson *et al.* (SIAM J. Computing 35(1) pp1–21).
2. We demonstrate upper and lower bounds on the query complexity of local testing for membership in *code formula* lattices. We instantiate our results for code formula lattices constructed from Reed-Muller codes to obtain nearly-matching upper and lower bounds on the query complexity of testing such lattices.
3. We contrast lattice testing to code testing by showing lower bounds on the query complexity of testing low-dimensional lattices. This illustrates large lower bounds on the query complexity of testing membership in the well-known *knapsack lattices*. On the other hand, we show that knapsack lattices with bounded coefficients have low-query testers if the inputs are promised to lie in the span of the lattice.

Key words. Lattices, Property Testing, Construction-D, Linear test, Locally testable codes

AMS subject classifications. 11H31, 94B05

1. Introduction. Local testing for properties of combinatorial and algebraic objects have widespread applications and have been intensely investigated in the past few decades. The main underlying goal in Local Property Testing is to distinguish objects that satisfy a given property from objects that are far from satisfying the property, using a small number of observations of the input object. Starting with the seminal works of [7, 12, 31], significant focus in the area has been devoted to locally testable error-correcting codes, called Locally Testable Codes (LTCs) [14]. LTCs are the key ingredients in several fundamental results in complexity theory, most notably in the PCP theorem [2, 3].

In this work we initiate the study of local testability for membership in *point lattices*, a class of infinite algebraic objects that form discrete subgroups of $\mathbb{R}^n$. Lattices are well-studied in mathematics, physics and computer science due to their rich algebraic structure [8]. Algorithms for various lattice problems have directly influenced the ability to solve integer programs [9, 22, 16]. Recently, lattices have found applications in modern cryptography due to attractive properties that enable efficient computations and security guarantees [26, 24, 29, 30]. Lattices are also used in practical communication settings to encode data in a redundant manner in order to protect it from channel noise during transmission [11].

A point lattice $L \subset \mathbb{R}^n$ of *rank* k and *dimension* n is specified by a set of linearly independent vectors $b_1, \ldots, b_k \in \mathbb{R}^n$ known as a basis, for some $k \leq n$. If $k = n$ the lattice is said to have full rank. The set L is defined to be the set of all vectors in $\mathbb{R}^n$ that are integer linear combinations of the basis vectors, i.e., $L := \{\sum_{i=1}^{k} \alpha_i b_i \mid \alpha_i \in \mathbb{Z} \ \forall\ i \in [k]\}$. Lattices are the analogues over $\mathbb{Z}$ of linear error-correcting codes over a finite field $\mathbb{F}$, which are generated as $\mathbb{F}$-linear combinations of a linearly independent set of basis vectors $b_1, \ldots, b_k \in \mathbb{F}^n$.

Given a basis for a lattice L, we are interested in testing if a given input $t \in \mathbb{R}^n$ belongs to L, or is far from all points in L by querying a small number of coordinates of t. We emphasize that this setting does not limit the computational space or time in pre-processing the lattice as well as the queried coordinates. The main goal is to design a tester that queries only a small number of coordinates of the input.

---

*The preliminary version of this work appeared in the Proceedings of FSTTCS'16.

†Email: karthe@illinois.edu.

‡Email: m.cheraghchi@imperial.ac.uk.

§Email: vgandiko@purdue.edu. Supported in part by a grant from the Purdue Research Foundation and by NSF CCF-1649515.

¶Email: elena-g@purdue.edu. Supported in part by a grant from the Purdue Research Foundation and by NSF CCF-1649515.

1.1. Motivation. Lattice-based communication. Lattices are a major technical tool in communication systems as the analogue of error-correcting codes over reals, for applications such as wireless communication and transmission over analog lines. In lattice-coding, the message m is mapped to a point c in a chosen lattice L. The codeword c is transmitted over an analog channel. If the encoded message gets corrupted by the channel, then the channel output may not be a lattice point, thus enabling transmission error detection. In order to correct errors, computationally expensive decoding algorithms are employed. Instead, the receiver may perform a local test for membership in the lattice beforehand, allowing the costly decoding computation to run only when there is a reasonably high chance of correct decoding.

Coding theory. Lattices can be seen as coding theoretic objects naturally bringing features of error-correcting codes from the finite field domain to the real domain. As such, a study of local testing (and correction) procedures for lattices naturally extends the classical notions of Locally Testable Codes (LTCs) and Locally Decodable Codes (LDCs), which are in turn of significance to computational complexity theory (for example in constructing probabilistically checkable proofs and hardness amplification, among numerous other applications). Characterizing local testability, explicitly initiated by Kaufman and Sudan [18], has been an intensely investigated direction in the study of LTCs. We believe that an analogous investigation of lattices is likely to bring new insights and new connections in property testing.

Integer Programming. Lattices are the fundamental structures underlying integer programming problems. An integer programming problem (IP) is specified by a constraint matrix $A \in \mathbb{R}^{n\times m}$ and a vector $b \in \mathbb{R}^n$. The goal is to verify if there exists an integer solution to the system $Ax = b, x \geq 0$. Although IP is NP-complete [17], its instances are solved routinely in practice using cutting planes and branch-and-cut techniques [33]. The relaxed problem of verifying integer feasibility of the system $Ax = b$ is equivalent to verifying whether b lies in the lattice generated by the columns of A. Thus, the relaxation problem is the membership testing problem in a lattice. It is solvable efficiently and is a natural pre-processing step to solving IPs. Furthermore, if the number of constraints n in the problem is very large, then it would be helpful to run a tester that reads only a partial set of coordinates of the input b to verify if b could lie in the lattice generated by the columns of A or is far from it. If the test rejects, then this saves on the computational effort to search for a non-negative solution.

Cryptography. In some cryptanalytic attacks on lattice-based cryptosystems, one needs to distinguish target vectors that are close to lattice vectors from those that are far from all lattice vectors. Hence, it is imperative to understand which lattices are difficult to test in order to ensure security of certain lattice-based cryptosystems.

We now give an informal description of our testing model motivated by its application in lattice-coding. The transmission of each coordinate of a lattice-codeword over the analog channel consumes power that is proportional to the square of the transmitted value. Thus the power consumption for transmitting the lattice-codeword $c \in L \subset \mathbb{R}^n$ is proportional to its squared ℓ_2 norm. The power consumption for transmitting a codeword over the channel is usually constrained by a power budget. The noise vector is also subject to a bound on its power. The power budget for transmission is typically formulated by considering the lattice-code $C(L)$ defined by the set of lattice points $c \in L$ that satisfy $\sum_{i=1}^n c_i^2 \leq \sigma n$ for some constant power budget $\sigma > 0$. In order to ensure that the receiver can tolerate adversarial noise budget δ per channel use, the shortest nonzero vector $v \in L$ should be such that $\sum_{i=1}^n v_i^2 \geq \delta n$. Thus, the *relative distance* of the lattice-code $C(L)$ is defined to be $\sum_{i=1}^n v_i^2/n$, where $v \in L$ is a shortest nonzero lattice vector. The *rate* of a lattice-code $C(L)$ is defined to be $(1/n)\log|C(L)|$ (note that this quantity could be larger than 1). An *asymptotically good family of lattices*, in this work, is one that achieves rate and relative distance that are both lower bounded by a positive constant. Such families are ideal for use in noisy communication channels.

We define a notion of a tester that will be useful as a pre-processor for decoding, and is similar to the established notion of code testing: An ℓ_2-tester of a lattice L for a given distance parameter $\epsilon > 0$ is a probabilistic procedure that given an input $t \in \mathbb{R}^n$, queries at most q coordinates of t, accepts with probability at least 2/3 if $t \in L$, and rejects with probability at least 2/3 if $\sum_{i=1}^n (t_i - w_i)^2 \geq \epsilon n$ for every $w \in L$.

In the next section we propose two main directions of research (Questions 1.2 and 1.3) analogous to similar directions that are fundamental to the study of locally testable codes.

1.2. Testing model. In the above application, we focused on ℓ_2 distances. We now formalize the notion of testing lattices for ℓ_p distances. We consider ℓ_p distances since these are natural notions for real-valued inputs [5]. The ℓ_p distance between $x, y \in \mathbb{R}^n$ is defined as $d_p(x,y) := \|x-y\|_p = (\sum_{i\in[n]} |x_i - y_i|^p)^{1/p}$. The distance from $v \in \mathbb{R}^n$ to L is $d_p(v,L) := \min_{u\in L} d_p(v,u)$. Denote the ℓ_p norm of the real vector 1^n by $\|1^n\|_p$.

In this work, we focus only on integral lattices, which are sub-lattices of $\mathbb{Z}^n$. Integral lattices are the most commonly encountered lattices in applications. Moreover, arbitrary lattices can be approximated by rational lattices and rational lattices can be scaled to integral lattices. Furthermore, it is imperative to investigate and understand integral lattices as a first step before attempting to address arbitrary lattices. For a lattice L, we denote the subspace of the lattice by $span(L)$.

Definition 1.1 (Local test for lattices). *An ℓ_p-tester $T(\epsilon, c, s, q)$ for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm that queries q coordinates of the input $t \in \mathbb{R}^n$, and*

- *(completeness)* accepts *with probability at least $1-c$ if $t \in L$,*
- *(soundness)* rejects *with probability at least $1-s$ if $d_p(t,L) \geq \epsilon \cdot \|1^n\|_p$ (we call such a vector t to be ϵ-far from L).*

If T always accepts inputs t that are in the lattice L then it is called 1-sided, otherwise it is 2-sided. If the queries performed by T depend on the answers to the previous queries, then T is called adaptive, otherwise it is called non-adaptive.

Our notion for a vector being "ϵ-far from the lattice" calls for a discussion. The notion of being 'ϵ-far' from the lattice should ideally be defined to suit the application of interest. For example, in applications like IP and cryptography, it is natural to ask for a notion of tester that ensures that scaling the lattice does not change the query complexity. An alternate definition of ϵ-far based on the *covering radius* of the lattice could be helpful to achieve this property. The covering radius of a lattice $L \subseteq \mathbb{R}^n$ (similar to codes) is the largest distance of any vector in $\mathbb{R}^n$ to the lattice. It is trivial to design a tester to verify if a point is in the lattice or at distance more than the covering radius from the lattice (simply accept all inputs). In order to have a tester notion where scaling preserves query complexity, we may define a vector as being ϵ-far from the lattice if the distance of the vector to every lattice point is at least ϵ times the covering radius of the lattice. We note that the covering radius of any *integral lattice* is $\Omega(\|1^n\|_p)$. Indeed, the densest possible integral lattice, namely the integer lattice $\mathbb{Z}^n$, has covering radius $(1/2)\|1^n\|_p$, as exhibited by the point $v = (1/2, \ldots, 1/2) \in \mathbb{R}^n$. Thus, by asking the tester to reject points at distance more than $\epsilon\|1^n\|_p$ in Definition 1.1, we have settled upon a strong notion of being ϵ-far from the lattice (i.e., the definition would in particular imply that vectors that are farther than ϵ times the covering radius would be rejected by the tester). Moreover, the definition of ϵ-far based on covering radius is essentially equivalent to the one in Definition 1.1 if the covering radius of the lattice is $\Theta(n)$.

A test $T(\epsilon, 0, 0, q)$ is a test with perfect completeness and perfect soundness. 1-sided testers (i.e., testers with perfect completeness) are useful as a pre-processing step, as mentioned earlier. An asymptotically good family of lattices $L(n)$ for ℓ_p distances is one that has ℓ_p-relative distance lower bounded by a constant (i.e., $\min_{v\in L(n)} \|v\|_p^p/n = \Omega(1)$) and has $2^{\Omega(n)}$ lattice points in the origin-centered ℓ_p-ball of radius $\|1^n\|_p$. By a simple packing argument[1], we note that such lattices have at most $2^{O(n)}$ lattice points in the origin-centered ℓ_p-ball of radius $\|1^n\|_p$. Similar to the application in lattice-coding and locally testable codes, a main question in ℓ_p-testing of lattices is the following:

Question 1.2. *Is there an asymptotically good family of lattices that can be tested for membership with constant number of queries?*

Motivated by the applications in IP and cryptography, we identify another fundamental question in ℓ_p-testing of lattices:

[1]Let $L(n)$ be an asymptotically good family of lattices with minimum distance $\lambda_1 := c\|1^n\|_p$ for some constant c. Let K be the number of lattice points inside the ℓ_p-ball of radius $\|1^n\|_p$, denoted by $B(0, \|1^n\|_p)$. The balls of radius $\lambda_1/2$ about each of those K lattice points are disjoint and the union of these balls are contained in $B(0, (1+c/2)\|1^n\|_p)$. Therefore, K is upper bounded by the ratio of the volumes of $B(0, (1+c/2)\|1^n\|_p)$ and $B(0, (c/2)\|1^n\|_p)$ which is at most $e^{2n/c}$.

Question 1.3. *What properties of a given lattice enable the design of ℓ_p-testers with constant query complexity?*

Tolerant Testing. Many applications can tolerate a small amount of noise in the input. Parnas *et al.* [28] introduced the notion of tolerant testing to account for a small amount of noise in the input. Tolerant testing has been studied in the context of codes (e.g. [15, 19]), and in the context of properties of real-valued data in the ℓ_p norm (e.g. [5]). We extend the tolerant testing model to lattices as follows.

Definition 1.4 (Tolerant local test for lattices). *An ℓ_p-tolerant-tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm that queries q coordinates of the input $t \in \mathbb{R}^n$, and*

- *(completeness)* accepts *with probability at least $1 - c$ if $d_p(t, L) \leq \epsilon_1 \cdot \|1^n\|_p$,*
- *(soundness)* rejects *with probability at least $1 - s$ if $d_p(t, L) \geq \epsilon_2 \cdot \|1^n\|_p$.*

Tolerant testing with parameter $\epsilon_1 = 0$ corresponds to the notion of testing given in Definition 1.1. Tolerant testing and distance approximation are closely related notions. In fact, in the Hamming space, the ability to perform tolerant testing for *every* choice of $\epsilon_1 < \epsilon_2$ can be exploited to approximate distances (using a binary search) [28].

Analogy with code testers. A common notion of testing for membership in *error-correcting codes* requires that inputs at *Hamming distance* at least ϵn from the code be rejected. (This notion is only relevant when the covering radius of the code is larger than ϵn.) We include the common definition here, and note that stronger versions of testing have also been considered in the literature [14, 15].

