Towards Lower Bounds on
Locally Testable Codes

Michael Viderman
Towards Lower Bounds on Locally Testable Codes

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Michael Viderman

Submitted to the Senate of
the Technion — Israel Institute of Technology
Tevet 5772 Haifa January 2012
The research thesis was done under the supervision of Assoc. Prof. Eli Ben-Sasson in the Computer Science Department.

I am deeply grateful to my supervisor, Eli Ben-Sasson, for his guidance, encouragement and help throughout the years. I thank Eli for pointing out many interesting research directions and for invaluable discussions. I would like to thank Eli for his willingness to teach and assist in any given moment. I really appreciate the good fortune of having the chance to work with you.

Furthermore, I am indebted to my collaborators in this work: Venkatesan Guruswami, Tali Kaufman and Madhu Sudan, for the pleasant and fruitful work together.

I am thankful to Eldar Fischer for many invaluable comments on an earlier version of this thesis.

I thank my dear family, and mostly my mother and farther, for their endless encouragement and support, during all these years.

Last but not least I would like to thank my girlfriend, Alexandra Dolzhenko, for her love, friendship and patience through the way. I couldn’t have made it without you.

The generous financial support of the Technion is gratefully acknowledged.
## Contents

Abstract

Abbreviations and Notations

1 General Introduction
   1.1 PCP theorem
   1.2 Property Testing
   1.3 Locally Testable Codes
      1.3.1 Random locally testable codes
      1.3.2 Algebraic Construction of LTCs
      1.3.3 Combinatorial Construction of LTCs
      1.3.4 Constructions of LTCs based on PCPs of proximity
   1.4 Locally Decodable Codes

2 Summary of results

3 Global Definitions and Preliminaries
   3.1 General Notations
   3.2 Codes invariant under groups
   3.3 Locally Testable Codes
   3.4 Locally Decodable (Correctable) Codes

4 Locally Testable Codes Require Redundant Testers
   4.1 Introduction
   4.2 Main results
      4.2.1 Statement of main results
      4.2.2 Main Technical Theorem
4.2.3 A simpler proof of the main result from [12] . . . . . . . . . 26
4.2.4 Proofs of main results . . . . . . . . . . . . . . . . . . 28
4.3 Proof of Main Technical Theorem 16 . . . . . . . . . . . . . . 30
  4.3.1 Overview for the proof of Theorem 16 . . . . . . . . . . . . 30
  4.3.2 The \((C,V)\)-representation of words in \(F^n\) . . . . . . . . 31
  4.3.3 Main Lemma and Proof of Main Theorem 16 . . . . . . . . 32
4.4 Tightness of Main Theorem 7 . . . . . . . . . . . . . . . . . . 36
4.5 Bounds on tester support size . . . . . . . . . . . . . . . . . . 37
  4.5.1 Proof of Proposition 27 . . . . . . . . . . . . . . . . . . 38
  4.5.2 Proof of Proposition 30 . . . . . . . . . . . . . . . . . . 39
4.6 Open questions and Discussion . . . . . . . . . . . . . . . . . . 40

5 Locally Testable vs. Locally Decodable Codes 42
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
    5.1.1 Main Results . . . . . . . . . . . . . . . . . . . . . . . 43
    5.1.2 On Sparse codes vs. Non-sparse codes . . . . . . . . . . . 44
    5.1.3 Locally Decodable vs. Locally Correctable Codes . . . . 44
  5.2 Two Transitivity with a Local Constraint implies Local Cor-
  rection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
    5.2.1 Proof of Theorem 35 . . . . . . . . . . . . . . . . . . . 47
  5.3 Non-sparse LDCs contain subcodes that are not LTCs . . . . 49
    5.3.1 Non-sparse LDCs contain many subcodes which are
    not uniform-LTCs . . . . . . . . . . . . . . . . . . . . . . 52

6 Towards lower bounds on Locally Testable Codes via density
  arguments 54
  6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
  6.2 Main Results . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
    6.2.1 Dense natural and regular LTCs cannot be asymptot-
    ically good . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
    6.2.2 Limiting the rate of weak 2-query LDCs . . . . . . . . . . 64
  6.3 Proof of Main Results for LTCs . . . . . . . . . . . . . . . . . 66
  6.4 Proof of Main Technical Lemma 63 . . . . . . . . . . . . . . . 69
    6.4.1 Equivalence Relation and Matchings . . . . . . . . . . . . 70
    6.4.2 Selection of columns to be removed from the generat-
    ing matrix . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
Abstract

Locally testable codes (LTCs) are error-correcting codes for which membership, in the code, of a given word can be tested by examining it in very few locations. The main open problem in the area of locally testable codes (LTCs) is whether there exists an asymptotically good family of LTCs, i.e., LTCs with constant rate, constant relative distance and constant query complexity. In this thesis we suggest an approach to refute the existence of asymptotically good linear LTCs.

Without loss of generality we may deal with the case of query complexity 3 (cf. Theorem 99). Our proof-strategy goes by way of contradiction and relies on proving the following pair of conjectures.

- If \( C \subseteq \mathbb{F}_q^n \) is an asymptotically good 3-query LTC then \( C \) has super-linear number of dual codewords of weight at most 3.

- If \( C \) is an asymptotically good 3-query LTC and has super-linear number of dual codewords of weight at most 3 then \( \text{rate}(C) = o(1) \).

Most known constructions of locally testable codes are linear codes, and give error-correcting codes whose duals have superlinearly many small weight codewords. Examining this feature appears to be one of the promising approaches to proving limitation results for (i.e., upper bounds on the rate of) LTCs. Unfortunately till now it was not even known if LTCs need to be non-trivially redundant, i.e., need to have one linear dependency among the low-weight codewords in its dual.

In this thesis we give the first lower bound of this form, by showing that every positive rate constant query LTC must have linearly many redundant low-weight codewords in its dual. We actually prove the stronger claim that the actual test itself must use a linear number of redundant dual codewords.
(beyond the minimum number of basis elements required to characterize the code); in other words, non-redundant (in fact, low redundancy) local testing is impossible. This result is based on [11]. Our hope is that this might be useful to prove the conjecture stated in the first item above.

On the other hand, we progress on the conjecture, stated in the second item, and show that if $C$ has superlinear number of dual codewords of weight at most 3 which are naturally distributed then rate$(C) = o(1)$. This result is based on [20].

One of the technical contribution of this thesis is an improvement of the combinatorial lemma of Goldreich et al. [38] which bounds the rate of 2-query locally decodable codes (LDCs) and is used in state-of-the-art rate-bounds for linear LDCs. The lemma of Goldreich et al. [38] bounds the rate of 2-query LDCs of blocklength $n$ in terms of the corruption parameter $\delta(n)$ — this is the maximal number of corrupted codeword bits for which a (2-query) decoder can recover correctly every message bit (with high probability). Our combinatorial lemma gives nontrivial rate bounds for any corruption parameter $\delta(n) = \omega(1)$, whereas the previous lemma works only for corruption parameter larger than $\log n$. The study of LDCs with sublinear corruption parameter is also motivated by Dvir’s observation [30] that sufficiently strong bounds on the rate of such LDCs imply explicit constructions of rigid matrices.
Abbreviations and Notations

(q, ε, δ)-LTCs — Locally Testable Codes, where q is the query complexity, ε is the rejection probability and δ is the distance threshold.

(q, ε, δ)-LDCs — Locally Decodable Codes, where q is the query complexity, ε is the recovery probability and δ is the error parameter.

(q, ε, δ)-LCCs — Locally Correctable Codes, where q is the query complexity, ε is the recovery probability and δ is the error parameter.

PCPs — Probabilistically Checkable Proofs

C — linear code
C⊥ — dual code
dim(C) — dimension of the code C
rate(C) — rate of the code C
Δ(C) — distance of the code C
δ(C) — relative distance of the code C
n — blocklength of a code
k — dimension of a code
F — field
[n] — a set {1, . . . , n}
Chapter 1

General Introduction

The main focus in this thesis is on aspects related to the area of locally testable codes. In this chapter we provide a brief introduction to the topics that are related to the content of this thesis. A more detailed introduction and bibliographic background, based on the articles in which these results were published, are provided later in the specific chapters.

1.1 PCP theorem

The PCP theorem [2, 3] is one of the major achievements of complexity theory. A Probabilistically Checkable Proof (PCP) is a proof that allows checking the validity of a mathematical claim by reading only a constant number of symbols of the proof. Moreover, the check can be performed by an efficient verifier. If the claim, supposedly being proven, is correct, then there exists a proof in the new form that the verifier accepts. On the other hand, if the claim is false, then no matter which proof is provided, the verifier rejects with at least some constant probability.

The PCP theorem asserts the existence of PCPs of polynomial length for any claim that can be stated as membership in an NP set. The theorem has found many applications, most notably in establishing lower bounds for approximation algorithms [33, 45, 61] but also for positive results, e.g., CS-proofs and their applications [6, 23, 54, 60, 65] and the length of the PCP affects the complexity of those applications. The discovery of PCPs of polynomial length, being remarkable by its own right, raises natural questions
such as how long should a proof be to enjoy local testability, and more general, what makes a property locally testable. Having shorter locally testable proofs also affects the various applications of PCPs. This consideration motivates the direct study of local testability, and the amount of redundancy it requires. On the other hand, we would like to understand in general what makes a property testable and which kind of constraints a property should satisfy to enjoy local testability. In this way the area of PCPs gave birth to Property Testing [37, 64], and Locally Testable Codes [41].

1.2 Property Testing

Property testing deals with the following relaxation of decision problems: Given a property $P$ and an input structure $S$, distinguish with high probability between the case where $S$ satisfies the property $P$ and the case where $S$ is $\varepsilon$-far from satisfying $P$. Roughly speaking, a combinatorial structure is said to be $\varepsilon$-far from satisfying the property $P$ if an $\varepsilon$-fraction of its representation has to be modified in order to make $S$ satisfy $P$.

The power of property testing lies in the ability to design algorithms, or property testers, which read only a small fraction of the input structure, and use this information to distinguish with high probability between the above two cases. We assume that our algorithms access the input structure using retrieval procedures that we call queries. The answer to each query is a value in the input structure. For example, if our inputs are binary vectors of a fixed length $n$, then a query to an input vector would be an index $i$, where the answer is the $i$th element of the vector. Property testing was initiated in the work of Blum, Luby and Rubinfeld [22], and given a general formulation by Rubinfeld and Sudan [64]. The latter were interested mainly in algebraic properties (such as linearity) of functions over finite fields and vector spaces. The study of property testing for combinatorial objects, and mainly for labelled graphs, was introduced by the seminal paper of Goldreich, Goldwasser and Ron [37]. A property in this respect is a collection of functions from a fixed combinatorial object to a finite set of labels, often $\{0, 1\}$, but in our work we consider larger finite sets as well.
1.3 Locally Testable Codes

Error-correcting codes are sets of strings (of equal length), typically, having a large pairwise distance. Equivalently, error-correcting codes are viewed as mappings from short strings ($\mathbb{F}^k$), called messages, to longer strings ($\mathbb{F}^n$), called codewords, such that the codewords are distant from one another. In this thesis we consider only linear codes. A linear code over a finite field $\mathbb{F}$ is a linear subspace $C \subseteq \mathbb{F}^n$. The dimension of $C$ is its dimension as a vector space, and its rate is the ratio of its dimension to $n$. The distance of $C$ is the minimal Hamming distance between two different codewords. One is typically interested in codes whose distance is a growing function of the block length $n$, ideally $\Omega(n)$.

Locally testable codes (LTCs) are error correcting codes for which distinguishing, when given oracle access to a purported word $w$, between the case that $w$ is a codeword and the case that it is very far from all codewords, can be accomplished by a randomized algorithm, called a tester. The tester reads a sublinear amount of information from $w$. Such codes are of interest in computer science due to their numerous connections to probabilistically checkable proofs (PCPs) and property testing (see the surveys [67, 36] for more information). In fact, the area of locally testable codes is a special case of property testing, where the underlying property is a given error-correcting code. LTCs were implicit already in [4] (cf. [36, Sec. 2.4]) and they were explicitly studied by Goldreich and Sudan [41].

By now several different constructions of LTCs are known including sparse random linear codes [19, 49, 58], codes based on low-degree polynomials over finite fields [22, 2] and affine invariant codes [50], constructions based on PCPs of proximity/assignment testers [9, 28] and combinatorial construction of LTCs (based on tensor products) [14, 59, 17].

In the following sections we provide a short introduction for each of the above groups for LTCs constructions.

1.3.1 Random locally testable codes

One of the most natural questions in the area of LTCs is whether one can obtain such codes at random. Kaufman and Sudan [49] answered positively this question by showing that randomly chosen code $C \subseteq \mathbb{F}_2^n$ such that $\text{dim}(C) \leq O(\log n)$ is locally testable with very high probability. Then Kop-
party and Saraf [58] proved that random codes with logarithmic dimension are locally testable in the high-error regime. In [57] it was conjectures that all linear codes with logarithmic dimension are locally testable. This conjecture was refuted by Ben-Sasson and Viderman [19] who pointed out that some codes with logarithmic dimension are not locally testable.

These results provide an understanding of the local testability feature among logarithmic dimension codes, i.e., most of the codes of logarithmic dimension are locally testable but not all. Kaufman and Sudan [49] also pointed out that if one picks randomly a code with dimension $\omega(\log n)$ then with probability close to 1 the code is not locally testable, i.e., most of the codes with higher dimension are not locally testable. Hence, LTCs with higher dimension should be constructed carefully to combine linear distance and the local testability feature.

1.3.2 Algebraic Construction of LTCs

This research line deals with a construction of LTCs based on low-degree polynomials. The general idea behind such constructions is viewing the messages as polynomials and the codewords are obtained via evaluation of these polynomials through all points of the underlying field.

Blum et al. [22] showed that the famous Hadamard code is locally testable. The big disadvantage of the Hadamard code is that its block-length is exponential in terms of the dimension. Alon et al. [1] showed that Reed-Muller codes of order $q$ are testable with $2^q$ queries. This result was later improved in [21]. However, Reed-Muller codes of order $q$ and block-length $n$ have dimension $\leq \log^3 n$, i.e., very low dimension. Kaufman and Sudan [50] generalized the results of testing of Reed-Muller codes to affine invariant locally testable codes. This research line followed by other works that studied affine invariant locally testable codes [42, 43, 39, 48, 13]. However, Ben-Sasson and Sudan [16] proved that affine invariant locally testable codes are subcodes of Reed-Muller codes of constant order. This fact implies that these codes have very bad rate.

1.3.3 Combinatorial Construction of LTCs

In this section we give an introduction on a different family of LTC constructions, namely, tensor codes. Given two linear error correcting codes
$C \subseteq \mathbb{F}^n$, $R \subseteq \mathbb{F}^m$ over a finite field $\mathbb{F}$, we define their tensor product to be the subspace $R \otimes C \subseteq \mathbb{F}^{n \times m}$ consisting of $n \times m$ matrices $M$ with entries in $\mathbb{F}$ having the property that every row of $M$ is a codeword of $R$ and every column is a codeword of $C$. If $C = R$ we use $C^2$ to denote $C \otimes C$ and for $i > 2$ define $C^i = C \otimes C^{i-1}$.

Ben-Sasson and Sudan suggested in [14] to use tensor product codes as a means to construct LTCs combinatorially. They showed that taking the three-wise tensor $C^3$ of any code $C \subseteq \mathbb{F}^n$ with sufficiently large distance results in a robust locally testable code. By robust we informally mean that the tester associated with $C^3$ has the property that given any word $w$ that is far from $C^3$, the local view selected by the tester will be far, on average, from being consistent with a local view of a codeword of $C^3$. More formally, denoting by $w|_I$ the projection of $w$ onto the set of queries $I \subset \{1, \ldots, n\}^3$ picked by the tester, and denoting by $C^3|_I = \{c|_I \mid c \in C^3\}$ the set of views that are consistent with $C^3$, the robustness of $C^3$ means that, on average, $w|_I$ will be far in Hamming distance from all elements of $C^3|_I$. This robustness allowed them to apply composition and prove that the repeated three-wise tensor product of $C$, namely, the code $C^{3^i}$, is locally testable. The ability to take the repeated tensor product is crucial for tensor-based constructions of LTCs. The repeated $m$-wise tensor product (for $m \geq 3$) was used in [14, 59] to construct new families of LTCs. In particular, Meir [59] showed a combinatorial construction of LTCs based on results of [14] that achieve the best known parameters. Ben-Sasson and Sudan also raised the question of whether the repeated two-wise tensor product of $C$ also leads to robust LTCs.

There is a surprising difference between two- and three-wise tensor products. For two-wise products, large distance is not sufficient to guarantee robustness (whereas for three-wise products it is). This phenomena was discovered by Paul Valiant who constructed in [68] a pair of codes $R, C$ with large distance whose tensor product is not robust. (See [25, 40] for generalizations of this result.) Nevertheless, in another surprising turn of events, Dinur et al. [29] and then Ben-Sasson and Viderman [18, 17] showed that for some base codes (including locally testable codes and expander codes) the associated tensor product is robust and the appropriate construction of LTCs can be achieved. It is interesting to point out that some non-locally testable codes (e.g., expander codes [12]) can be used in the tensor product
operations to construct locally testable codes.

In particular, the results of [17] provide a construction of LTCs with query complexity $n^\varepsilon$ and rate $1 - \varepsilon$ for any $\varepsilon > 0$ based on repeated two-wise tensors, where $n$ is a blocklength of the constructed code. Best to our knowledge this range of parameters was not shown before and was achieved in [17] at the first time.

1.3.4 Constructions of LTCs based on PCPs of proximity

Recall that a standard PCP is given an explicit input, which is supposedly in some $\text{NP}$ language, as well as access to an oracle that is supposed to encode a “probabilistically verifiable” $\text{NP}$ witness. The PCP verifier uses oracle queries in order to probabilistically verify whether the input, which is explicitly given to it, is in the language. In contrast, a PCP of proximity is given access to two oracles, one representing an input and the other being a redundant encoding of an $\text{NP}$-witness (as in a PCP). Indeed, the verifier may query both the input oracle and the proof oracle, but its queries to the input oracle are also counted in its query complexity. A verifier for a PCP of proximity is only required to accept inputs that are in the language and reject inputs that are far from the language.

The notion of a PCP of proximity generalizes the notion of holographic proofs set forward by Babai et al. [4] and were implicit in the low-degree testers that utilize auxiliary oracles [2, 3]. On the other hand, PCPs of proximity extend “property testing” [37, 64] by providing the testers with oracle access to a proof. The notion was explicitly defined by Ben-Sasson et al. [9] and independently by Dinur and Reingold [28].

It turns out that combined with any good code, any PCP of proximity yields a locally testable code [9]. This can be done by appending each codeword with a PCP of proximity proving the codeword is indeed an encoding of a message. Then every codeword should be repeated many times, so that a PCP of proximity constitutes only a small fraction of the total length. Hence the distance of the new code is roughly preserved. Moreover, the blocklength of the new code is, roughly speaking, close to the proof length of the PCP of proximity attached to it. In this way, shorter proofs in the constructions of PCPs of proximity yields shorter locally testable codes and, equivalently, lower bounds on the blocklength of locally testable codes gives
lower bounds on the proof length in PCPs of proximity.

1.4 Locally Decodable Codes

Locally decodable codes (LDCs) are in some sense complimentary to local testable codes. Informally, a code is locally decodable if there exists a decoder that given a slightly corrupted codeword (i.e., a string close to some unique codeword), and is required to recover individual bits of the encoded information based on a constant number of probes (per recovered bit). That is, whenever relatively few location are corrupted, the decoder is able to recover each information-bit, with high probability, based on a constant number of probes to the (corrupted) codeword. The problem is related to the construction of (information theoretic secure) Private Information Retrieval schemes, introduced in [24]. The best known construction of LDCs was initiated by the breakthrough results of Yekhanin [71] who showed a (conditional) subexponential construction of 3-query LDCs. Later Efremenko [32] showed unconditional subexponential construction of LDCs. Specifically, $k$ information bits can be encoded by codewords of length $n = \exp(k^{o(1)})$ that are locally decodable using three bit-probes.

Katz and Trevisan [47] were first who defined formally LDCs and showed that LDCs have superlinear blocklength. Goldreich et al. [38] showed that linear 2-query LDCs have exponential blocklength. This result was generalized by Dvir and Shpilka [31] for all arbitrarily large fields. Obata [62] showed asymptotically tight (exponential) lower bounds on the blocklength of 2-query LDCs. Kerenidis and de Wolf [53] showed exponential lower bounds for 2-query LDC and improved superlinear lower bound for $q$-query LDCs, where $q \geq 3$. Then Woodruff [69, 70] improved this result for odd $q$ and showed that $q$-query LDCs ($q \geq 3$) with $k$ message bits and blocklength $n$ have $n \geq \Omega(k^{1+\frac{1}{q/2-1}})/\log(k)$ and for 3-query linear LDCs showed that $n \geq \Omega(k^2)$. The known lower bounds for $q$-query LDCs for $q \geq 3$ seems to be very far from tight.
Chapter 2

Summary of results

This thesis is dedicated to the study of limitations of linear locally testable codes. The main open problem in the area of locally testable codes is whether there exist asymptotically good LTCs, i.e., locally testable codes that have constant query complexity, constant rate and constant relative distance. For linear codes, one can assume without loss of generality [12] that the tester picks a low-weight dual codeword $c^\perp$ from some distribution, and checks that the input $x$ is orthogonal to $c^\perp$. All known constructions of LTCs in fact have duals which have super-linearly many low-weight dual codewords. In other words, there must be a substantial number of linear dependencies amongst the low-weight dual codewords. Examining whether this feature is necessary might be one of the promising approaches to proving limitations (i.e., upper bounds on the rate) of LTCs, as it imposes strong constraints on the dual code.

In Chapter 4 (based on [11]) we give the first lower bound of this form, by showing that every positive rate constant query LTC must have $\Omega(n)$ redundant low-weight dual codewords. We point out that our main theorem (Theorem 7) is actually just a special case of a more general statement given in Theorem 16. For instance, the more general theorem can be used to provide a different and arguably simpler, proof of the main result of [12] stating that testing of random low density parity check (LDPC) codes require linear query complexity (see Section 4.2.3). But Theorem 16 goes even further and we believe it may be instrumental in proving limitations on the rate of other families of LTCs in the future. We informally describe this result. Let $B$ be any basis for $C^\perp$ comprised of words of small support. Such
a basis must exist if \( \mathcal{C} \) is to be locally testable. Theorem 16 says that any tester for \( \mathcal{C} \) must use (many) dual words that are each a linear combination of a constant fraction of \( B \). In plain words, \( \mathcal{C}^\perp \) must have a high level of redundancy and cancellation to allow for large sums of small-support words in \( B \) to result in words that are also of small support. We hope that this result will help to prove that all \( q \)-query LTCs have superlinear number of duals of weight at most \( q \).

In Chapter 5 (based on [51]) we study a relation between LTCs and LDCs. Both these families of error correcting codes are explicitly studied, for survey see e.g. [67]. In spite of the fact, the distinction between the two families of the codes was not made. Namely, it is well-known that there is an intersection between the two families of codes, e.g. the famous Hadamard code is 3-query LTC and 2-query LDC. Moreover, it is well-known that LTCs do not imply LDCs, i.e., there are LTCs which are not LDCs. This follows simply by comparing the upper and lower bounds on the blocklength of these families of codes. If \( \mathcal{C} \subseteq \mathbb{F}_q^n \) is a \( q \)-query LDCs then \( n \geq \Omega(\dim(\mathcal{C})^{q/(q-1)}) \) (by Katz and Trevisan [47]), while there exist (best known) LTCs such that \( n \leq O(\dim(\mathcal{C}) \cdot (\text{poly log}(\dim(\mathcal{C})))) \) [15, 26, 59]. However, the other direction, i.e., whether LDCs imply LTCs, was not known.

We show that LDC does not imply LTC, and in fact there are inherent differences between LDCs and LTCs. In particular, we show that some LDCs are not LTCs.

Nevertheless, in Chapter 6 (based on [20]) we use some relation between LTCs and LDCs in the non-standard range of parameters. We prove tight lower bounds for 2-query locally decodable codes with small corruption parameter and use this result to obtain some lower bounds for LTCs. Looking into known “base-constructions” of \( q \)-query LTCs they all share a few properties that we formalize in this thesis. First, they are \( q \)-regular, i.e., every codeword-bit sees the same number of dual codewords of weight \( q' \leq q \) (see Definition 53). Second, they are all \( q \)-dense, by which we mean that the number of dual codewords of weight at most \( q \) is super-linear in the code blocklength. Indeed, a popular belief (stated formally in Conjecture 51) says that all \( q \)-query LTCs are \( q \)-dense (see Definition 48).

Our main result is that families of 3-dense and 3-regular LTCs cannot be asymptotically good. We bound the rate of the code as a function of 3-density and show that even arbitrarily slowly growing 3-density implies...
vanishing rate (cf. Theorem 54 and Corollary 55). We then put forth a conjecture stating that all 3-query LTCs are dense and have a “natural” distribution of dual codewords of weight $\leq 3$ (Conjecture 58) and show that under this conjecture there are no asymptotically good families of LTCs whatsoever (cf. Theorem 56 and Corollary 59).
Chapter 3

Global Definitions and Preliminaries

3.1 General Notations

We start with a few definitions. Let $F$ be a finite field and $[n]$ be the set $\{1, \ldots , n\}$. In this thesis, we consider only linear codes. Let $C \subseteq F^n$ be a linear code over $F$. The dimension of $C$ is denoted by $\text{dim}(C)$ and its rate is denoted by $\text{rate}(C)$ and defined to be $\text{rate}(C) = \text{dim}(C)/n$. For $w \in F^n$, let $\text{supp}(w) = \{i \in [n] \mid w_i \neq 0\}$ and $|w| = |\text{supp}(w)|$. We define the distance between two words $x, y \in F^n$ to be $\Delta(x, y) = |\{i \mid x_i \neq y_i\}|$ and the relative distance to be $\delta(x, y) = \frac{\Delta(x, y)}{n}$. The distance of the code $C$ is denoted by $\Delta(C)$ and defined to be $\Delta(C) = \min_{x \neq y \in C} \Delta(x, y)$. The relative distance of a code is denoted by $\delta(C)$ and defined to be $\delta(C) = \frac{\Delta(C)}{n}$.

We use the standard notation for describing linear error correcting codes. A $[n, k, d]_F$-code is a $k$-dimensional subspace $C \subseteq F^n$ of distance $d$, defined next.

For $x \in F^n$ and $C \subseteq F^n$, let $\delta(x, C) = \min_{y \in C} \{\delta(x, y)\}$ denote the relative distance of $x$ from the code $C$. If $\delta(x, C) \geq \varepsilon$, we say that $x$ is $\varepsilon$-far from $C$ and otherwise $x$ is $\varepsilon$-close to $C$. We note that $\Delta(C) = \min_{\varepsilon \in C \setminus \{0\}} \{|\varepsilon|\}$. The vector inner product between $u = (u_1, u_2, \ldots , u_n) \in F^n$ and $v = (v_1, v_2, \ldots , v_n) \in F^n$ is denoted by $\langle u, v \rangle$ and defined to be $\langle u, v \rangle = \sum_{i=1}^n u_iv_i$. The dual code
$C^\perp$ is defined as $C^\perp = \{u \in \mathbb{F}^n \mid \forall c \in C : \langle u, c \rangle = 0\}$. In a similar way we define $C^\perp_{\leq t} = \{u \in C^\perp \mid |u| \leq t\}$ and $C^\perp_t = \{u \in C^\perp \mid |u| = t\}$. For $w \in \mathbb{F}^n$ and $S = \{j_1, j_2, \ldots, j_m\} \subseteq [n]$, where $j_1 < j_2 < \ldots < j_m$, let $w|_S = (w_{j_1}, w_{j_2}, \ldots, w_{j_m})$ be the restriction of $w$ to the subset $S$. For $V \subseteq \mathbb{F}^n$ let $V|_S = \{v|_S \mid v \in V\}$ denote the restriction of the subspace $V$ to the subset $S$. For $T \subseteq \mathbb{F}^n$ and $w \in \mathbb{F}^n$ we say that $w \perp T$ if for all $t \in T$ we have $\langle w, t \rangle = 0$. For $S \subseteq \mathbb{F}^n$, where $S$ is not a vector space, with some abuse of notation we let $\dim(S) = \dim(\text{span}(S))$. For any integer $n \geq 2$ let $\binom{n}{2} = \{\{i, j\} \mid 1 \leq i < j \leq n\}$.

### 3.2 Codes invariant under groups

Let $G$ be a group of permutations over $[n]$.

For $\pi \in G$ and $w = (w_1, w_2, \ldots, w_n) \in \mathbb{F}^n$ with some abuse of notation we let $\pi(w) = (w_{\pi^{-1}(1)}, \ldots, w_{\pi^{-1}(n)})$ be a $\pi$-permuted word. Note that since $G$ is a group and $\pi \in G$ we have $\pi^{-1} \in G$. A linear code $C$ is invariant under $G$ if for every $\pi \in G$ and $c \in C$ we have $\pi(c) \in C$. Note that if $C$ is invariant under $G$ then also $C^\perp$ is invariant under $G$. $G$ is called 2-transitive if for all $i \neq j \in [n]$ and $i' \neq j' \in [n]$ we have $\pi \in G$ such that $\pi(i) = i'$ and $\pi(j) = j'$. A linear code $C$ is 2-transitive if it is invariant under some 2-transitive permutation group $G$.

### 3.3 Locally Testable Codes

In this section, we define LTCs following [41, 11].

**Definition 1 (Tester)** Suppose $C$ is a $[n, k, d]_q$-code. A $q$-query test for $C$ is an element $u \in C^\perp_{\leq q}$ and a $q$-query tester $T$ for $C$ is defined by a distribution $p$ over $q$-query tests. When $C$ is clear from context we omit reference to it. The support of $T$, denoted $S = S_T$, is the support of $p$, i.e., the set $S = S_T = \{u \in C^\perp_{\leq q} \mid p(u) > 0\}$. When $p$ is uniform over a subset of $C^\perp_{\leq q}$ we say the tester is uniform and may identify the tester with $S$.

Invoking the tester $T$ on a word $w \in \mathbb{F}^n$ is done by sampling a test $u \in S_T$ according to the distribution $p$ and outputting accept if $\langle u, w \rangle = 0$, in which case we say that $u$ (and $T$) accept $w$, denoted $T[w] =$ accept, and outputting
reject, denoted $T[w] = \text{reject}$, if $\langle u,w \rangle \neq 0$. Clearly any such tester always accepts $w \in \mathcal{C}$.

- A $(q, \varepsilon')$-strong tester is a $q$-query tester $T$ satisfying for all $w \in \mathbb{F}^n$
  $$\Pr[T[w] = \text{reject}] \geq \varepsilon' \cdot \delta(w, \mathcal{C}).$$

- A $(q, \varepsilon, \rho)$-tester is a $q$-query tester $T$ satisfying for all $w \in \mathbb{F}^n$ that is $ho$-far from $\mathcal{C}$
  $$\Pr[T[w] = \text{reject}] \geq \varepsilon.$$

The probability in both equations above is according to the distribution $p$ associated with $T$.

**Definition 2 (Locally Testable Code (LTC) [41])** A $[n, k, d]_F$-code $\mathcal{C}$ is said to be a $(q, \varepsilon')$-strong locally testable code if it has a $(q, \varepsilon')$-strong tester, and $\mathcal{C}$ is a $(q, \varepsilon, \rho)$-locally testable code if it has $(q, \varepsilon, \rho)$-tester $T$. The parameter $q$ is known as the query complexity of $T$, $\varepsilon$ is its rejection probability and $\rho$ is its distance threshold.

Note that a $(q, \varepsilon')$-strong LTC is also a $(q, \rho \cdot \varepsilon', \rho)$ LTC for every $\rho > 0$. Moreover, if $T$ is a $(q, \varepsilon' > 0)$-strong tester for a $[n, k, d]_F$-code then, letting $S_T$ denote the support of $T$, we have $\dim(S_T) = \dim(C^\perp) = n - k$.

We say that a family of codes $\{C^{(n)} \mid n \in \mathbb{Z}\}$ is locally testable if there exist constants $q, \varepsilon, \delta > 0$ such that for infinitely many $n$ it holds that $C^{(n)} \subseteq \mathbb{F}^n$ is a $(q, \varepsilon, \delta)$-LTC, where $\delta \leq \delta(C^{(n)})/3$.

Note that if $C$ is a perfect code and $\delta > \delta(C)/2$ then $C$ is a $(0, 1, \delta)$-LTC, i.e., $C$ is locally testable with $0$ queries if the distance threshold is more than $\delta(C)/2$. This is true since there are no words which are $\delta$-far from the code. Hence, to avoid trivial cases we must require the distance threshold $\delta$ to be at most $\delta(C)/2$. In the area of locally testable codes we usually require $\delta \leq \delta(C)/3$. E.g., all known constructions of LTCs satisfy this requirement (see e.g., [41, 49, 50, 59, 26]). On the other side, if for all constants $q, \varepsilon > 0$ the code $C$ is not $(q, \varepsilon, \delta(C)/3)$-LTC we say that $C$ is not locally testable (see e.g., [5, 11, 42]).
Remarks on definitions of testers. Our definition of a tester, and an LTC is somewhat different from previous definitions (notably [12] and [41]). We clarify the differences here.

We start with Definition 2. The definition of strong LTCs we use is the same as that in [41]. The weak notion is weaker than their definition of a weak tester (which simply allowed the rejection probability of a weak tester to be smaller by a $o(1)$ additive amount compared to the strong case). Our definition on the other hand only requires rejection probability to be positive when the word is very far (constant relative distance) from the code. Since our goal is to prove “impossibility” results, doing so with weaker definitions makes our result even stronger.

We now discuss Definition 1. For linear LTCs it was shown in [12] (see also references therein) that the tester might as well pick a collection of low-weight dual codewords and verify that the given word $w$ is orthogonal to all of them. On the other hand, our definition (Definition 1) requires the tester to pick only one dual codeword and test orthogonality to it. Our definition is more convenient to use when defining and analyzing the redundancy of tests (defined below). We first note that our restricted forms of tests may only alter the soundness of the test by a constant factor. For this we recall the assertion from [12] that showed that without loss of generality a $q$-query “standard” tester for a $[n, k, d]$-code is defined by a distribution over subsets $I \subseteq [n], |I| \leq q$. The test associated with $I$ accepts a word $w$ if and only if $\langle w, u \rangle = 0$ for all $u \in C^\perp$ such that $\text{supp}(u) \subseteq I$. (The soundness and the distance threshold of a “standard” tester are defined as in Definition 1.) To convert this “standard” tester to one that only tests one dual codeword, consider a tester that, given $I$, samples uniformly from the set $U_I = \{ u \in C^\perp \mid \text{supp}(u) \subseteq I \}$ and accepts iff $\langle u, w \rangle = 0$. This resulting tester conforms to our Definition 1. Furthermore, if the soundness of the “standard” tester is $\rho$ then the soundness of the tester that samples uniformly from $U_I$ is at least $\frac{|I|-1}{|I|} \cdot \rho \geq \frac{1}{2} \rho$. To see this, notice that $U_I$ forms a linear space over $\mathbb{F}$. And the set \{ $u \in U_I \mid \langle u, w \rangle = 0$ \} is a linear subspace of $U_I$. Thus, whenever $w$ is rejected by some $u \in U_I$ we actually know that $w$ is rejected by at least a fraction $\frac{|I|-1}{|I|}$ of $U_I$ because the set of rejecting words is the complement of a subspace of $U_I$. Hence, using our definition of a tester is equivalent to the most general definition of a tester, up to a

\footnote{The idea of choosing a random element of the set $U_I$ is folklore.}
constant loss in the soundness parameter.

**Definition 3 (Linearly independent tester and tester redundancy)**
Suppose $\mathcal{C}$ is a $[n, k, d]_F$-code. A $q$-query tester $T$ for $\mathcal{C}$ is said to be a linearly independent tester if its support $S_T \subseteq \mathcal{C}_\perp_q$ is a set of linearly independent vectors. If $T$ is a linearly independent tester and its support $S_T$ is of size $|S_T| = \dim(\mathcal{C}_\perp) = n - k$ then we call it a basis tester because $S_T$ forms a basis for $\mathcal{C}_\perp$. In case $S_T$ has size larger than $\dim(\text{span}(S_T))$ we define the redundancy of $T$ to be $|S_T| - \dim(\text{span}(S_T))$. (Notice that a linearly independent tester has redundancy 0.)

**Definition 4 (Expected query complexity)** The expected query complexity of a tester for $\mathcal{C}$ with distribution $p$ over its support $S \subseteq \mathcal{C}_\perp$ is defined to be $\mathbb{E}_{u \sim p}[|u|]$.

### 3.4 Locally Decodable (Correctable) Codes

Now we define Locally Decodable Codes (LDCs) following [24].

**Definition 5 (LDCs)** Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code of dimension $k$. Let $E_{\mathcal{C}}$ be the encoding function, i.e., $\mathcal{C} = \{ E_{\mathcal{C}}(x) \mid x \in \mathbb{F}^k \}$. Then $\mathcal{C}$ is a $(q, \varepsilon, \delta)$-LDC if there exists a randomized decoder ($D$) that reads at most $q$ entries and the following condition holds:

- For all $x \in \mathbb{F}^k$, $i \in [k]$ and $\hat{c} \in \mathbb{F}^n$ such that $\Delta(E_{\mathcal{C}}(x), \hat{c}) \leq \delta n$ we have $\Pr[D(x)_i = x_i] \geq \frac{1}{|\mathbb{F}|} + \varepsilon$, i.e., with probability at least $\frac{1}{|\mathbb{F}|} + \varepsilon$ entry $x_i$ will be recovered correctly.

We say that a family of codes $\{C_n\}_{n \in \mathbb{Z}}$ over the field $\mathbb{F}$ is a $(q, \varepsilon, \delta)$-locally decodable if for infinitely many $n$ it holds that $C_n \subseteq \mathbb{F}^n$ is a $(q, \varepsilon, \delta)$-LDC.

Now we define locally self-correctable codes (LCCs).

**Definition 6 (LCCs)** Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code of dimension $k$. Then $\mathcal{C}$ is a $(q, \varepsilon, \delta)$-LCC if there exists a self-corrector ($\mathcal{S}_C$) that reads at most $q$ entries and the following condition holds:

- For all $c \in \mathcal{C}$, $i \in [n]$ and $\hat{c} \in \mathbb{F}^n$ such that $\Delta(c, \hat{c}) \leq \delta n$ we have $\Pr[\mathcal{S}_C^c[i] = c_i] \geq \frac{1}{|\mathbb{F}|} + \varepsilon$, i.e., with probability at least $\frac{1}{|\mathbb{F}|} + \varepsilon$ entry $c_i$ will be recovered correctly.
We say that a code $\mathcal{C}$ is locally self-correctable when $q, \varepsilon, \delta > 0$ are constants. Note that the definition implies that $\delta < \delta(\mathcal{C})/2$. 
Chapter 4

Locally Testable Codes Require Redundant Testers
4.1 Introduction

In this chapter of the thesis, we exhibit some limitations of locally testable linear codes.

One of the outstanding open questions in the subject is whether there are asymptotically good LTCs, i.e., LTCs that have dimension $\Omega(n)$ and distance $\Omega(n)$. Our understanding of the limitations of LTCs is, however, quite poor (in fact, practically non-existent), and approaches that may rule out the existence of asymptotically good LTCs have been elusive. Essentially the only negative results on LTCs concern binary codes testable with just 2-queries [10, 44] (which is a severe restriction), random LDPC codes [12], and cyclic codes [5]. In fact, we cannot even rule out the existence of binary LTCs meeting the Gilbert-Varshamov bound (which is the best known rate for codes without any local testing restriction). So, for all we know, the strong testability requirement of LTCs may not “cost” anything extra over LDPC codes!

This chapter is a (modest) initial attempt at addressing our lack of knowledge concerning lower bound results for LTCs. For linear codes, one can assume without loss of generality [12] that the tester picks a low-weight dual codeword $c^\perp$ from some distribution, and checks that the input $x$ is orthogonal to $c^\perp$. It is thus necessary that if $C$ is a $q$-query LTC of dimension $k$, then its dual $C^\perp$ has a basis of $n - k$ codewords each of weight at most $q$. All known constructions of LTCs in fact have super-linearly many low-weight dual codewords. In other words, there must be a substantial number of linear dependencies amongst the low-weight dual codewords. Examining whether this feature is necessary might be one of the promising approaches to proving limitations (i.e., upper bounds on the rate) of LTCs, as it imposes strong constraints on the dual code. Nevertheless, till now it was not even

---

1 The last result rules out asymptotically good cyclic LTCs; the existence of asymptotically good cyclic codes has been a longstanding open problem, and the result shows the “intersection” of these questions concerning LTCs and cyclic codes has a negative answer.

2 To be precise, only when $C$ is a strong LTC, as per Definition 2, need $C^\perp$ be spanned by words of weight $q$. Non-strong LTCs have the property that the set of low-weight words in the dual code must span a large dimensional subspace of $C^\perp$ (see Proposition 30 for an exact statement).

3 We remark that information on the dual weight distribution is useful, for example, in the linear programming bounds on the rate vs. distance trade-off of a linear code. For LDPC codes whose dual has a low weight basis, stronger upper bounds on distance are known compared to general linear codes of the same rate [8].

---
known if the dual of a LTC has to be non-trivially redundant, i.e., if it must have at least one linear dependency among its low-weight codewords.

In this chapter, we give the first lower bound of this form, by showing that every positive rate constant query LTC must have $\Omega(n)$ redundant low-weight codewords. The result is actually stronger — it shows that the actual test itself must use $\Omega(n)$ extra redundant dual codewords (beyond the minimum $n - k$ basis elements). In other words, non-redundant testing is impossible.

While this might sound like an intuitively obvious statement, we remark that even for Hadamard codes (whose dual has $\Theta(n^2)$ weight 3 codewords), a nonredundant test consisting of a basis of weight 3 dual codewords was not ruled out prior to our work. Also, without the restriction on number of queries, every code does admit a basis tester (which makes at most $k + 1$ queries).

We also note that a known upper bound [7, Proposition 11.2] shows that $O(n)$ redundancy suffices for testing. Bellare, Goldreich, and Sudan [7] prove this in the context of PCPs, but the technique extends to LTCs as well. For completeness, in Section 4.5 we include a proof showing that for every $q$-query LTC, there is an $O(q)$-query tester that picks a test uniformly from at most $3(n - k) = O(n)$ dual codewords. The quantity $n - k$ (as opposed to $n$) is significant in that this is the dimension of the dual code, and our lower bound shows that every tester (for any code) must have a support of size at least $n - k$.

We point out that our main theorem (Theorem 7) is actually just a special case of a more general statement given in Theorem 16. For instance, the more general theorem can be used to provide a different, and arguably simpler, proof of the main result of [12] that says that testing of random LDPC codes requires linear query complexity (see section 4.2.3). But Theorem 16 goes even further, and we believe it may be instrumental in proving limitations on the rate of other families of LTCs in the future. We end this section by informally describing this result. Let $B$ be any basis for $C^\perp$ composed of words of small support. Such a basis must exist if $C$ is to be locally testable. Theorem 16 says that any tester for $C$ must use (many) dual words that are each a linear combination of a constant fraction of $B$. In other words, $C^\perp$ must have a high level of redundancy and cancellation to allow for large sums of small-support words in $B$ to result in words that

22
are also of small support.

4.2 Main results

This section contains four parts. We start by stating our main results — Theorem 7 and Corollaries 10 and 11. Then, we discuss the main technical contribution of this paper — Theorem 16 — which implies all of our main results. We go on to show another application of Theorem 16, namely, a generalization and simplification of the main result from [12] stating that testing random low-density-parity-check (LDPC) codes require linear query complexity. Finally, we provide the proofs of our main results assuming Theorem 16. The proof of Theorem 16 appears in the next section.

4.2.1 Statement of main results

**Theorem 7 (Linearly independent tester)** If a \([n,k,d]\)_F-code \(C\) has a \((q,\rho,\frac{\delta(C)}{3})\)-linearly independent tester then

\[ \rho \leq \frac{q}{k}. \]

**Remark 8** Theorem 7 (Linearly independent tester) holds even for a basis tester that has only expected query complexity \(\leq q\) (and all other parameters are as in the statement of the theorem). Recall that a tester has expected query complexity at most \(q\) if \(\mathbb{E}_{u \sim D}[|u|] \leq q\) where the expectation is taken with respect to the probability \(D\) associated with the tester.

**Remark 9** The choice of \(\frac{\delta(C)}{3}\) as the distance threshold is crucial in Theorem 16, Theorem 7 and their corollaries.

The first corollary of our main theorem says that \(\Omega(n)\) redundancy is necessary for uniform testing of all codes that have nontrivial (i.e., super-constant) size.

**Corollary 10 (Uniform testers for LTCs require lin. redundancy)**

Let \(C\) be a \([n,k,d]\) code that is \((q,\rho,\frac{1}{3}\delta(C))\)-locally testable by a uniform tester using a set \(S \subseteq C_{\leq q}^\perp\). Then

\[ |S| \geq \left( \frac{1 - q/k}{1 - \rho} \right) \cdot \dim(\text{span}(S)) = \left( \frac{1 - q/k}{1 - \rho} \right) \cdot \Omega(n). \]
In words, $S$ has redundancy at least $\frac{\rho - q/k}{1 - \rho} \cdot \dim(\text{span}(S))$.

For instance, if $k = \dim(C) = \omega(1)$ and $\rho, q$ are constants then the previous corollary says that a uniform tester for $C$ requires a linear amount ($\Omega(n)$) of redundancy. The fact that $\dim(\text{span}(S)) = \Omega(n)$ will be proven below (Proposition 30).

Our second corollary shows that non-trivial redundancy is necessary for general (i.e., for nonuniform) testing.

**Corollary 11 (Testers for LTCs with const. rate require lin. red.)**

Let $C$ be a $[n, k, d]$ code that is $(q, \rho, \frac{1}{3}d(C))$-locally testable by a tester that is distributed over a set $S \subseteq C^\perp_{\leq q}$. Then

$$|S| \geq \dim(\text{span}(S)) + \frac{\rho k}{q} - 1.$$ 

In words, $S$ has redundancy at least $\frac{\rho k}{q} - 1$.

For instance, if $k = \Theta(n)$ and $\rho, q$ are constants (i.e., when $C$ comes from an asymptotically good family of error correcting codes) then, once again, a linear amount of redundancy is required by any constant-query tester for $C$. For the state of the art LTCs [15, 26, 59] $k = \Theta(n/poly\log n)$ and our result implies that $\Theta(n/poly\log n)$ redundancy is necessary in such cases.

**Remark 12** Looking at Corollary 11 one might conjecture that a stronger bound on the redundancy of a tester should hold, one that depends on the blocklength $n$. However, as shown in [19] this conjecture is false. In particular, for every $k$ ranging from $\omega(1)$ to $O(n/poly(\log n))$ [19, Section 4.3] gives a construction of LTCs of dimension $k$ along with constant-query testers with redundancy $k \cdot poly(\log k)$. This result also shows that the lower bound on redundancy is nearly tight, up to a multiplicative $poly(\log k)$ factor. Stated differently, the said result of [19] shows an inherent difference between the redundancy of uniform and non-uniform testers. Namely, the redundancy of a non-uniform tester is at least proportional to the dimension of the code whereas the redundancy of a uniform tester for LTCs with non-constant dimension is at least proportional to the blocklength of the code.

The main observation of [19] is that taking repetitions of a code increases its block length without affecting the testability of the code and the
redundancy of the tester. More specifically, take any LTC $C$ of block length $n = k \cdot \text{poly}(\log k)$ and consider the code $C'$ that contains all the strings that consist of $m$ consecutive copies of some codeword of $C$. [19, Section 4.3] proves that $C'$ has block length $m \cdot n$, and that its “natural” tester has the same redundancy as the tester of $C$. By taking $m$ to be much larger than $n$, we obtain the required.

Later on in the paper we show that our main theorem is almost tight in two respects. In Section 4.4 we show that there do exist codes of constant size that can be strongly tested by a uniform basis tester of $O(1)$ query complexity, and that every code can be strongly tested by a uniform basis tester that has large query complexity. We conclude by showing in Section 4.5 that if $C \subseteq F^n_q$ is a $(q, \rho, \varepsilon)$-LTC then it has a $(\frac{10q}{\rho}, \frac{1}{100}, \varepsilon)$-tester that is uniform over a multiset $S$ with a small (linear) amount of redundancy, i.e., with $|S| \leq 3 \dim(C^\perp)$ and $\dim(S) \geq \dim(C^\perp) - 3\varepsilon n$.

4.2.2 Main Technical Theorem

Theorem 7 follows from the theorem stated next, which is the main technical contribution of this paper. To state the theorem we need a couple of preliminary definitions.

**Definition 13 (Support size of a test)** Let $T$ be a tester for $C$ and $S \subseteq C^\perp$ be its support. Let $B \subseteq S$ be a basis for $S$ and $u \in S$. Then let $\{u\}_B$ be the subset of $B$ needed to represent $u$ in the basis $B$. Formally, if $u = \sum_{v \in B} a_v \cdot v$ then

$$\{u\}_B = \{v \in B | a_v \neq 0\}.$$ 

We let $|u|_B = |\{u\}_B|$ be the support size of $u$ with respect to the basis $B$.

**Example 14** For $u \in S$ of the form $u = u_1 + u_2 + u_3$ for $u_1, u_2, u_3 \in B$ we have $\{u\}_B = \{u_1, u_2, u_3\}$ and $|u|_B = |\{u\}_B| = 3$.

It will be convenient to work with the following measure.

**Definition 15 (Average weight)** Given $u \in S \subseteq C^\perp$ and a basis $B$ we let

$$\text{avg}(\{u\}_B) = \frac{\sum_{u \in \{u\}_B} |u_i|}{|u|_B}.$$
to denote the average weight of the words in $\{u\}_B$.

**Theorem 16 (Main Technical Theorem)** If a $[n,k,d]_F$-code $C$ has a $(\cdot, \rho, \frac{\delta(C)}{3})$-tester which is a distribution $D$ over $S \subseteq C^\perp$ then for every basis $B$ of $S$ it holds that

$$\mathbf{E}_{u \sim D}[|u|_B \cdot \text{avg}(\{u\}_B)] \geq \rho k.$$

In particular,

- If for every $u \in S$ we have $|u|_B \leq c$ then $\mathbf{E}_{u \sim D}[\text{avg}(\{u\}_B)] \geq \frac{\rho k}{c}$.
- If for every $u \in S$ we have $\text{avg}(\{u\}_B) \leq q$ then $\mathbf{E}_{u \sim D}[|u|_B] \geq \frac{\rho k}{q}$.

### 4.2.3 A simpler proof of the main result from [12]

Ben-Sasson et al. showed in [12] that a family of randomly chosen low-density-parity-check (LDPC) codes requires, with high probability, linear query complexity. To explain the significance of this result, let us say that a code $C$ has **characterization weight** $w$ if $C^\perp$ is spanned by words of weight at most $w$. The result of [12] shows a huge gap between characterization weight — which, there, equals 3 — and query complexity, which, there, is shown to be linear in the block length of the code. All other upper bounds on the rate of families of locally testable codes are obtained by ruling out a small-weight characterization of the code. For example, the results of [10, 44] that rule out 2-query LTCs do this by (roughly) showing that any code that is characterized by 2-query words must be of small size. Similarly, the results of [5] show that any cyclic code with constant rate cannot be characterized by constant weight words.

In this section we use our main result to present an arguably simpler proof of the main result of [12]. In particular, we show that to obtain the same qualitative bounds as for random LDPC codes [12], we only replace one of the three conditions required there by a requirement that holds for all LDPC codes, namely, that $C^\perp$ has constant characterization weight. Now for the details.

We start by stating the following result of [12], which is the combination of Definition 3.4 and Theorem 3.5 there.
Theorem 17 (Some locally-character. codes require large q.c.) Let $C$ be a $[n,k,d]$-code over the two element field $\mathbb{F}_2$ such that $C^\perp$ has a basis $B$ satisfying the following two conditions for some $0 < \varepsilon, \mu < 1/2$ and some integer $q$:

- Every $w \in \mathbb{F}^n_2$ that is orthogonal to all but one constraint in $B$ satisfies $|w| \geq \varepsilon n$.
- Every $u \in C^\perp$ that is the sum of at least $\mu |B|$ constraints of $B$ must satisfy $|u| \geq q$.

Then any tester as per Definition 1 that rejects words that are $\varepsilon$-far from $C$ with probability at least $2\mu$ must have query complexity $\geq q$.

This Theorem implies the main result of [12] because a family of random LDPC codes of constant rate will satisfy the conditions of the previous theorem for some $0 < \varepsilon, \mu < 1/2$ and $q = \delta n$ for some $\delta > 0$.

Our work can be used to replace Theorem 17. In particular, the following statement does not require a basis for $C^\perp$ (any set $S$ spanning $C^\perp$ suffices), but we do need $S$ to be comprised of low weight dual words. More importantly, we completely remove the need for the first bullet in Theorem 17.

Theorem 18 Let $C$ be a $[n,k,d]$-code over the two element field $\mathbb{F}_2$ such that $C^\perp$ is spanned by a set $S \subseteq C^\perp_{\leq q}$ satisfying the following condition for some $0 < \mu < 1$ and some integer $q$:

- Every $u \in C^\perp$ that is the sum of at least $\mu \dim(C)$ words of $S$ must satisfy $|u| \geq q$.

Then any tester as per Definition 1 that rejects words that are $\frac{\delta(C)}{3}$-far from $C$ with probability at least $\mu$ must have query complexity $\geq q$.