Definition 1.5 (Local test for codes). *A tester $T(\epsilon, c, s, q)$ for an error-correcting code $C \subseteq \mathbb{F}^n$ is a probabilistic algorithm that makes q queries to the input $t \in \mathbb{F}^n$, and*

- *(completeness)* accepts *with probability at least $1 - c$ if $t \in C$, and*
- *(soundness)* rejects *with probability at least $1 - s$ if $d_H(t, C) \geq \epsilon \cdot n$, where $d_H(u, v) := |\{i \in [n] : u(i) \neq v(i)\}|$ denotes the Hamming distance between u and v, and $d_H(t, C) := \min_{c \in C} d_H(t, c)$ (we call such a vector t to be ϵ-far from C).*

1.3. Our contributions. We initiate the study of membership testing in point lattices from the perspective of sublinear algorithms aiming to lay the ground work for further advances towards resolving Question 1.2 and Question 1.3. Our contributions draw on connections between lattices and codes, and on well-known techniques in property testing.

1.3.1. Upper and lower bounds for testing specific lattice families. Motivated by applications in lattice-based communication, we focus on an asymptotically good family of sets constructed from linear codes, via the so-called "code formula" [11]. We show upper and lower bounds on the query complexity of ℓ_1-testers for code formulas, as a function of the query complexity of the constituent code testers.

Code formula lattices. For simplicity, in what follows we will slightly abuse notation and use binary linear code $C \subseteq \{0,1\}^n$ to denote both the code viewed over the field $\mathbb{F}_2 = \{0, 1\}$ and the code embedded into $\mathbb{R}^n$ via the trivial embedding $0 \mapsto 0$ and $1 \mapsto 1$. All the arithmetic operations in the code formula refer to operations in $\mathbb{R}^n$. For two sets A and B of vectors we define $A + B := \{a + b \mid a \in A, b \in B\}$. The constituent codes used in all code formula lattices considered in this work are binary linear codes.

Definition 1.6 (Code Formula). *Let $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq C_m = \mathbb{F}_2^n$ be a family of nested binary linear codes. Then the code formula constructed from the family is defined as*

$$C_0 + 2C_1 + \cdots + 2^{m-1}C_{m-1} + 2^m\mathbb{Z}^n.$$

Here, m is the height *of the code-formula.*

If the family satisfies the Schur product condition*, namely, $c_1 * c_2 \in C_{i+1}$ for all codewords $c_1, c_2 \in C_i$, where the '$*$' operator is the coordinate-wise (Schur) product $c_1 * c_2 = \langle (c_1)_i \cdot (c_2)_i \rangle_{i \in [n]}$, then the code-formula forms a* lattice *(see [20]) and we denote it by $L(\langle C_i \rangle_{i=0}^{m-1})$.*

Significance of code formula lattices. Code formula lattices with height one already have constant rate if the constituent code C_0 has minimum Hamming distance $\Omega(n)$. Unfortunately, these lattices have

tiny relative minimum distance (since $2\mathbb{Z}^n$ has constant length vectors). However, code formulas of larger height achieve much better relative distance. In particular, it is easy to see that code formula lattices of height $m \geq \log n$ in which each of the constituent codes C_i has minimum Hamming distance $\Omega(n)$ give asymptotically good families of lattices [13, 8]. The code formula lattice constructed from a family of codes that satisfies the Schur-product condition is equivalent to the lattice constructed from the same family of codes by Construction D [21, 8, 20]. Construction-D lattices are primarily used in communication settings, e.g. see Forney [11].

In this work we design a tester for code formula lattices using testers for the constituent codes.

THEOREM 1.7. *Let $0 < \epsilon, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a 1-sided tester $T_i(\epsilon/m2^{i+1}, 0, s, q_i)$. Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right) + \sum_{i=1}^{m-1} q_i.$$

Next, we show a lower bound on the query complexity for testing membership in code formula lattices, using lower bounds for testing membership in the constituent codes.

THEOREM 1.8. *Let $0 < \epsilon, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Let $q_i = q_i(\epsilon, c, s)$ be such that any (possibly adaptive, 2-sided) ℓ_1-tester $T_i(\epsilon, c, s, q')$ for C_i satisfies $q' = \Omega(q_i)$, for every $i = 0, 1, \ldots, m-1$. Then every (possibly adaptive, 2-sided) ℓ_1-tester $T(\epsilon, c, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ has query complexity*

$$q = \Omega\left(\max\left\{\frac{1}{\epsilon}\log\frac{1}{s}, \max_{i=0,1,\ldots,m-1} q_i\right\}\right).$$

Code formula lattices from Reed-Muller codes. We instantiate the upper and lower bounds on the query complexity for a common family of code formula lattices constructed using Reed-Muller codes [11] to obtain nearly matching upper and lower bounds. We recall Reed-Muller codes below.

DEFINITION 1.9 (Reed Muller Codes). *Each codeword of a binary Reed-Muller code $RM(k, r) \subseteq \mathbb{F}_2^{2^r}$ corresponds to a polynomial $p(x) \in \mathbb{F}_2[x]$ in r variables of degree at most k evaluated at all 2^r possible inputs $x \in \mathbb{F}_2^r$.*

For the family of Reed-Muller codes in $\mathbb{F}_2^{2^r}$, it is well-known that $RM(0, r) \subseteq RM(1, r) \subseteq RM(2, r) \subseteq RM(3, r) \subseteq \cdots \subseteq RM(r-1, r) \subseteq RM(r, r) = \mathbb{F}_2^{2^r}$. A particular family of RM codes that leads to code formula lattices is $\langle RM(k_i, r)\rangle_{i=0}^{\log r}$, with $k_i = 2^i$. Indeed, it can be easily verified that this family satisfies the Schur product condition since Reed-Muller codewords are evaluation tables of multivariate polynomials over the binary field and product of two degree k polynomials is a degree $2k$ polynomial. Hence for height $m \leq \log r$ the construction $\langle RM(2^i, r)\rangle_{i=0}^{m-1}$ gives rise to a lattice. We note these lattices have small relative minimum distance and are not asymptotically good families of lattices.

COROLLARY 1.10. *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, s < 1$ and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for L with query complexity

$$q(\epsilon, s) = O\left(2^{k_{m-1}} \cdot \frac{1}{\epsilon}\log\frac{1}{s}\right).$$

In particular, when the height m and the degrees are constant, the query complexity of the tester is a constant.

For the lower bound, we obtain the following corollary using known lower bounds for testing Reed-Muller codes.

COROLLARY 1.11. *Let $0 \le k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, c, s < 1$ be constants and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then, every (possibly 2-sided, adaptive) ℓ_1-tester $T(\epsilon, c, s, q)$ for L has query complexity

$$q = \Omega(2^{k_{m-1}}).$$

We note that for code formula lattices obtained from Reed-Muller codes, Corollaries 1.10 and 1.11 show matching bounds (up to a constant factor depending on ϵ, s).

Random lattices. There exists a distribution of random lattices which are impossible to test with small number of queries. This follows from Theorem 1.8 and considering random codes, which typically need at least a linear number of queries to test. We illustrate a concrete example by considering the following distribution of random lattices [10, 4]: For constants $b < a$, let $m = nb/a$ and let $H \in \mathbb{F}_2^{m\times n}$ be a random matrix such that each row and column has exactly a and b non-zeroes respectively. Consider the linear code $C_{a,b} := \{x \in \mathbb{F}_2^n : Hx = 0 (\mathrm{mod}\ 2)\}$ and the code formula lattice $L(C_{a,b})$ associated with the linear code $C_{a,b}$.

THEOREM 1.12. *There exist constants a, b, ϵ, c, s such that every (possibly 2-sided, adaptive) ℓ_1-tester $T(\epsilon, c, s, q)$ for $L(C_{a,b})$ has query complexity $q = \Omega(n)$.*

The above theorem follows as an immediate corollary of Theorem 1.8 and Theorem 3.7 of [4].

1.3.2. Tolerant testing code formulas. We also obtain upper bounds for tolerantly testing code formula lattices.

THEOREM 1.13. *Let $0 < \epsilon_1, \epsilon_2, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a tolerant tester $T_i(2\epsilon_1, \frac{\epsilon_2}{m2^{i+1}}, \frac{c}{m+1}, s, q_i)$. Let $\gamma = \min\{c/(m+1), s\}$, $\epsilon_2 > m2^{m+1}\epsilon_1$. Then there exists an ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for the lattice $L(\langle C_i\rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{(\epsilon_2 - 2\epsilon_1)^2}\log\left(\frac{1}{\gamma}\right)\right) + \sum_{i=0}^{m-1} q_i.$$

COROLLARY 1.14. *Let $0 \le k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then there exists a ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, 1/3, 1/3, q)$ for L for all $\epsilon_1 \le \frac{c_1'}{2^{k_{m-1}}}$, $\epsilon_2 \ge \frac{c_2'm}{2^{k_0-1}}$ (for some constants c_1' and c_2') with query complexity $q = O(2^{k_{m-1}} \cdot \log m)$.

1.3.3. A canonical/linear test for lattices. We show a reduction from any given arbitrary test to a *canonical linear test*, thus suggesting that it is sufficient to design *canonical linear tests* for achieving low query complexity. In order to describe the intuition behind a canonical linear test, we first illustrate how to solve the membership testing problem when all coordinates of the input are known. For a given lattice L, its *dual lattice* is defined as

$$L^\perp := \{u \in span(L) \mid \langle u, v\rangle \in \mathbb{Z}, \text{ for all } v \in L\}.$$

It is easy to verify that $(L^\perp)^\perp = L$. Furthermore, a vector $v \in L$ if and only if for all $u \in L^\perp$, we have $\langle u, v\rangle \in \mathbb{Z}$. Thus, to test membership of t in L in the classical decision sense, it is sufficient to verify whether t has integer inner products with a set of basis vectors of the dual lattice $L^\perp$. Inspired by this observation, we define a canonical *linear test* for lattices as follows. For a lattice $L \subseteq \mathbb{R}^n$ and $J \subseteq [n]$, let $L_J^\perp := \{x \in L^\perp \mid supp(x) \subseteq J\}$, where $supp(x)$ is the set of non-zero indices of the vector x.

DEFINITION 1.15 (Linear Tester). *A* linear tester *for a lattice $L \subseteq \mathbb{Z}^n$ is a probabilistic algorithm which queries a subset $J = \{j_1, \ldots, j_q\} \subseteq [n]$ of coordinates of the input $t \in \mathbb{R}^n$ and accepts t if and only if $\langle t, x \rangle \in \mathbb{Z}$ for all $x \in L_J^\perp$.* [2]

Remark. By definition, the probabilistic choices of a linear tester are only over the set of coordinates to be queried: upon fixing the coordinate queries, the choice of the algorithm to accept or reject is fully determined. Furthermore, a linear tester is 1-sided since if the input t is a lattice vector, then for every dual vector $u \in L^\perp$, the inner product $\langle u, t \rangle$ is integral, and so it will be accepted with probability 1.

We show that non-adaptive linear tests are nearly as powerful as 2-sided adaptive tests for a full-rank lattice. We reduce any (possibly 2-sided, and adaptive) test for a full-rank lattice to a non-adaptive linear test for the same distance parameter ϵ, with a small increase in the query complexity and the soundness error.

THEOREM 1.16. *Let $L \subseteq \mathbb{Z}^n$ be a lattice with rank$(L) = n$. If there exists an adaptive 2-sided ℓ_p-tester $T(\epsilon, c, s, q)$ with query complexity $q = q_T(\epsilon, c, s)$, then there exists a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, c + s, q')$ with query complexity $q' = q_T(\epsilon/2, c, s) + O((1/\epsilon^p) \log(1/s))$.*

Furthermore, if we are guaranteed that the inputs are in $\mathbb{Z}^n$, then the query complexity of the test T' above can be improved to be identical to that of T (up to a constant factor in the ϵ parameter). The increase in the query complexity comes from an extra step used to verify the integrality of the input.

Theorem 1.16 suggests that, for the purposes of designing a tester with small query complexity, it is sufficient to design a non-adaptive linear tester, i.e., it suffices to only identify the probability distribution for the coordinates that are queried. Moreover, this theorem makes progress towards Question 1.3, since it shows that a lower bound on the query complexity of non-adaptive linear tests for a particular lattice implies a lower bound on the query complexity of all tests for that lattice. Thus in order to understand the existence of low query complexity tester for a particular lattice, it is sufficient to examine the existence of low query complexity *non-adaptive linear* tester for that lattice.

We note that Theorem 1.16 is the analogue of the result of [4] for linear error-correcting codes. In Section 2, we comment on the comparison between our proof and that in [4].

1.3.4. Testing membership of inputs outside the span of the lattice. We also observe a stark difference between the membership testing problem for a linear code, and the membership testing problem for a lattice. In the membership testing problem for a linear code $C \subseteq \mathbb{F}^n$ defined over a finite field that is specified by a basis, the input is assumed to be a vector in $\mathbb{F}^n$ and the goal is to verify whether the input lies in the span of the basis (see Definition 1.5). As opposed to codes, for a lattice $L \subseteq \mathbb{R}^n$, the input is an arbitrary real vector, and the goal is to verify whether the input is a member of L, and not to verify whether the input is a member of the span of the lattice. Thus, the inputs to the lattice membership testing problem could lie either in *span*(L), or outside *span*(L). Interestingly, for some lattices it is easy to show strong lower bounds on the query complexity if the inputs are allowed to lie outside *span*(L), thus suggesting that such inputs are hard to test.

THEOREM 1.17. *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in span$(L)^\perp$. Let $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for L for inputs in $\mathbb{R}^n$ has query complexity*

$$q = \Omega(|P|).$$

On the other hand, testers for inputs in the *span*(L) can be lifted to obtain testers for all inputs (including inputs that could possibly lie outside *span*(L)).

THEOREM 1.18. *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in span$(L)^\perp$. Let $0 < \epsilon, c, s < 1$, and suppose L has an ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs $t \in$ span(L). Then L has a tester $T'(2\epsilon, c, s, q')$ for inputs in $\mathbb{R}^n$ with query complexity*

$$q' \leq q + |P|.$$

[2] Verifying whether $\langle t, x \rangle \in \mathbb{Z}$ for all $x \in L_J^\perp$ can be performed efficiently by checking inner products with a set of basis vectors of the lattice $L_J^\perp$.

Theorem 1.18 implies that for lattices L of rank at most $n-1$, if the membership testing problem for inputs that lie in $span(L)$ is solvable using a small number of queries and if $span(L)^{\perp}$ is supported on few coordinates, then the membership testing problem for all inputs (including those that do not lie in $span(L)$) is solvable using a small number of queries.