For instance, in the case of random LDPC codes take $S$ to be the set of rows of the parity check matrix of the code. We get $q^* = O(1)$ and, following the analysis of random expanders as in [12], one can verify that the assumption of the theorem holds for any sufficiently small $\mu > 0$ and for $q = \mu' n$ where $\mu' > 0$ depends only on $\mu$. This implies that testing random LDPC codes requires linear query complexity and thus we recover the main result of [12].
Proof. By way of contradiction assume that for \( q' < q \) the code \( C \) is a \((q', \mu, \frac{\delta(C)}{3})\)-LTC with a tester having distribution \( D \). Pick any basis \( B \subseteq S \) and then by Theorem 16 it holds that \( E_{u \sim D}[|u|_B] \geq \mu \frac{\dim(C)}{q'} \), where \( D(u) > 0 \) implies \( |u| \leq q' < q \). This implies the existence of \( u \in C^\perp \) such that \( |u| < q \) and \( |u|_B \geq \frac{\mu \dim(C)}{q'} \). But this contradicts the assumption of our theorem, and the proof is complete. ■

4.2.4 Proofs of main results

We end this section by proving Theorem 7 and its corollaries using Theorem 16.

Proof of Theorem 7. By assumption \( C \) has a \((q, \frac{\delta(C)}{3}, \rho)\)-linearly independent tester which is a distribution \( D \) over some set \( B' \) (support of the tester). We consider a distribution \( D \) as a function from \( C^\perp \) to \([0, 1]\) such that \( D(u) > 0 \) iff \( u \in B' \). We have \( E_{u \sim D}[|u|] \leq q \). Since \( B' \) contains only linearly independent vectors it can be completed to a basis \( B \) for \( C^\perp \) by adding some \( u \in C^\perp \) such that \( D(u) = 0 \). Notice that we still have \( E_{u \sim D}[|u|] \leq q \) since distribution was not changed. We know that for any \( u \in B \) it holds that \( |u|_B = 1 \) and \( \text{avg}(u_B) = |u| \). Thus by the first bullet of Theorem 16 we have

\[
q \geq E_{u \sim D}[|u|] \geq \rho k.
\]

Proof of Corollary 10. By assumption \( \dim(S) \leq \dim(C^\perp) = n - k \). Partition \( S \) into \( B \cup S' \) where \( B \) is a basis for \( S \subseteq C^\perp \), \( |B| = \dim(S) \leq n - k \) and \( S' = S \setminus B \) is the set of redundant tests. We bound the size of \( S' \) from below.

Consider a basis tester defined by \( B \). By Theorem 7 this tester is not very sound, i.e., there exists a word \( w \in \mathbb{F}^n \) that is \((\frac{1}{3}\delta(C))\)-far from \( C \) and is rejected by at most a fraction \( \rho_B \leq \frac{q}{k} \) of the constraints in \( B \). The overall number of constraints rejecting \( w \) is at least \( \rho|S| = \rho(|B| + |S'|) \) because \( S \) is a uniform tester for \( C \) and \( w \) is far from \( C \). Taking the most extreme case that all words in \( S' \) reject \( w \) we get

\[
\rho(\dim(S) + |S'|) \leq \rho_B|B| + |S'| \leq \frac{q}{k} \cdot \dim(S) + |S'|
\]

28
which implies

$$|S'| \geq \frac{\rho - (q/k)}{1 - \rho} \cdot (\dim(S)),$$

thus completing the proof of Corollary 10. \[\blacksquare\]

Proof of Corollary 11. The high level idea is to partition $S$ into a basis $B$ for $S \subseteq C^\perp$ and a set of redundant tests $S'$ such that, roughly speaking, the probability of sampling from $B$, according to the distribution $p$ associated with the test $T$, is large. Then we continue as in the proof of Corollary 10.

To construct the said partition start with an arbitrary partition $S = B \cup S'$ with $B$ a basis for $S \subseteq C^\perp$. Iteratively modify the partition as follows. If there exists $u \in S'$ represented in the basis $B$ as $\sum_{b \in B} \alpha_b b$ and $p(b) < p(u)$ for some $b \in B$ with $\alpha_b \neq 0$, then replace $b$ with $u$, i.e., set $B$ to be $(B \cup \{u\}) \setminus \{b\}$ and $S'$ to be $(S' \cup \{b\}) \setminus \{u\}$. Repeat the process until no such $u \in S'$ exists. Notice the process must terminate because $\sum_{b \in B} p(b)$ is bounded by 1 and there exists $\gamma > 0$ such that with each iteration this sum increases by at least $\gamma$.

At the end of the process we have partitioned $S$ into a basis $B$ for $S$ and a redundant set $S'$ with the following property that will be crucial to our proof. For $u \in S'$,

$$p(u) \leq p(b) \text{ for all } b \in \{u\}_B.$$

(Recall that $\{u\}_B$ is minimal set of basis vectors in $B$ whose span contains $u$.)

We continue with our proof. Consider the basis tester $T'$ defined by taking the conditional distribution of the tester $T$ on $B$. Let $p'$ denote the resulting conditional distribution on $B$. Note that $p'(b) \geq p(b)$ for all $b \in B$. By Theorem 7 there exists $w$ that is $\frac{1}{3} \delta(C)$-far from $C$ and is rejected by $T'$ with probability at most $q/k$. Let $B' \subseteq B$ be the set of tests that reject $w$ and notice that $p(B') \leq p'(B') \leq q/k$.

Consider a word $u \in S'$ that rejects $w$ and represent $u$ as a linear combination of elements of $\{u\}_B \subseteq B$. Note that if the test $u$ rejects $w$, then there must be some $b$ in $\{u\}_B$ that also rejects $w$ (and hence belongs to $B'$). By the special properties of our partition which were discussed in the previous paragraph we have

$$p(u) \leq p(b) \leq p(B') \leq q/k.$$
Thus, every test that rejects \( w \) from \( S' \) has probability at most \( q/k \) of being performed, and, furthermore, the probability of rejecting \( w \) using an element of \( B \) is at most \( q/k \) as well. Summing up, we get

\[
\rho \leq \Pr[T[w] = \text{reject}] \leq q/k + |S'| \cdot q/k,
\]

which after rearranging the terms gives \( |S'| \geq \frac{\rho k}{q} - 1 \) as claimed.

4.3 Proof of Main Technical Theorem 16

4.3.1 Overview for the proof of Theorem 16

We recall that \( C \) is a \((q, \varepsilon, \delta(C))\)-LTC. Let \( S \) be the support of the tester of \( C \) and let \( B = \{u_1, \ldots, u_{n-k}\} \subseteq C_{\leq q}^\perp \) be a basis for \( C_{\leq q}^\perp \) obtained by starting with a basis for \( S \) and completing it to a basis for \( C_{\leq q}^\perp \) in an arbitrary manner.

Now, assume that \( B \) is a basis for \( C_{\leq q}^\perp \) and contains words of size at most \( q \).

In Section 4.3.2 we define the set of vectors \( V = \{v_1, v_2, \ldots, v_{n-k}\} \), where \( v_i \in \mathbb{F}^n \). Then we prove in Claim 21 that \( \dim(\text{span}(C \cup V)) = n \) and \( \dim(\text{span}(V)) = n - k \). Claim 21 shows that every \( w \in \mathbb{F}^n \) has a unique representation as a sum of a single element from \( C \), denoted \( c(w) \), and a linear combination of \( v_j \)'s, denoted \( v(w) \). We say \((c(w), v(w))\) is the \((C,V)\)-representation of \( w \). We denote by \( \Gamma(w) \subseteq [n-k] \) the set of indices \( (j) \) of \( v_j \)'s participating in \( v(w) \). Formally, if \( v(w) = \sum_{j=1}^{n-k} \alpha_j v_j \) then \( \Gamma(w) = \{j \mid \alpha_j \neq 0\} \).

In particular, this is true for every singleton vector \( e_i \in \mathbb{F}^n \), i.e., for every \( i \in [n] \) we have \( e_i = c(e_i) + v(e_i) \).

Using this fact, in Lemma 23 we show that there exist \( k \) distinct indices \( i_1, \ldots, i_k \in [n] \) and \( k \) corresponding words \( w_{i_1}, \ldots, w_{i_k} \in \mathbb{F}^n \) such that for every \( i \) the following two conditions hold:

- \( \delta(w_{i_j}, C) \geq \frac{\delta(C)}{3} \).
- For every \( u \in B \) we have that if \( i_j \notin \text{supp}(u) \) then \( \langle u, w_{i_j} \rangle = 0 \).

Then, without loss of generality we may assume that \( \{i_1, \ldots, i_k\} = [k] \). For \( i \in [k] \) we define the set \( B_{w_i} = \{u \in B \mid \langle u, w_i \rangle \neq 0\} \). These sets are used to prove Theorem 16.
4.3.2 The \((C, V)\)-representation of words in \(F^n\)

Let \(B = \{u_1, \ldots, u_{n-k}\} \subseteq C_{\leq q}^\perp\) be a basis for \(C^\perp\) obtained by starting with a basis for \(S\) and completing it to a basis for \(C^\perp\) in an arbitrary manner.

The first part of our proof shows that every word in \(F^n\) can be represented uniquely as the sum of a codeword in \(C\) and a subset of a set of vectors \(V = \{v_1, \ldots, v_{n-k}\}\) where the rejection probability of \(w\) is related to its representation structure. We start by defining \(V\).

**Definition 19** For \(i \in [n-k]\) let \(v_i\) be a word of minimal weight that satisfies
\[
\langle v_i, u_j \rangle = \begin{cases} 
1 & i = j \\
0 & j \in [n-k] \setminus \{i\}
\end{cases}
\]
and let \(V = \{v_1, \ldots, v_{n-k}\}\).

**Proposition 20** For all \(v_i \in V\) we have \(\frac{\text{wt}(v_i)}{n} = \delta(v_i, C)\).

**Proof.** We have \(\delta(v_i, C) \leq \frac{\text{wt}(v_i)}{n}\) because \(\delta(v_i, 0^n) = \frac{\text{wt}(v_i)}{n}\) and \(0^n \in C\). On the other hand, for every \(c \in C\) we must have \(\delta(v_i, c) \geq \frac{\text{wt}(v_i)}{n}\) because if \(\delta(v_i, c) < \frac{\text{wt}(v_i)}{n}\) then setting \(v'_i = v_i - c\) we have \(\text{wt}(v'_i) < \text{wt}(v_i)\) but \(v'_i\) satisfies (4.1) (with respect to index \(i\)), thus contradicting the minimal weight of \(v_i\).

The following claim states that \(F^n\) is the direct sum of the code \(C\) and \(\text{span}(V)\).

**Claim 21** \(\dim(\text{span}(C \cup V)) = n\) and \(\dim(V) = n - k\).

**Proof.** Let \(Z = C \cup V\). To prove both equalities stated in our claim it is sufficient to show that \(Z^\perp = \{0^n\}\), i.e., that \(\dim(Z^\perp) = 0\), because \(\dim(C) = k\) and \(|V| = n - k\). Assume by way of contradiction that \(u \in Z^\perp\) is nonzero. Then in particular \(u \in C^\perp\) because \(C \subseteq Z\) which implies \(C^\perp \supseteq Z^\perp\). Thus, \(u\) is a nonzero linear combination of vectors from \(B\) because \(B\) is a basis for \(C^\perp\). Suppose \(u_i\) appears in the representation of \(u\) under \(B\). Then from (4.1) we conclude \(\langle u, u_i \rangle \neq 0\) which implies \(u \notin V^\perp\) which gives \(u \notin Z^\perp\), contradicting the assumption \(u \in Z^\perp\). So \(\dim(Z^\perp) = 0\) and this completes our proof.

Claim 21 shows that every \(w \in F^n\) has a unique representation as a sum of a single element from \(C\), denoted \(c(w)\), and a linear combination of \(v_j\)'s,
denoted \( v(w) \). We say \((c(w), v(w))\) is the \((\mathcal{C}, V)\)-representation of \( w \). We denote by \( \Gamma(w) \subseteq [n-k] \) the set of indices \((j)\) of \( v_j \)'s participating in \( v(w) \). Formally, if \( v(w) = \sum_{j=1}^{n-k} \alpha_j v_j \) then

\[
\Gamma(w) = \{ j \mid \alpha_j \neq 0 \}.
\]

The next claim relates the rejection probability of \( w \) by our basis tester to the structure of \( v(w) \). For \( i \in [n-k] \) let \( p(i) = p(u_i) \) denote the probability of \( u_i \) under the distribution associated with our basis tester. For \( I \subseteq [n-k] \) the set of indices of \( B' \subseteq B \) let \( p(I) = p(B') = \sum_{i \in I} p(i) = \sum_{u_i \in B'} p(u_i) \).

**Claim 22 (Rejection probability is related to \((\mathcal{C}, V)\)-representation)**

For all \( w \in \mathbb{F}_n \) we have

\[
\Gamma(w) = \{ j \in [n-k] \mid \langle u_j, w \rangle \neq 0 \}.
\]

Consequently, we have

\[
\Pr[T[w] = \text{reject}] = p(\Gamma(w)).
\]

**Proof.** Consider the \((\mathcal{C}, V)\)-representation of \( w \):

\[
w = c(w) + \sum_{j \in \Gamma(w)} \alpha_j v_j, \text{ where } \alpha_j \neq 0.
\]

By assumption for all \( u_i \in B \) we have \( \langle u_i, c(w) \rangle = 0 \) and by (4.1) we have \( \langle u_i, v(w) \rangle \neq 0 \) if and only if \( i \in \Gamma(w) \). This implies (4.2). The consequence follows because, by definition, the probability of rejecting \( w \) is the probability of the event \( \langle u_i, w \rangle \neq 0 \) where \( u_i \) is selected from \( B \) with probability \( p(i) \). This completes the proof.

### 4.3.3 Main Lemma and Proof of Main Theorem 16

The following lemma is the main part of our proof. Assuming it we shall promptly complete the proof of Theorem 16 and the proof of the lemma comes after the proof of the theorem. In what follows the singleton vector \( e_i = 0^{i-1}10^{n-i} \) is the characteristic vector of the singleton set \( \{i\} \subset [n] \).
Lemma 23 (Main Lemma) If $C$ is an $[n,k,d]_F$-code and $B = \{u_1, \ldots, u_{n-k}\}$ is a basis for $C^\perp$, then there exist $k$ distinct indices $i_1, \ldots, i_k \in [n]$ and $k$ corresponding words $w_{i_1}, \ldots, w_{i_k} \in \mathbb{F}^n$ such that for every $i_j$ the following two conditions hold:

- $\delta(w_{i_j}, C) \geq \frac{\delta(C)}{3}$.
- For every $u \in B$ we have that if $i_j \notin \text{supp}(u)$ then $\langle u, w_{i_j} \rangle = 0$.

Proof of Theorem 16. We apply Main Lemma 23, and without loss of generality we may assume that $\{i_1, \ldots, i_k\} = [k]$. Recall that we have $w_1, \ldots, w_k$ such that for every $i \in [k]$ it holds that $\delta(w_i, C) \geq \frac{\delta(C)}{3}$ and for every $u \in B$ we have it holds that if $i \notin \text{supp}(u)$ then $\langle u, w_i \rangle = 0$. For $i \in [k]$ let

$$B_{w_i} = \{u \in B \mid \langle u, w_i \rangle \neq 0\}.$$ 

Note that $B_{w_i} \subseteq \{u \in B \mid i \in \text{supp}(u)\}$, so we have the following immediate claim (proof omitted), which will be used later in the proof of the theorem.

Claim 24 For every $i \in [k]$ and $u \in B$ it holds that if $u \in B_{w_i}$ then $i \in \text{supp}(u)$ and thus $u$ can belong to at most $|\text{supp}(u)| = |u|$ different $B_{w_j}$s.

We continue the proof of Theorem 16. For all $i \in [k]$ we have

$$\Pr_{u \in D}[(\langle u, w_i \rangle \neq 0)] \geq \rho$$

because $D$ is a $(q, \frac{\delta(C)}{3}, \rho)$-tester of $C$. Hence for all $i \in [k]$ we have

$$\Pr_{u \in D} [|\{u\}_B \cap B_{w_i}| \geq 1] \geq \rho.$$ 

So by linearity of expectation:

$$\mathbb{E}_{u \in D} [|\{u\}_B \cap B_{w_1}| + |\{u\}_B \cap B_{w_2}| + \ldots + |\{u\}_B \cap B_{w_k}|] \geq \rho k.$$ 

Let us consider

$$|\{u\}_B \cap B_{w_1}| + |\{u\}_B \cap B_{w_2}| + \ldots + |\{u\}_B \cap B_{w_k}|.$$
Let $m = |\{u\}_B|$ and $\{u\}_B = \{u_1, ..., u_m\}$. Note that $m$, $u$ and $u_1, ..., u_m$ are random variables. Let $X_{i,j}$ to be an indicator variable for the event “$u_i \in B_{w_j}$”, i.e. $X_{i,j}$ equals 1 if $u_i \in B_{w_j}$ and equals 0 otherwise. Then

$$|\{u\}_B \cap B_{w_1}| + |\{u\}_B \cap B_{w_2}| + \ldots + |\{u\}_B \cap B_{w_k}| = \sum_{j=1}^{k} \sum_{i=1}^{m} X_{i,j} = \sum_{i=1}^{m} \sum_{j=1}^{k} X_{i,j}$$

Note that $B_{w_1} \cup \ldots \cup B_{w_k} \subseteq B$. By Claim 24 $u_i$ is contained in at most $|u_i|$ sets $B_{w_j}$ and thus we have

$$\sum_{j=1}^{k} \sum_{i=1}^{m} X_{i,j} \leq |u_i|$$

So,

$$|\{u\}_B \cap B_{w_1}| + |\{u\}_B \cap B_{w_2}| + \ldots + |\{u\}_B \cap B_{w_k}| = \sum_{j=1}^{k} \sum_{i=1}^{m} X_{i,j} = \sum_{i=1}^{m} \sum_{j=1}^{k} X_{i,j} \leq \sum_{i=1}^{m} |u_i| = \sum_{u_i \in \{u\}_B} |u_i|$$

Thus

$$\mathbb{E}_{u \in \mathcal{D}^S} \left[ \text{avg}(\{u\}_B) \cdot |u|_B \right] = \mathbb{E}_{u \in \mathcal{D}^S} \left[ \sum_{u_i \in \{u\}_B} |u_i| \right] \geq \rho k.$$ 

This completes the proof of Theorem 16 from Lemma 23. ■

**Proof of Lemma 23.** We start by showing that there exist $k$ distinct singleton vectors, denoted without loss of generality $e_1, \ldots, e_k$, such that $c(e_1), \ldots, c(e_k)$ are linearly independent, hence distinct and nonzero.

Since every word in $\mathbb{F}^n$ has a unique $(\mathcal{C}, V)$-representation we get $e_i \in \{c(e_i) + v \mid v \in \text{span}(V)\}$. This implies

$$\{e_1, \ldots, e_n\} \subseteq \text{span}(\{c(e_1), \ldots, c(e_n)\} \cup V).$$
Counting dimensions, we have
\[
    n = \dim(\text{span}(\{e_1, \ldots, e_n\})) \\
    \leq \dim(\text{span}(\{c(e_1), \ldots, c(e_n)\} \cup V)) \\
    \leq \dim(\text{span}(\{c(e_1), \ldots, c(e_n)\})) + \dim(\text{span}(V)).
\]

By Claim 21 we have \(\dim(\text{span}(V)) = n - k\), so we conclude that (without loss of generality) \(c(e_1), \ldots, c(e_k)\) are linearly independent, as claimed.

Next, we argue that for \(i \in [k]\) we have \(|v(e_i)| \geq d - 1\). This is because \(e_i = c(e_i) + v(e_i)\) and \(|e_i| = 1\) and \(|c(e_i)| \geq d\) because \(c(e_i)\) is a nonzero word in a linear code with minimal distance \(d\).

So far we have shown that for every \(v(e_i), i \in [k]\) we have \(|v(e_i)| \geq d - 1\).

Let \(i \in [k]\) and let us show that there exists \(w_i \in \text{span}(\{v_j \mid j \in \Gamma(e_i)\})\) such that \(\delta(w_i, \mathcal{C}) \geq \frac{\delta(\mathcal{C})}{3}\). Note that in this case for all \(u \in B\) we have that if \(i \notin \text{supp}(u)\) then \(\langle u, w_i \rangle = 0\). This is true since Claim 22 implies that for every \(j \in \Gamma(e_i)\) and \(u \in B\) we have that if \(i \notin \text{supp}(u)\) then \(\langle u, v_j \rangle = 0\).

Now, if \(|v_j| \geq \frac{1}{3}d\) for some \(j \in \Gamma(e_i)\) then setting \(w_i = v_j\) completes the proof because Proposition 20 implies that \(v_j\) is \(\frac{d}{3n}\)-far from \(\mathcal{C}\).

From here on we assume \(|v_j| < \frac{1}{3}d\) for all \(j \in \Gamma(e_i)\). Let \(t = |\Gamma(e_i)|\) and assume without loss of generality that \(\Gamma(e_i) = [t]\). Denote the \((\mathcal{C}, V)\)-representation of \(e_i\) by \(c(e_i) + \sum_{j=1}^{t} \alpha_j v_j\) where \(\alpha_j \neq 0\). Let \(w_\ell = \sum_{j=1}^{\ell} \alpha_j v_j\). We know the following:

- \(|w_1| < \frac{1}{3}d|.

- \(|w_\ell| = |v(e_i)| \geq d - 1\) by the second bullet in Lemma 23.

- \(|w_{\ell+1}| \leq |w_\ell| + |v_{\ell+1}| < |w_\ell| + \frac{1}{3}d| \) for all \(1 \leq \ell < t\), by the triangle inequality.

This implies the existence of some \(\ell \in [t]\) such that \(\frac{1}{3}d < |w_\ell| \leq \frac{2d}{3}\) and notice \(w_1 = w_\ell\) is \(\frac{d}{3n}\)-far from \(\mathcal{C}\). To see this note that \(\delta(w_\ell, 0) = \frac{|w_\ell|}{n} \geq \frac{d}{3n}\) and for every \(c \in \mathcal{C}\) such that \(c \neq 0^n\), we have \(\delta(w, c) \geq |c| - |w_\ell| \geq d - \frac{2d}{3} = \frac{d}{3}\), where the last inequality follows since \(|c| \geq \Delta(\mathcal{C}) \geq d\). ■
4.4 Tightness of Main Theorem 7

In this section we argue that the bound \((k \leq \frac{q}{\rho})\) obtained in Theorem 7 is close to tight. In Proposition 25 we show that there are codes with constant relative distance and constant dimension which have a basis tester, and in Proposition 26 we show that all codes have a basis tester whose query complexity equals to the dimension of the code plus one. Propositions 25 and 26 are folklore.

Proposition 25 (The repet. code has strong uniform basis tester)

For any finite field \(F\) and constant \(c \in \mathbb{N}^+\) there exists a \([n = cm, k = c, d = m]_F\)-code \(C\) which has a \((2,1)\)-strong basis tester.

Proof. Let \(C\) be the \([n = cm, k = c, d = m]_F\) repetition code where a \(c\)-symbol message \(a_1, \ldots, a_c\) is encoded by repeating each symbol \(m\) times, i.e., \(a_1, \ldots, a_c \mapsto a_1^m, \ldots, a_c^m\). Consider the tester that compares a random position in a block to the first symbol in the block. Formally, the tester is defined by the uniform distribution over the following set \(B\) of words of weight 2:

\[
B = \{e_{im+1} - e_{im+j} \mid i \in \{0, \ldots, c - 1\}, j \in \{2, \ldots, m\}\},
\]

where \(e_\ell\) has a 1 in the \(\ell\)th coordinate and is zero elsewhere.

It can be readily verified that \(B\) is a basis for \(C^\perp\), has query complexity 2 and rejects a word \(w\) with probability \(\delta(w, C)\) because if the rejection probability is \(\varepsilon\) this means that at most an \(\varepsilon\) fraction of symbols need to be changed to reach a word that is constant on each of its \(c\) blocks. ■

Proposition 26 (Every code has a basis tester with large q.c.) Let \(F\) be a finite field and \(C\) be a \([n, d]_F\) code. Then \(C\) has a \((k + 1,1)\)-strong uniform basis tester.

Proof. Let \(G \in F^{n \times k}\) be a generating matrix for \(C\), i.e., \(C = \{Gm \mid m \in F^k\}\). Assume without loss of generality that the first \(k\) rows of \(G\) are linearly independent. This means that after querying the first \(k\) symbols of a word \(w_1, \ldots, w_k\), one can interpolate to obtain any other symbol of the codeword that is the encoding of the message \(m \in F^k\) such that \((Gm)[k] = (w_1, \ldots, w_k)\). For \(k < i \leq n\) let \(u_i\) be the constraint that
queries the first $k$ symbols of $w$ and accepts iff $w_i$ is equal to the $i$th symbol of the encoding of $m$. It can be readily verified that $B = \{u_{k+1}, \ldots, u_n\}$ is a basis for $C^\perp$ and has query complexity $k + 1$.

Consider the soundness of the uniform tester over $B$. If $\Pr[T[w] = \text{reject}] \leq \rho$ then $w$ is $\rho$-close to the codeword of $C$ that is the encoding of $m$, implying that $\delta(w, C) \leq \delta(w, G_m) \leq \rho$. ■

### 4.5 Bounds on tester support size

We show that every binary linear code $C$ can be tested with linear redundancy, by proving the following statement. We point out that [7] implicitly showed already that every code can be tested with a linear amount of redundancy. The added value of the following statement is that it shows that the amount of redundancy can be as small as twice the dimension of $C^\perp$.

**Proposition 27** If $C \subseteq \mathbb{F}_2^n$ is a $[n, k, d]_{\mathbb{F}_2}$-code that is a $(q, \rho, \varepsilon)$-LTC, then it has a $(\frac{10\log q}{\rho}, \frac{1}{100}, \varepsilon)$-tester whose support is over a set $U$ of size at most $3 \dim(C^\perp)$.

The proof of the above result appears in Section 4.5.1.

**Remark 28** Inspection of the proof of Proposition 27 reveals that $C$ can be tested by a $(c \cdot q, 1/c, \varepsilon)$-tester whose support is over $U$ of size $\leq (4 \ln 2 + \eta) \cdot (n-k)$ for any $\eta > 0$, where $c > 1$ is a constant that depends on $\eta$ and goes to infinity as $\eta$ goes to 0. Recalling $4 \ln 2 = 2.77258 \ldots$, we preferred to round this constant up to the closest integer in the statement of the proposition above.

**Remark 29** The assumption that the code is binary is not crucial, and Proposition 27 can be stated for the fields of larger cardinality, but the size of $U$ will increase respectively.

While the support of a non-strong tester need not span $C^\perp$, we can prove that every tester’s support must at least span a large subspace of $C^\perp$.

**Proposition 30** Let $T$ be a $(q, \rho, \varepsilon)$-tester for a linear code $C \subseteq \mathbb{F}^n$ such that $\varepsilon \leq \frac{\delta(C)}{3}$. Let $U \subseteq C_{\leq q}^\perp$ denote the support of $T$. Then $\dim(U) \geq \dim(C^\perp) - 3\varepsilon n$.

The proof of the above result appears in Section 4.5.2.
4.5.1 Proof of Proposition 27

Let us state a couple of inequalities in probability that will be used later on in the proof.

Claim 31 (Chernoff Bound) If $X = \sum_{i=1}^{m} X_i$ is a sum of independent \{0,1\}-valued random variables, where $\Pr[X_i = 1] = \gamma$, then

$$\Pr[X < (1 - \sigma)\gamma m] \leq \exp(-\sigma^2\gamma m/2).$$

Claim 32 If $X = \sum_{i=1}^{m} X_i$ is a sum of independent \{0,1\}-valued random variables, where $\Pr[X_i = 1] = \gamma$, then

$$\Pr[X \equiv 0 \pmod{2}] \leq \frac{1}{2}(1 + \exp(-2\gamma m)).$$

Proof of Proposition 27. Let $t = \frac{10}{\rho}$ and $m = 3 \dim(C^\perp) = 3(n - k)$. Let $T$ be the assumed $(q, \rho, \varepsilon)$ tester for $C$. Pick $U = \{u_1, \ldots, u_m\}$ where each $u_i$ is obtained by taking the sum of $t$ independent samples from $T$. $U$ is a multiset and the distribution $p$ associated with our tester is the uniform distribution over $U$. The query complexity of $U$ is bounded by $t q = \frac{10q}{\rho}$.

To analyze soundness, fix a word $w$ that is $\varepsilon$-far from $C$. Let $X_i$ be the indicator random variable for the event $\langle w, u_i \rangle \neq 0$. By Claim 32 it holds that $\Pr[X_i = 0] \leq \frac{1}{2}(1 + e^{-2\rho t})$ and $\Pr[X_i = 1] \geq \frac{1}{2}(1 - e^{-2\rho t})$.

Let $U_{\text{bad}} = \{u \in U \mid \langle u, w \rangle \neq 0\}$. We know that $E[|U_{\text{bad}}|] > \frac{m}{2}(1 - e^{-2\rho t}) \geq \frac{m}{2}(0.999)$. Let $\sigma = 0.979$ and note that $\frac{m}{2}(0.999)(1 - \sigma) = 0.021m < \frac{m}{100}$. Then by the Chernoff bound (Claim 31) we have

$$\Pr\left[\frac{|U_{\text{bad}}|}{m} < \frac{1}{100}\right] = \Pr\left[|U_{\text{bad}}| < \frac{m}{100}\right] \leq e^{-0.979^2(\frac{1}{2}(1 - e^{-2\rho t}))m/2}$$

We take a union bound over all words that are $\varepsilon$-far from $C$. Notice that $\mathbb{F}_2^n$ can be partitioned into $2^{n-k}$ affine shifts of (the linear space) $C$. For each such affine shift, which has the form $v + C = \{v + c \mid c \in C\}$, any two words from $v + C$ differ only by a word from $C$ which has inner product 0 with all tests. Thus, it suffices to take a union bound over one representative per affine shift, and there are at most $2^{n-k}$ of them.

Continuing with the proof, the probability that there exists a $\varepsilon$-far word that is rejected with probability less than $\frac{1}{100}$ is at most $e^{-0.979^2(\frac{1}{2}(1 - e^{-2\rho t}))m/2}$.
The inequality follows since by construction we have \( m > 2.95(n - k) \) and hence \( m > 2 \cdot 773(\bar{n} - k) \). So \(-0.979^2(\frac{1}{2}(1 - e^{-2\rho^2}))m/2 + \ln(2)(n - k) < 0 \) and \( e^{-0.979^2(\frac{1}{2}(1 - e^{-2\rho^2}))m/2 + \ln(2)(n - k)} < 1 \).

Hence we showed that there is a positive probability to pick the set \( U \) such that every \( \varepsilon \)-far word is rejected with probability at least \( \frac{1}{100} \) and this completes the proof. ■

4.5.2 Proof of Proposition 30

Proof of Proposition 30. Assume by way of contradiction that \( \dim(U) < \dim(C^\perp) - 3\varepsilon n \). We call a word \( w \) a coset leader if \( w \) has minimal weight in \( w + C = \{ w + c \mid c \in C \} \). (If there is more than one minimal weight word in \( w + C \), arbitrarily pick one of them to be the coset leader.) The proof of Proposition 20 implies that if \( w \) is a coset leader, then \( \frac{\text{wt}(w)}{n} = \delta(w, C) \).

Let

\[ V = \{ w \in \mathbb{F}^n \setminus C \mid \text{for all } u \in U : \langle u, w \rangle = 0 \text{ and } w \text{ is a coset leader of } w + C \}; \]

i.e., \( V \) contains all non-codewords that are coset leaders and are accepted by all tests in \( U \). Notice that for all \( w \in V \) we have \( \delta(w, C) = \delta(w, 0) = |w| \) because \( w \) is a coset leader of \( w + C \).

We argue that \( \dim(V) \geq 3\varepsilon n \) and thus \( \sum_{v \in V} (\text{supp}(v)) \geq \dim(V) \geq 3\varepsilon n \). We show that \( \dim(V) \geq 3\varepsilon n \). It sufficient to prove that \( \dim(V) \geq \dim(C^\perp) - \dim(U) \geq 3\varepsilon n \). Fix any basis \( U_B = \{ u_1', \ldots, u_n' \} \) for \( U \) and complete it to a basis for \( C^\perp \) in an arbitrary manner, obtaining \( U_B \cup R_B \), where \( R_B = \{ u_1, \ldots, u_h \} \); i.e., \( U_B \cap R_B = \emptyset \) and \( \text{span}(U_B \cup R_B) = C^\perp \).

Note that \( h \geq 3\varepsilon n \). We argue that \( \dim(V) = h \). This is true since for every \( u_j \in R_B \) we have unique coset leader \( v_j \in \mathbb{F}^n \) (up to multiplication by a nonzero element of \( \mathbb{F} \)) such that \( \langle u_j, v_j \rangle \neq 0 \) and \( \langle u, v_j \rangle = 0 \) for all \( u \in (R_B \setminus \{ u_j \}) \cup U_B \). Moreover, all these \( v_j \)'s are linearly independent by definition.

In addition, for all \( v \in V \) we have \( |\text{supp}(v)| = \delta(w, C) < \varepsilon n \) because

\[ \Pr[T[v] = \text{reject}] = 0. \]

Let \( w_1, \ldots, w_s \) be an arbitrary ordering of the elements of \( V \). Let \( \mu(\ell) \) be
the maximal weight of an element in $\text{span}(w_1, \ldots, w_\ell)$. We have $\mu(1) \leq \varepsilon n$ and $\mu(s) \geq \frac{3}{2}\varepsilon n$ because the expected weight of a word in $\text{span}(V)$ is (exactly) $\frac{|F| - 1}{|F|} \left| \bigcup_{w \in V} (\text{supp}(w)) \right|$. To see that the expected weight of a word $w_{\text{exp}} \in \text{span}(V)$ is as claimed, note that $w_{\text{exp}}$ is picked by a random linear combination of vectors in $V$, where each vector $v \in V$ is taken independently with probability $1 - \frac{1}{|F|}$. Hence if $i \in \bigcup_{w \in V} (\text{supp}(w))$, then $i \in \text{supp}(w_{\text{exp}})$ with probability $\frac{|F| - 1}{|F|} \geq \frac{1}{2}$.

Finally, we have $\mu(\ell + 1) < \mu(\ell) + \varepsilon n$. We conclude there must exist $\ell$ for which $\varepsilon n < \mu(\ell) \leq 2\varepsilon n$. Let $w'$ be a word of maximal weight in $\text{span}(w_1, \ldots, w_\ell)$. We have that $\varepsilon n < |w'| \leq 2\varepsilon n$. We conclude that $\Delta(w, 0) = |w'| > \varepsilon n$ and for all $c \in C \setminus \{0^n\}$ we have $\Delta(w, c) \geq |c| - |w| \geq d - 2\varepsilon n \geq d - \frac{2d}{3} = \frac{d}{3}$.

We see that $w'$ is $\varepsilon$-far from $C$ but accepted by $T$ with probability 1. Contradiction.

4.6 Open questions and Discussion

In this section we discuss a possible approach to obtain the upper bounds on the rate of LTCs. Recall that the main open question in the subject of locally testable codes is whether there are asymptotically good LTCs. We believe that there is no asymptotically good LTC and suggest a way that may lead to proving that these codes do not exist. One of the ways to disprove the existence of a certain kind of LTCs is to assume its existence and its tester, and then show a counter-example $w$, which will be far from the code but rejected with very small probability. The main challenge in this way is to construct a specific counter-example that “cheats” the tester for the given LTC. In particular, the problem is to argue that $w$ will satisfy almost all local constraints which can be selected by the tester.

We suggest to consider the basis for a dual code, and to construct many different counter-examples that will be far from each other and far from the code, but each of them will satisfy all but one basis constraint. More formally, we suggest to show that that if $C \subseteq F^n$ is an $(q, \rho, \varepsilon)$-asymptotically good LTC and $B \subseteq C_{\leq q}^\perp$ is a basis for $C_{\leq q}^\perp$ then there exist $\Omega(n)$ different words $v_j$, where every word is $\Omega(1)$ far from $C$ and from the other $v_j$-s, but satisfies all but one constraints from $B$. This creates a large space of potential counter-examples in $F^n$. If this approach succeeds then one might
show that at least one of \( v_j \)-s satisfies almost all local constraints of the tester. Now to the details.

Assume the existence of asymptotically good \((q, \rho, \varepsilon)\)-LTC \( C \subseteq \mathbb{F}^n \) whose tester has support \( S \). Note that \( k = \dim(C) = \Omega(n) \). Let \( B \subseteq C_{\leq q}^\perp \) be a corresponding basis.

The technique of the Main Theorem 16 implies that there are \( \Omega(k) \) different \( v_i \) such that each one is \( \Omega(1) \) far from \( C \). To see this note that for each \( i \in [k] \) we have:

\[
e_i = c_i + \sum_{j \in J_i} v_j; c_i \in C \setminus \{0\}
\]

Let \( h = \frac{2q(n-k)}{k} \). Note that \( h = O(1) \) since \( k = \Omega(n) \). We say that \( j \in [k] \) has high degree if for at least \( h \) different \( u \in B \) it holds that \( j \in \text{supp}(u) \).

The number of high degree indices \( j \in [k] \) is bounded above by \( \frac{q(n-k)}{h} = \frac{k}{2} \). Thus the number of low degree indices \( j \in [k] \) is at least \( k - k/2 = k/2 \).

Without loss of generality we assume that all \( i \in [k/2] \) have low degree, i.e.

\[
e_i = c_i + \sum_{j \in J_i} v_j; c_i \in C \setminus \{0\} \text{ and } |J_i| \leq h
\]

Hence that all \( i \in [k/2] \) we have \( \sum_{j \in J_i} |v_j| \geq \Delta(C) - 1 \) and there exists \( v_j \) \(|v_j| \geq \frac{\Delta(C)-1}{h} \) and \( j \in J_i \).

Each \( v_j \) can be counted at most \( q \) times, since \(|\text{supp}(u_j)| \leq q \). We conclude that there are at least \( \frac{k}{2q} \) different \( v_j \) such that for every \( v_j \) we have \( \Delta(v_j, C) \geq \frac{\Delta(C)-1}{h} \). We believe that a constant fraction of them should be also far from each other and that this should somehow result in additional restrictions on LTCs. Hence, assuming the existence of asymptotically good LTC \( C \), one should get \( \Omega(n) \) different \( v_j \), where each one is \( \Omega(1) \) far from \( C \) and from the other \( v_j \)-s.
Chapter 5

Locally Testable vs. Locally Decodable Codes
5.1 Introduction

In this chapter we study the relation between locally testable and locally decodable codes. Locally testable codes (LTCs) are error-correcting codes for which membership of a given word in the code can be tested probabilistically by examining it in very few locations. Locally decodable codes (LDCs) allow to recover each message entry with high probability by reading only a few entries of a slightly corrupted codeword. A linear code $C \subseteq \mathbb{F}_2^n$ is called sparse if $n \geq 2^{\Omega(\dim(C))}$.

LDCs are related to private information retrieval protocols, initiated by [24], while LTCs are related to PCPs [41]. Both these families of error correcting codes are explicitly studied, for survey see e.g. [67]. In spite of the fact, the distinction between the two families of the codes was not made. Namely, it is well-known that there is an intersection between the two families of codes, e.g., the famous Hadamard code is 3-query LTC and 2-query LDC. Moreover, it is well-known that LTCs do not imply LDCs, i.e., there are LTCs which are not LDCs. This follows simply by comparing the upper and lower bounds on the blocklength of these families of codes. If $C \subseteq \mathbb{F}_n^q$ is a $q$-query LDCs then $n \geq \Omega(\dim(C)^{q/(q-1)})$ (by Katz and Trevisan [47]), while there exist (best known) LTCs such that $n \leq O(\dim(C) \cdot (\text{poly log}(\dim(C))))$ [15, 26, 59]. However, the other direction, i.e., whether LDCs imply LTCs, was not known.

5.1.1 Main Results

We show that LDCs do not imply LTCs, and in fact there are inherent differences between LDCs and LTCs. Specifically we show the following results.

- In Theorem 35 we show that codes invariant under two-transitive groups that obey a local constraint are LDCs, while they are not necessarily LTCs. This provides a general proof to the local decodability of polynomial codes such as Hadamard code, Reed-Muller codes and dual-BCH codes. Combining this with a recent result of [42], we obtain an explicit family of linear codes which is locally decodable but is not locally testable.

- In Theorem 40 we show that every non-sparse code contains a large
subcode which is not LTC, while every subcode of an LDC remains LDC (Corollary 43). Hence, every non-sparse LDC contains a subcode that is LDC but is not LTC. Moreover, we show (Theorem 45) that if we consider uniform-LTCs (for which a tester picks every possible local constraint with the same probability) then, in fact, every non-sparse LDC has many large subcodes which are not uniform-LTCs (but still LDCs).

The above results demonstrate inherent differences between LDCs and LTCs, in particular, they imply that LDCs do not imply LTCs.

5.1.2 On Sparse codes vs. Non-sparse codes

Recall that a code $C \subseteq \mathbb{F}_2^n$ is called sparse if $n \geq 2^\Omega(\dim(C))$, otherwise the code is non-sparse. A sparse code $C$ is called unbiased if all nonzero codewords $c \in C$ have relative weight ranging in $(\frac{1}{2} - \gamma, \frac{1}{2} + \gamma)$ for some constant $\gamma > 0$. Kaufman and Sudan [49] showed that all sparse unbiased codes are LTCs and LDCs. Since every subcode of a sparse unbiased code is a sparse unbiased code we conclude that it is an LTC and LDC. However, sparse codes have exponential blocklength.

Our Theorem 40 shows that every non-sparse LDC contains a large subcode which is not LTC. Hence, every non-sparse LDC contains a subcode that is LDC but is not LTC. This demonstrates an inherent difference between sparse and non-sparse codes. In sparse codes local testability is preserved in subcodes, while in non-sparse codes local testability is not preserved in subcodes. In contrast to local testability, local decodability of all codes is always preserved in their subcodes (Corollary 43).

5.1.3 Locally Decodable vs. Locally Correctable Codes

The following folklore claim says that LCCs imply LDCs with the same parameters.

**Claim 33** If $C \subseteq \mathbb{F}_2^n$ is a $(q, \varepsilon, \delta)$-LCC then $C$ is a $(q, \varepsilon, \delta)$-LDC.

**Proof.** Let $k = \dim(C)$. We pick a generator matrix $G \in \mathbb{F}^{n \times k}$ for $C$, i.e., $C = \{Gm \mid m \in \mathbb{F}^k\}$ such that the first $k$ rows of $G$ form identity
matrix\(^1\). Hence the first \(k\) symbols of the code are message symbols, i.e., for all \(m \in \mathbb{F}^k\) we have \((Gm)|[k] = m\).

Let \(\mathbb{S}\mathcal{C}\) be a self-corrector for a code \(\mathcal{C}\) that for every \(i \in [n]\) reads at most \(q\) symbols and recovers the symbol \(i\) with probability at least \(\frac{1}{|\mathcal{F}|} + \varepsilon\) even if at most \(\delta\)-fraction of the symbols was adversely corrupted. In particular, \(\mathbb{S}\mathcal{C}\) recovers with probability at least \(\frac{1}{|\mathcal{F}|} + \varepsilon\) every coordinate \(i \in [k]\), i.e., every message symbol. We conclude that \(\mathcal{C}\) is a \((q, \varepsilon, \delta)\)-LDC.

We stress that LDCs do not imply LCCs. To see this, let \(C \subseteq \mathbb{F}_2^n\) be an LDC. Append to it one entry (with coordinate \((n+1)\)) obtaining \(C' \subseteq \mathbb{F}_2^{n+1}\), such that this entry will not be involved in low-weight constraints of \(C'\) and thus could not be recovered with constant query complexity after the codeword will be corrupted. However, the extended code remains LDC. Claim 34 provides a formal proof to the above intuition.

\textbf{Claim 34} There exist constants \(q, \varepsilon, \delta > 0\) and a code \(C' \subseteq \mathbb{F}_2^{n+1}\) such that \(C'\) is a \((q, \varepsilon, \delta)\)-LDC, but for any constants \(q', \varepsilon', \delta' > 0\) it holds that \(C'\) is not a \((q', \varepsilon', \delta')\)-LCC.

\textbf{Proof.} Let \(C \subseteq \mathbb{F}_2^n\) be a \((q, \varepsilon, \delta)\)-LDC with \(\dim(C) = \omega(\log(n))\) (such codes exist, e.g. [32]). Claim 44 implies that there exists a word \(u \in \mathbb{F}_2^n\) such that \(\Delta(u, C^\bot) \geq \omega(1)\). Let \(C' \subseteq \mathbb{F}_2^{n+1}\) be such that \(C'|[n] = C\) and for every \(c' \in C'\) we have \((c')_{(n+1)} = \langle u, c'|[n]\rangle\), i.e., the first \(n\) coordinates of the code \(C'\) are identical to the code \(C\) and the last bit of the code \(C'\) is a sum of the bits indexed by \(\text{supp}(u)\).

Let \(q', \varepsilon', \delta' > 0\) be constants. We argue that there is no \(u' \in (C')^\bot\) such that \(n+1 \in \text{supp}(u')\) and \(|u'| \leq q' + 1 = O(1)\). Assume the contrary, and let \(u' \in (C')^\bot\) be such that \(n+1 \in \text{supp}(u')\) and \(|u'| \leq q' + 1\). This implies that the last symbol \((n+1)\) of the code \(C'\) is a sum of only \(q'\) symbols of \(C'|[n]\). Recall that \(C'|[n+1]\) was defined as a sum of symbols indexed by \(\text{supp}(u)\). We conclude that for all \(c' \in C'|[n]\) we have \(\langle u, c'\rangle = \langle u', c'\rangle\) which implies that \(\langle u + u', c'\rangle = 0\). Since \(C'|[n] = C\) we get that \(u + u' \in C^\bot\). This means \(\Delta(u, C^\bot) \leq |u'| \leq q' + 1\) contradicting our assumption that \(\Delta(u, C^\bot) \geq \omega(1) > q' + 1\).

We conclude that the last bit of \(C'\) is not involved in the constraints of weight at most \(q' + 1\). Hence any “local view” which contains only \(q'\) queries

\(^1\)\(\mathcal{C}\) need not be systematic but it can be easily converted into one as was stated.
but does not contain the last bit has no information about the last bit of $C'$ and thus $C'$ is not $(q',\varepsilon',\delta')$-LCC. However, $C'$ is a $(q,\varepsilon,\delta - \frac{1}{n+1})$-LDC because if $(\delta - \frac{1}{n+1}) \cdot (n+1) \leq \delta n$ symbols are corrupted then at most $\delta n$ symbols from the first $n$ symbols are corrupted. Recall that $C'|_{[n]} = C$ and $C$ is a $(q,\varepsilon,\delta)$-LDC. So, every message bit can be recovered with probability at least $\frac{1}{2} + \varepsilon$ and only $q$ queries.

\section{Two Transitivity with a Local Constraint implies Local Correction}

In this section we show (Theorem 35) that 2-transitive codes with local constraints imply LCCs and hence also LDCs. However, there exists a family of two-transitive codes with local constraints which is not locally testable, due to [42]. We conclude in Corollary 36 that a family of codes $\{C(n)\}_{n \in \mathbb{Z}}$ (explicitly) shown in [42] is locally correctable (and locally decodable) but is not locally testable.

\textbf{Theorem 35 (2-transitivity implies LCCs)} If $C \subseteq \mathbb{F}^n$ is a 2-transitive code such that $C_q^\perp \neq \emptyset$ then $C$ is a $(q-1,\frac{1}{6},\frac{1}{3q})$-LCC (LDC).

Moreover, there exists a family of 2-transitive codes $\{C(n)\}_{n \in \mathbb{Z}}$, where $C(n) \subseteq \mathbb{F}^n$ and $(C(n))^\perp \neq \emptyset$, which is not $(q',\varepsilon',1/7)$-LTC for all constants $q',\varepsilon' > 0$.

The following corollary follows immediately from Theorem 35.

\textbf{Corollary 36} There exists a family of linear codes $\{C_n\}_{n \in \mathbb{N}}$, where $C_n \subseteq \mathbb{F}^n$, which is a $(7,\frac{1}{6},\frac{1}{21})$-LCC (LDC) but is not $(q',\varepsilon',1/7)$-LTC for all constants $q',\varepsilon' > 0$.

Since by Claim 33 a $(q,\varepsilon,\delta)$-LCC is also a $(q,\varepsilon,\delta)$-LDC then Theorem 35 and the lower bound on the blocklength of LDCs by Kerenidis and de Wolf [53] imply the next corollary.

\textbf{Corollary 37} Let $C \subseteq \mathbb{F}^n$ be a 2-transitive linear code and $k = \dim(C)$. If $C_q^\perp \neq \emptyset$ then $n \geq \Omega(k/\log(k))^{1+1/(\frac{2}{3}-1)}$.

Notice that under the famous conjecture that LDCs have superpolynomial blocklength we have that 2-transitive codes with constant weight dual codewords have superpolynomial blocklength.
5.2.1 Proof of Theorem 35

Let us first recall the main result of Grigorescu et al. [42]. For additional details see [42]. Let $Tr$ be a trace function from $F_{2^n}$ to $F_2$, i.e.,

$$\text{Tr}(x) = x + x^2 + x^{2^2} + \ldots + x^{2^{n-1}}.$$ 

For positive integers $k < s$ let

$$F_{k,s}^* = \{ f : F_{2^n} \to F_2 \mid \exists \beta, \beta_0, \ldots, \beta_k \in F_{2^n} : f(x) = \text{Tr}(\beta + \beta_0 x + \sum_{i=1}^{k} \beta_i x^{2^{i+1}}) \}.$$ 

**Theorem 38 ([42])** Let $C = F_{k,s}^*$ be a linear code such that $k = \omega(1)$ and $s > 2k + 1$. Then $C$ is 2-transitive, $C_k^* \neq \emptyset$, but for all constants $q, \varepsilon > 0$ it holds that $C$ is not a $(q, \varepsilon, 1/7)$-LTC.

Now we prove Proposition 39. Then we prove Theorem 35. The proof of Proposition 39 is inspired by [5, Section 7].

**Proposition 39** Let $G$ be a 2-transitive group and $G(i) = \{ g \in G \mid g(i) = i \}$. Then $G(i)$ is a group of permutations such that for all $i' \neq i$ and $j' \neq i$ there exists $g \in G(i)$ such that $g(i') = j'$. Furthermore, for any $i' \neq i$ and $j' \neq i$ we have $\Pr_{g \in G(i)} [g(i') = j'] = \frac{1}{n-1}$.

**Proof.** Let $id \in G$ be the identity permutation, i.e., for all $j \in [n]$ we have $id(j) = j$. We know that $id \in G(i)$. Moreover, for every $g \in G(i)$ there exists $g^{-1} \in G(i)$. Furthermore, we know that if $h_1, h_2 \in G(i)$ then $h_1 \circ h_2 \in G(i)$. We conclude that $G(i)$ is a group of permutations.

For any $i' \neq i$ and $j' \neq i$ there exists $g \in G$ such that $g(i) = i$ and $g(i') = j'$ because $G$ is 2-transitive, and moreover, $g \in G(i)$.

We argue that for any $i' \neq i$ and $j' \neq i$ we have $\Pr_{g \in G(i)} [g(i') = j'] = \frac{1}{n-1}$. It is sufficient to show that for any $i', j'_1, j'_2 \neq i$ we have $\Pr_{g \in G(i)} [g(i') = j'_1] = \Pr_{g \in G(i)} [g(i') = j'_2]$.

Assume by a way of contradiction that $\Pr_{g \in G(i)} [g(i') = j'_1] > \Pr_{g \in G(i)} [g(i') = j'_2]$. Let $h \in G(i)$ such that $h(j'_1) = j'_1$. Since $G(i)$ is a group it
follows that random $g$ is distributed in $G(i)$ exactly as $hg$ is distributed in $G(i)$. Thus

$$\Pr_{g \in G(i)} [g(i') = j'_1] > \Pr_{g \in G(i)} [g(i') = j'_2] = \Pr_{g \in G(i)} [h(g(i')) = j'_1] = \Pr_{g \in G(i)} [g(i') = j'_1].$$

Contradiction.

Now we prove Theorem 35.

**Proof of Theorem 35.** Assume $C$ (and thus $C^\perp$) is invariant under a 2-transitive permutations group $G$ (note that $G \neq \emptyset$, e.g., $G$ contains the identity permutation). Let $u \in C_q^\perp$ (note $C_q^\perp \neq \emptyset$) and let $\text{supp}(u) = \{i_1, i_2, \ldots, i_q\}$. Hence for every $i \in [n]$ there exists $u' \in C_q^\perp$ such that $i \in \text{supp}(u')$, e.g., pick $g \in G$ such that $g(i_1) = i$ and let $u' = g(u)$.

We define the self-corrector of entry $i \in [n]$ ($SC_i$) which on word $w$

- picks random $g \in G$ such that $g(i) = i$
- queries all entries of $w|_{\text{supp}(g(u')) \setminus \{i\}}$
- and recovers the entry $w|_{(i)}$ by $-\frac{\sum_{j \in (\text{supp}(g(u')) \setminus \{i\})} w|_j \cdot g(u')|_j}{u'(i)}$.

This self-corrector queries only $q - 1$ entries and has perfect completeness, i.e., for all $c = (c_1, \ldots, c_n) \in C$ and $i \in [n]$ it holds that $SC_i[c]$ returns $c_i$. Assume the self-corrector $SC_i$ is given a word $w$ such that for some $c \in C$ we have $\delta(w, c) \leq \frac{1}{4q}$. Let $I = \text{supp}(w - c)$ and note that $|I| \leq \frac{1}{24q}$. Think of $I$ as a set of corrupted coordinates. Notice that if $SC_i$ picks $g \in G$, $g(i) = i$ such that $(\text{supp}(g(u')) \setminus \{i\}) \cap I = \emptyset$ then $SC_i$ recovers correctly the entry $w_i$, i.e., $SC_i[w] = c_i$. This is true because

$$-\frac{\sum_{j \in (\text{supp}(g(u')) \setminus \{i\})} w|_j \cdot g(u')|_j}{u'(i)} = -\frac{\sum_{j \in (\text{supp}(g(u')) \setminus \{i\})} c|_j \cdot g(u')|_j}{u'(i)} = c_i,$$

where the last equality follows because $\langle c, g(u') \rangle = 0$. In other words, whenever all the coordinates of $g(u')$ are correct but possibly the $i$'s coordinate, $SC_i$ recovers correctly the entry $w_i$. Proposition 39 implies that for $j \neq i$ and random $g \in G$ such that $g(i) = i$ we have that $g(j)$ is uniformly distributed
We conclude that
\[
\Pr_{g \in G, g(i) = i} [(\text{supp}(g(u')) \cap I) \setminus \{i\} \neq \emptyset] \leq \frac{(q - 1)|I|}{n - 1} \leq \frac{q|I|}{n}.
\]

It follows that the probability that \(\text{SC}_i\) picks \(g \in G, g(i) = i\) such that \(\text{supp}(g(u')) \cap I \subseteq \{i\}\) is at least \(1 - \frac{2}{3q} = \frac{2}{3}\). So, with probability at least \(2/3\) the self-corrector \(\text{SC}_i\) picks \(g \in G\) such that \(g(i) = i\) and \(|\text{supp}(g(u')) \setminus \{i\} \cap I| = 0\) and the correction succeeds.