Knapsack Lattices. Theorem 1.17 implies a linear lower bound for non-adaptively testing a well-known family of lattices, known as *knapsack lattices*, which have been investigated in the quest towards lattice-based cryptosystems [23, 32, 27]. We recall that a knapsack lattice is generated by a set of basis vectors $B = \{b_1, \ldots, b_{n-1}\}, b_i \in \mathbb{R}^n$ that are of the form

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, a_1) \\ b_2 &= (0, 1, \ldots, 0, a_2) \\ &\vdots \\ b_{n-1} &= (0, 0, \ldots, 1, a_{n-1}) \end{aligned}$$

where $a_1, \ldots, a_{n-1}$ are integers. We denote such a knapsack lattice by $L_{a_1,\ldots,a_{n-1}}$.

COROLLARY 1.19. *Let $a_1, \ldots, a_n$ be integers and $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for $L_{a_1,\ldots,a_{n-1}}$ has query complexity*

$$q = \Omega(n).$$

However, knapsack lattices with bounded coefficients are testable with a constant number of queries if the inputs are promised to lie in $span(L)$.

THEOREM 1.20. *Let $a_1, \ldots, a_{n-1}$ be integers with $M = \max_{i\in[n]} |a_i|^p$ and $0 < \epsilon, s < 1$. There exists a non-adaptive ℓ_p-tester $T(\epsilon, 0, s, q)$ for $L_{a_1,\ldots,a_{n-1}}$ with query complexity $q = O\left(\frac{M}{\epsilon^p} \cdot \log \frac{1}{s}\right)$, if the inputs are guaranteed to lie in $span(L)$.*

Theorem 1.20 indicates that the large lower bound suggested by Theorem 1.17 could be circumvented for certain lattices if we are promised that the inputs lie in $span(L)$.

2. Overview of the proofs.

2.1. Upper and lower bounds for testing general code formula lattices. The constructions of a tester for Theorem 1.7 and a tolerant tester for Theorem 1.13 follow the natural intuition that in order to test the lattice one can test the underlying codes individually. The proof relies on a triangle inequality that can be derived for such lattices. The application to code-formula lattices constructed from Reed-Muller codes follows from the tight analysis of Reed-Muller code testing from [6], which guarantees constant rejection probability of inputs that are at distance proportional to the minimum distance of the code. We note that the time complexity of the code-formula tester is given by the sum of the run-times of the component code testers. Since the component code testers can be assumed to be linear, and hence efficient, the code-formula lattice tester is also efficient.

While the tester that we construct from code testers for the purposes of proving Theorem 1.7 is an adaptive linear test, there is a simple variant that is a non-adaptive linear test with at least as good correctness and soundness. (see Remark 5.16 in Section 5).

The lower bound (Theorem 1.8) relies on the fact that if an input t is far from the code C_k in the code formula construction, then the vector $2^k t$ is far from the lattice . Moreover, if $t \in C_k$ then $2^k t$ belongs to the lattice. Therefore a test for the lattice can be turned into a test for the constituent codes.

2.2. From general tests to canonical tests. We briefly outline our reduction for Theorem 1.16. Suppose $T(\epsilon, c, s, q)$ is a 2-sided, adaptive tester with query complexity $q = q_T(\epsilon, c, s)$ for a full rank integral lattice L. Such a tester handles all real-valued inputs. We first restrict T to a test that processes only integral inputs in the bounded set $\mathcal{Z}_d = \{0, 1, \ldots, d-1\}$ (for some carefully chosen d), and so the restricted test inherits all the parameters of T. We remark that $\mathcal{Z}_d \subset \mathbb{Z}$ is a subset of integers, and it should not be confused with $\mathbb{Z}_d$, the ring of integers modulo d.

A key ingredient in our reduction is choosing the appropriate value of d in order to enable the same guarantees as that of codes. We choose d such that $d\mathbb{Z}^n \subseteq L$. Such a d always exists [25]. This choice of d allows us to add any vector in $V = L \bmod d$ (embedded in $\mathbb{R}^n$) to any vector $x \in \mathbb{R}^n$ without changing the distance of x to L in any ℓ_p-norm (see Proposition 5.2).

Since our inputs are now integral and bounded, any adaptive test can be viewed as a distribution over deterministic tests, which themselves can be viewed as decision trees. This allows us to proceed along the same lines as in the reduction for codes over finite fields of [4].

We exploit the property that adding any vector in V to any vector $x \in \mathbb{R}^n$ does not change the distance to L. In the first step of our reduction we add a random vector in V to the input and perform a probabilistic *linear* test. The idea is that one can relabel the decision tree of any test according to the decision tree of a linear test, such that the error shifts from the positive (yes) instances to the negative (no) instances (see Lemma 5.3). A simple property of lattices used in this reduction is that if the set of queries I and answers a_I do not have a local witness for non-membership in the lattice (in the form of a dual lattice vector v supported on I such that $\langle w_I, v_I \rangle \notin \mathbb{Z}$), then there exists $w \in L$ that extends a_I to the remaining set of coordinates (i.e., $a_I = w_I$).

In the next step we remove the adaptive aspect of the test to obtain a non-adaptive linear test for inputs in $\mathcal{Z}_d^n$ (see Lemma 5.4). We obtain this tester by performing the adaptive queries on a randomly chosen vector in V (and not on the input itself) and rejecting/accepting according to whether there exists a local witness for the non-membership of the input queried on the same coordinates.

We then lift this test to a non-adaptive linear test for inputs in $\mathbb{Z}^n$, by simulating the test over $\mathcal{Z}_d^n$ on the same queried coordinates but using the answers obtained after taking modulo d. Owing to the choice of d, this does not change the distance of the input to the lattice (see Lemma 5.5).

Finally, we extend this test to a non-adaptive linear test for inputs in $\mathbb{R}^n$ by performing some additional queries to rule out inputs that are not in $\mathbb{Z}^n$. For this, we design a tester for the integer lattice $\mathbb{Z}^n$ with query complexity $O((1/\epsilon^p)\log(1/s))$. This final step of testing integrality increases the overall query complexity to $q_T(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$ (see Lemma 5.6).

Organization. We prove the upper bound and lower bound for testing code formula constructions (Theorem 1.7, Theorem 1.8) and its instantiations to Reed-Muller codes (Corollary 1.10, Corollary 1.11) in Section 3. The upper bound for tolerant testing code-formula constructions (Theorem 1.13) and its instantiations to Reed-Muller codes (Corollary 1.14) are proved in Section 4. We present the formal lemmas needed to prove Theorem 1.16 and their proofs in Section 5. We address non-full-rank lattices and prove Theorems 1.17, 1.18, Corollary 1.19 and Theorem 1.20 in Section 6.

3. Testing Code-Formula Lattices.

3.1. Upper Bounds for Code-Formula Lattices. In this section we construct a tester for testing membership in lattices obtained from the code formula construction using a tester for the constituent codes.

THEOREM 1.7. *Let $0 < \epsilon, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a 1-sided tester $T_i(\epsilon/m2^{i+1}, 0, s, q_i)$. Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right) + \sum_{i=1}^{m-1} q_i.$$

Proof. Let $L := L(\langle C_i \rangle_{i=0}^{m-1})$. First, we use Lemma 5.6 to reduce the task to testing integral inputs for distance parameter $\epsilon/2$. According to the lemma, it suffices to show that all such inputs can be tested with 1-sided error and using $\sum_{i=1}^{m-1} q_i$ queries.

Now, let $w \in \mathbb{Z}^n$ denote the input. We may assume that all coordinates of w are non-negative integers less than 2^m. Otherwise, we can shift each coordinate by an appropriate (possibly different) multiple of 2^m to make sure this condition holds. Observe that each such operation would correspond to shifting w by an integer multiple of the lattice vector $2^m e_i$, where e_i is the i^{th} basis vector, and that translating a vector by

a lattice point does not affect its distance to the lattice. Moreover, observe that this transformation can be applied implicitly, locally, and efficiently by the testing algorithm as the queries are made.

Let $w_0, \ldots, w_{m-1} \in \{0,1\}^n$ where $w_i(j)$ is the $(i+1)^{th}$ least significant bit in the binary decomposition of the j^{th} coordinate of w. Thus we have $w = \sum_{i=0}^{m-1} 2^i w_i$. Once again, the coordinates of w_i can be computed implicitly, locally and efficiently by the algorithm as the queries are made.

The tester T would now proceed as follows: Run $T_i(\epsilon/(m2^{i+1}), 0, s, q_i)$ on w_i for every $i = 0, 1, \ldots, m-1$. Accept if and only if all tests accept.

The overall query complexity of this tester is $\sum_{i=1}^{m-1} q_i$.

The completeness of this tester is easy to deduce. Indeed, if $w \in L(\langle C_i \rangle_{i=0}^{m-1})$, then by definition of the code formula construction, there exist $\tilde{w}_i \in C_i$ for every $i = 0, 1, \ldots, m-1$ and an integer $\tilde{w}_m \in 2^m\mathbb{Z}^n$ such that $w = \sum_{i=0}^{m-1} 2^i \tilde{w}_i + \tilde{w}_m$. Since entries of w are non-negative integers less than 2^m, we must have $\tilde{w}_m = 0$. Moreover, since $\tilde{w}_i \in \{0,1\}^n$ and the binary representation is unique, it must be that $\tilde{w}_i \in C_i$ for $i = 0, \ldots, m-1$. That is, each of the w_i embedded in $\mathbb{F}_2^n$ are in C_i and therefore, each $T_i(\epsilon/(2^{i+1}m), 0, s, q_i)$ (and thus the overall tester) will accept w_i.

Before analyzing the soundness, we observe the following simple inequality.

Claim 3.1. $d_1(w, L) \leq d_1(w_0, C_0) + 2d_1(w_1, C_1) + \cdots + 2^{m-1} d_1(w_{m-1}, C_{m-1})$.

Proof. Let $c_i \in \{0,1\}^n$ be the closest codeword to w_i in C_i for every $i = 0, 1, \ldots, m-1$. From the definition of the code formula construction, we know that the vector $v = c_0 + 2c_1 + \cdots + 2^{m-1}c_{m-1}$ is a lattice vector. Therefore,

$$\begin{aligned}
\sum_{i=0}^{m-1} 2^i d_1(w_i, C_i) &= \sum_{i=0}^{m-1} 2^i \|w_i - c_i\|_1 \\
&\geq \left\| \sum_{i=0}^{m-1} 2^i (w_i - c_i) \right\|_1 \qquad \text{(by the triangle inequality)} \\
&= d_1(w, v) \\
&\geq d_1(w, L).
\end{aligned}$$

∎

Now, if $d_1(w, L) \geq \epsilon n/2$, then by Claim 3.1 we have

$$d_1(w_0, C_0) + 2d_1(w_1, C_1) + \cdots + 2^{m-1} d_1(w_{m-1}, C_{m-1}) \geq \epsilon n/2.$$

Therefore, by averaging, for some $i \in \{0, 1, \ldots, m-1\}$ we must have $d_1(w_i, C_i) \geq (\epsilon/(m2^{i+1}))n$. Thus the tester $T_i(\epsilon/(m2^{i+1}), 0, s, q_i)$ will reject with probability at least $1 - s$. Hence the soundness follows. ∎

We now apply the result of Theorem 1.7 to the lattice obtained by applying code formula on a nested family of Reed-Muller codes. In order to do so, we use the following result.

Theorem 3.2. *[6] For any $0 \leq k \leq r$ and $0 < s < 1$, $RM(k, r)$ has a 1-sided tester $T(\epsilon, 0, s, q(\epsilon, s))$ with query complexity $q(\epsilon, s) = O((\log \frac{1}{s})(2^k + \frac{1}{\epsilon}))$ whose queries are each uniformly distributed.*

Using the above result in Theorem 1.7, we obtain Corollary 1.10 whose proof we present next.

Corollary 1.10. *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, s < 1$ and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1} RM(k_{m-1}, r) + 2^m \mathbb{Z}^{2^r}.$$

Then, there exists an ℓ_1-tester $T(\epsilon, 0, s, q)$ for L with query complexity

$$q(\epsilon, s) = O\left(2^{k_{m-1}} \cdot \frac{1}{\epsilon} \log \frac{1}{s}\right).$$

Proof. From Theorem 1.7, for any $\epsilon > 0$, there is a $T(\epsilon, 0, s, q(\epsilon, s))$ tester for the code formula lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ with query complexity $q(\epsilon, s) = O\left(\frac{1}{\epsilon} \log \frac{1}{s}\right) + \sum_{i=0}^{m-1} q_i\left(\frac{\epsilon}{m2^{i+1}}, s\right)$, where $q_i(\epsilon, s)$ is the query

complexity of testing the code C_i (with distance parameter ϵ and soundness error s). By Theorem 3.2, each $RM(k_i, r)$ has a 1-sided tester $T_i(\epsilon_i, 0, s, q_i(\epsilon_i, s))$ with query complexity $q_i(\epsilon_i, s) = O(\log(1/s))(2^{k_i} + 1/\epsilon_i)$. Therefore, the 1-sided tester $T(\epsilon, 0, s, q)$ for the code formula lattice $L(\langle RM(i,r)\rangle_{i=k_0}^{k_{m-1}})$ has query complexity

$$\begin{aligned} q(\epsilon, s) &= O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right) + \sum_{i=0}^{m-1} q_i\left(\frac{\epsilon}{m2^{i+1}}, s\right) \\ &= O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right)\sum_{i=0}^{m-1}\left(2^{k_i} + m2^{i+1}\right) \\ &= O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right)\left(m2^m + \sum_{i=0}^{m-1} 2^{k_i}\right) \\ &= O\left(\frac{1}{\epsilon}\log\frac{1}{s}\right)\left(m2^m + 2^{k_{m-1}}\right) \\ &= O\left(\frac{1}{\epsilon}\log\frac{1}{s}\cdot 2^{k_{m-1}}\right). \end{aligned}$$

To see the last step of the above equation, recall that in order for $L(\langle RM(i,r)\rangle_{i=k_0}^{k_{m-1}})$ to be a lattice, we must have $k_i \geq 2^{i-1}$ for $i > 0$, and in particular $k_{m-1} \geq 2^{m-2}$. So $2^{k_{m-1}} \geq 2^{2^{m-2}} \geq m2^m$, for $m \geq 5$. ∎

3.2. Lower bounds for Code-Formula Lattices. In this section, we prove Theorem 1.8. We will use the following lemma.

LEMMA 3.3. *Let $C_0, C_1, \ldots, C_{m-1}$ be a family of codes satisfying the Schur product condition and $L = L(\langle C_i\rangle_{i=0}^{m-1})$. Let $t \in \{0,1\}^n$ and $k \in \{0, 1, \ldots, m-1\}$. Then*

$$d_1(t, C_k) \leq d_1(2^k t, L) \leq 2^k d_1(t, C_k).$$

Proof. Since $2^k C_k$ is contained in L, $d_1(2^k t, L) \leq d_1(2^k t, 2^k C_k) = 2^k d_1(t, C_k)$. So, the distance of $2^k t$ to the lattice is at most $2^k d_1(t, C_k)$. We now show the inequality

$$d_1(2^k t, L) \geq d_1(t, C_k).$$

Let $v = \sum_{j=0}^{m-1} 2^j c_j + 2^m z$ for some arbitrary $c_j \in C_j$ (for every $j \in \{0, 1, \ldots, m-1\}$) and some $z \in \mathbb{Z}^n$, be an arbitrary lattice vector. We will show that $d_1(2^k t, v) \geq d_1(t, C_k)$. Let $u = c_k - t$, and $S \subseteq [n]$ be the support of u, then $|S| \geq d_1(t, C_k)$.

By Claim 3.4, $d_1(2^k t, v) \geq \sum_{i\in S} |v(i) - 2^k t(i)| \geq |S|$ (where $v(i)$ and $t(i)$ represent the ith entry in the respective vectors), which completes the proof. ∎

CLAIM 3.4. *For every $i \in S$, $|v(i) - 2^k t(i)| \geq 1$.*

Proof. Let $i \in S$. Since $2^k u = 2^k c_k - 2^k t$, we have $2^k |u(i)| = 2^k$. We also have

$$|v(i) - 2^k t(i)| = \left|\sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i) + \sum_{j=k+1}^{m-1} 2^j c_j(i) + 2^m z(i)\right|.$$

Since $c_j(i) \in \{0,1\}$ for every $j \in \{0, 1, \ldots, k-1\}$, the first term in the above sum is at least zero and at most $2^k - 1$. The maximum is achieved when all $c_j(i) = 1$ for all $j \in [k-1]$, and $u(i) = 1$. Hence, $1 \leq \left|\sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i)\right| \leq 2^{k+1} - 1$. Since $c_{k+1}(i) \in \{0,1\}$, we have

$$1 \leq \left|\sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i) + 2^{k+1} c_{k+1}(i)\right| \leq \left|\sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i)\right| + \left|2^{k+1} c_{k+1}(i)\right| \leq 2^{k+2} - 1.$$

Proceeding similarly, since $c_j(i) \in \{0,1\}$ for $j = k+2, \ldots, m-1$, we have

$$1 \leq \left| \sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i) + \sum_{j=k+1}^{m-1} 2^j c_j(i) \right| \leq 2^m - 1.$$

Since $z_m \in \mathbb{Z}$, we conclude that $\left| \sum_{j=0}^{k-1} 2^j c_j(i) + 2^k u(i) + \sum_{j=k+1}^{m-1} 2^j c_j(i) + 2^m z(i) \right| \geq 1$. ∎

THEOREM 1.8. *Let $0 < \epsilon, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Let $q_i = q_i(\epsilon, c, s)$ be such that any (possibly adaptive, 2-sided) ℓ_1-tester $T_i(\epsilon, c, s, q')$ for C_i satisfies $q' = \Omega(q_i)$, for every $i = 0, 1, \ldots, m-1$. Then every (possibly adaptive, 2-sided) ℓ_1-tester $T(\epsilon, c, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ has query complexity*

$$q = \Omega\left(\max\left\{ \frac{1}{\epsilon} \log \frac{1}{s}, \max_{i=0,1,\ldots,m-1} q_i \right\} \right).$$

Proof. Let $T(\epsilon, c, s, q)$ be a test for the code lattice, and let $k \in \{0, 1, \ldots, m-1\}$. We construct a tester $T_k(\epsilon, c, s, q)$ for C_k as follows: On input $w \in \{0,1\}^n$, run $T(\epsilon, c, s, q)$ on $2^k w$ and accept if and only if T accepts. The query complexity of T_k is the same as the query complexity of T. If the input w is a codeword in C_k, then by the definition of the lattice, $2^k w$ is a lattice vector, and T_k will accept w with probability at least $1 - c$. If $d_1(w, C_k) \geq \epsilon n$, then by Lemma 3.3, we have that $d_1(2^k w, L(\langle C_i \rangle)_{i=0}^{m-1}) \geq \epsilon n$. Therefore, T_k will reject w with probability at least $1 - s$. Finally, since $L \subseteq \mathbb{Z}^n$ we have that $d(w, \mathbb{Z}^n) \leq d(w, L)$ and so we could use T to test membership in $\mathbb{Z}^n$. By Claim 3.5 testing $\mathbb{Z}^n$ requires $q = \Omega(\frac{1}{\epsilon} \log(1/s))$ queries. ∎

CLAIM 3.5. *Any test $T_k(\epsilon, c, s, q)$ for $\mathbb{Z}^n$ requires $q = \Omega(\frac{1}{\epsilon} \log(1/s))$ queries.*

Proof. First, we use Yao's duality theorem [34] which is a standard tool in proving lower bounds and assume that the testing algorithm is, without loss of generality, deterministic (but possibly adaptive). We exhibit two distributions on the inputs which the algorithm is expected to distinguish but cannot do so without making sufficiently many queries. The *yes case* distribution is the deterministic distribution on all-zeros input (which is a lattice point). Given an input from this distribution, the algorithm should accept. The *no case* distribution would consists of the all-zeros vector but with a uniformly random set S of $2\epsilon n$ coordinate positions changed from 0 to 1/2. Indeed, all vectors on the support of this distribution are ϵ-far from the lattice. Assuming that $q \leq n/2$ (otherwise there is nothing to prove), each time the algorithm queries a position that has not been queries before, there is at most a 4ϵ chance that it hits any position in S (even conditioned on the past query outcomes). Thus the probability that the algorithm ever succeeds in finding a position in S is at most $1 - (1 - 4\epsilon)^q$, which, on the other hand by the soundness condition, should be at least $1 - s$. Therefore, the soundness error is at least $s \geq (1 - 4\epsilon)^q$ or, in other words, in order to achieve a given s we must have $q = \Omega(\frac{1}{\epsilon} \log(\frac{1}{s}))$. ∎

In the case of code formula lattices generated from Reed-Muller codes of order r, we note that $n = 2^r$. We need the following known lower bound on the query complexity of testing Reed-Muller codes. For completeness we reproduce the exact statement that we need in this work and include a proof.

THEOREM 3.6 ([1]). *Let $T(\epsilon, c, s, q)$ be a (possibly 2-sided and adaptive) tester for the code $RM(k, r)$ where $k \leq r/(2 \log r)$, $\epsilon < 1/2 - \Omega(1)$, and $c + s < 1 - \Omega(1)$ (where $\Omega(1)$ hides arbitrarily small positive absolute constants). Then, $q \geq 2^k$.*

Proof. Using the reduction from 2-sided, adaptive testers to non-adaptive, 1-sided tests for any linear code (applicable to RM codes) of [4], it is sufficient to focus on the latter tests. Let $\mathcal{C}$ be the code $RM(k, r)$. First we note that the length of the code is $R := 2^r$ and its dimension is

$$\log |\mathcal{C}| = \sum_{i=0}^{k} \binom{r}{i} \leq 1 + kr^k.$$

Therefore, noting that $k \leq r/(2 \log r)$,

$$|\mathcal{C}| = O(2^{kr^k}) = O(2^{r^{k+1}}) = O(2^{(\log R)\sqrt{R}}) = 2^{o(R)}.$$

Let V be the number of points in a Hamming ball of radius ϵR in $\{0,1\}^R$. Thus we have $V \leq 2^{h(\epsilon)R}$, where $h(\cdot)$ denotes the binary entropy function. Let S be the set of points in $\{0,1\}^R$ that have Hamming distance at most ϵr with the code. Of course we have

$$|S| \leq V|\mathcal{C}| = O(2^{(h(\epsilon)+o(1))R}).$$

Since $\epsilon < 1$ is a fixed constant, this implies $|S|/2^R = o(1)$. Now we run the tester with the following two input distributions:

Case 1. The tester is given a uniformly random string in $\{0,1\}^R$ as the input.

Case 2. The tester is given a uniformly random codeword of $\mathcal{C}$ as the input.

Since the dual distance of $\mathcal{C}$ is 2^{k+1}, a standard coding theoretic fact implies that a uniformly random codeword of $\mathcal{C}$ is t-wise independent for $t = 2^{k+1}$; i.e., any local view of up to t coordinates of the random codeword is exactly the uniform distribution. Therefore, since the tester makes no more than t queries, its output distribution is exactly the same in the above two cases. Let p be the acceptance probability of the tester with respect to the (common) output distribution. In order to satisfy completeness, the tester should accept with probability at least $1 - c$ in the second case. Therefore, we must have $p \geq 1 - c$.

On the other hand, a uniform random string in $\{0,1\}^R$ is ϵ-far from the code with probability $1 - o(1)$ according to the above bound on $|S|$. In the conditional world where this string actually becomes ϵ-far from the code, the acceptance probability of the code would thus remain within $p(1 \pm o(1))$. However, in this case the soundness implies that the tester should accept with probability at most s, and thus, $p \leq s(1 + o(1))$. Thus the two distributions provided by the above two cases would violate requirements of the local tester assuming that $c + s \leq 1 - \Omega(1)$. ∎

COROLLARY 1.11. *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let $0 < \epsilon, c, s < 1$ be constants and L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1}RM(k_{m-1}, r) + 2^m\mathbb{Z}^{2^r}.$$

Then, every (possibly 2-sided, adaptive) ℓ_1-tester $T(\epsilon, c, s, q)$ for L has query complexity

$$q = \Omega(2^{k_{m-1}}).$$

Proof. Suppose we have a tester $T(\epsilon, c, s, q)$ for $L(\langle RM(k_i, r)\rangle_{i=0}^{m-1})$. By Theorem 1.8, we have

$$q \geq \max_{i=0,1,\ldots,m-1} q_i(\epsilon, c, s).$$

By Theorem 3.6, it follows that $q = \Omega(2^{k_{m-1}})$. ∎

4. Tolerant Testing Code Formula Lattices . We first give a tolerant tester for testing membership in $\mathbb{Z}^n$. We will use this tester in the design of a tolerant tester for testing membership in lattices obtained from the code formula construction.

LEMMA 4.1. *Let $\epsilon_1, \epsilon_2, c, s > 0$ such that $\epsilon_2 > \epsilon_1$ and $\gamma = \min\{c, s\}$. There is a tolerant tester $T_Z(\epsilon_1, \epsilon_2, c, s, q_Z)$ for $\mathbb{Z}^n$ which uses $q_Z = O(1/(\epsilon_2 - \epsilon_1)^2 \cdot \log(\frac{1}{\gamma}))$ queries.*

Proof. The tester estimates the distance of the input from $\mathbb{Z}^n$ by querying $O(1/(\epsilon_2 - \epsilon_1)^2 \log(\frac{1}{\gamma}))$ coordinates uniformly at random. If the estimated distance is at least $\frac{(\epsilon_1+\epsilon_2)}{2}n$, then it rejects, otherwise it accepts. The correctness and soundness follow from Chernoff bounds. We describe the test formally as follows:

1. Query $q := C/(\epsilon_2 - \epsilon_1)^2 \cdot \log(\frac{1}{\gamma})$ coordinates of the input t uniformly at random, for some constant C to be determined later. Let $I \subseteq [n]$ be the indices of the queried coordinates.
2. Let $\delta := \frac{\sum_{i\in I}|t_i - \lfloor t_i \rceil|}{q}$.
3. If $\delta \leq \frac{\epsilon_1+\epsilon_2}{2}$ then Accept.
4. Else Reject.

Suppose $d(t,\mathbb{Z}^n) \le \epsilon_1 n$, then $d(t,\mathbb{Z}^n)/n = \sum_{i=1}^n |t_i - \lfloor t_i \rceil|/n \le \epsilon_1$. Therefore, $\mathbb{E}[\delta] \le \epsilon_1$. By a Chernoff bound, it follows that $\Pr[\delta - \epsilon_1 > \frac{\epsilon_2-\epsilon_1}{2}] \le e^{-q(\epsilon_2-\epsilon_1)^2/2} \le c$ for $q \ge C/(\epsilon_2-\epsilon_1)^2 \cdot \log(\frac{1}{\gamma})$ and a constant $C > 0$.

Now suppose $d(t,\mathbb{Z}^n) > \epsilon_2 n$. Then, $d(t,\mathbb{Z}^n)/n = \sum_{i=1}^n |t_i - \lfloor t_i \rceil|/n \ge \epsilon_2$. Again, by a Chernoff bound, and suitable choice of the constant C, it follows that $\Pr[\epsilon_2 - \delta \ge \frac{\epsilon_2-\epsilon_1}{2}] \le e^{-q(\epsilon_2-\epsilon_1)^2/4} \le s$ for q chosen as above. ∎

We now describe a tolerant tester for code formula lattices.

THEOREM 1.13. *Let $0 < \epsilon_1, \epsilon_2, c, s < 1$ and $C_0 \subseteq C_1 \subseteq \cdots \subseteq C_{m-1} \subseteq \{0,1\}^n$ be a family of binary linear codes satisfying the Schur product condition. Suppose every C_i has a tolerant tester $T_i(2\epsilon_1, \frac{\epsilon_2}{m2^{i+1}}, \frac{c}{m+1}, s, q_i)$. Let $\gamma = \min\{c/(m+1), s\}$, $\epsilon_2 > m2^{m+1}\epsilon_1$. Then there exists an ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for the lattice $L(\langle C_i \rangle_{i=0}^{m-1})$ with query complexity*

$$q = O\left(\frac{1}{(\epsilon_2 - 2\epsilon_1)^2} \log\left(\frac{1}{\gamma}\right)\right) + \sum_{i=0}^{m-1} q_i.$$

Proof. We use the tolerant testers T_i for the codes C_i and the tolerant tester T_Z for $\mathbb{Z}^n$ to construct a tolerant tester for L.

Let $\lfloor t \rceil$ denote the vector obtained by rounding each coordinate of t to its nearest integer and for any vector x, let $x(j)$ denote the j^{th} coordinate of x. Let $t_0, ..., t_{m-1} \in \{0,1\}^n$ where $t_i(j)$ is the $(i+1)^{th}$ least significant bit in the binary decomposition of the j^{th} coordinate of $\lfloor t \rceil$. Define $t_m = \frac{1}{2^m}(t - \sum_{i=0}^{m-1} 2^i t_i) \in \mathbb{R}^n$. Therefore, t can be written as $t = t_0 + 2t_1 + \cdots + 2^{m-1}t_{m-1} + 2^m t_m$, where $t_i \in \{0,1\}^n$ for all $i \in [m-1]$ and $t_m \in \mathbb{R}^n$. Moreover, $t_m \in \mathbb{Z}^n$ if and only if $t \in \mathbb{Z}^n$.