The second part of Theorem 35 follows immediately from Theorem 38 that presents a construction of a family of 2-transitive codes \(\{C^{(n)}\}_{n}\) such that \(C^{(n)}_8 \neq \emptyset\) which is not \((q', \varepsilon', 1/7)\)-LTC for all constants \(q', \varepsilon' > 0\).

\[\Box\]

5.3 Non-sparse LDCs contain subcodes that are not LTCs

In this section we prove Theorem 40 that shows that non-sparse LDCs contain subcodes that are not LTCs. This demonstrates an important difference between LTCs and LDCs. It turns out that reducing dimension of LDCs remains LDCs (Corollary 43), however LTCs are not stable to the dimension reduction. This leads to an interesting observation that every non-sparse LDC has a large subcode which is not LTC (but is an LDC). Notice that non-sparse LDCs include Reed-Muller codes of low degree as well as subexponential LDCs that were recently discovered by Yekhanin [71] and Efremenko [32].

**Theorem 40 (Non-sparse LDCs contain non-LTCs as subcodes)**

Let \(q > 0\) and \(0 < \varepsilon, \delta < 1\) be constants. Then for every linear code \(C \subseteq \mathbb{F}^n\) that is a \((q, \varepsilon, \delta)\)-LDC with \(\dim(C) \geq \omega(\log(n))\) and any constants \(q', \varepsilon' > 0\) there exists a linear subcode \(C' \subset C\) such that \(\dim(C') \geq \omega(\log(n))\), \(C'\) is a \((q, \varepsilon, \delta)\)-LDC but \(C'\) is not a \((q', \varepsilon', \delta(C))\)-LTC.

In the following we show that every subcode of an LDC remains an LDC.

**Claim 41** Let \(C \subseteq \mathbb{F}^n\) be a linear code and a \((q, \varepsilon, \delta)\)-LDC. Assume that \(C' \subset C\) is a linear subcode of \(C\). Then \(C'\) is a \((q, \varepsilon, \delta)\)-LDC.
Proof. Assume $\mathcal{C}$ has the message space $\mathbb{F}^k$ and has the decoder $D$. Let $E_C$ be an encoding function for $\mathcal{C}$, i.e., $\mathcal{C} = \{E_C(x) \mid x \in \mathbb{F}^k\}$. Let $T \subset \mathbb{F}^k$ be a (linear) message space for $\mathcal{C}'$, i.e., $\mathcal{C}' = \{E_C(x') \mid x' \in T\}$.

For a linear subspace $M \subset \mathbb{F}^k$ and a subset $S \subseteq [k]$ we say that $S$ is an $M$-independent subset if there is no $u \in M^\perp \setminus \{0^k\}$ such that $\text{supp}(u) \subseteq S$. We say that $S \subseteq [k]$ is a maximal $M$-independent subset if $S$ is $M$-independent and for all $i \in [k] \setminus S$ it holds that $S \cup \{i\}$ is not $M$-independent subset. Note that if $S$ is a maximal $M$-independent subset then $|S| = \dim(M)$.

Recall that $T \subset \mathbb{F}^k$ is a linear subspace of $\mathbb{F}^k$. Let $T' \subseteq [k]$ be a maximal $T$-independent subset (clearly, such subset exists). Let $k' = \dim(T) = |T'|$ and without loss of generality (renaming of bits) we assume that $T' = [k']$. Note that $k' < k$. Let $E_T : \mathbb{F}^{k'} \to \mathbb{F}^k$ be a (linear) encoding function for the linear subspace $T$, i.e., $T = \{E_T(x') \mid x' \in \mathbb{F}^{k'}\}$. This linear encoding exists because $T' = [k']$ is a maximal $T$-independent subset. Note that for all $x' \in \mathbb{F}^{k'}$ we have $E_T(x')|_{[k]} = x'$, namely, the encoding function $E_T$ preserves all “input” bits. Then the code $\mathcal{C}'$ has message space $\mathbb{F}^{T'} = \mathbb{F}^{k'}$ that means $\mathcal{C}' = \{E_C(E_T(x')) \mid x' \in \mathbb{F}^{k'}\}$. We conclude that for every $x' \in \mathbb{F}^{k'}$ there exists $x \in \mathbb{F}^k$ such that $x|_{[k]} = x'$ and $E_C(x) = E_C(E_T(x'))$.

Recall that $\mathcal{C} \subseteq \mathbb{F}^n$ is a $(q, \varepsilon, \delta)$-LDC and has the decoder $D$. We argue that $\mathcal{C}'$ has the same decoder $D$. Let $w \in \mathbb{F}^n$ be $\delta$-close to $\mathcal{C}'$. But we know that $\mathcal{C}' \subset C$. That means there exists $x' \in \mathbb{F}^{k'}$ such that $\delta(w, E_C(E_T(x'))) \leq \delta$, and moreover, $\delta(w, E_C(x)) \leq \delta$, where $x = E_T(x') \in \mathbb{F}^k$. Note that for all $i \in [k']$ we have $x'_i = x_i$. The fact that $\mathcal{C}$ is a $(q, \varepsilon, \delta)$-LDC implies that for all $i \in [k'] \subset [k]$ the decoder $D$ recovers correctly the message entry $(x'_i = x_i)$ with probability at least $\frac{1}{2} + \varepsilon$. We conclude that $\mathcal{C}'$ is a $(q, \varepsilon, \delta)$-LDC.

**Remark 42** The special case of reducing dimension is a removing of columns from the generator matrix. E.g., given a code $\mathcal{C} = \{Gx \mid x \in \mathbb{F}^k\}$, where $G \in \mathbb{F}^{n \times k}$ is a generator matrix for $\mathcal{C}$. Let $G' \in \mathbb{F}^{n \times (k-1)}$ be obtained by removing the last column of $G$. Then $\mathcal{C}' = \{G'x \mid x \in \mathbb{F}^{k-1}\}$ is a linear subcode of $\mathcal{C}$ and $\dim(\mathcal{C}') = \dim(\mathcal{C}) - 1$. In this case message space of $\mathcal{C}'$ is $\mathbb{F}^{k-1}$, while the message space of $\mathcal{C}$ is $\mathbb{F}^k$.

**Corollary 43** (LDCs are stable for dimension reduction) Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code and a $(q, \varepsilon, \delta)$-LDC. Take any sequence of linear subcodes: $\mathcal{C}_1 \subset \mathcal{C}_2 \subset \ldots \subset \mathcal{C}_f = \mathcal{C}$. Then for all $i \in [f]$ it holds that $\mathcal{C}_i$ is a $(q, \varepsilon, \delta)$-LDC.
Now we show the auxiliary claim and then prove Theorem 40.

**Claim 44** Let $C \subseteq \mathbb{F}^n$ be a linear code such that $\dim(C) = \omega(\log(n))$. Then there exists $w \in \mathbb{F}^n$ such that $\Delta(w, C^\perp) \geq \omega(1)$.

**Proof.** For integer $R$ let $V(n, R) = \sum_{i=0}^{R} \binom{n}{i} \cdot (|\mathbb{F}| - 1)^i$ be the volume of a sphere in $\mathbb{F}^n$ of radius $R$. Let $k = \dim(C) \geq \omega(\log(n))$ and $S = C^\perp$. Then $\dim(S) = n - k$ and $|S| = |\mathbb{F}|^{n-k} = |\mathbb{F}|^n / |\mathbb{F}|^k$. Recall that a covering radius of a code $S$ is $R_S = \max_{w \in \mathbb{F}^n} \Delta(w, S)$, i.e., the largest Hamming distance of any word in $\mathbb{F}^n$ from $S$. Note that if $R_S$ is constant then $V(n, R_S)$ is polynomial in $n$, and vice versa, if $V(n, R_S)$ is super-polynomial in $n$ then $R_S$ goes to infinity with $n$. Assume by a way of contradiction that there exists a constant $t > 0$ such that for all $w \in \mathbb{F}^n$ we have $\Delta(w, S) \leq t$, i.e., $R_S \leq t = O(1)$.

The covering radius bound\(^2\) states that

$$|S| \cdot V(n, R_S) \geq |\mathbb{F}|^n.$$ 

But then $V(n, R_S) \geq |\mathbb{F}|^k$, where $k \geq \omega(\log(n))$. Hence $V(n, R_S)$ must be super-polynomial in $n$, and $R_S \geq \omega(1)$. Contradiction. \(\blacksquare\)

**Proof of Theorem 40.** Claim 44 implies that there exists $u \in \mathbb{F}^n$ such that $\Delta(u, C^\perp) \geq \omega(1) > q'$. Let $S = \text{span}(C^\perp \cup \{u\})$. Note that for all $u' \in S$ if $|u'| \leq q'$ then $u' \in C^\perp$. Let $C' = S^\perp$ and then $C'^\perp = S$. We have $C^\perp \subseteq C'^\perp$, $C' \subset C$ and in particular, $\dim(C') = \dim(C) - 1$. We argue that $C'$ is not $(q', \varepsilon', \delta(C))$-LTC. We have $c \in C \setminus C'$ since $C'' \subset C$. However $c \perp (C')_{\leq q'}$ because $(C')_{\leq q'} \subseteq C^\perp$ by construction. Hence $c$ is $\delta(C)$-far from $C'$ but will be accepted with probability 1 by any $q'$-query tester of $C'$.

We conclude that $C' \subset C$ and $\dim(C') = \dim(C) - 1$ but $C'$ is not $(q', \varepsilon', \delta(C))$-LTC. Claim 41 guarantees that $C'$ is a $(q, \varepsilon, \delta)$-LDC since $C'$ is a linear subcode of $C$. The Theorem follows. \(\blacksquare\)

\(^2\)For any code $C \subseteq \mathbb{F}^n$ (whether linear or not) the covering bound states that the covering radius $R$ of $C$ relates to $n$ and $|C|$ by $|C| \cdot V(n, R) \geq |\mathbb{F}|^n$.
5.3.1 Non-sparse LDCs contain many subcodes which are not uniform-LTCs

Theorem 40 shows that every non-sparse LDC $C$ contains a single sequence of linear subcodes $C_1 \subseteq C_2 \subseteq C_3 \subseteq \cdots \subseteq C$ which are all not LTCs. In the following (Theorem 45) we show that every non-sparse LDC and every long enough sequence of linear subcodes $C_1 \subseteq C_2 \subseteq C_3 \subseteq \cdots \subseteq C_\ell = C$ contains at least one subcode $C_i$ which is not uniform LTC.

**Theorem 45** Let $q, q', \varepsilon, \varepsilon', \delta > 0$ be constants. Let $C \subseteq \mathbb{F}^n$ be a linear code such that $\dim(C) \geq \omega(\log(n))$. Then every sequence of $\ell$ linear subcodes $C_1 \subseteq C_2 \subseteq C_3 \subseteq \cdots \subseteq C_\ell = C$, where $\ell \geq (q' \log(n))/\varepsilon'$, contains at least one code $C_i$ which is not $(q', \varepsilon', \delta(C)/2)$-uniform LTC. Moreover, if $C$ is a $(q, \varepsilon, \delta)$-LDC then all linear subcodes $C_i$ in the sequence are $(q, \varepsilon, \delta)$-LDCs.

Note that if $\dim(C) \geq \omega(\log(n))$ then $C$ contains sequences of subcodes of length $\omega(\log(n))$. Now we prove two simple claims that will be useful in the proof of Theorem 45.

**Claim 46** Let $C \subseteq \mathbb{F}^n$ be a linear code. Moreover, let a linear code $C' \subseteq C$ be a $(q, \varepsilon, \delta(C))$-uniform LTC. Then $|C_{\leq q}^\perp| \leq (1 - \varepsilon)|C_{\leq q}^\perp|$. 

**Proof.** Let $D$ be the uniform distribution over $C_{\leq q}^\perp$. We know that $C_{\leq q}^\perp \subseteq C_{\leq q}^\perp$. Consider any $w \in C \setminus C'$ (note that $C \setminus C' \neq \emptyset$). Since $C'$ is a $(q, \varepsilon, \delta(C))$-uniform LTC and $\delta(w, C') \geq \delta(C)$ it holds that $\Pr_{u \sim D}[\langle u, w \rangle \neq 0] \geq \varepsilon$. Notice that if for $u \in C_{\perp}^\perp$ it holds that $\langle u, w \rangle \neq 0$ then $u \notin C_{\perp}^\perp$. So, there are at least $\varepsilon|C_{\leq q}^\perp|$ words in $C_{\leq q}^\perp$ that are not in $C_{\leq q}^\perp$. Thus we have $|C_{\leq q}^\perp| \leq (1 - \varepsilon)|C_{\leq q}^\perp|$. 

**Claim 47** Let $q', \varepsilon' > 0$ be constants. Let $\ell$ be the minimal integer such that $\ell \geq \frac{q' \log n}{\varepsilon'}$ and $C \subseteq \mathbb{F}^n$ be a linear code such that $\dim(C) > \frac{q' \log n}{\varepsilon'}$. Then at least one of the codes in the sequence of the linear subcodes $C_1 \subseteq C_2 \subseteq \cdots \subseteq C_\ell = C$ is not $(q', \varepsilon', \delta(C)/2)$-uniform LTC.

**Proof.** Note that for all $i \in [\ell - 1]$ we have $\dim(C_i) < \dim(C_{i+1})$. Assume that for all $i \in [\ell]$, $C_i$ is a $(q', \varepsilon', \delta(C)/2)$-uniform LTC. If $(C_\ell)_{\leq q}^\perp = \emptyset$ then for any word $w \in \mathbb{F}^n$ such that $|\text{supp}(w)| = \delta(C)/2$ (i.e., $\delta(w, C_\ell) \geq \delta(C)/2$) it holds that $w \perp (C_\ell)_{\leq q}^\perp$. Contradiction. We conclude that $|(C_\ell)_{\leq q}^\perp| \geq 1$. 

52
Claim 46 implies that for all $i \in [\ell - 1]$ we have that $|\langle C_{i+1}\rangle_{\leq q'}| \leq (1 - \varepsilon') \cdot |\langle C_i\rangle_{\leq q}|$. Then it holds that

$$|\langle C_{\ell}\rangle_{\leq q'}| \leq (1 - \varepsilon')^\ell \cdot |\langle C_1\rangle_{\leq q'}| < e^{-\varepsilon' \ell} \cdot n^q' \leq 1,$$

where $\ell \geq \frac{q' \log n}{\varepsilon'}$. We conclude that $\langle C_{\ell}\rangle_{\leq q'} = \emptyset$. Contradiction. ■

**Proof of Theorem 45.** Assume $C_1 \subset C_2 \subset \ldots \subset C_\ell = C$, where $\ell \geq (q' \log(n))/\varepsilon'$. Claim 47 says that at least one of the codes in the sequence is not $(q', \varepsilon', \delta(C)/2)$-uniform LTC. Corollary 43 implies that for all $i \in [\ell]$ the code $C_i$ is a $(q, \varepsilon, \delta)$-LDC. ■
Chapter 6

Towards lower bounds on Locally Testable Codes via density arguments
6.1 Introduction

This chapter of the thesis is motivated by one of the most important open problems regarding locally testable codes (LTCs), whether there exists an asymptotically good family of LTCs with constant query complexity.

To avoid repeating what is recounted in the previous chapters, it suffices to say that for all the work that has gone into the study of LTCs, our understanding of their rate is very limited. The only negative results on LTCs rate concern special families of codes testable with just 2-queries [10, 44, 56, 55], random low density parity check (LDPC) codes [12], cyclic codes [5], solvable codes [52] and affine-invariant codes [16]. In fact, we cannot even rule out the existence of binary LTCs meeting the Gilbert-Varshamov bound (which is the best known rate for codes without any local testing restriction). So, for all we know, the strong testability requirement of LTCs may not “cost” anything extra over LDPC codes!

We suggest a strategy to disprove the existence of an asymptotically good family of linear LTCs. Without loss of generality we may deal with the case of query complexity 3 (cf. Theorem 99). Our proof-strategy goes by way of contradiction and relies on proving the following pair of conjectures.

- If $C$ is an asymptotically good 3-query LTC then $C$ has a super-linear number of dual codewords of weight at most 3.
- If $C$ is an asymptotically good 3-query LTC and has a super-linear number of dual codewords of weight at most 3 then rate($C$) = $o(1)$.

The result of Ben-Sasson et al. [11] seems to lead in the direction of proving the first item as it shows that all LTCs have more small-weight dual codewords than what is needed to characterize the code and the small-weight dual codewords display nontrivial dependencies among them. In this paper we make initial progress on the second item and show that a broad family of 3-query LTCs (including all “base constructions” of LTCs) cannot have both constant rate and a super-linear number of dual codewords of weight at most 3.

Roughly speaking, LTCs are invariably constructed by starting with a decent “base-construction” of an LTC (such as a Hadamard, Reed-Muller, or constant-blocklength code) and modifying it by various techniques like repetition [63], concatenation [2, 3], tensoring [15], gap-amplification [26],...
taking direct-products [27, 46] and PCPP-composition [9, 28]. These operations improve the LTC-related parameters of the code, they increase soundness and/or reduce query complexity but none of them increases rate. In fact, the improvement in LTC-related parameters of the afore-mentioned operations comes at the price of reduced rate. So if asymptotically good LTCs are to be constructed one should start with a “base-construction” that is asymptotically good, or come up with a new set of LTC-related techniques that do not decrease code-rate.

Looking into known “base-constructions” of \( q \)-query LTCs they all share a few properties that we formalize in this paper. First, they are \( q \)-regular, i.e., every codeword-bit sees the same number of dual codewords of weight \( q' \leq q \) (see Definition 53). Second, they are all \( q \)-dense, by which we mean that the number of dual codewords of weight at most \( q \) is super-linear in the code blocklength. Indeed, a popular belief (stated formally in Conjecture 51) says that all \( q \)-query LTCs are \( q \)-dense (see Definition 48).

Our main result is that families of 3-dense and 3-regular LTCs cannot be asymptotically good. We bound the rate of the code as a function of 3-density and show that even arbitrarily slowly growing 3-density implies vanishing rate (cf. Theorem 54 and Corollary 55). We then put forth a conjecture stating that all 3-query LTCs are dense and have a “natural” distribution of dual codewords of weight \( \leq 3 \) (Conjecture 58) and show that under this conjecture there are no asymptotically good families of LTCs whatsoever (cf. Theorem 56 and Corollary 59). We go on to say that regular codes strongly generalize symmetric\(^1\) LTCs, i.e., LTCs which are invariant under a group of permutations that is 1-transitive. A subclass of these codes — so-called “2-transitive” codes — was suggested by Alon et al. [1] as possibly being locally testable, and this family was first studied systematically for the special case of affine-invariant codes by Kaufman and Sudan [50]. As a corollary of our main results we show that 3-query symmetric LTCs with super-constant density are not asymptotically good.

**Improved rate bounds for weak 2-query LDCs** Our analysis of 3-query LTCs relies on a new upper bound on the rate of families of locally decodable codes (LDCs) with a rather weak requirement on their decoding

\(^1\)These codes are called symmetric since every coordinate of the code participates in a similar set of dual codewords.
capabilities described next. This LTC-to-LDC reduction is especially interesting in light of the fact that LTCs and LDCs seem to be two very different kinds of codes (cf. Chapter 5).

Recall that $q$-query LDCs allow to recover each message entry with constant probability by reading only $q$ entries of the codeword even if a “large” number of codeword bits are adversely corrupted. Best known upper bounds on the rate of linear $q$-query LDCs [69, 70] and non-linear $q$-query LDCs [53] go by reducing the problem to that of showing rate bounds for 2-query LDCs. And the best rate bounds for 2-query linear LDCs follow from the so-called “combinatorial lemma” of Goldreich et al. [38, Lemma 3.3] (see also [31]). Our main technical contribution is an improvement of this combinatorial lemma as described next. The combinatorial lemma of [38] bounds the rate of a 2-query LDC in terms of the corruption parameter — the number of bits which can be adversarially corrupted. All things considered, as the corruption parameter decreases, it should get easier to construct LDCs (because the adversary is more restricted) and consequently is should get harder to prove upper bounds on the rate of such LDCs. Indeed, the rate bound given by the combinatorial lemma becomes trivial when the number of corrupted bits is roughly logarithmic in the blocklength of the code. Our improved combinatorial lemma (Lemma 63) gives nontrivial rate bounds for any super-constant corruption parameter (see Section 6.2.2).

Two additional remarks regarding our combinatorial lemma should be made. First, given that state-of-the-art bounds on rate of $q$-query LDCs for $q \geq 3$ rely on rate bounds 2-query LDCs with a sublinear number of errors [69, 70] shows that proving rate bounds for smaller values should result in improved bounds for $q$-query LDCs even for larger values of $q$. Second, the recent work of Dvir [30] shows that proving sufficiently strong lower bounds on locally decodable codes which can be corrected from a sublinear number of corruptions would result in explicit constructions of rigid matrices, giving further motivation for our lemma.

We end with a few words on our proof of the rate-bound on 2-query LDCs (Lemma 63) and how it differs from the proof method of Goldreich et al in [38]. They provided two different proofs, the first uses an isoperimetric inequality statement regarding the hypercube and the second is an information-theoretic argument due to Alex Samorodnitsky. Our proof goes by removing carefully selected columns from the generating matrix of a lo-
cally decodable code. This removal, we argue, partitions the rows of the matrix into sets of identical rows. We study how the sets identical rows grow in size with the removal of additional columns and perform a careful amortized analysis of this process (see Section 6.4 for details).

**Organization of the chapter.** In the following section we state our main results (Theorems 54, 56, 65). We prove our main results on LTCs in Section 6.3. We go on to prove the main technical Lemma 63 in Section 6.4. Finally, in Section 6.5 we prove our improved bound for 2-query LDCs over arbitrary fields (Theorem 65).

### 6.2 Main Results

Our main motivation is the study of rate limitations of families of LTCs and the results regarding this question are presented in Section 6.2.1. The main tool used in our proofs is a new bound on the rate of weak 2-query LDCs. We present this bound and discuss its implications to LDCs in Section 6.2.2. We start by stating the popular belief about density of locally testable codes and for this we need first to define the notion of “dense” codes.

The results presented in this section deal with linear codes over the binary field. These results can be extended to any finite field but for simplicity we prefer to state them for the binary case.

**Definition 48 (q-density)** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and $q > 0$. Let $\Delta_q(C) = |C_{\leq q}^\perp|$ be the number of dual codewords of weight at most $q$ and $\Delta_{q,i}(C) = \left| \{ u \in C_{\leq q}^\perp | i \in \text{supp}(u) \} \right|$ be the number of small-weight dual codewords that “touch” the index $i$. The $q$-density of $C$ is defined as $\sigma_q(C) = \frac{\Delta_q(C)}{n}$.

Now we define the notion of “sunflower”.

**Definition 49 (sunflower)** For a linear code $C \subseteq \mathbb{F}_2^n$ and $q > 0$ we say that a set $M \subseteq C_{\leq q}^\perp$ (respectively, $M \subseteq C_q^\perp$) is a $(\leq q, i, C)$-sunflower (respectively, $(q, i, C)$-sunflower) if for every $u_1 \neq u_2 \in M$ it holds that $\text{supp}(u_1) \cap \text{supp}(u_2) = \{i\}$.

We let

$$M_{\leq q,i}(C) = \max \{ |M| \mid M \text{ is a } (\leq q, i, C) - \text{sunflower} \} \quad \text{and}$$
\[ M_{q,i}(C) = \max \{|M| \mid M \text{ is a } (q,i,C) - \text{sunflower}\}, \]

where \(|M|\) denotes the size of \(M\).

**Remark 50** The repetition code \(C = \{0^n, 1^n\}\) is a 3-query LTC but \(|C_3^\perp| = 0\). This example shows that the above definition of density which counts all words of weight at most \(q\) should not be replaced with the finer definition which counts all words of weight exactly \(q\).

Popular belief says that \(q\)-query LTCs have a superlinear number of dual codewords of weight at most \(q\) (e.g., see Chapter 4). Recall that to rule out the existence of asymptotically good LTCs it is sufficient to rule out 3-query asymptotically good LTCs (cf. Theorem 99). The main point of this paper is to show that if the following conjecture is proven to be true then there are no asymptotically good natural families of LTCs.

**Conjecture 51 (LTCs are dense)** Let \(\varepsilon, \delta > 0\) be constants. Then there exists a function \(\sigma : \mathbb{N} \rightarrow \mathbb{N}\) such that \(\sigma(n) = \omega_{\varepsilon, \delta}(1)\) such that the following condition holds.

\[ \text{If } C \subseteq \mathbb{F}_2^n \text{ is a } (3, \varepsilon, \delta/3)\text{-LTC and } \delta(C) \geq \delta \text{ then } \sigma_3(C) \geq \sigma(n). \quad (6.1) \]

**Remark 52** To rule out the existence of asymptotically good families of LTCs it is sufficient to make the weaker assumption that the family of codes in the conjecture above is asymptotically good and then prove (6.1) for such families. Indeed, all our results regarding asymptotically good codes work under this weaker assumption.

The results presented in Chapter 4 may be useful in this context as they showed that LTCs have many linear dependencies in their small weight dual codewords and this number increases with the rate of the code.

6.2.1 Dense natural and regular LTCs cannot be asymptotically good

To state our main results about LTCs we formalize the notion of \(q\)-regular, and natural, codes. (Recall that we have argued in the introduction that all base LTCs are natural, and even regular.) We note that \(q\)-regular codes are similar to regular LDPC codes introduced by Gallager [34, 35]. The main
difference is that regular LDPC codes are defined by the regular structure of
the parity check matrix, while our $q$-regular codes assume a regular structure
in the subspace of all dual codewords of weight at most $q$. Later on we shall
argue that the class of regular codes strictly contains the class of symmetric
codes, suggested as candidate LTCs in [1] and first studied systematically
in [50].

The notion of a natural code should be viewed as a weaker definition of
regularity. It does not require that all codeword coordinates participate in
the exact same number of small-weight dual words. Rather, it suffices that
an independent set of indices (a notion we define next) each participate in
a large number of dual words of small weight. We say that $I \subseteq [n]$ is a set
of independent indices of a code $C \subseteq \mathbb{F}_2^n$ if $C\vert_I = \mathbb{F}^I$, or equivalently, there
is no $u \in C^\perp$ such that $\text{supp}(u) \subseteq I$. It can be easily verified that $C$ has at
least one set of independent indices of size $\dim(C)$.