The tolerant tester $T(\epsilon_1, \epsilon_2, c, s, q)$ on input $t \in \mathbb{R}^n$ now proceeds as follows: Run $T_Z(\epsilon_1, \frac{\epsilon_2}{2}, \frac{c}{m+1}, s, q_Z)$ on t and $T_i(2\epsilon_1, \frac{\epsilon_2}{m2^{i+1}}, \frac{c}{m+1}, s, q_i)$ on t_i for all $i \in \{0, 1, \ldots, m-1\}$. Accept if and only if all tests accept. The query complexity of $T(\epsilon_1, \epsilon_2, c, s, q)$ is therefore:

$$q(\epsilon_1, \epsilon_2, c, s) = \sum_{i=0}^{m-1} q_i + q_Z,$$

where we recall that q_Z is the query complexity of $T_Z(\epsilon_1, \frac{\epsilon_2}{2}, \frac{c}{m+1}, s, q_Z)$. From Lemma 4.1, we know that $q_Z = O(\frac{1}{(\epsilon_2-2\epsilon_1)^2} \log(\frac{1}{\gamma}))$, where $\gamma = \min\{\frac{c}{m+1}, s\}$. We now analyze the soundness and completeness of this test.

Soundness. Suppose $d(t,L) \ge \epsilon_2 n$. We first show that either t is far from $\mathbb{Z}^n$ or the closest integer vector to t is far from the lattice.

CLAIM 4.2. $d(t,L) \le d(\lfloor t \rceil, L) + d(t,\mathbb{Z}^n)$

Proof. Let u be the closest lattice vector to $\lfloor t \rceil$. Then

$$d(t,L) \le \|t-u\|_1 = \|(t - \lfloor t \rceil) + (\lfloor t \rceil - u)\|_1 \le \|t - \lfloor t \rceil\|_1 + \|\lfloor t \rceil - u\|_1$$

Since $\|t - \lfloor t \rceil\|_1 = d(t,\mathbb{Z}^n)$, it follows that $d(t,L) \le d(\lfloor t \rceil, L) + d(t,\mathbb{Z}^n)$. ∎

Therefore, if $d(t,L) \ge \epsilon_2 n$, then from Claim 4.2, either $d(\lfloor t \rceil, L) \ge \epsilon_2 n/2$ or $d(t,\mathbb{Z}^n) \ge \epsilon_2 n/2$. If $d(t,\mathbb{Z}^n) \ge \epsilon_2 n/2$, then T_Z rejects t with probability at least $1-s$. If $d(\lfloor t \rceil, L) \ge \epsilon_2 n/2$, then from Claim 3.1 proved in Section 3 we can conclude that there exists some $i \in \{0, 1, \ldots, m-1\}$ such that $2^i d(t_i, C_i) \ge \epsilon_2 n/2m$, and $T_i(t_i, 2\epsilon_1, \epsilon_2/m2^{i+1})$ will reject t_i with probability at least $1-s$. Thus, if $d(t,L) \ge \epsilon_2 n$, then T rejects t with probability with at least $1-s$.

Completeness. Suppose $d(t,L) \le \epsilon_1 n$. Then $d(t,\mathbb{Z}^n) \le \epsilon_1 n$, since $L \subseteq \mathbb{Z}^n$. So, $T_Z(\epsilon_1, \frac{\epsilon_2}{2}, \frac{c}{m+1}, s, q_Z)$ will accept t with probability at least $1 - \frac{c}{m+1}$. We now show that each t_i is also close to the corresponding linear code C_i.

For the sake of contradiction, suppose $d(t_i, C_i) > 2\epsilon_1 n$ for some $i \in [m-1]$. We will show that $d(t,L) > \epsilon_1 n$. We do this in two steps. First we show in Lemma 4.3 that $d(\lfloor t \rceil, L) > 2\epsilon_1 n$. Then by Claim 4.5 and the fact that $d(t,\mathbb{Z}^n) \le \epsilon_1 n$, we have that $d(t,L) > \epsilon_1 n$, a contradiction.

LEMMA 4.3. *If $d(t_i, C_i) > 2\epsilon_1 n$ for some $i \in [m-1]$, then $d(\lfloor t \rceil, L) > 2\epsilon_1 n$.*

Proof. Let $v = \sum_{i=0}^{m-1} 2^i v_i + 2^m v_m \in L$ be the closest lattice vector to $\lfloor t \rceil$. By definition of the lattice, each $v_i \in C_i$ for $i \in [m-1]$ and $v_m \in \mathbb{Z}^n$. Consider the vectors $t_0, t_1, \ldots, t_m$ as defined above (for which $\lfloor t \rceil = \sum_{i=0}^{m-1} 2^i t_i + 2^m t_m$). So, each $t_i \in \{0,1\}^n$ and $t_m \in \mathbb{Z}^n$. The following property of vectors with bounded entries will be used to prove the claim.

CLAIM 4.4. *Let $a_0, a_1, \ldots, a_{m-1} \in \{-1, 0, +1\}^n$ and $a_m \in \mathbb{Z}$. Define $u = a_0 + 2a_1 + \cdots + 2^m a_m$. If there exists some $k \in [m-1]$ such that $\|a_k\|_1 > s$, then $\|u\|_1 > s$.*

Proof. Since $\|a_k\|_1 > s$, and $a_k \in \{-1, 0, +1\}^n$, there exist at least s coordinates such that $|a_k(i)| = 1$. Let S be the set of those indices, $S = \{i \in [n] \colon |a_k(i)| = 1\}$. Since $\|a_k\|_1 > s$, we know that $|S| > s$. We now show that for all $i \in S, |u(i)| \geq 1$. Therefore, $\|u\|_1 > s$.

Let $i \in S$. For each such coordinate, we can express $u(i) = \sum_{j=0}^{m} 2^j a_j(i)$. Let $h \in [k]$ be the smallest integer such that $a_h(i) \neq 0$. We know that such h exists since $a_k(i) \neq 0$. Therefore, we know that $u(i) \bmod 2^{h+1} (= a_h)$, is non-zero. Therefore, $u(i)$ is also non-zero. Since $u(i) \in \mathbb{Z}$, we have that $|u(i)| \geq 1$.

Therefore, $|u(i)| \geq 1$ for all $i \in S$. and $\|u\|_1 \geq |S| > s$. ∎

Define $a_i = (t_i - v_i)$ for all $i \in [m]$. We note that each $a_i \in \{-1, 0, +1\}^n$ for $i \in [m-1]$ and that $a_m \in \mathbb{Z}^n$. The proof now follows from Claim 4.4 for $s = 2\epsilon_1 n$. ∎

The next claim is a straightforward application of the triangle inequality.

CLAIM 4.5. $d(t, L) \geq d(\lfloor t \rceil, L) - d(t, \mathbb{Z}^n)$

Proof. Let u be the closest lattice vector to t.

$$d(\lfloor t \rceil, L) \leq \|\lfloor t \rceil - u\|_1 = \|\lfloor t \rceil - t + t - u\|_1.$$

By the triangle inequality, we have $\|\lfloor t \rceil - t + t - u\|_1 \leq \|\lfloor t \rceil - t\|_1 + \|t - u\|_1$. Since u is the closest lattice vector to t, $d(t, L) = \|t - u\|_1$. Also, $\|\lfloor t \rceil - t\| = d(t, \mathbb{Z}^n)$, Therefore, $d(\lfloor t \rceil, L) \leq d(t, L) + d(t, \mathbb{Z}^n)$. ∎

Thus, if $d(t, L) \leq \epsilon_1 n$, then T_Z accepts t with probability at least $1 - \frac{c}{m+1}$ and each code tester T_i accepts t_i with probability at least $1 - \frac{c}{m+1}$. Therefore, from the union bound, T accepts t with probability at least $1 - \sum_{i=0}^{m} \frac{c}{m+1} = 1 - c$. ∎

We next instantiate Theorem 1.13 for code-formula lattices obtained from Reed-Muller codes. We first recall a simple observation made in [28] that any local test with individual queries uniformly distributed is also a tolerant test.

CLAIM 4.6 ([28]). *If a code $C \subseteq \{0,1\}^n$ has a one-sided local test $T(\epsilon, 0, 1/3, q)$ whose queries are each uniformly distributed, then C has a tolerant test $T(\epsilon_1, \epsilon_2, 1/3, 1/3, q)$, with $\epsilon_1 \leq \frac{1}{3q}$ and $\epsilon_2 \geq \epsilon$.*

Using Claim 4.6 and Theorem 3.2, and by appropriately amplifying the success probability, we get a tolerant test for Reed-Muller codes.

COROLLARY 4.7. *For any $k, r, c, s > 0$ and $\gamma = \min\{c, s\}$, there exists a tolerant test $T(\epsilon_1, \epsilon_2, c, s, q)$ for $RM(k, r)$ such that $\epsilon_1 \leq c_1 \frac{1}{2^k}$, $\epsilon_2 \geq c_2 \frac{1}{2^k}$ and $q = O(2^k \log(\frac{1}{\gamma}))$, for some $c_1, c_2 > 0$.*

Proof. By Theorem 3.2 we know that there is a 1-sided tester $T(\epsilon, 0, 1/3, q)$ for $RM(k, r)$ and $\epsilon = O(1/2^k)$ with query complexity $q = O(2^k)$. From Claim 4.6, we know that we can obtain a tolerant tester $T(\epsilon_1, \epsilon_2, 1/3, 1/3, q)$ with $O(2^k)$ queries for any $\epsilon_1 \leq c_1/2^k$ and $\epsilon_2 \geq c_2/2^k$. By independently repeating the tester multiple times and taking majority vote to amplify the success probability, for any $0 < c, s \leq 1$ and $\gamma = \min\{c, s\}$ we get a tolerant tester $T(\epsilon_1, \epsilon_2, c, s, q)$ for $RM(n, k)$ with $q = O(2^k \log(\frac{1}{\gamma}))$ queries. ∎

Using Corollary 4.7 and Theorem 1.13 we obtain the following immediate corollary.

COROLLARY 1.14. *Let $0 \leq k_0 < k_1 < \cdots < k_{m-1} < r$ be integers such that the family of Reed-Muller codes $RM(k_0, r) \subseteq RM(k_1, r) \subseteq \cdots \subseteq RM(k_{m-1}, r)$ satisfies the Schur product condition. Let L be the lattice obtained from this family of codes using the code formula construction:*

$$L = RM(k_0, r) + 2RM(k_1, r) + \cdots + 2^{m-1} RM(k_{m-1}, r) + 2^m \mathbb{Z}^{2^r}.$$

Then there exists a ℓ_1-tolerant-tester $T(\epsilon_1, \epsilon_2, 1/3, 1/3, q)$ for L for all $\epsilon_1 \leq \frac{c_1'}{2^{k_{m-1}}}$, $\epsilon_2 \geq \frac{c_2' m}{2^{k_0-1}}$ (for some constants c_1' and c_2') with query complexity $q = O(2^{k_{m-1}} \cdot \log m)$.

Proof. From Corollary 4.7, every RM_{k_i} has a tolerant tester $T_i(2\epsilon_1, \frac{\epsilon_2}{m2^{i+1}}, \frac{1}{3(m+1)}, \frac{1}{3}, q_i)$ with query complexity $q_i = O(2^{k_i} \log(m+1))$ for $2\epsilon_1 \leq \frac{c_1}{2^{k_i}}$ and $\frac{\epsilon_2}{m2^{i+1}} \geq \frac{c_2}{2^{k_i}}$ for some constants $c_1, c_2 > 0$.

Using Theorem 1.13, we therefore conclude that L has a tolerant tester $T(\epsilon_1, \epsilon_2, \frac{1}{3}, \frac{1}{3}, q)$ with query complexity $q = O(\frac{1}{(\epsilon_2 - 2\epsilon_1)^2} \log(m+1)) + \sum_{i=0}^{m-1} O(2^{k_i} \log(m+1)) = O(2^{k_{m-1}} \log m)$ for $2\epsilon_1 \leq \min_i\{\frac{c_1}{2^{k_i}}\}$ and $\frac{\epsilon_2}{m2^{i+1}} \geq \max_i\{\frac{c_2}{2^{k_i}}\}$. ∎

5. Reducing an arbitrary test to a non-adaptive linear test. In this section we sketch the proof of Theorem 1.16. Throughout this section, we focus on full-rank integral lattices. Given a 2-sided adaptive ℓ_p-tester $T(\epsilon, c, s, q)$, with $q = q_T(\epsilon, c, s)$ for an integral lattice L, we construct a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, c+s, q')$ with query complexity $q' = q_T(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$. We reduce the inputs to a bounded set using the following property of integral lattices.

FACT 5.1. *[25] Given any full rank integral lattice L, there exists $d \in \mathbb{Z}$ such that $d \cdot \mathbb{Z}^n \subseteq L$. In particular $|\det(L)| \cdot \mathbb{Z}^n \subseteq L$ for any lattice (where $\det(L)$ denotes the determinant of a lattice, a parameter that can be computed given a basis of the lattice). For instance, we can take $d = 2^m$ for the lattices of height m obtained using the code formula construction.*

Let $V = L \bmod d$ embedded in $\mathbb{Z}^n$ (i.e., we treat V as a set of vectors in $\mathbb{Z}^n$ each of which is obtained by taking coordinate-wise modulo d of some lattice vector). Thus, $V \subseteq \mathcal{Z}_d^n$. We will need the following properties of V.

PROPOSITION 5.2. *Let $L \subseteq \mathbb{Z}^n$ be a full-rank lattice, $d \in \mathbb{Z}_+$ such that $d\mathbb{Z}^n \subseteq L$, and let $V = L \bmod d \subseteq \mathbb{Z}^n$. Then V satisfies the following properties:*

1. *$v \in L$ if and only if $v \bmod d \in V$.*
2. *$V = L \cap \mathcal{Z}_d^n$.*
3. *$(v + V) \bmod d \subseteq V$ if and only if $v \in L$.*
4. *For any $v \in \mathbb{Z}^n$, $d_p(v, L) = d_p(v \bmod d, L)$.*

Proof.

1. If $v \in L$, then $v \bmod d \in V$ by definition. For the opposite direction, let $v \in \mathbb{Z}^n$ be such that $u = v \bmod d \in V$. Then by the definition of V there exists $v' \in L$ such that $v' = u = v \pmod d$. Then $v - v' \in d\mathbb{Z}^n \subseteq L$, and so $v \in L$.
2. By definition $L \cap \mathcal{Z}_d^n \subseteq V$. To show that $V \subseteq L$ note that by 1), if $v \in V$ there exists $v' \in L$ such that $v' = v \bmod d$. As before, this implies that $v \in L$.
3. This statement follows by the fact that $V \subseteq L$ and from the fact that lattices are closed under addition.
4. Note that $d_p(v, L) = \min_{u \in L} d_p(u, v) = \min_{u \in L} \|v - u\|_p$. If $v = dv_1 + v_2$, since $dv_1 \in d\mathbb{Z}^n \subseteq L$, it follows that $\min_{u \in L} \|v - u\|_p = \min_{u \in L} \|v_2 - u\|_p$, since a lattice is closed under addition.

Theorem 1.16 will immediately follow by combining Lemmas 5.3, 5.4, 5.5, and 5.6. We now state the lemmas and prove them in the subsequent subsections.

LEMMA 5.3. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$ has an adaptive 2-sided ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$. Then L has an adaptive linear ℓ_p-tester $T'(\epsilon, 0, c+s, q)$ for inputs from the domain $\mathcal{Z}_d^n$.*

LEMMA 5.4. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$ has an adaptive linear ℓ_p-tester $T(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$. Then L has a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$.*

LEMMA 5.5. *Let $L \subseteq \mathbb{Z}^n$ be a full-rank lattice with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$. Then, L has a non-adaptive linear ℓ_p-tester $T(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$ if and only if L has a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, s, q)$ for inputs from the domain $\mathbb{Z}^n$.*

LEMMA 5.6. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ has a non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathbb{Z}^n$. Then there exists a non-adaptive ℓ_p-tester $T'(\epsilon, c, s, q')$ for inputs in $\mathbb{R}^n$ with query complexity $q' = q(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$. Moreover, if T is a linear tester, then so is T'.*

The proof of Lemma 5.6 uses the following tester for integer lattices which is based on querying a random collection of coordinates and verifying whether all of them are integral.

LEMMA 5.7. *For every $0 < \epsilon \leq 1$ and every $0 < s \leq 1$, there exists a non-adaptive linear ℓ_p-tester $T_p(\epsilon, 0, s, q_Z)$ for $\mathbb{Z}^n$ with query complexity*

$$q_Z = O\left(\frac{1}{\epsilon^p}\log\frac{1}{s}\right).$$

5.1. 2-sided to Linear Tester. In this section, we prove Lemma 5.3. Given a 2-sided adaptive tester $T(\epsilon, c, s, q)$ for inputs x from the domain $\mathcal{Z}_d^n$, we build an adaptive linear (thus one-sided) test $T'(\epsilon, 0, c+s, q)$ for inputs from the same domain $\mathcal{Z}_d^n$ with the same query complexity as that of T in this section.

For an index set $J \subseteq [n]$ and a vector $w \in \mathcal{Z}_d^n$, let $X(w, J) := \{x \in \mathcal{Z}_d^n \mid (\forall\ i \in J)\ x_i = w_i\ \}$. For a subset of coordinates $J \subseteq [n]$ and a vector $w \in \mathcal{Z}_d^n$, we say that *there exists a dual witness for* $X(w, J)$ if there exists $\alpha \in L_J^\perp$ such that $\langle \alpha, w\rangle \notin \mathbb{Z}$. That is, a dual witness α is a dual vector entirely supported on J that proves none of the vectors in $X(w, J)$ (and thus w) can be in the lattice. Recall that $V := L \bmod d$ where $d\mathbb{Z}^n \subseteq L$.

If the input vector x is from the domain $\mathcal{Z}_d^n$, then each coordinate of the input has d possible choices. Thus, any 2-sided adaptive tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathbb{Z}_d^n$, can be viewed as a distribution over deterministic decision trees with each leaf being labeled 1 if accepting and 0 if rejecting. Therefore we will express the tester as $T = (\Upsilon_T, D_T)$, where Υ_T is the set of all decision trees (with at most q queries) and D_T is a distribution over Υ_T.

Let l be a leaf of a decision tree. We denote the coordinates queried along the path to l by $\mathrm{var}(l)$. We denote the vector that is consistent with the queried coordinates along the path to l and has zeros in the non-queried coordinates by s_l. Let us define V_l to be the set of lattice vectors u which are consistent with the queries along the path to l. Similarly, let V_l^x be the set of vectors in $(x + V) \bmod d$ which are consistent with the queries along the path to l, i.e., $V_l = X(s_l, \mathrm{var}(l)) \cap V$ and $V_l^x = X(s_l, \mathrm{var}(l)) \cap ((x + V) \bmod d)$. We need the following claim about the sizes of V_l and V_l^x.

CLAIM 5.8. *For every leaf l in the decision tree Γ, if both V_l and V_l^x are non-empty, then $|V_l| = |V_l^x|$.*

Proof. Let U denote the set of all the lattice vectors in $\mathcal{Z}_d^n$ which have all 0's in the positions queried along the path to l. We know that U is non-empty because the all zeros vector is in U.

For every $v \in V_l$ and $u \in U$, we have that $(v + u) \bmod d$ is also in V_l since we are only adding 0's at the queried coordinates. Similarly, for every vector $v' \in V_l^x$ and $u \in U$, we have that $(v' + u) \bmod d$ is also in V_l^x. Therefore, we know that $(U + v) \bmod d \subseteq V_l$ for every $v \in V_l$ and similarly, $(U + v') \bmod d \subseteq V_l^x$ for every vector $v' \in V_l^x$.

Further, for every two vectors $u, v \in V_l$, we have that $(u - v) \bmod d$ is in U and since u and v are both consistent along the path to l, the vector $u - v$ has all zeros at the queried coordinates. So, $(u - v) \bmod d \in U$. Therefore, $(V_l - v) \bmod d \subseteq U$ for every $v \in V_l$ and hence $V_l \subseteq (U + v) \bmod d$ for every $v \in V_l$. Similarly, for every vector $v' \in V_l^x$, we have that $(V_l^x - v') \bmod d \subseteq U$ and hence $V_l^x \subseteq (U + v') \bmod d$.

Therefore, if V_l and V_l^x are non-empty, then $(U + v) \bmod d = V_l$ for every vector $v \in V_l$ and $(U + v') \bmod d = V_l^x$ for every vector $v' \in V_l^x$. Hence, $|V_l| = |U| = |V_l^x|$. ∎

We now show that if a linear test accepts, then there exists a lattice vector that is consistent with the queried coordinates. In other words, if there is no dual witness then there is a lattice vector that is accepted by the test.

In the following, let $\mathrm{proj}_J(u) \in \mathbb{R}^J$ denote the projection of vector u to the coordinates in J and $\mathrm{proj}_J(S)$ denote the set of vectors obtained by projecting the vectors in S to the coordinates in J. We note that the projection of a rational lattice to a set of coordinates gives a lattice again.

PROPOSITION 5.9. *Let $J \subseteq [n]$, $w \in \mathcal{Z}_d^n$. If $\langle \alpha, proj_J(w) \rangle \in \mathbb{Z}$ for every $\alpha \in proj_J(L_J^\perp)$, then $V \cap X(w, J) \neq \emptyset$.*

Proof. We recall that the dual of a projection of a lattice is the set of vectors in the projected space which have integral dot products with all points in the projected lattice. The following proposition shows that the dual of a projected lattice is the projection of the set of vectors in the dual lattice whose support is contained in the projection.

PROPOSITION 5.10. *Let $J \subseteq [n]$. Then*

$$(proj_J(L))^\perp = proj_J\left(L_J^\perp\right).$$

Proof. Let $\alpha_J \in \text{proj}_J(L_J^\perp)$. Let us extend the vector α_J to $\alpha \in \mathbb{R}^n$ by setting the coordinates that are not in J to zero. We note that $\alpha \in L_J^\perp$. Hence $\langle \alpha, x \rangle \in \mathbb{Z}$ for every $x \in L$. Therefore $\langle \alpha_J, x_J \rangle \in \mathbb{Z}$ for every $x_J \in \text{proj}_J(L)$. Thus, $\alpha_J \in (\text{proj}_J(L))^\perp$.

Let $\alpha_J \in (\text{proj}_J(L))^\perp$. Then for every $v_J \in \text{proj}_J(L)$, we have $\langle \alpha_J, v_J \rangle \in \mathbb{Z}$. Consequently for every $v \in L$, we have $\langle \alpha_J, \text{proj}_J(v) \rangle \in \mathbb{Z}$. Let us extend the vector α_J to $\alpha \in \mathbb{R}^n$ by setting the coordinates that are not in J to zero. Then $\langle \alpha, v \rangle \in \mathbb{Z}$ for every $v \in L$. Therefore $\alpha \in L^\perp$ and hence $\alpha_J \in \text{proj}_J(L_J^\perp)$. ∎

We have that $\langle \alpha, \text{proj}_J(w) \rangle \in \mathbb{Z}$ for every $\alpha \in \text{proj}_J(L_J^\perp)$. Therefore $\text{proj}_J(w) \in (\text{proj}_J(L_J^\perp))^\perp$. By Proposition 5.10, we have that $\text{proj}_J(w) \in \text{proj}_J(L)$. Hence, there exists $x \in X(w, J) \cap L = X(w, J) \cap V$. ∎

Note that it is possible to determine if there exists a dual witness for $X(w, J)$ and if so, find one efficiently as shown in Proposition 5.11.

PROPOSITION 5.11. *Given $w \in \mathcal{Z}_d^n$ and $J \subseteq [n]$, we can find a dual witness for $X(w, J)$ if one exists or confirm that no dual witness for $X(w, J)$ exists in time $O(|J|^\omega)$, where $O(m^\omega)$ is the time to compute the inverse of a $m \times m$ real matrix.*

Proof. A basis for $\text{proj}_J(L)$ can be obtained by projecting the basis for L. Now a basis for the dual of the projected lattice, namely $\text{proj}_J(L)^\perp = \text{proj}_J(L_J^\perp)$, can be computed in time $O(|J|^\omega)$. We observe that for every $\alpha \in L_J^\perp$, we have $\langle \alpha, w \rangle \in \mathbb{Z}$ if and only if for every basis vector b of $\text{proj}_J(L_J^\perp)$, we have $\langle b, \text{proj}_J(w) \rangle \in \mathbb{Z}$. Hence it is sufficient to only verify the inner product of $\text{proj}_J(w)$ with the basis vectors of $\text{proj}_J(L_J^\perp)$. ∎

We now have the ingredients needed to prove Lemma 5.3.

LEMMA 5.3. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$ has an adaptive 2-sided ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$. Then L has an adaptive linear ℓ_p-tester $T'(\epsilon, 0, c + s, q)$ for inputs from the domain $\mathcal{Z}_d^n$.*

Proof. We first relabel the decision tree according to the rule required for a linear test: Given a decision tree Γ for the tester T, we say that it is *optimally labeled* if the label of any leaf l is 0 whenever there exists a dual witness for $X(s_l, \text{var}(l))$ and 1 otherwise. We denote the tree obtained from Γ by optimally relabeling to be Γ_{OPT} (the relabeling for a given leaf of a tree Γ can be done efficiently by Proposition 5.11). We build a tester T' as follows:

1. On input $x \in \mathcal{Z}_d^n$, choose a tree Γ according to D_T.
2. Choose a uniformly random vector v in V (recall that $V := L \bmod d$).
3. Answer according to the relabeled decision tree Γ_{OPT} on input $(x + v) \bmod d$.

It is clear that T' is a linear test and has the same query complexity as that of T. We now show that the probability of acceptance by T' of any vector w which is ϵ-far from L, does not exceed $c + s$. Let us define the following for a tester $\bar{T}$:

$$\rho^{\bar{T}} := \operatorname*{avg}_{y \in V} Pr[\bar{T}(y) = 1],$$

$$\rho_x^{\bar{T}} := \operatorname*{avg}_{y \in (x+V) \bmod d} Pr[\bar{T}(y) = 1].$$

Due to the randomness in the choice of the tester T', we have

$$\rho^{T'} = Pr[T'(x) = 1 \mid x \in V],$$
$$\rho_x^{T'} = Pr[T'(x) = 1].$$

Since T' is a 1-sided tester, we have that $\rho^{T'} = 1$. Since T accepts lattice vectors with probability at least $1 - c$, we have $\rho^T \geq 1 - c$. Let $x \in \mathcal{Z}_d^n$ be ϵ-far from L. For every $v \in V$, we have that $(x + v) \bmod d$ is also ϵ-far from L by Proposition 5.2. Therefore, $\rho_x^T \leq s$. Using Claim 5.12, we have

$$\rho_x^{T'} \leq \rho^{T'} - \rho^T + \rho_x^T \leq 1 - (1 - c) + s = c + s. \qquad \square$$

CLAIM 5.12. *For every $x \in \mathcal{Z}_d^n$,*

$$\rho_x^{T'} \leq \rho^{T'} - \rho^T + \rho_x^T.$$

Proof. Let x be a vector in $\mathcal{Z}_d^n$. We analyze the effect of relabeling a single leaf l of the decision tree Γ. We show that relabeling l optimally preserves the claim and hence by repeated relabeling, we can deduce the claim.

Case (i). There exists a dual witness for $X(s_l, \text{var}(l))$. Then the leaf l is relabeled from 1 to 0. If input $y \in X(s_l, \text{var}(l))$, then y cannot be a lattice vector (if y is a lattice vector, then there cannot exist a dual witness for $X(s_l, \text{var}(l))$). Therefore, the probability of acceptance of lattice vectors is not changed due to relabeling, i.e., $\rho^{T'} = \rho^T$. If the leaf l is reached for input $y \in \mathcal{Z}_d^n \setminus V$, then T' rejects. Thus, relabeling does not increase the probability of acceptance of non-lattice vectors, i.e., $\rho_y^{T'} \leq \rho_y^T$. Therefore, $\rho_x^{T'} \leq \rho^{T'} - \rho^T + \rho_x^T$ holds for this case.

Case (ii). There does not exist a dual witness for $X(s_l, \text{var}(l))$. Then the leaf l is relabeled from 0 to 1.

The set of vectors in $V_l \cup V_l^x$ were rejected by T and, after optimal relabeling of the leaf l, are now accepted by T'. The rest of the vectors in V and $(x + V) \bmod d$ are rejected/accepted equally by both T and T'.