Definition 53 (Regular and natural codes) We say that a code $C \subseteq \mathbb{F}_2^n$ is
$q$-regular if for all $q' \leq q$ and $i, j \in [n]$, we have
$$\left| \left\{ u \in C^\perp_{q'} \mid i \in \text{supp}(u) \right\} \right| = \left| \left\{ u \in C^\perp_{q} \mid j \in \text{supp}(u) \right\} \right|.$$ We say that $C$ is $(\alpha, \Delta)$-natural if there exists a set of independent in-
dices $I \subseteq [n]$ such that $|I| \geq \alpha \cdot \dim(C)$ and for every $i \in I$ it holds that
$$M_{\leq 3,i}(C) \geq \Delta.$$ Our first main result demonstrates a tight relation between the density
and the rate of 3-regular codes.

Theorem 54 (3-density limits rate of regular codes) Let $C \subseteq \mathbb{F}_2^n$ be
a 3-regular code such that $\sigma_3(C) \geq 2$. Then
$$\text{rate}(C) \leq \frac{6 \log(\sigma_3(C)) + 2}{\sqrt{\sigma_3(C')}}.$$ Spielman [66] suggested to use dense regular expander codes for con-
structing LTCs. The next corollary says that dense 3-regular codes cannot
be asymptotically good even without any expansion assumptions. Furthermore,
this corollary limits the rate of 3-regular LTCs under Conjecture 51.

Corollary 55 (No asymptotically good regular 3-query LTCs) Let
$C = \{C_n\}_{n \in \mathbb{Z}}$ is a family of 3-regular codes, where $C_n \subseteq \mathbb{F}_2^n$. 60
• If $\sigma_3(C_n) = \omega(1)$ then
  \[
  \text{rate}(C_n) \leq \frac{6 \log(\sigma_3(C_n)) + 2}{\sqrt{\sigma_3(C_n)}} = o(1).
  \]

• Let $\varepsilon, \delta > 0$ be constants. Under Conjecture 51, if $C_n \subseteq \mathbb{F}_2^n$ is a $(3, \varepsilon, \delta/3)$-LTC and $\delta(C_n) \geq \delta$ then $\text{rate}(C_n) = o(1)$.

**Proof.** The first bullet follows from Theorem 54. For the second bullet, assume the contra-positive, i.e., $\text{rate}(C_n) \geq \rho$ for some constant $\rho > 0$. Conjecture 51 says that $\sigma_3(C_n) = \omega(1)$. Theorem 54 then implies that $\text{rate}(C_n) = o(1)$. ■

**Natural codes** Next we present limits on the rate of natural LTCs. We then present a believable conjecture that is stronger than Conjecture 51 and show that it implies there are no asymptotically good LTCs. We now show that natural LTCs have bounded rate.

**Theorem 56 (Natural 3-query LTCs have bounded rate)** Let $C \subseteq \mathbb{F}_2^n$ be $(\alpha, \Delta)$-natural. Then
  \[
  \text{rate}(C) \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta)}{\Delta/2}.
  \]

**Corollary 57 (No asymptotically good natural dense codes)** Let $\alpha > 0$ be constant and $\Delta : \mathbb{N} \rightarrow \mathbb{N}$ be a function such that $\Delta(n) = \omega(1)$. Let $C = \{C_n\}_{n \in \mathbb{Z}}$ be a family of codes, where $C_n \subseteq \mathbb{F}_2^n$ is an $(\alpha, \Delta(n))$-natural code. Then
  \[
  \text{rate}(C_n) \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta(n))}{\Delta(n)/2} = o(1).
  \]

The following conjecture says that if $C$ is an asymptotically good 3-query LTC then $C$ is a natural code with super-linear density. Note that Conjecture 51 implies that 3-query LTCs have a superlinear number of dual codewords of weight at most 3.

**Conjecture 58 (3-query LTCs are natural and dense)** Let $\varepsilon, \delta, \rho > 0$ be constants. Then there exists a function $\Delta : \mathbb{N} \rightarrow \mathbb{N}$ such that $\Delta(n) = \omega_{\varepsilon, \delta, \rho}(1)$ and constant $\alpha > 0$ which depends only on $\varepsilon, \delta, \rho$ such that the
following condition holds. If $C \subseteq \mathbb{F}_2^n$ is a $(3, \varepsilon, \delta/3)$-LTC, where $\delta(C) \geq \delta$ and rate$(C) \geq \rho$ then $C$ is $(\alpha, \Delta(n))$-natural.

Under this conjecture we can rule out the existence of asymptotically good LTCs altogether.

**Corollary 59 (No Asymptotically good LTCs)** Under Conjecture 58 there is no family of asymptotically good 3-query LTCs. Consequently (cf. Theorem 99) there is no asymptotically good family of linear LTCs.

**Proof.** Assume the contrary, i.e., there exists a family $C = \{C_n\}_{n \in \mathbb{Z}}$, where $C_n \subseteq \mathbb{F}_2^n$ is a $(3, \varepsilon, \delta/3)$-LTC, $\delta(C_n) \geq \delta$ and rate$(C_n) \geq \rho$ for some constants $\varepsilon, \delta, \rho > 0$. Conjecture 58 implies that there exist a function $\Delta(n) : \mathbb{N} \to \mathbb{N}$ such that $\Delta(n) = \omega_{\varepsilon, \delta, \rho}(1)$ and constant $\alpha > 0$ which depends only on $\varepsilon, \delta, \rho$ such that $C_n$ is $(\alpha, \Delta(n))$-natural.

Theorem 56 implies that rate$(C_n) \leq \frac{\dim(C_n)}{\beta n} \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta(n))}{\Delta(n)/2} \leq o(1)$. Contradiction. ■

**LTCs vs. LDCs** In Chapter 5 it was shown that locally testable and locally decodable codes are two very different kinds of error-correcting codes and, in particular, some LDCs are not LTCs and some LTCs are not LDCs. In spite of this fact we show that Conjecture 58 implies the following corollary, which says that 3-query LTCs contain large subcodes that are 2-query “weak” LDCs.

**Corollary 60 (3-query LTCs contain 2-query “weak” LDCs)** Let $\varepsilon, \delta, \rho > 0$ be constants. Then, under Conjecture 58, there exists a function $\Delta : \mathbb{N} \to \mathbb{R}$ such that $\Delta(n) \cdot n = \omega_{\varepsilon, \delta, \rho}(1)$ and constant $\alpha > 0$ which depends only on $\varepsilon, \delta, \rho$ such that the following condition holds. If $C \subseteq \mathbb{F}_2^n$ is a $(3, \varepsilon, \delta/3)$-LTC, where $\delta(C) \geq \delta$ and rate$(C) \geq \rho$ then there exists a linear subcode $C' \subseteq C$ such that $C'$ is a $(2, 1/6, \Delta(n))$-LDC and $\dim(C') \geq \alpha \cdot \dim(C)$.

**Proof.** Let $C \subseteq \mathbb{F}_2^n$ be a $(3, \varepsilon, \delta/3)$-LTC, where $\delta(C) \geq \delta$ and rate$(C) \geq \rho$. Conjecture 58 implies that $C$ is $(\alpha, \Delta'(n))$-natural, where $\alpha > 0$ is a constant and $\Delta'(n) = \omega(1)$. This implies the existence of a subset of independent indices $I \subseteq [n]$ such that $|I| \geq \alpha n$ and for every $i \in I$ we
have a \((\leq 3, i, C)\)-sunflower \(M_i\) of size at least \(\Delta'(n)\). Let \(C'\) be a linear subcode of \(C\) whose message bits indices are \(I\). Clearly, for every \(i \in I\) we have a \((\leq 3, i, C')\)-sunflower of size at least \(\Delta'(n)\) (the same sunflower \(M_i\)). This implies that \(C'\) is a \((2, 1/6, \Delta(n))\)-LDC, where \(\Delta(n) = \frac{\Delta'(n)/3}{n}\) and note that \(\Delta(n) \cdot n = \omega(1)\). To see this let \(i \in I\) be an index of a message bit. Let \(M_i\) be a \((\leq 3, i, C)\)-sunflower. Consider a decoder of message bit \(i\), which picks (uniformly) random \(u \in M_i\) (recall that \(i \in \text{supp}(u)\)) and on the given word \(w \in \mathbb{F}_2^n\) (a corrupted version of some \(c \in C\)) recovers the \(i\)th message symbol of \(c\) by \(\langle w|_{S}, u|_{S} \rangle\), where \(S = \text{supp}(u) \subseteq \{i\}\). If at most \(\Delta(n) \cdot n\) bits of \(c\) was corrupted then at most \(\Delta(n) \cdot n = \Delta'(n)/3\) vectors of \(M_i\) lead to the wrong recovery of the \(i\)th message bit of \(C\). Hence the recovery succeeds with probability at least \(2/3 = 1/2 + 1/6\). 

Symmetric codes are regular We end this section by focusing on an interesting class of regular codes that has been investigated intensively in recent years (cf. [1, 50]) — the class of symmetric, or 1-transitive, LTCs.

Let \(G\) be a group of permutations over \([n]\). For \(\pi \in G\) and \(w = (w_1, w_2, ..., w_n) \in \mathbb{F}_2^n\) with some abuse of notation we let \(\pi(w) = (w_{\pi^{-1}(1)}, ..., w_{\pi^{-1}(n)})\) be a \(\pi\)-permuted word. Note that since \(G\) is a group and \(\pi \in G\) we have \(\pi^{-1} \in G\). A linear code \(C\) is invariant under \(G\) if for every \(\pi \in G\) and \(c \in C\) we have \(\pi(c) \in C\). Note that if \(C\) is invariant under \(G\) then also \(C^\perp\) is invariant under \(G\). \(G\) is called 1-transitive if for all \(i, j \in [n]\) we have \(\pi \in G\) such that \(\pi(i) = j\). A linear code \(C\) is 1-transitive if it is invariant under some 1-transitive permutation group \(G\).

All relevant LTCs based on the “invariance” approach are regular. This is true since 1-transitivity is a minimal possible requirement for such LTCs and all 2-transitive codes, affine-invariant codes, linear invariant codes etc. are 1-transitive (for further information see [50]). It is not hard to show that 1-transitive codes are \(q\)-regular for every \(q > 0\) (cf. Claim 101) and this leads to the following corollary. Moreover, the next corollary shows that under Conjecture 51 there is no asymptotically good 1-transitive 3-query LTCs.

**Corollary 61 (Dense 1-transitive LTCs are not asymptot. good)**

Let \(C = \{C_n\}_{n \in \mathbb{Z}}\) be a family of codes, where \(C_n \subseteq \mathbb{F}_2^n\) is 1-transitive.
• If $\sigma_3(C_n) = \omega(1)$ then
  \[
  \text{rate}(C_n) \leq \frac{2\log(\sigma_3(C_n)) + 2}{\sqrt{\sigma_3(C_n)}} = o(1).
  \]

• Under Conjecture 51, if $C_n$ is a $(3, \varepsilon, \delta/3)$-LTC and $\delta(C_n) \geq \delta$ then $\text{rate}(C_n) = o(1)$.

**Proof.** The first bullet follows from Claim 101 (which implies that $C_n$ is 3-regular) and Corollary 55. The second bullet follows from the first bullet.

\[\blacksquare\]

### 6.2.2 Limiting the rate of weak 2-query LDCs

The proof of our main theorems regarding LTCs, presented in the previous section, follow from an improved version of the rate-bound on 2-query LDCs due to [38]. In this section we present this improved version and discuss its corollaries for locally decodable codes.

The following lemma is due to Goldreich et al. [38], stated there as Lemma 3.3. This lemma had a crucial role in proving lower bounds for LDCs (see, e.g., the results of Goldreich et al. [38], Dvir and Shpilka [31], Obata [62], Woodruff [69, 70]). The lemma is used as a combinatorial core which analyzes the relation between the rate of a LDC and the number of tuples used in the decoding.

Let us first recall the definition of a singleton vector: let $e_i = 0^{i-1}10^{k-i}$ for $i \in [k]$. For a matrix $G$ we let $G_i$ denote the $i$th row of $G$. In this section we think of $G \in \mathbb{F}_2^{n \times k}$ as a generator matrix for some 2-query LDC $C$. We also relate $k$ to $\dim(C)$ and $n$ to the blocklength of $C$.

**Lemma 62** (Lemma 3.3 in [38]) Let $G \in \mathbb{F}_2^{n \times k}$ be a matrix and $\Delta \geq 1$. Suppose for every $i \in [k]$ there is a matching $M_i \subseteq \binom{[n]}{2}$, i.e., a set of disjoint pairs of indices $(j_1, j_2)$, such that $G_{j_1} + G_{j_2} = e_i$. Moreover, suppose it holds that $\sum_{i=1}^{k} |M_i| \geq \Delta$. Then $k \leq \frac{n(\log n)}{2\Delta}$.

Goldreich et al. [38] prove the lemma using the assumption that $\sum_i |M_i|$ is large. They go on to point out that in the context of LDCs one has a stronger assumption, namely, that every single matching $M_i$ is large but this stronger assumption is not used. The following lemma, which is the main
technical contribution of this paper, improves upon Lemma 62 by using the stronger assumption on the size of individual matchings.

**Lemma 63 (Main technical lemma)** Let $G \in \mathbb{F}_2^{n \times k}$ be a matrix and $\Delta \geq 1$. Suppose for every $i \in [k]$ there is a matching $M_i \subseteq \binom{[n]}{2}$, i.e., a set of disjoint pairs of indices $(j_1, j_2)$, such that $G_{j_1} + G_{j_2} = e_i$. Moreover, suppose for every $i \in [k]$ it holds that $|M_i| \geq \Delta$. Then $k \leq \frac{n(\log \Delta) + n}{\Delta}$.

Notice that this lemma implies Lemma 62 and works for smaller densities. In particular, for any super-constant function $\Delta(n) \geq \omega(1)$ our lemma gives $\frac{k}{n} = o(1)$ but for $\Delta(n) \leq (\log n)/2$ Lemma 62 gives no nontrivial bounds.\(^2\)

In Section 6.4 we prove Lemma 63 and in Section 6.5.1 we generalize it to arbitrary fields. The tightness of Lemmata 63, 62 is shown in Section 6.4.4.

Next we use Lemma 63 to limit the rate of weak 2-query LDCs, i.e., LDCs that allow correct decoding of message bits under the weak assumption that a super-constant (but sublinear) number of codeword bits are corrupted. We believe that Theorem 65 might be useful for improving the existing rate bounds of $q$-query locally decodable codes with $q \geq 3$ and subconstant corruption parameter $\delta = o(1)$. The point is that the best known lower bounds for $q$-query LDCs ($q \geq 3$) are obtained by way of reduction to 2-query LDC (with worse parameters) and applying the lower bound for 2-query LDC (see e.g. [69], [70]). However, the parameter $\delta$ of an LDC is strongly decreased in such a reduction and becomes $o(1)$ even if initially we have started the reduction from a $q$-query LDC with $\delta = \Omega(1)$.

The best known lower bound for 2-query LDCs is due to Goldreich et al. [38] who proved it for binary fields (see also [62]), it was generalized to general fields in [31]:

**Theorem 64 ([31])** Let $\mathbb{F}$ be any field. Let $C \subseteq \mathbb{F}^n$ be a linear $(2, \varepsilon, \delta)$-LDC with $k = \dim(C)$. Then $n \geq 2^{\frac{k\varepsilon}{\delta} - 1}$.

Previous lower bounds on LDCs with $\delta = o(1)$ were not achieved because of lack of tight lower bounds on 2-query LDCs with very small but non-

\(^2\)Recall that we think of $k$ as $\dim(C)$ and $n$ is a blocklength of $C$, where $C$ is a linear code. Hence $\dim(C) = k \leq n$ is a trivial bound in this case, in contrast to the bound $k/n = o(1)$.
trivial $\delta$, i.e., where $\omega(1) \leq \delta n \leq \log n$ (see Dvir [30] for motivation for such bounds). In Theorem 65 we give such a lower bound.

**Theorem 65 (Main Theorem on LDCs)** Let $\mathbb{F}$ be any field. If $C \subseteq \mathbb{F}^n$ is a $(2, \varepsilon, \delta)$-LDC with $k = \dim(C)$ then

$$n \geq 2^{\frac{\delta k}{1-\varepsilon}} \cdot \frac{1 - \varepsilon}{\delta}.$$ 

**Corollary 66** Let $\mathbb{F}$ be any field, $\varepsilon > 0$ and $\delta : \mathbb{N} \rightarrow \mathbb{R}$ be a function such that $\delta(n) \cdot n \geq \omega(1)$. Let $C = \{C_n\}_{n \in \mathbb{Z}}$ be a family of codes, where $C \subseteq \mathbb{F}^n$ is a $(2, \varepsilon, \delta(n))$-LDC. Then $\text{rate}(C_n) \leq O\left(\frac{\log(\delta(n) \cdot n)}{\delta(n) \cdot n}\right) = o(1)$.

**Proof.** Let $k = \dim(C_n)$. Theorem 65 implies that $n \geq 2^{\frac{\delta(n)k}{1-\varepsilon}} \cdot \frac{1 - \varepsilon}{\delta(n)} \geq 2^{\frac{\delta(n)k}{1-\varepsilon}} \cdot \frac{1}{\delta(n)}$. Hence $\delta(n) n \geq 2^{\frac{\delta(n)k}{1-\varepsilon}}$. We conclude that $\text{rate}(C_n) = k/n \leq O\left(\frac{\log(\delta(n) \cdot n)}{\delta(n) \cdot n}\right) = o(1)$.

**Remark 67** The above corollary says that there is no constant rate 2-query LDC such that $\delta(n) \cdot n = \omega(1)$. In contrast, the best known lower bound for 2-query LDCs (by Dvir and Shpilka [31]) does not give any non-trivial bound when $\delta(n) \cdot n \leq \log n$.

### 6.3 Proof of Main Results for LTCs

In this section we prove our main results regarding LTCs — Theorems 54 and 56 — and show how they follow from the main technical Lemma 63.

We first prove a Claim 68 which is the main place where Lemma 63 is used. Then we show how this claim implies Theorem 56.

**Claim 68** Let $C \subseteq \mathbb{F}_2^n$ be a linear code and let $I \subseteq [n]$ be a subset of independent indices. Assume that for every $i \in I$ we have $(3, i, C)$-sunflower $M_i$ such that $|M_i| \geq \Delta$. Then $|I|/n \leq \frac{\log(\Delta)+1}{\Delta}$.

**Proof.** Let $k = \dim(C)$. Let $G \in \mathbb{F}_2^{n \times k}$ be a generator matrix for $C$ and assume without loss of generality (reordering of indices) that $I = \{1, 2, \ldots, |I|\}$. Assume without loss of generality that the first $|I|$ rows and the first $|I|$ columns of $G$ form an identity matrix. ^3

^3 Do Gaussian elimination on columns to get the identity matrix in the first $|I|$ rows, since rank($G|_{|I| \times k}$) = $|I|$ the submatrix $G|_{|I| \times k}$ will contain the identity submatrix $|I| \times |I|$.
Let \( G' \in \mathbb{F}_2^{n \times |I|} \) be a submatrix of \( G \) obtained by removing all columns of \( G \) which have \( c|_I = 0^{||I||} \) (there are \( k - |I| \) such columns). Without loss of generality it holds that the top \(|I|\) rows of \( G' \) form an identity matrix \(|I| \times |I|\). Moreover, for all \( u \in C^\perp \) it holds that \( u^T \cdot G' = 0 \) since \( G' \) contains only the columns of \( G \) (i.e., the codewords of \( C \)). For the rest of the proof let \( e_i \) be a singleton vector in \( \mathbb{F}_2^{||I||} \). Note that for all \( i \in [|I|] \) it holds that \( G'_i = e_i \).

We conclude that for all \( i \in I \) we have a set \( M_i \subseteq ([n]/2) \) of disjoint pairs such that \( |M_i| \geq \Delta \) and for all \((j_1,j_2) \in M_i\) we have \( G'_{j_1} + G'_{j_2} = e_i \). Lemma 63 implies that \( |I|/n \leq \frac{\log(\Delta)+1}{\Delta} \).

Now we prove Theorem 56. We need the following definition of the repetition.

**Definition 69 (Bounded repetition)** Let \( C \subseteq \mathbb{F}_2^n \) be a linear code. For \( i_1,i_2 \in [n] \) we say that \( i_1 \) is a repetition of \( i_2 \) if for all \( c \in C \) we have \( c_{i_1} = c_{i_2} \), which happens if and only if there exists \( u \in C^\perp \) such that \( \text{supp}(u) = \{i_1,i_2\} \). We say that \( C \) is \( t \)-repetitive if for every \( i \in [n] \) it holds that \(|\{j \mid j \text{ is a repetition of } i\}| \leq t \).

**Proof of Theorem 56.** The fact that \( C \) is \((\alpha,\Delta)\)-natural implies that there exists a set of independent indices \( I \subseteq [n] \) such that \(|I| \geq \alpha \cdot \dim(C)\) and for every \( i \in I \) it holds that \( M_{\leq 3,i}(C) \geq \Delta \). Note that the fact that \( I \) contains only independent indices implies that there are no \( i \in I \) and \( u \in C^\perp \) such that \( \text{supp}(u) = \{i\} \).

If at least for \(|I|/2\) indices \( i \in I \) it holds that \(|\{u \in C^\perp \mid i \in \text{supp}(u)\}| \geq \Delta/2 \) then \( \text{rate}(C) = \frac{\dim(C)}{n} \leq \frac{1}{\alpha} \cdot \frac{4}{\Delta} \). To see this note that since \( I \) is a subset independent indices repetitions of different indices of \( I \) are different.

Otherwise, for at least \(|I|/2\) indices \( i \in I \) there exists a \((3,i,C)\)-sunflower \( M_i \subseteq C^\perp \) such that \(|M_i| \geq \Delta/2 \). Claim 68 says that \(|I|/n \leq \frac{\log(\Delta/2)+1}{\Delta/2} \).

Thus \( \text{rate}(C) = \frac{\dim(C)}{n} \leq \frac{1}{\alpha} \cdot \frac{\log(\Delta/2)+1}{\Delta/2} \).

We proceed to prove an auxiliary Proposition 70. Then we prove Theorem 54.

**Proposition 70** Let \( C \subseteq \mathbb{F}_2^n \) be a \( t \)-repetitive code and let \( I \subseteq [n] \) be a set of independent indices. Assume that for every \( i \in I \) it holds that \(|\{u \in C^\perp \mid i \in \text{supp}(u)\}| \geq \Delta \). Then, \(|I|/n \leq \frac{\log(\Delta/(2t))+(1)}{\Delta/(2t)} \).
Proof. We start from showing the following claim.

**Claim 71** For every $i \in I$ there exists $M_i \subseteq \binom{[n]}{2}$ such that $|M_i| \geq \Delta/(2t)$ and the following condition holds. For every $(j_1,j_2) \in M_i$ we have $u \in C^+_\Delta$, where $\text{supp}(u) = \{i,j_1,j_2\}$ and for every $(j_1,j_2) \neq (j'_1,j'_2) \in M_i$ we have $\{j_1,j_2\} \cap \{j'_1,j'_2\} = \emptyset$.

**Proof.** Let $i \in I$. We construct the subset $M_i$ iteratively. With some abuse of notation, for $S \subseteq [n]$ we say that $S \cap M_i = \emptyset$ if for all $x \in M_i$ we have $S \cap x = \emptyset$.

- $M_i := \emptyset$
- While there exists $u \in C^+_\Delta$ such that $i \in \text{supp}(u)$ and $\text{supp}(u) \cap M_i = \emptyset$
  
  
  $M_i := M_i \cup (\text{supp}(u) \setminus \{i\})$

The construction of $M_i$ implies that for every $(j_1,j_2), (j'_1,j'_2) \in M_i$ we have $u \in C^+_\Delta$, where $\text{supp}(u) = \{i,j_1,j_2\}$ and $\{j_1,j_2\} \cap \{j'_1,j'_2\} = \emptyset$. If $|M_i| \geq \Delta/(2t)$ we are done.

Assume that $|M_i| < \Delta/(2t)$. With some abuse of notation let $\text{supp}(M_i) = \{j \mid \exists j' \in [n]: (j,j') \in M_i\}$. We have $|\text{supp}(M_i)| = 2|M_i| < \Delta/t$. By assumption, it holds that $|\{u \in C^+_\Delta \mid i \in \text{supp}(u)\}| \geq \Delta$ and by construction for every $u \in C^+_\Delta$ such that $i \in \text{supp}(u)$ we have $(\text{supp}(u) \setminus \{i\}) \cap \{j_1,j_2\} \neq \emptyset$ for some $(j_1,j_2) \in M_i$. Let $T_{i,j} = \{u \in C^+_\Delta \mid i,j \in \text{supp}(u)\}$.

By the pigeonhole principle we conclude that there exists $j \in [n]$ such that $j \in \text{supp}(M_i)$ and $|T_{i,j}| > t$. Note that if $u_1, u_2 \in T_{i,j}$ and $u_1 \neq u_2$ then $\text{supp}(u_1) \cap \text{supp}(u_2) = \{i,j\}$ but $|\text{supp}(u_1)| = |\text{supp}(u_2)| = 3$. Clearly, $u_1 + u_2 \in C^+_\Delta$ (recall that the field is binary). Letting $i_1, i_2 \in [n]$ be such that $\{i_1\} = \text{supp}(u_1) \setminus \{i,j\}$ and $\{i_2\} = \text{supp}(u_2) \setminus \{i,j\}$ we have that $i_2$ is a repetition of $i_1$. Hence for every $u \in T_{i,j}$ letting $i' \in [n]$ be such that $\{i',i\} = \text{supp}(u) \setminus \{i,j\}$ it holds that $i'$ is a repetition of $i_1$, so there are $|T_{i,j}| > t$ repetitions of $i_1$. Contradiction.

Claim 71 implies that for every $i \in I$ there exists a subset $M_i \subseteq \binom{[n]}{2}$ of disjoint pairs such that $|M_i| \geq \Delta/(2t)$ and for all $(j_1,j_2) \in M_i$ we have $u \in C^+_\Delta$ such that $\text{supp}(u) = \{i,j_1,j_2\}$. Equivalently, for every $i \in I$ there exists a $(3,i,C)$-sunflower of size at least $\Delta/(2t)$. Claim 68 implies that $|I|/n \leq \frac{\log(\Delta/(2t)) + 1}{\Delta/(2t)}$.

We are ready to prove Theorem 54.
Proof of Theorem 54. Let $\sigma = \sigma_3(C)$. Note that $|C_1^\perp| = 0$ since otherwise $C = \{0^n\}$ ($C$ is 3-regular and hence in particular 1-regular). The fact that $C$ is 3-regular implies that every index $i \in [n]$ has the same number of repetitions in $C$ (see Definition 69). Let $t$ be the number of repetitions per index. Let $k = \dim(C)$. Then there exists an independent set $I \subseteq [n]$ such that $|I| = k$, and in particular, all indices in $I$ are not repetitions of each other. So, $|I| \cdot t = k \cdot t \leq n$. If $t \geq \sqrt{\sigma}/6$ then $\frac{k}{n} \leq \frac{1}{t} \leq \frac{6}{\sqrt{\sigma}}$ and we are done. Otherwise, $t < \sqrt{\sigma}/6$ and hence $C$ is $(\sqrt{\sigma}/6)$-repetitive. We argue that for every $i \in I$ it holds that $\Delta_{3,i}(C) \geq \sigma/3$, because every index $i \in [n]$ it holds that $\Delta_{3,i}(C) \geq \sigma_3(C)/3 = \sigma/3$. We conclude that for every $i \in I$ it holds that $|\{u \in C_3^\perp \mid i \in \text{supp}(u)\}| \geq \sigma/3 - \sqrt{\sigma}/6 \geq \sigma/6$.