Now, if y was a lattice vector, then the probability of accepting a lattice vector increases because of the relabeling of l. Among the vectors in V, the vectors in V_l are precisely the ones which were rejected before relabeling and are now accepted after relabeling. Since we average over all possible vectors $y \in V$ in the definition of ρ^T, the fractional change in the acceptance probability given that T' and T chose the decision tree Γ is exactly $|V_l|/|V|$. Therefore,

$$\rho^{T'} = \rho^T + D_T(\Gamma)\frac{|V_l|}{|V|}.$$

Among the vectors in $(x + V) \bmod d$, the vectors in V_l^x are the only vectors which were rejected before relabeling and are now accepted after relabeling. Thus, the fractional change in the acceptance probability of $(x + v) \bmod d$ given that T' and T chose the decision tree Γ is exactly $|V_l^x|/|V|$. Therefore,

$$\rho_x^{T'} = \rho_x^T + D_T(\Gamma)\frac{|V_l^x|}{|V|}.$$

Combining the two equations, we get

$$\rho_x^{T'} = \rho^{T'} - \rho^T + \rho_x^T + \frac{D_T(\Gamma)}{|V|}(|V_l^x| - |V_l|).$$

Using Claim 5.8, we know that $|V_l^x| \leq |V_l|$ if V_l is non empty. Since there does not exist a dual witness for l, by Proposition 5.9, we have that V_l is non-empty. Hence the claim follows. $\square$

5.2. Adaptive to Non-adaptive. In this section we show that given an adaptive linear tester for a lattice, we can construct a non-adaptive linear tester from it without increasing the query complexity or the acceptance probability of non-lattice vectors.

LEMMA 5.4. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$ has an adaptive linear ℓ_p-tester $T(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$. Then L has a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$.*

Proof. Let $T(\epsilon, 0, s, q)$ be an adaptive linear tester for inputs from the domain $\mathcal{Z}_d^n$ with query complexity q. We construct a non-adaptive linear tester $T'(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$ as follows:

1. On input $x \in \mathcal{Z}_d^n$, choose a random vector $v \in V$.
2. Run T on input v. Let J denote the set of coordinates that are queried.
3. Query x on all the coordinates in J.
4. Reject if and only if there exists a dual witness for $X(x, J)$.

We note that T' is a linear test and the query complexity of T' is the same as the query complexity of T. Since the queries depend only on a random $v \in V$ and not on the input x, the test T' is non-adaptive. It remains to bound the acceptance probability of non-lattice vectors by T'. We will show that there is no dual witness for $X(x, J)$ if and only if there exists a vector $y \in (x + V) \bmod d$ that is consistent with the queried coordinates of v. As a consequence, we will show that the probability that T' accepts x is identical to the average acceptance probability of $x + v$ for random vectors $v \in V$ by T. Before analyzing the acceptance probability, we introduce a few notations and observations.

For a decision tree $\Gamma \in \Upsilon_T$, we denote the set of leaves of Γ which are labeled 1 by $l_1(\Gamma)$. For a leaf l of Γ and a vector $x \in \mathcal{Z}_d^n$, let I_l^x be a boolean (indicator) variable which takes a value of 1 if and only if $\langle \alpha, x \rangle \in \mathbb{Z}$ for every $\alpha \in L_{var(l)}^{\perp}$.

Let $\bar{\Gamma}$ be the decision tree chosen by the tester T' on input x. The random vector $v \in V$ chosen by T' corresponds to a leaf labeled 1 in $\bar{\Gamma}$. This is because T is a linear test and hence a lattice vector v cannot have any dual witness. Therefore, $v \in V_{\bar{l}}$ for some $\bar{l} \in l_1(\bar{\Gamma})$. Since T' is a linear test it is clear that T' accepts x if and only if $I_{\bar{l}}^x = 1$.

Claim 5.13. *Let l be a leaf of a decision tree $\Gamma \in \Upsilon_T$, $x \in \mathcal{Z}_d^n$ and $y \in (x + V) \bmod d$. We have that $I_l^x = 1$ if and only if $I_l^y = 1$.*

Proof. If $y \in (x + V) \bmod d$, then $x - y \in L$. If $x, y \in \mathbb{Z}^n$ belong to the same coset of L, then for every $a \in L^{\perp}$, we have that $\langle x, a \rangle \in \mathbb{Z}$ if and only if $\langle y, a \rangle \in \mathbb{Z}$. Therefore, there exists $\alpha \in L_{var(l)}^{\perp}$ such that $\langle \alpha, x \rangle \notin \mathbb{Z}$ if and only if there exists $\alpha \in L_{var(l)}^{\perp}$ such that $\langle \alpha, y \rangle \notin \mathbb{Z}$. Hence $I_l^x = 1$ if and only if $I_l^y = 1$ for every $y \in (x + V) \bmod d$. ∎

Claim 5.14. *Let $x \in \mathcal{Z}_d^n$ and l be a leaf of a decision tree $\Gamma \in \Upsilon_T$ such that $l \in l_1(\Gamma)$. Then $|V_l^x| = I_l^x |V_l|$.*

Proof. We know that for every leaf l which is labeled 1, the set V_l is non-empty since T is a linear tester (using Proposition 5.9). By Claim 5.8 we know that $|V_l^x| = |V_l|$ if V_l^x is also non-empty. Therefore it is sufficient to show that $I_l^x = 1$ if and only if V_l^x is non-empty

If V_l^x is non-empty, then by definition, there is a vector $y \in (x + V) \bmod d$ which is consistent with all the queries along the path to l. Since l is labeled 1, we know that T accepts y. Since T is a linear tester, this implies that there does not exist an $\alpha \in L_{\mathrm{var}(l)}^{\perp}$ such that $\langle \alpha, x \rangle \notin \mathbb{Z}$. Hence $I_l^y = 1$. By Claim 5.13, we know that I_l^x is also 1.

If $I_l^x = 1$, then for every $\alpha \in L_{\mathrm{var}(l)}^{\perp}$, we have $\langle \alpha, x \rangle \in \mathbb{Z}$. By Proposition 5.9, there exists a vector $v \in V \cap X(x, \mathrm{var}(l))$. Hence, we have a vector $v \in V$ whose entries are identical to that of x at the coordinates in $\mathrm{var}(l)$. We observe that the vector $(x - v) \bmod d$ has all 0 entries at the coordinates in $\mathrm{var}(l)$. Further, V_l is non-empty since l is labeled 1. Let $u \in V_l$. Then $((x - v) + u) \bmod d$ is consistent with all queries along the path to l, and is in $(x + V) \bmod d$. Therefore V_l^x is non-empty. ∎

We now show that the acceptance probability of T' is equal to the average acceptance probability of T. Let

$$\rho_x := \operatorname*{avg}_{v \in V} \Pr[T((x + v) \bmod d) = 1].$$

We note that this quantity is 1 if $x \in V$ and is at most s if x is ϵ-far from the lattice L. The following claim shows that T' accepts an input vector x with probability 1 if $x \in V$ and with probability at most s if x is ϵ-far from the lattice L. ∎

Claim 5.15. *Let $x \in \mathcal{Z}_d^n$. Then $\Pr[T'(x) = 1] = \rho_x$.*

Proof. The average acceptance probability of T can be viewed as follows: we pick a decision tree Γ according to D_T. Then we pick a leaf l labeled 1 with probability proportional to the fraction of vectors in

$x + V \bmod d$ that are consistent with the queries along the path to l. Therefore,

$$\rho_x = \sum_{\Gamma \in \Upsilon_T} D_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l^x|}{|V|} \right).$$

We have seen that $T'(x) = 1$ if and only if for the random vector $v \in V$ chosen by T', and a leaf $\bar{l} \in l_1(\bar{\Gamma})$ such that $v \in V_{\bar{l}}$, we have $I_{\bar{l}}^x = 1$. Thus the execution of T' can be treated as follows: First, pick a decision tree $\Gamma \in \Upsilon_T$ according to $D_T(\Gamma)$, then choose a leaf l labeled 1 in Γ with probability proportional to the fraction of vectors in V that are consistent with the queries to the coordinates in l. Finally, query x on the variables in $\mathrm{var}(l)$ and accept if and only if $I_l^x = 1$. Therefore, the acceptance probability of T is given by

$$\Pr[T(x) = 1] = \sum_{\Gamma \in \Upsilon_T} D_T(\Gamma) \left(\sum_{l \in l_1(\Gamma)} \frac{|V_l|}{|V|} \cdot I_l^x \right).$$

By Claim 5.14, we see that $\rho_x = \Pr[T'(x) = 1]$. ∎

5.3. Handling real-valued inputs. In this section, we build a tester for real-valued inputs using a tester for bounded integral inputs. We first show how to handle all integral inputs using a tester for integral inputs from a bounded domain.

LEMMA 5.5. *Let $L \subseteq \mathbb{Z}^n$ be a full-rank lattice with $d\mathbb{Z}^n \subseteq L$ for $d \in \mathbb{Z}_+$. Then, L has a non-adaptive linear ℓ_p-tester $T(\epsilon, 0, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$ if and only if L has a non-adaptive linear ℓ_p-tester $T'(\epsilon, 0, s, q)$ for inputs from the domain $\mathbb{Z}^n$.*

Proof. If we have a tester $T'(\epsilon, c, s, q)$ for integral inputs, then the same tester can be applied to inputs in $\mathcal{Z}_d^n$ with the same completeness and soundness parameters and the same query complexity. Given a tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathcal{Z}_d^n$, we construct the tester $T'(\epsilon, c, s, q)$ for arbitrary integral inputs as follows: On input $x \in \mathbb{Z}^n$ run $T(\epsilon, c, s, q)$ on $w := x \bmod d$, and output the result.

If x is a lattice vector, then from Proposition 5.2, we know that w is also a lattice vector, and therefore T' accepts x with probability at least $1 - c$. If x is ϵ-far from the lattice, then again from Proposition 5.2, we know that w is also ϵ-far from the lattice and T' will accept x with probability at most s. We note that the query complexity of T' is identical to that of T. ∎

To address the case of real inputs, we will design a tester for the integer lattice.

LEMMA 5.7. *For every $0 < \epsilon \leq 1$ and every $0 < s \leq 1$, there exists a non-adaptive linear ℓ_p-tester $T_p(\epsilon, 0, s, q_Z)$ for $\mathbb{Z}^n$ with query complexity*

$$q_Z = O\left(\frac{1}{\epsilon^p} \log \frac{1}{s} \right).$$

Proof. The test queries $O((1/\epsilon^p)\log(1/s))$ coordinates of the input uniformly at random and accepts iff all the queried coordinates are integral.

If the input is in the lattice, then all the queried coordinates will be integral, and hence the tester will accept. If the input w is at ℓ_p distance at least $\epsilon \cdot \|1^n\|_p$, then at least $\epsilon^p n$ coordinates of the input are non-integral. Thus the tester will reject with probability at least $1 - s$.

We note that the tester is a linear test: the test described can be viewed as picking independent uniform random standard basis vectors $e_i \in \mathbb{Z}^n \subseteq L^\perp$ (where e_i is the indicator vector of the index i), for $i \in [n]$, and testing if the input w satisfies $\langle w, e_i \rangle \in \mathbb{Z}$.

LEMMA 5.6. *Suppose a full-rank lattice $L \subseteq \mathbb{Z}^n$ has a non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs from the domain $\mathbb{Z}^n$. Then there exists a non-adaptive ℓ_p-tester $T'(\epsilon, c, s, q')$ for inputs in $\mathbb{R}^n$ with query complexity $q' = q(\epsilon/2, c, s) + O((1/\epsilon^p)\log(1/s))$. Moreover, if T is a linear tester, then so is T'.*

Proof. Suppose we have a ℓ_p-tester $T(\epsilon, c, s, q)$ for integer inputs. We can build a tester T' for real valued inputs as follows:

1. On input $x \in \mathbb{R}^n$, run the ℓ_p-tester $\bar{T}(\epsilon/2, 0, s, q_Z)$ for $\mathbb{Z}^n$ from Lemma 5.7 on input x. If the tester $\bar{T}$ rejects, then reject.
2. Else, run $T(\epsilon/2, c, s, q')$ on x where $q' = q(\epsilon/2, c, s)$ and reject immediately if any of the coordinates queried are not integers; otherwise output the result of T.

If x is a lattice vector, then the acceptance probability of T' is the same as that of T since the tester used in step 1 is a linear tester. If $d_p(x, L) \geq \epsilon \cdot \|1^n\|_p$, then $d_p(x, \lfloor x \rceil) + d_p(\lfloor x \rceil, L) \geq d_p(x, L) \geq \epsilon \cdot \|1^n\|_p$ and therefore either $d_p(x, \lfloor x \rceil)$ or $d_p(\lfloor x \rceil, L)$ is at least $\frac{1}{2}\epsilon \cdot \|1^n\|_p$. If $d_p(x, \lfloor x \rceil) \geq \frac{1}{2}\epsilon \cdot \|1^n\|_p$, then $d_p(x, \mathbb{Z}^n) = d_p(x, \lfloor x \rceil) \geq \frac{1}{2}\epsilon \cdot \|1^n\|_p$ and therefore step 1 rejects with probability at least $1 - s$. If $d_p(\lfloor x \rceil, L) \geq \frac{1}{2}\epsilon \cdot \|1^n\|_p$, then step 2 rejects with probability at least $1 - s$.

The number of queries made by the tester T' is $q(\epsilon/2, c, s) + O((1/\epsilon^p) \log(1/s))$. We note that since the tester used in step 1 is a non-adaptive linear tester, T' would be a non-adaptive linear tester if T is a non-adaptive linear tester. ∎

REMARK 5.16. *We note that the test described in the proof of Theorem 1.7 is not a linear test by definition. We now describe a linear test for the code-formula lattice which is equivalent to the test described in Section 3.1.*

Let T_p denote the tester for $\mathbb{Z}^n$. We assume that each code tester T_i for the code C_i is linear [4] (i.e T_i queries the input $t_i \in \{0,1\}^n$ at $I_i = \{i_1, \cdots, i_q\} \subseteq [n]$ coordinates according to some distribution and accepts it if and only if $\langle t_i, v \rangle \equiv 0 \bmod 2$ for every $v \in C_{I_i}^{\perp}$). Consider the following variant of the test, that we call T_{linear}, which by definition is a linear test:

1. *Let each T_i query $I_i \subseteq [n]$ coordinates and let T_p query I_p coordinates.*
2. *Let $I = \cup_i I_i \cup I_p$.*
3. *Accept t if $\langle t, x \rangle \in \mathbb{Z}$ for all $x \in (L^{\perp})_I$*
4. *Reject otherwise.*

Note that the query complexity of T_{linear} is upper bounded by the query complexity of T.

If the input is a lattice vector; i.e., $t \in L$, then by definition, the inner product of t with every dual lattice vector would be an integer. Therefore, the test is 1-sided.