Proposition 70 implies that rate($C$) $\leq \frac{\log(\sigma/12t) + 1}{\sigma/12t} \leq \frac{6\log \sigma + 2}{\sqrt{\sigma}}$. $\blacksquare$

6.4 Proof of Main Technical Lemma 63

In this section we prove Lemma 63. We end the section by showing that Lemmas 63 and 62 are tight (Section 6.4.4).

Overview of proof We study the generating matrix $G \in \mathbb{F}_2^{n \times k}$ of a 2-query LDC of dimension $k$ and blocklength $n$. Let $e_i \in \mathbb{F}_2^k$ be a singleton vector, i.e., $e_i = 0^{i-1}10^{k-i}$. Notice that when the first column of $G$ is removed, for each pair of indices $i \neq j$ used to decode the first message bit (i.e., $G_i + G_j = e_1$) we now have that the $i$ and $j$ rows of the smaller $n \times (k - 1)$ matrix are identical. In other words, after removing column 1 we may partition the rows of the residual matrix, denoted $G|_{n \times ([k]\backslash\{1\})}$, into sets of equal rows. Typically such sets will have size either 2 or 1. The former correspond to rows participating in a query for decoding the first message bit and the latter correspond to all other rows. Now, if we go on to remove the second column from $G$ we may expect to see in the residual matrix sets of equivalent rows of sizes between 1 and 4. The former sets correspond to rows not participating in any decoding of bits 1, 2 and the latter include rows that participate both in decoding message-bit number 1 and number 2. Continuing in this manner we would expect the size of sets of equivalent rows to double with every removal of an additional column from $G$ and this would show that after $\approx \log n$ column-removals all rows are equivalent, which means $k = O(\log n)$. 69
Of course, the description above is a gross oversimplification of what actually happens when columns are removed. The problem is that the size of different sets of equivalent rows can grow in arbitrary ways. To prove our lemma we rely on a simple fact — that whenever two equivalence classes “merge” into one larger class after removing a column of $G$, then at least one of them (the smaller) must double in size. This observation leads us to measure size of sets on a logarithmic scale and carry out an amortized analysis of the number of times sets (of equivalent rows) are merged upon removal of columns of the generating matrix. We shall explain how we remove columns from $G$ after making a few preliminary definitions and claims used in our proof.

### 6.4.1 Equivalence Relation and Matchings

With some abuse of notation consider every set as a multiset if not stated otherwise. The size of the multiset is the number of elements in it including repetitions. We recall that for $w \in F^n$ and $S \subseteq \{1, \ldots, n\}$ we let $w|S$ to be the restriction of $w$ to the subset $S$.

We define an equivalence relation over the set of rows of $G$.

**Definition 72 (Equivalence relation and class)** Let $J \subseteq \{1, \ldots, k\}$. For any $i, j \in \{1, \ldots, n\}$ we say that $G_i \approx_J G_j$ if and only if $G_i|\{k\}\setminus J = G_j|\{k\}\setminus J$. Since $\approx_J$ is an equivalence relation over $G$ it defines equivalence classes. Let $[G_i]_J$ be the equivalence class of $G_i$ under $J$, i.e., $[G_i]_J = \{G_j \mid G_i \approx_J G_j\}$.

We let $P_J$ denote the quotient set of the multiset $G$ by $\approx_J$, i.e., $P_J = \{[G_{i_1}]_J, \ldots, [G_{i_m}]_J\}$, where $G_{i_1}, \ldots, G_{i_m} \in G$ are arbitrarily chosen representatives for the quotient set. It holds that $P_J$ is a partition of the multiset $G$ hence we will also say that $P_J$ is a $J$-partition of $G$.

Now we define the important concept, called **valid matchings**. The concepts “equivalence classes” and “valid matchings” are central in the proof of Lemma 63.

**Definition 73 (i-Matching)** Let $J \subseteq \{1, \ldots, k\}$ and $i \in \{1, \ldots, n\}$. Let $M \subseteq \binom{\{1, \ldots, n\}}{2}$. We say that $M$ is an $i$-matching if for all pairs $(i_1, i_2) \in M$ it holds that $G_{i_1} + G_{i_2} = e_i$. We say that the matching $M$ is valid for $J$ if for all pairs $(i_1, i_2) \in M$ it holds that $G_{i_1}|\{k\}\setminus J + G_{i_2}|\{k\}\setminus J = (e_i)|\{k\}\setminus J$. 70
For $a \in [n]$ we say that an element $G_a \in G$ appears in the pair $(i_1, i_2)$ if either $a = i_1$ or $a = i_2$. We say that $G_a$ appears in the matching $M$ if it appears in at least one pair of $M$.

Recall that for every $i \in [k]$ we have an $i$-matching $M_i$ such that $|M_i| \geq \Delta$. Note that for every $i \in [k]$ it holds that every element of $G$ appears at most once in the matching $M_i$. The following two simple claims summarize the effect of projection on the equivalence classes and matchings.

**Claim 74 (Projection does not affect non-projected matchings)**

Let $J \subseteq [k]$ and $i \in [k] \setminus J$. If $M$ is an $i$-matching then $M$ is valid for $J$.

**Proof.** This is true since for all pairs $(i_1, i_2) \in M$ we have $G_{i_1} + G_{i_2} = e_i$ hence $G_{i_1}|_{[k] \setminus J} + G_{i_2}|_{[k] \setminus J} = (e_i)|_{[k] \setminus J}$. $\blacksquare$

**Claim 75 (Projection implies Collapse of Equivalence Classes)**

Let $J \subseteq [k]$ and $e_i = G_j + G_{j'}$. If $i \in J$ then $G_j \approx_J G_{j'}$, or equivalently, $[G_j]_J = [G_{j'}]_J$.

**Proof.** If $G_j + G_{j'} = e_i$ then $G_j|_{[k] \setminus J} + G_{j'}|_{[k] \setminus J} = 0$. So, $G_j|_{[k] \setminus J} = G_{j'}|_{[k] \setminus J}$ hence $G_j \approx_J G_{j'}$. $\blacksquare$

### 6.4.2 Selection of columns to be removed from the generating matrix

In this section we describe the process by which columns of $G$ are removed. We start with an explanation of the intuition behind this selection process. Recall that our goal is to upper-bound the number $k$. We start from the definition of small multisets and good matchings.

**Definition 76 (Small Multisets and Good Matching)** A multiset $S$ is called small if $|S| < \Delta$ and otherwise it is called large. We say that the $i$-matching $M_i$ is $J$-good if $i \in [k] \setminus J$ and for all edges $(j, j') \in M_i$ it holds that at least one of $[G_j]_J$, $[G_{j'}]_J$ is a small multiset. 

Let $J = \{i_1, i_2, ..., i_h\} \subseteq [k]$ and for $t \leq h$ let $J(t) = \{i_1, i_2, ..., i_t\}$ and $J(0) = \emptyset$. Assume that for all $t \leq h$ it holds that the $i_t$-matching $M_{i_t}$ is $J(t - 1)$-good. By Definition 76 all pairs of $M_{i_t}$ “touch” small subsets in $P_{J(t-1)}$ and note that Claim 75 implies that a large number of pairs of
multisets \([G_{j_1}]_{J(t-1)}\) and \([G_{j_2}]_{J(t-1)}\) collapse into the single multiset \([G_{j_1}]_{J(t)}\), i.e., \([G_{j_1}]_{J(t)} = [G_{j_2}]_{J(t)}\). In this way, we can expect that for all \(t \leq h\) the size of \(P_{J(t)}\) will be much smaller than the size of \(P_{J(t-1)}\). Note also that \(|P_{J(h)}| \geq 1\) and \(|P_{J(0)}| \leq n\). Hence the subset \(J\) cannot be too large. Later on in the proof we will upper-bound \(|J|\) and on the other side we will argue that \(|[k] \setminus J|\) is small, obtaining the upper bound on \(k\).

The following algorithm constructs the set \(J \subseteq [k]\). Roughly we maintain an iteration number \(t\) and set \(J(t)\) which grows slowly. For analysis it is better to denote sets separately.

\[
\begin{align*}
\text{Construction of } J \\
\quad & \cdot t := 0 \\
\quad & \cdot J(t) := \emptyset \\
\quad & \cdot \text{While there exists } i \in [k] \setminus J(t) \text{ such that the matching } M_i \text{ is } J(t)\text{-good} \\
\quad & \hspace{1cm} - J(t + 1) := J(t) \cup \{i\} \\
\quad & \hspace{1cm} - t := t + 1 \\
\quad & \cdot J := J(t) \\
\quad & \text{return } J
\end{align*}
\]

For the rest of the proof, we assume that the algorithm returns the subset \(J = \{i_1, i_2, ..., i_h\}\), where \(i_t\) is the element added in the \(t\)th iteration of the algorithm. Notice \(J(t) = \{i_1, i_2, ..., i_t\}\) and \(J(0) = \emptyset\). We have two immediate but crucial properties, stated formally in Claims 77 and 78.

\textbf{Claim 77} For every \(t \in [h]\) it holds that the \(i_t\)-matching \(M_{i_t}\) is \(J(t-1)\)-good.

\textbf{Claim 78} For every \(i \in [k] \setminus J\) it holds that the \(i\)-matching \(M_i\) is not \(J\)-good, i.e., there exists a pair \((j, j') \in M_i\) such that both multisets \([G_j]_J\) and \([G_{j'}]_J\) are large.

\textbf{Proof.} The claim follows from the construction of \(J\). If for some \(i \in [k] \setminus J\) the \(i\)-matching is \(J\)-good then the construction of \(J\) would not stop.

\[\blacksquare\]
6.4.3 Completing the proof of Main Technical Lemma 63

In this section we present Lemmas 79 and 80. The proof of the Combinatorial Lemma 63 will follow immediately from these two lemmas. The rest of this section is devoted to the proofs of the two sub-lemmas stated next.

Lemma 79 (Bound on \( k - |J| \)) It holds that \( k - |J| \leq \frac{n\Delta}{\Delta} \).

Lemma 80 (Bound on \( |J| \)) It holds that \( |J| \leq \frac{n\log \Delta}{\Delta} \).

The proof of Lemma 63 follows by a combination of Lemmas 79 and 80.

Proof of Lemma 63.

We have

\[
k = |J| + (k - |J|) \leq n\log \Delta + \frac{n\Delta}{\Delta}.\]

\(\blacksquare\)

In Sections 6.4.3 and 6.4.3 we prove Lemmas 79 and 80, correspondingly.

Proof of Lemma 79

Let \( m = k - |J| \) and assume without loss of generality that \( J = \{m+1, m+2, \ldots, k\} \). Let \( r \) be the number of large multisets in \( P_J \) and assume without loss of generality that the large multisets of \( P_J \) are \( [G_1]_J, \ldots, [G_r]_J \). We have that \( r \leq n/\Delta \) since the number of rows is \( |G| = n \) and every large subset has size at least \( \Delta \).

Claim 78 says that for every \( i \in [m] = [k] \setminus J \) the matching \( M_i \) is not \( J \)-good, i.e., there exists at least one edge \((j, j') \in M_i\) such that both \([G_j]_J\) and \([G_{j'}]_J\) are large, meaning \( |[G_j]_J| \geq \Delta \) and \( |[G_{j'}]_J| \geq \Delta \). Note that in this case \( G_j|_{[m]} + G_{j'}|_{[m]} = e_i|_{[m]} \), i.e., \( e_i|_{[m]} \in \text{span}\{G_j|_{[m]} | j \in [r]\} \).

We conclude that for every \( i \in [m] \) it holds that \( e_i|_{[m]} \in \text{span}\{G_j|_{[m]} | j \in [r]\} \). We argue that \( m \leq r \). To see this recall that \( G_1|_{[m]}, \ldots, G_r|_{[m]} \in \mathbb{F}_2^m \) and note that for every \( i \in [m] \) we have \( e_i|_{[m]} \in \text{span}\{G_j|_{[m]} | j \in [r]\} \). Thus \( m = \dim(\text{span}\{e_i|_{[m]} | i \in [m]\}) \leq \dim(\text{span}\{G_1|_{[m]}, \ldots, G_r|_{[m]}\}) \leq r \).

We conclude that \( k - |J| = m \leq r \leq n/\Delta \) and this completes the proof of Lemma 79.

Proof of Lemma 80

In this section we prove that \( |J| \leq \frac{n\log \Delta}{\Delta} \). We first define the valence of a multiset.
Definition 81 (Valence of the Multiset) Given a multiset \( S \neq \emptyset \) its valence \( v(S) \) is defined as \( \lfloor \log |S| \rfloor \).

Remark 82 Recall from Definition 76 that a multiset \( S \) is large if \( v(S) \geq \log \Delta \).

Definition 83 (Consumptions - Edges vs. Rows) Let \( t \leq h \). For \( i_t \in J(t) \setminus J(t-1) \) let \( M_{i_t} \) be the \( i_t \)-matching and \( e = (m, m') \in M_{i_t} \). In this case, we say that the edge \( e \) was consumed in iteration \( t \).

If \( |[G_m]_{J(t-1)}| \leq |[G_{m'}]_{J(t-1)}| \) we say that \( G_m \) consumed the edge \( e \) in iteration \( t \). Note that if \( |[G_m]_{J(t-1)}| = |[G_{m'}]_{J(t-1)}| \) then both \( G_m \) and \( G_{m'} \) consumed \( e \) in iteration \( t \).

Now let us define the indicator variables for row and edge consumptions.

Definition 84 (Indicators for the consumptions) Since every row of \( G \) appears in any given matching at most once, we know that every row consumes at most one edge in iteration \( t \). Hence we can define the following indicator variables. Let \( E_{(m,m'),t} \) be the indicator for the event that the edge \( (m, m') \) was consumed in iteration \( t \). Let \( E_t = \sum_{e \in M_t} E_{e,t} \) be the number of edges that were consumed in time \( t \) and let \( E_{\leq t} = \sum_{i=1}^{t} E_i \) be the number of edges that were consumed up to time \( t \).

Similarly, for \( l \in [n] \) we let \( R_{l,t} \) be the indicator for the event that the row \( G_l \) consumes some edge in time \( t \). Let \( R_t = \sum_{l \in [n]} R_{l,t} \) be the number of rows that consume an edge in iteration \( t \), and let \( R_{\leq t} = \sum_{i=1}^{t} R_i \) be the number of consumptions which happen up to time \( t \).

The intuition behind this definition is as follows. The consumption of edges is tightly related to the consumption by rows. The numbers are roughly equal since when an edge is consumed, it is consumed by at least one (and at most two) rows. So, on the one side there are many edges that were consumed and on the other side, as shown in Proposition 86, every row can not consume too many edges, since the valence of an equivalence class containing the row is increased at least by one after consumption.

We go on to present Claim 85 and Propositions 86 and 87. Then we prove Lemma 80. We end this section by proving Claim 85 and Propositions 86 and 87.
Claim 85 (Consumption implies increase of valence) Let $t < h$, $M_{t+1}$ be the $i_{t+1}$-matching and $(j,j') \in M_{t+1}$. It holds that

- at least one of $G_j, G_{j'}$ consumes the edge $(j,j')$ in iteration $t+1$, and
- if $G_j$ consumes the edge $(j,j')$ in iteration $t$ then $v([G_j]_{J(t+1)}) \geq 1 + v([G_j]_{J(t)})$.

Proposition 86 (Row Consumption) For every $t \leq h$ it holds that $R_{\leq t} \leq n \log \Delta$.

Proposition 87 (Edge Consumption) For every $t \leq h$ we have $E_{\leq t} \geq \Delta \cdot |J(t)| = \Delta \cdot t$.

We are ready now to prove the Lemma 80, which says $|J| \leq \frac{n \log(\Delta)}{\Delta}$. 

Proof of Lemma 80. Recall that $h = |J|$. Proposition 86 implies that $R_{\leq h} \leq n \log \Delta$. Proposition 87 implies that the total number of edge consumptions $E_{\leq h}$ is at least $h \cdot \Delta$. Claim 85 implies that $E_{\leq h} \leq R_{\leq h}$. We conclude that $h \cdot \Delta \leq E_{\leq h} \leq R_{\leq h} \leq n \cdot \log \Delta$, and thus $|J| = h \leq \frac{n \log(\Delta)}{\Delta}$.

Now we prove Claim 85 and Propositions 86 and 87.

Proof of Claim 85. Claim 74 implies that $M_{t+1}$ is valid for $J(t)$ and, in particular,

$G_j|_{J(t)} + G_{j'}|_{J(t)} = e_{i_{t+1}}$.

Hence $[G_j]_{J(t)} \neq [G_{j'}]_{J(t)}$ since otherwise, if $[G_j]_{J(t)} = [G_{j'}]_{J(t)}$ then it holds that $G_j|_{J(t)} + G_{j'}|_{J(t)} = 0 \neq e_{i_{t+1}}$. Clearly, either $|[G_j]_{J(t)}| \leq ||G_{j'}]_{J(t)}|$ or $|[G_{j'}]_{J(t)}| \leq ||G_j]_{J(t)}|$ hence, by definition, at least one of $G_j, G_{j'}$ consumes the edge in iteration $t + 1$. This completes the proof of the first bullet.

For the second bullet, by assumption, we have $|[G_j]_{J(t)}| \leq ||G_{j'}]_{J(t)}|$. Claim 75 implies that

$[G_j]_{J(t+1)} = [G_{j'}]_{J(t+1)}$ but $[G_j]_{J(t)} \neq [G_{j'}]_{J(t)}$.

This means $[G_j]_{J(t)} \cup [G_{j'}]_{J(t)} \subseteq [G_j]_{J(t+1)}$ and so, $|[G_j]_{J(t)}| + ||G_{j'}]_{J(t)}| \leq ||G_j]_{J(t+1)}|$. The fact that $|[G_j]_{J(t)}| \leq ||G_{j'}]_{J(t)}|$ implies that

$2|[G_j]_{J(t)}| \leq ||G_j]_{J(t)}| + ||G_{j'}]_{J(t)}| \leq ||G_j]_{J(t+1)}|$. 

75
It follows that

\[ 1 + |\log |G_J|,J(t)|| = 1 + \log |G_J|,J(t)| = |\log 2| |G_J|,J(t)|| \leq |\log |G_J|,J(t+1)|| . \]

We conclude that

\[ 1 + v(|G_J|,J(t)|) = 1 + \log |G_J|,J(t)|| \leq |\log |G_J|,J(t+1)|| = v(|G_J|,J(t+1)|) . \]

**Proof of Proposition 86.** We first claim that for every row \( G_t \in G \) it holds that \( \sum_{i=1}^h R_{it} \leq \log \Delta \). Note that for all \( v(|G_t|,J(t)|) \) is monotonically non-decreasing, i.e., \( v(|G_t|,J(0)|) \leq v(|G_t|,J(1)|) \leq \cdots \leq v(|G_t|,J(h)|) \). This is true because \( |G_t|,J(0)| \subseteq |G_t|,J(1)| \subseteq \cdots \subseteq |G_t|,J(h)| \).

We argue that if for some time \( t \leq h \) we have \( v(|G_t|,J(t)|) \geq \log \Delta \) then for every \( t' \) such that \( t < t' \leq h \) we have \( R_{it'} = 0 \) and \( v(|G_t|,J(t)|) \geq \log \Delta \). Assume the contrary. Clearly, for every \( t' > t \) we have \( v(|G_t|,J(t')|) \geq v(|G_t|,J(t)|) \geq \log \Delta \) since \( |G_t|,J(t)| \subseteq |G_t|,J(t')| \). So, there exists \( t' > t \) such that \( v(|G_t|,J(t'-1)|) \geq \log \Delta \) but \( R_{it'} = 1 \). Note that \( i_t \in J \). From the definition of “consumption” (Definition 83) it follows that there exists an edge \( (l',t') \in M_{i_t} \) such that \( |G_t|,J(t'-1)| \leq |G_{t'}|,J(t'-1)| | \leq \Delta \). In this case, the matching \( M_{i_{t'}} \) is not \( J(t'-1)-\text{good} \), contradicting Claim 77. We conclude that if for some time \( t \leq h \) we have \( v(|G_t|,J(t)|) \geq \log \Delta \) then for every \( t' \) such that \( t < t' \leq h \) we have \( R_{it'} = 0 \) and \( v(|G_t|,J(t')|) \geq v(|G_t|,J(t)|) \geq \log \Delta \).

Now, in iteration 0 the valence of \( |G_t|,J(0)| \) is at least 0. Claim 85 implies that if \( G_t \) consumes an edge in iteration \( t' \leq h \) then \( v(|G_t|,J(t')|) \geq v(|G_t|,J(t'-1)|) + 1 \). This means that if \( R_{it'} = 1 \) then \( v(|G_t|,J(t')|) \geq v(|G_t|,J(t'-1)|) + R_{it'} \). Note that if \( R_{it} = 0 \) it is also true that \( v(|G_t|,J(t')|) \geq v(|G_t|,J(t'-1)|) + R_{it'} \). Hence for every \( t' \leq h \) we have \( v(|G_t|,J(t')) \geq v(|G_t|,J(t'-1)|) + R_{it'} \). Recalling \( v(|G_t|,J(0)|) \geq 0 \) it follows that for every \( t' \leq h \) we have \( \sum_{i=1}^t R_{it'} \leq v(|G_t|,J(t')|) \).

We conclude that for every row \( G_t \in G \) it holds that \( \sum_{i=1}^t R_{it} \leq \log \Delta \). Recalling that \( |G| = n \), we have

\[ R_{\leq t} = \sum_{i=1}^t \sum_{t \in [n]} R_{it} \leq \sum_{t \in [n]} \sum_{i=1}^t R_{it} \leq \sum_{i=1}^t \log \Delta = n \log \Delta . \]
Proof of Proposition 87. Recall that \( J(t) = \{i_1, i_2, \ldots, i_t\} \) is an ordered set. By construction of \( J \), for every \( t \leq h \) it holds that \( i_t \in [k] \setminus J(t-1) \). Claim 85 implies that all edges of the \( i_t \)-matching \( M_{i_t} \) are consumed in iteration \( t \). Thus for every \( t \leq h \) we have \( |E_t| \geq |M_{i_t}| \geq \Delta \). Recalling that \( |M_{i_l}| \geq \Delta \) we conclude

\[
E_{\leq t} = \sum_{l=1}^{t} E_l = \sum_{l=1}^{t} |M_{i_l}| \geq t \cdot \Delta.
\]

6.4.4 Tightness of Lemmas 63 and 62

We end our discussion of Lemmas 63 and 62 by showing that each of them is tight.

Lemma 88 (Tightness of Lemma 63) Let \( \Delta : \mathbb{N} \to \mathbb{N} \) be a function such that \( \Delta(n) \leq n/2 \). Then there exists a matrix \( G \in \mathbb{F}_2^{n \times k} \) and for every \( i \in [k] \) there exists a set of disjoint pairs of indices \( M_i \subseteq \binom{[n]}{2} \) such that

- For every \( (i_1, i_2) \in M_i \) we have \( G_{i_1} + G_{i_2} = e_i \),
- For every \( i \in [k] \) we have \( |M_i| \geq \Delta(n) \),
- Furthermore, it holds that \( k = \frac{n \log \Delta(n) + n}{2 \Delta(n)} \).

Remark 89 We assume that \( \Delta(n) \), \( n/2 \Delta(n) \) and \( \log \Delta(n) \) are integers. Otherwise, we would work in terms of \( \lceil \Delta(n) \rceil \), \( \lfloor n/2 \Delta(n) \rfloor \) and \( \lfloor \log \Delta(n) \rfloor \).

Proof of Lemma 88. Let \( k = \frac{n(\log(\Delta(n)) + 1)}{2 \Delta(n)} \). Let \( k_1 = \log(\Delta(n)) + 1 \) and \( n_1 = 2^{k_1} = 2 \Delta(n) \). Let \( H \in \mathbb{F}_2^{n_1 \times k_1} \) be the generator matrix of the Hadamard code (with blocklength \( n_1 \) and dimension \( k_1 \)).

We show how to construct the required matrix \( G \in \mathbb{F}_2^{n \times k} \). Informally, \( G \) will be constructed from \( \frac{n}{2 \Delta(n)} \) copies of matrix \( H \) and they will be located along the diagonal of the matrix \( G \).

1. Initialization:
   - \( G := 0^{n \times k} \)
• row := 1
• column := 1

2. While (row ≤ n)
   • While (column ≤ k)
     (a) Copy the matrix $H$ to the submatrix of $G$ with coordinates

     $[$row, ..., row + n − 1$] \times [column, ..., column + k − 1$]

     (b) column := column + k
   • column := 1
• row := row + n

We argue that for every $i \in [k]$ there are at least $\Delta(n)$ disjoint pairs

$(i_1, i_2) \in [n] \times [n]$ such that $G_{i_1} + G_{i_2} = e_i$. Let $i \in [k]$. Assume without loss of
generality that $i \in [k]$. It is sufficient to show that there are $n_1/2 = \Delta(n)$
disjoint pairs $(i_1, i_2) \in [n_1] \times [n_1]$ such that $G_{i_1} + G_{i_2} = e_i$. Recall that

$G|_{[n_1] \times [k]} = H \in \mathbb{F}_2^{n_1 \times k}$ is the generating matrix for the Hadamard code

hence contains $n_1/2 = \Delta(n)$ disjoint pairs $(i_1, i_2) \in [n_1] \times [n_1]$ such that

$H_{i_1} + H_{i_2} = e_i$. This true for $G|_{[n_1] \times [k]}$ as well since $G|_{[n_1] \times [k]}$ is zero outside
the submatrix $[n_1] \times [k]$.

We now show that it is crucial to take into account the fact that every
matching, not just an average on, is large. In particular, we show that if
this fact is not taken into account then the lower bound of Goldreich et al.
[38] is tight.

**Lemma 90 (Tightness of Lemma 62)** Let $\Delta : \mathbb{N} \to \mathbb{N}$ be a function
such that $\Delta(n) \leq n/2$. Then there exists matrix $G \in \mathbb{F}_2^{n \times k}$ and for every
$i \in [k]$ there exists a set of disjoint pairs of indices $M_i \subseteq \binom{[n]}{2}$ such that

- For every $(i_1, i_2) \in M_i$ we have $G_{i_1} + G_{i_2} = e_i$,
- $\sum_{i=1}^{k} |M_i| = k \cdot \Delta(n)$, i.e., in the average $|M_i| = \Delta(n)$,
- Furthermore, it holds that $k = \frac{n \log n}{2\Delta(n)}$.

\footnote{It can be assumed without loss of generality since the matrix $G$ was constructed in a completely symmetric way.}
Remark 91  Once again, we assume that $\Delta(n)$, $\frac{n}{2\Delta(n)}$ and $\log \Delta(n)$ are integers. Otherwise, we would work in terms of $[\Delta(n)]$, $[n/2\Delta(n)]$ and $[\log \Delta(n)]$.