We now show that T_{linear} rejects all inputs t which are rejected by T and hence, T_{linear} performs at least as well as T. If T rejects t, then there is some $i \in \{0, 1, \ldots, m-1\}$ such that t_i is rejected by T_i or t is rejected by T_p. We note that each code tester T_i and also T_p are linear. Therefore, if T_i rejects t_i, then there are no codewords of C_i which agree with t_i on the coordinates I_i queried by T_i. So, for the set I which contains I_i, there are no codewords of C_i which agree with t_i on the coordinates in I. If T_p rejects t, then there is some non-integral coordinate in I_p and hence in I. By definition of the code formula construction, t is a lattice vector if and only if for each $i = 0, \ldots m-1$, t_i is a codeword in C_i and $t \in \mathbb{Z}^n$. Hence, no lattice vector of L agrees with t on those set of coordinates. Therefore, there exists a dual lattice vector supported on I, which does not have an integral inner product with t. Therefore, T_{linear} also rejects t (and thus, has at least as good a soundness as the original test T).

6. Testing membership of inputs outside the span of the lattice. In this section we prove Theorems 1.17, 1.18, Corollary 1.19 and Theorem 1.20. We first recall the definitions. Let L be a rank k lattice in $\mathbb{Z}^n$. Let S denote the $span(L)$ and $S^{\perp}$ be the subspace orthogonal to S. Let $U = [u_1, \cdots, u_{n-k}]^T \in \mathbb{R}^{(n-k)\times n}$ be an orthonormal basis for $S^{\perp}$. Let $P \subseteq [n]$ be the set of coordinates that support the vectors in $S^{\perp}$ i.e.,

$$P := \bigcup_{i \in [n-k]} supp(u_i).$$

THEOREM 1.17. *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in $span(L)^{\perp}$. Let $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for L for inputs in $\mathbb{R}^n$ has query complexity*

$$q = \Omega(|P|).$$

Proof. To show the $\Omega(|P|)$ lower bound, we use Yao's principle: we setup a distribution $\mathcal{D}$ on far inputs such that every deterministic algorithm requires $\Omega(|P|)$ queries to distinguish whether the input is $0 \in L$ or is far from L. We define $\mathcal{D}$ as follows: pick j uniformly at random from P, and set $t^{(j)} := De_j$, where $D \geq \frac{\epsilon \cdot \|1^n\|_p}{\min_{i,j: u_{i,j} \neq 0} |u_{i,j}|}$. The following claim shows that the distance of each such $t^{(j)}$ from L is at least $\epsilon \cdot \|1^n\|_p$.

CLAIM 6.1. *$d_p(t^{(j)}, L) \geq \epsilon \cdot \|1^n\|_p$ for every $j \in P$.*

Proof. It is sufficient to show that $t^{(j)}$ is far from S, since $L \subseteq S$. Let $t^{(j)} = t^{(j)\|} + t^{(j)\perp}$, where $t^{(j)\|}$ is the component of $t^{(j)}$ in S and $t^{(j)\perp}$ is the component of t in $S^\perp$. By definition,

$$t^{(j)\perp} = proj_{S^\perp}(t^{(j)}) = \sum_{\ell \in [n-k]} \langle t^{(j)}, u_\ell \rangle u_\ell.$$

Since U is an orthonormal basis of $S^\perp$,

$$\|t^{(j)\perp}\|_p^p = \sum_{\ell \in [n-k]} |\langle t^{(j)}, u_\ell \rangle|^p = \sum_{\ell \in [n-k]} (Du_{\ell,j})^p \geq (\epsilon \cdot \|1^n\|_p)^p.$$

The last inequality follows from the choice of D and the fact that there exists at least one $u_{\ell,j} \neq 0$ since $j \in P$. Therefore, the distance of $t^{(j)}$ from S and hence from L, is at least $\epsilon \cdot \|1^n\|_p$. ∎

By the choice of the distribution, every deterministic test fails on inputs drawn from $\mathcal{D}$ with probability $1/|P|$. Thus any randomized test requires $\Omega(|P|)$ queries in order to succeed with constant probability. ∎

THEOREM 1.18. *Let $L \subseteq \mathbb{Z}^n$ be a lattice of rank k. Let $P \subseteq [n]$ be the support of the vectors in $span(L)^\perp$. Let $0 < \epsilon, c, s < 1$, and suppose L has an ℓ_p-tester $T(\epsilon, c, s, q)$ for inputs $t \in span(L)$. Then L has a tester $T'(2\epsilon, c, s, q')$ for inputs in $\mathbb{R}^n$ with query complexity*

$$q' \leq q + |P|.$$

Proof. Let $T(\epsilon, c, s, q)$ be an ℓ_p-tester for L for inputs in $span(L)$ with query complexity $q = q(\epsilon)$. We now design a tester $T'(\epsilon', c', s', q')$ for L for inputs $t = (t_1, t_2, \cdots, t_n) \in \mathbb{R}^n$. By making an additional $|P|$ queries, T' can compute the coordinates of the projection of t onto S. If t is far from L, then either (i) t is far from S or (ii) t is close to S but far from L. The coordinates in P would identify if t is far from S and enable rejection. If t is close to S but far from L, the tester T would reject the projection of t onto S thus enabling rejection. We now formalize this intuition.

Let $t = (t_1, t_2, \cdots, t_n) \in \mathbb{R}^n$ be the input to the tester T'. We compute the projection of t on $span(L)$ by querying all the coordinates in P. Let $t^\perp$ be the projection of t onto $S^\perp$. Since U is an orthonormal basis for $S^\perp$, we have

$$t^\perp = \sum_{\ell \in [n-k]} \langle t, u_\ell \rangle u_\ell.$$

Each inner product in this expression can be computed using only the coordinates in P and therefore, $t^\perp$ can be computed from t by querying just $|P|$ coordinates. If $\|t^\perp\|_p \geq \epsilon'/2 \cdot \|1^n\|_p$, then T' rejects t immediately. So we now assume $\|t^\perp\|_p < \epsilon'/2 \cdot \|1^n\|_p$ The projection of t onto S is:

$$t_j^{\|} = \begin{cases} t_j & \text{if } j \notin P \\ t_j - t_j^\perp & \text{if } j \in P \end{cases}$$

Now we run the tester for T on input $t^{\|}$ for distance parameter $\epsilon = \epsilon'/2$ and accept t if and only if T accepts.

If $t \in L$, then $t^\perp = 0$ and T would accept with probability at least $1 - c$. If $d_p(t, L) \geq \epsilon' \cdot \|1^n\|_p$, then

$$d_p(t^{\|}, L) \geq \epsilon' \cdot \|1^n\|_p - d_p(t^\perp, S) \geq \epsilon'/2 \cdot \|1^n\|_p = \epsilon \cdot \|1^n\|_p.$$

Therefore, T would reject with probability at least $1 - s$. Finally, note that $q'(\epsilon') \leq q(\epsilon'/2) + |P|$. ∎

6.1. Testing Knapsack Lattices.

COROLLARY 1.19. *Let $a_1, \ldots, a_n$ be integers and $0 < \epsilon, c, s < 1$. Every non-adaptive ℓ_p-tester $T(\epsilon, c, s, q)$ for $L_{a_1,\ldots,a_{n-1}}$ has query complexity*

$$q = \Omega(n).$$

Proof. We note that L has rank $n - 1$ and the vector $(a_1, a_2, \ldots, a_{n-1}, -1)$ generates the subspace orthogonal to $span(L)$, hence the set P of elements in the support of this space has size $|P| = n$, and the lower bound follows from Theorem 1.17. ∎

We now prove Theorem 1.20, namely that knapsack lattices can be tested with a constant number of queries if the inputs come from the span of the lattice. In fact, we will show that testing such lattices simply reduces to testing membership in $\mathbb{Z}^n$.

THEOREM 1.20. *Let $a_1, \ldots, a_{n-1}$ be integers with $M = \max_{i\in[n]} |a_i|^p$ and $0 < \epsilon, s < 1$. There exists a non-adaptive ℓ_p-tester $T(\epsilon, 0, s, q)$ for $L_{a_1,\ldots,a_{n-1}}$ with query complexity $q = O\left(\frac{M}{\epsilon^p} \cdot \log \frac{1}{s}\right)$, if the inputs are guaranteed to lie in span(L).*

Proof. Let $L = L_{a_1,\ldots,a_{n-1}}$. Let $w \in \text{span}(L)$ denote the input. Any vector $w \in span(L)$ is of the form

$$w = \left(\alpha_1, \cdots, \alpha_{n-1}, \sum_{i=1}^{n-1} a_i\alpha_i\right)$$

for some real values $\alpha_1, \ldots, \alpha_{n-1}$. Let $w' \in \mathbb{R}^{n-1}$ denote the projection of w on the first $n-1$ coordinates. Let $T_p(\epsilon', 0, s', q')$ denote the ℓ_p-tester for $\mathbb{Z}^{n-1}$, where $q' = O\left(\left(\frac{1}{\epsilon'^p}\right) \log \frac{1}{s'}\right)$.

The tester proceeds as follows: Run the tester $T_p(\epsilon' = \epsilon/(M+1)^{1/p}, 0, s, q = O\left(\left(\frac{M}{\epsilon^p}\right) \log \frac{1}{s}\right))$ on input w'. Accept if and only if the tester T_p accepts.

The query complexity of the tester is immediate. If $w \in L$, then each coordinate is integral. Therefore the test accepts w with probability 1. We use the following claim to analyze the soundness of the test.

CLAIM 6.2. *Let $w \in span(L)$, and $w' = (w_1, \cdots, w_{n-1}) \in \mathbb{R}^{n-1}$ then,*

$$d(w, L)^p \leq (M+1) \cdot d(w', \mathbb{Z}^{n-1})^p$$

Proof. Consider the following vector $v \in L$:

$$v = (\lfloor w_1 \rceil, \cdots, \lfloor w_{n-1} \rceil, \sum_{i=1}^{n-1} a_i \lfloor w_i \rceil)$$

where $\lfloor w_i \rceil$ denotes the rounding of w_i the nearest integer. We now upper bound the distance of w from L using this lattice vector v.

$$\begin{aligned}
d(w, L)^p &\leq d(w, v)^p = \|w - v\|_p^p \\
&= \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p + |w_n - \sum_{i=1}^{n-1} a_i \lfloor w_i \rceil|^p \\
&= \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p + |\sum_{i=1}^{n-1} a_i w_i - \sum_{i=1}^{n-1} a_i \lfloor w_i \rceil|^p \\
&\leq \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p + \sum_{i=1}^{n-1} |a_i(w_i - \lfloor w_i \rceil)|^p \\
&\leq \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p + M \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p \\
&= (M+1) \cdot \sum_{i=1}^{n-1} |w_i - \lfloor w_i \rceil|^p \\
&= (M+1) \cdot d(w', \mathbb{Z}^{n-1})^p
\end{aligned}$$

□

It remains to bound the soundness error probability. If $d(w, L) \geq \epsilon \|1^n\|_p$, then from Claim 6.2, we get that $d(w', \mathbb{Z}^{n-1}) \geq (\epsilon/(M+1)^{1/p}) \|1^n\|_p$. Therefore, the tester T_p rejects w with probability at least $1 - s$.

Acknowledgments. We thank Chris Peikert for mentioning to us about the potential application to cryptanalysis, and anonymous reviewers for helpful comments and pointers.

REFERENCES

[1] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.

[2] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[3] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[4] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005. Earlier version in STOC'03.

[5] P. Berman, S. Raskhodnikova, and G. Yaroslavtsev. L_p-testing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173, 2014.

[6] A. Bhattacharyya, S. Kopparty, G. Schoenebeck, M. Sudan, and D. Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497, 2010.

[7] 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.

[8] J. Conway, N. Sloane, and E. Bannai. *Sphere Packings, Lattices and Groups*. A series of comprehensive studies in mathematics. Springer, 1999.

[9] F. Eisenbrand. Fast integer programming in fixed dimension. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, pages 196–207, 2003.

[10] U. Erez, S. Litsyn, and R. Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005.

[11] G. D. Forney. Coset codes-I: Introduction and geometrical classification. *IEEE Transactions on Information Theory*, 34(5):1123–1151, 1988.

[12] K. Friedl and M. Sudan. Some improvements to low-degree tests. In *Proceedings of the 3rd Annual Israel Symposium on Theory and Computing Systems*, 1995.

[13] P. Gaborit and G. Zémor. On the construction of dense lattices with a given automorphisms group. *Annales de l'institut Fourier*, 57(4):1051–1062, 2007.

[14] O. Goldreich. Short locally testable codes and proofs: A survey in two parts. In *Property Testing - Current Research and Surveys*, pages 65–104, 2010.

[15] V. Guruswami and A. Rudra. Tolerant locally testable codes. In *Proceedings of RANDOM/APPROX 2005*, pages 306–317, 2005.

[16] R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.

[17] R. M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.

[18] T. Kaufman and M. Sudan. Algebraic property testing: The role of invariance. In *STOC*, pages 403–412, 2008.

[19] S. Kopparty and S. Saraf. Tolerant linearity testing and locally testable codes. In *Proceedings of RANDOM*, pages 601–614, 2009.

[20] W. Kositwattanarerk and F. E. Oggier. Connections between construction D and related constructions of lattices. *Des. Codes Cryptography*, 73(2):441–455, 2014.

[21] J. Leech and N. Sloane. Sphere packings and error-correcting codes. *Canad. J. Math*, 23(4):718–745, 1971.

[22] H. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[23] R. C. Merkle and M. E. Hellman. Hiding information and signatures in trapdoor knapsacks. *Information Theory, IEEE Transactions on*, 24(5):525–530, 1978.

[24] D. Micciancio. *The LLL Algorithm: Survey and Applications*, chapter Cryptographic functions from worst-case complexity assumptions, pages 427–452. Information Security and Cryptography. Springer, December 2009. Prelim. version in Proc. of LLL25, 2007.

[25] D. Micciancio. Lecture notes on lattice algorithms and applications, Winter 2012, Lecture 2, 2012.

[26] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, March 2002.

[27] A. M. Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and computational number theory*, 42:75–88, 1990.

[28] M. Parnas, D. Ron, and R. Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

[29] O. Regev. Lattice-based cryptography. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, pages 131–141, 2006.

[30] O. Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204, 2010.

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

[32] A. Shamir. A polynomial time algorithm for breaking the basic merkle-hellman cryptosystem. In *Advances in Cryptology*, pages 279–288. Springer, 1983.

[33] L. A. Wolsey and G. L. Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

[34] A. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of Annual IEEE Symposium*

on Foundations of Computer Science, pages 222–227, 1977.