Proof of Lemma 90.  Recall that by assumption we have $\Delta(n) \leq n/2$. Let $k = \frac{n \log n}{2\Delta(n)}$. Let $k_1 = \log n$ and $k_2 = k - k_1$. Let $H \in \mathbb{F}_2^{n \times k_1}$ be the Hadamard generator matrix. Let $L = 0^{n \times k_2}$ be a zero matrix. Let $G = H \circ L$ (we took $H$ and appended $L$). We argue that for every $i \in [k_1]$ there are $n/2$ distinct pairs $G_{i_1}, G_{i_2} \in G$ such that $G_{i_1} + G_{i_2} = e_i$. This is true since for every $i \in [k_1]$ there are $n/2$ distinct pairs $H_{i_1}, H_{i_2} \in H$ such that $H_{i_1} + H_{i_2} = e_i$ and $L$ is a zero matrix and hence does not affect this property when it is appended to $H$. Note also that $\sum_{i=1}^{k_1} |M_i| = 0$ because of zero matrix $L$. Hence $\sum_{i=1}^{k_1} |M_i| = \sum_{i=1}^{k_1} |M_i| = (n/2) \cdot k_1 = n \log(n)/2 = k \cdot \Delta(n)$.

6.5  Limiting the rate of weak 2-query LDCs —  
Proof of Theorem 65

In this section we prove Theorem 65. We first present Lemmas 92 and 93. The proof of Theorem 65 will follow by a combination of these lemmas.

Lemma 92 (Combinatorial Lemma for General Field) Let $\mathbb{F}$ be any field and let $G \in \mathbb{F}^{n \times k}$. For every $i \in [k]$ let $M_i \subseteq [n] \times [n]$ be a set of disjoint pairs of indices such that $e_i \in \text{span} \{ G_{j_1}, G_{j_2} \}$ for every $(j_1, j_2) \in M_i$. Assume that for all $i \in [k]$ we have $|M_i| \geq \Delta$, where $\Delta \geq 1$. Then,

$$k \leq \frac{16n \log \Delta + 16n}{\Delta}$$

The proof of Lemma 92 is postponed to Section 6.5.1. The following Lemma 93 is due to Obata [62]. The main result of [62] (Lemma 93) provides a tight analysis of the number of matchings which has a 2-query LDC. Although Obata [62] proved this result over the binary field but generalization to arbitrary fields is straightforward. To make the paper self-contained we give a proof-sketch of Lemma 93, stated for all fields, in Section 6.5.2.

Lemma 93  Let $C \subseteq \mathbb{F}^n$ be a $(2, \varepsilon, \delta)$-LDC and $k = \dim(C)$. Let $G \in \mathbb{F}^{n \times k}$ be a generator matrix for $C$. Then for every $i \in [k]$ there exists $M_i \subseteq [n] \times [n]$
of disjoint pairs such that \(|M_i| \geq \frac{1}{2} \cdot \frac{\delta n}{1 - \varepsilon} - \frac{\delta n \cdot |F|}{|F| - 1} \cdot \varepsilon\) and \(e_i \in \text{span}\{G_{j_1}, G_{j_2}\}\) for every \((j_1, j_2) \in M_i\).

We are ready to prove Theorem 65.

**Proof of Theorem 65.** Let \(G \in \mathbb{F}^{n \times k}\) be a generator matrix for \(C\).

Let \(\Delta = \frac{1}{2} \cdot \frac{\delta n}{1 - \varepsilon} - \frac{\delta n \cdot |F|}{|F| - 1} \cdot \varepsilon\). Lemma 93 implies that for every \(i \in [k]\) there is a set \(M_i \subseteq [n] \times [n]\) of disjoint pairs such that \(|M_i| \geq \Delta\) and for every \((j_1, j_2) \in M_i\) we have \(e_i \in \text{span}\{G_{j_1}, G_{j_2}\}\). Then Lemma 92 implies that \(k \leq \frac{16n \log \Delta + 16n}{\Delta}\).

We conclude that \(k \leq \frac{16n \log(\frac{1}{2} \cdot \frac{\delta n}{1 - \varepsilon}) + 16n}{(\frac{1}{2} \cdot \frac{\delta n}{1 - \varepsilon})} \leq \frac{32 \log(\frac{\delta n}{1 - \varepsilon}) + 32}{\frac{\delta}{1 - \varepsilon}}\). Hence \(\frac{\delta k}{32(1 - \varepsilon)} - 1 \leq \log(\frac{\delta n}{1 - \varepsilon})\) and \(n \geq 2^{\frac{6k}{(1 - \varepsilon)\cdot \delta^2}} \cdot \frac{1 - \varepsilon}{\delta}\).  

### 6.5.1 Proof of Lemma 92 – General field \(\mathbb{F}\)

In this section we prove Lemma 92. We need the following Lemma 94 due to Dvir and Shpilka [31, Lemma 2.5].

**Lemma 94 ([31])** Let \(\mathbb{F}\) be any field and let \(G \in \mathbb{F}^{n \times k}\). For every \(i \in [k]\) let \(M_i \subseteq [n] \times [n]\) be a set of disjoint pairs of indices, such that \(e_i \in \text{span}\{G_{j_1}, G_{j_2}\}\) for every \((j_1, j_2) \in M_i\). Then, there exist \(G'' \in \mathbb{F}^{n \times k}\) and \(k\) sets \(M''_1, \ldots, M''_k \subseteq \binom{[n]}{2}\) of disjoint pairs, such that:

- For every \((j_1, j_2) \in M''_i\) it holds that \(G''_{j_1} \oplus G''_{j_2} = e_i\),
- \(\sum_{i=1}^{k} |M_i| \leq 2 \sum_{i=1}^{k} |M''_i| + n\),
- For every \(i \in [k]\) it holds that \(M''_i \subseteq M_i\)

**Remark 95** The only difference between [31, Lemma 2.5] and Lemma 94 is that the third bullet was not explicitly stated in [31, Lemma 2.5]. However, it can be readily verified that for all \(i \in [k]\) it holds that \(M''_i \subseteq M_i\). We briefly explain this and refer a reader to [31, Lemma 2.5] for notation and definitions.

This is true since Dvir and Shpilka [31, Lemma 2.5] showed the reduction from a general field \(\mathbb{F}\) to binary field in two steps. In the first step some pairs were removed from the matchings \(M_1, \ldots, M_k\) resulting in the matchings \(M'_1, \ldots, M'_k\) such that \(M'_i \subseteq M_i\). In the second step they suggested a
transformation from $\mathbb{F}$ to $\mathbb{F}_2$ such that for all $i \in [k]$ some pairs from $M'_i$ were removed resulting in $M''_i$. So, they obtained matchings $M''_1, \ldots, M''_k$ such that $M''_i \subseteq M'_i$.

**Proof of Lemma 92.** Let $M_1, \ldots, M_k \subseteq [n] \times [n]$ be matchings such that for every $i \in [k]$ we have $|M_i| \geq \Delta$. We can assume w.l.o.g. that for every $i \in [k]$ we have $|M_i| = \Delta$ (otherwise remove some pairs from $M_i$).

Lemma 94 implies the existence of $G'' \in \mathbb{F}_2^{n \times k}$ and matchings $M''_i$ such that $\Delta \cdot k = \sum_{i=1}^k |M_i| \leq 2 \sum_{i=1}^k |M''_i| + n$ and for every $i \in [k]$ it holds that $M''_i \subseteq M_i$, which means $|M''_i| \leq |M_i| \leq \Delta$. Moreover, for every $(j_1, j_2) \in M''_i$ it holds that $G''_{j_1} \oplus G''_{j_2} = e_i$. We say that the matching $M''_i$ is bad if $|M''_i| < \Delta/4$. If the number of bad matchings is more than $3\Delta/4$ then $\Delta k \leq 2 \sum_{i=1}^k |M''_i| + n \leq 2((3\Delta/4)(\Delta/4) + (\Delta/4)\Delta) + n \leq (14/16)\Delta^2 + n = (7/8)\Delta^2 + n$. In this case we get $k \leq 8n/\Delta$ and we are done since $8n/\Delta \leq \frac{16n \log \Delta + 16n}{\Delta}$. Otherwise, the number of bad matchings is less than $3\Delta/4$, hence there are at least $k/4$ good matchings (those with $|M''_i| \geq \Delta/4$).

Assume w.l.o.g. that for all $i \in [k/4]$ the matching $M''_i$ is good, i.e., $|M''_i| \geq \Delta/4$. Consider $A'' = G''|_{n \times (k/4)} \in \mathbb{F}_2^{n \times (k/4)}$ and note that for every $i \in [k/4]$ and $(j_1, j_2) \in M''_i$ we have $A''_{j_1} \oplus A''_{j_2} = e_i$. Lemma 63 implies that $k/4 \leq \frac{n \log (\Delta/4) + n}{\Delta} \leq \frac{4n \log \Delta + 4n}{\Delta}$. We conclude that $k \leq \frac{16n \log \Delta + 16n}{\Delta}$. ■

### 6.5.2 Proof of Lemma 93

In this section we give a sketch of the proof of Lemma 93. We start from the (non-standard) definition of non-redundant matchings.

**Definition 96 (Non-redundant Edges and Matching)** Let $G \in \mathbb{F}^{n \times k}$ and let $i \in [k]$. We say that $(j_1, j_2) \in [n] \times [n]$ is a non-redundant $i$-edge if we have $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$, and moreover, if $e_i \in \text{span}\{G_{j_1}\}$ or $e_i \in \text{span}\{G_{j_2}\}$ then $j_1 = j_2$. We say that $E_i \subseteq [n] \times [n]$ is an $i$-set of non-redundant edges if for every $(j_1, j_2) \in E_i$ we have that $(j_1, j_2)$ is a non-redundant $i$-edge. We say that $M_i \subseteq E_i$ is a non-redundant $i$-matching if every $j \in [n]$ appears in at most one edge of $M_i$.

Note that Definition 96 allows self-loops in the non-redundant matchings, and we demonstrate this in the next example.
Example 97 Let $G \in \mathbb{F}_2^{3 \times 3}$ such that $G = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, i.e., $G_1 = (100)$ is a first row, $G_2 = (111)$ is a second row and $G_3 = (011)$ is a third row. Then, $E_1 = \{(1,1),(2,3)\}$ is a 1-set of (non-redundant) edges and a non-redundant 1-matching $M_1 = E_1$. Note that $M_1$ is a (legal) non-redundant 1-matching, e.g., 1 appears only in the single edge $(1,1)$ of $M_1$. Moreover, a 2-set and a 3-set of non-redundant edges are empty, i.e., $E_2 = E_3 = \emptyset$.

The intuition behind the definition of “non-redundant” edges (Definition 96) is as follows. Let $C$ be a 2-query linear LDC and $G$ be its generator matrix. Without loss of generality [38], a 2-query decoder for $C$ recovers the message bit $i$ by querying (at most) two bits indexed by $j_1,j_2$ such that $e_i \in \text{span}(G_{j_1}, G_{j_2})$. However, if it holds that $e_i \in \text{span}(G_{j_1})$ or $e_i \in \text{span}(G_{j_2})$ we can assume w.l.o.g. [38] that the decoder queries (at the same invocation) at most one from $j_1,j_2$. So, if $e_i \in \text{span}(G_{j_1}, G_{j_2})$ but $e_i \notin \text{span}(G_{j_1})$ and $e_i \notin \text{span}(G_{j_2})$ then $(j_1,j_2)$ is a non-redundant $i$-edge; and if $e_i \in \text{span}(G_{j_1})$ (or $e_i \in \text{span}(G_{j_2})$) then $(j_1,j_1)$ (or $(j_2,j_2)$) is a non-redundant $i$-edge.

We continue by recalling an implicit argument from [38] (see also [31]).

Claim 98 (Implicit in [38]) Let $C \subseteq \mathbb{F}^n$ be a $(2,\varepsilon,\delta)$-LDC and $k = \dim(C)$. Let $G \in \mathbb{F}^{n \times k}$ be a generator matrix for $C$. The decoder $\mathcal{D}$ for $C$ is associated with a list of distributions $\{\mathcal{D}_i\}_{i \in [k]}$, where $\mathcal{D}_i$ is a distribution over the $i$-set of non-redundant edges $E_i$. On a word $w$ and input $i \in [k]$ the decoder $\mathcal{D}$ picks a pair $(j_1,j_2) \in E_i$ according to the distribution $\mathcal{D}_i$ and recovers the $i$th message entry in the following way. If $j_1 \neq j_2$ then $c' \cdot G_{j_1} + c'' \cdot G_{j_2} = e_i$ for some $c',c'' \in \mathbb{F} \setminus \{0\}$, and the message bit is recovered by $c' \cdot w_{j_1} + c'' \cdot w_{j_2}$. Otherwise, $j_1 = j_2$ and then $c' \cdot G_{j_1} = e_i$ for some $c' \in \mathbb{F} \setminus \{0\}$, and the message bit is recovered by $c' \cdot w_{j_1}$.

We are ready to prove Lemma 93.

Proof of Lemma 93. Let $i \in [k]$. Let $E_i$ be an $i$-set of non-redundant edges as in Claim 98. Let $T_i = ([n],E_i)$ be an undirected graph, where $[n]$ is a set of nodes and $E_i$ is a set of edges. Let $\mathcal{D}_i$ be a distribution over $E_i$ as in Claim 98, i.e., the probability that the edge $(j_1,j_2)$ is chosen is $\mathcal{D}_i(j_1,j_2)$.

Let $L \subseteq [n]$ be a maximal independent set in the graph $T_i$ and let $\alpha_i > 0$ be such that $|L| = \alpha_i n$. Let $R = [n] \setminus L$ and note that $|R| = (1 - \alpha_i)n$. 82
Notice that \((L, R)\) is a partition of \([n]\) and by definition there are no edges going from \(L\) to \(L\). \(^5\)

We argue that \(1 - \alpha_i \geq \frac{\delta}{1 - \frac{|F|}{|F| - 1} \cdot \varepsilon}\). We consider the following sampling. A set \(R_0 \subseteq R\) is selected uniformly at random such that \(|R_0| = \min\{|R|, \delta n\}\), and independently, the edge \((j_1, j_2) \in T_i\) is sampled according to \(D_i\). Let \(\text{Ind}\) be an indicator variable for the event \(R_0 \cap (j_1, j_2) \neq \emptyset\). Then,

\[
\mathbb{E}[\text{Ind}] = \Pr[\text{Ind} = 1] = \sum_{(j_1, j_2) \in E_i} D_i(j_1, j_2) \cdot \Pr[j_1 \in R_0 \lor j_2 \in R_0] = \\
\sum_{(j_1, j_2) \in E_i} D_i(j_1, j_2) \cdot \frac{\delta n}{(1 - \alpha_i)n} = \frac{\delta}{1 - \alpha_i} \cdot \sum_{(j_1, j_2) \in E_i} D_i(j_1, j_2) = \frac{\delta}{1 - \alpha_i}.
\]

Let \(R_0\) such that \(|R_0| \leq \delta n\) be a subset which achieves (at least) this expectation, i.e.,

\[
\sum_{(j_1, j_2) \in E_i} D_i(j_1, j_2) \cdot \Pr[j_1 \in R_0 \lor j_2 \in R_0] \geq \frac{\delta}{1 - \alpha_i}.
\]

Change every symbol in \(R_0\) independently to uniformly chosen random element of \(F\), i.e., every symbol from \(R_0\) is independently and uniformly distributed in \(F\). Then, the probability that the decoder will not recover correctly the \(i\)th message symbol is at least \(\frac{|F| - 1}{|F|} \cdot \frac{\delta}{1 - \alpha_i}\). \(^6\) But the mistake of the decoder must be at most \(1 - \left(1 - \frac{1}{|F|} + \varepsilon\right) = \frac{|F| - 1}{|F|} - \varepsilon\). Hence \(\frac{|F| - 1}{|F|} \cdot \frac{\delta}{1 - \alpha_i} \leq \frac{|F| - 1}{|F|} - \varepsilon\). Hence \(\frac{\delta}{1 - \alpha_i} \leq 1 - \frac{|F|}{|F| - 1} \cdot \varepsilon\). We conclude that \(1 - \alpha_i \geq \frac{\delta}{1 - \frac{|F|}{|F| - 1} \cdot \varepsilon}\).

Let \(M_i \subseteq E_i\) be a maximal matching (self loops are allowed). We argue that \(|M_i| \geq (1 - \alpha_i)n/2\). The vertices left uncovered by \(M_i\) must be an independent set, since for an edge between any of these vertices would allow us to increase the size of the matching at least by one. Since the size of the maximal independent set is \(\alpha_i n\) it follows that the number of vertices covered by \(M_i\) is at least \((1 - \alpha_i)n\). Since every edge of \(M_i\) covers at most two vertices (self-loop covers only one vertex) we have \(|M_i| \geq (1 - \alpha_i)n/2\).

\(^5\)Note that the graph might be not bipartite.

\(^6\)Here we used an assumption that \(E_i\) is an \(i\)-set of non-redundant edges, since a decoder uses both endpoints of an edge to recover a message bit and a change in any of this endpoint affects its recovery output.
Thus $|M_i| \geq \frac{(1-\alpha_i)n}{2} \geq \frac{1}{2} \cdot \frac{\delta n}{1-\frac{\delta n}{|p|+1}}$ and recall that for every $(j_1, j_2) \in M_i$ we have $e_i \in \text{span}\{G_{j_1}, G_{j_2}\}$.

\section{6.6 Proofs of folklore statements}

This section contains two statements used earlier in the paper, the proofs of which we view as folklore. We present these results and their proofs for the sake of completeness.

\subsection{6.6.1 Query reduction}

The following theorem (its proof is folklore) stresses the importance of obtaining lower bounds on 3-query LTCs.

\textbf{Theorem 99 (Folklore)} If there exists an asymptotically good family of LTCs then there exists an asymptotically good family of 3-query LTCs. Equivalently, if there is no asymptotically good family of 3-query LTCs then there is no asymptotically good family of LTCs.

The proof of Theorem 99 follows from the following folklore proposition (related claims appeared, e.g., in \cite[Lemma 3.8]{12}, \cite[Proposition 25]{11}, \cite[Theorem 6.11]{59}). For the sake of completeness we present the proof sketch of Proposition 100.

For linear codes $C' \subseteq \mathbb{F}^{n'}$ and $C \subseteq \mathbb{F}^n$ we say that $C$ is a puncturing of $C'$ if there exists a set $I \subseteq [n']$ such that $C = C'|_I$.

\textbf{Proposition 100 (Query Reduction)} If $C \subseteq \mathbb{F}^n$ is a $(q, \varepsilon, \delta)$-LTC and $k = \dim(C)$ then there exist constants $\alpha, \varepsilon', \delta > 0$, which depend only on $q, \varepsilon, \delta$, and a $(3, \varepsilon', 1.01\delta)$-LTC $C'$ such that $\text{rate}(C') \geq \alpha \cdot \text{rate}(C)$ and $\delta(C') \geq 0.99 \cdot \delta(C)$. Moreover, the code $C$ is a puncturing of $C'$.

\textbf{Proof Sketch.} By Definition 2 there exists a distribution $\mathcal{D}$ over $C_{\leq q}^\perp$ such that for every $w \in \mathbb{F}^n$ such that $\delta(w, C) \geq \delta$ we have $\Pr_{u \sim \mathcal{D}}[\langle u, w \rangle \neq 0] \geq \varepsilon$. Let $t = \frac{100n}{\varepsilon}$ and let $S_{\mathcal{D}} = \{u_1, ..., u_t\} \subseteq C_{\leq q}^\perp$ be a multiset, where each $u_i$ is obtained by sampling independently from $\mathcal{D}$. Let $\mathcal{D}'$ be a uniform distribution over the multiset $S_{\mathcal{D}}$. Fix any $w \in \mathbb{F}^n$ such that $\delta(w, C) \geq \delta$. Note that for every $u_i \in S_{\mathcal{D}}$ the event $\langle u_i, w \rangle \neq 0$ occurs
with probability at least \( \varepsilon \). The Chernoff bound and the Union bound imply the existence of a set \( S_D \subseteq C^{1 \leq q} \) and a distribution \( D' \) such that for every \( w \in \mathbb{F}^n \), \( \delta(w, C) \geq \delta \) we have \( \Pr_{w \sim D'}[(u, w) \neq 0] \geq \varepsilon/10 \) and \( |S_D| \leq t \). Fix such \( S_D \) and \( D' \).

Notice that every constraint \( u \in S_D \) can be represented as \( \gamma_1 \cdot X_{i_1} + \gamma_2 \cdot X_{i_2} + \ldots + \gamma_{q'} \cdot X_{i_{q'}} = 0 \), where \( q' \leq q \), \( \text{supp}(u) = \{i_1, \ldots, i_{q'}\} \) and \( \gamma_1, \ldots, \gamma_{q'} \in \mathbb{F} \).

Let us define new symbols. For every such constraint \( u \in S_D \) we define at most \( q - 3 \) new codeword symbols \( Y_1, Y_2, \ldots \) and the constraints \( \gamma_1 \cdot X_{i_1} + \gamma_2 \cdot X_{i_2} + Y_1 = 0 \), \(-Y_1 + \gamma_3 \cdot X_{i_3} + Y_2 = 0 \), \(-Y_2 + \gamma_3 \cdot X_{i_3} + Y_3 = 0 \) etc. We say that these new constraints are used to split the original constraint \( u \).

Let us call the codeword symbols of \( C \) — those indexed by \([n]\) — the original symbols and the symbols \( Y_1, \ldots \) defined above are the new symbols. Since \( |S_D| \leq t \) we conclude that the total number of new symbols is bounded by \( t \cdot q = O(n) \).

The code \( C_1 \) is obtained from \( C \) by repeating \((1000/\delta) \cdot q \cdot (t/n)\) times all codeword symbols of \( C \) (original symbols). Note that \( \delta(C_1) = \delta(C) \) and \( \text{rate}(C') \geq \frac{\text{rate}(C)}{(1000/\delta)q \cdot (t/n)} \). Then there exists \( \varepsilon_1 > 0 \) which depends only on \( \varepsilon, \delta \) such that \( C_1 \) is a \((q, \varepsilon_1, 1.001\delta)\)-LTC, where the tester of \( C_1 \) with probability \( 1/2 \) invokes the original \( q \)-query tester to check the validity of original symbols and with probability \( 1/2 \) checks that a repetition is “legal” (using only 2 queries) (see [19]). Note that \( C_1|_{[n]} = C \). Let \( R \) be a set of all codeword indices of the code \( C_1 \).

The code \( C' \) is obtained from \( C_1 \) by appending all new symbols to the code \( C_1 \). Clearly \( \delta(C') \geq 0.99\delta(C) \) and \( \text{rate}(C') \geq \frac{\text{rate}(C)}{10^{\delta \cdot q} \cdot (t/n)} \). We have \( C'|_R = C_1 \). We argue that \( C' \) is a \((3, \varepsilon', 1.01\delta)\)-LTC, where \( \varepsilon' > 0 \) depends only on \( q, \varepsilon, \delta \). The tester of \( C' \) will sample the tester of \( C_1 \). If the tester of \( C_1 \) picks a constraint of weight at most 3 then the tester of \( C' \) checks the same constraint. Otherwise, the tester of \( C_1 \) picks a constraint of weight \( q \geq 4 \) (let us call it \( w \)), then the tester of \( C' \) checks only one random constraint of weight 3 used to split the constraint \( w \). Let \( w \) be a word that is \( 1.01\delta \)-far from \( C' \). Then \( \delta(w|_R, C'|_R) = \delta(w|_R, C_1) \geq 1.001\delta \). Hence the tester of \( C_1 \) rejects \( w|_R \) with probability at least \( \varepsilon_1 \). Thus the tester of \( C' \) rejects \( w \) with probability at least \( \varepsilon_1/q \). Let \( \varepsilon' = \varepsilon_1/q \). We conclude that the tester of \( C' \) rejects \( w \) with probability at least \( \varepsilon' > 0 \) and recall that \( \varepsilon' \) depends only on \( q, \varepsilon, \delta \). \( \blacksquare \)

85
Proposition 100 implies that every LTC can be converted to a 3-query LTC over the same field (with only a constant factor loss in parameters). Hence we conclude Theorem 99.

6.6.2 Transitive codes are regular

Claim 101 Let \( C \subseteq \mathbb{F}_2^n \) be a code. If \( C \) is 1-transitive then \( C \) is \( q \)-regular for every \( q > 0 \).

Proof. For \( l \in [n] \) and \( q > 0 \) let \( T^q_l = \{ u \in C_q^\perp \mid l \in \text{supp}(u) \} \). It is sufficient to argue that for every \( i, j \in [n] \) and \( q > 0 \) we have \( |T^q_i| = |T^q_j| \).

Assume the contrary, i.e., there exist \( i, j \in [n] \) and \( q > 0 \) such that \( |T^q_i| > |T^q_j| \). Let \( G \) be a 1-transitive group such that \( C \) is invariant under \( G \). For \( \pi \in G \) let \( \pi(T^q_i) = \{ \pi(u) \mid u \in T^q_i \} \). Note that for all \( \pi \in G \) we have \( |T^q_i| = |\pi(T^q_i)| \). Let \( \pi \in G \) be such that \( \pi(i) = j \) (such \( \pi \) exists since \( G \) is 1-transitive). It holds that \( \pi(T^q_i) \subseteq T^q_j \) since for all \( u \in \pi(T^q_i) \) we know that \( j \in \text{supp}(u) \) and \( u \in C_q^\perp \). This implies that \( |T^q_i| = |\pi(T^q_i)| \leq |T^q_j| \). Contradiction. \( \blacksquare \)
Chapter 7

Bibliography


Bibliography


90


لكראת חטמי תחתונים על קודים
הניתנים לבדיקת לוקליות

מיכאל ידרמן
לקראת חסמים תחתונים על קדימה
הניתנים לבדיקהconciliation

חיבורי על מחקר

לשם מלולי חלקי של הדרישה לƙבבלת התנהר
דוקטור לפילוסופיה

מיכאל ידרמן

הנהל לסנט הטכנולוגי - מכון טכנולוגי לישראל
гибしたら ניוואר 2012
חיפה שבט תשע"ב

Technion - Computer Science Department - Ph.D. Thesis PHD-2012-02 - 2012
המחקר נעשה בהנחיית פרופסור ח erb אל־בריהון בפקולטה למדעי המחשב.

אני מודע לטכניון על התמיכת התמחויות הנדיבת בשיתוף הון.

Technion - Computer Science Department - Ph.D. Thesis PHD-2012-02 - 2012
תקציר

הוא אחד hakkו כっこ מספר בולטים של תורת הסיבוכיות. ההוכחה שיתוף PCP
משפט נתחיה שיתוף PCP אשר ההוכחה מספקת עבור קיומו של מודא בודק
שהוכחה שיתוף PCP לא מצאה את ההוכחה. המודא ידיע את הביטים בכוקה אפקטיבית. ממספח PCP
הוכחה PCP בפרוג נולמי לכל скаענו של גרף זרום שיתוף השפה - NP.

המפה מאפיו פומקבר ריבי כנין סמסים הורותים לאלאורית הירוב [33, 45]自贸ת
יחסיום כנין PCP וsembledיםיה [6, 23, 54, 60] шן פז
PCP נattivitàים כנין הוכחות [37, 64] ד"ה PCP

jisת בדיאת המסה את הקסנון של להכנתו של הדיקוק
משפס ביי שחר בראים פילחה, אנו קוס שטומיתים ממקוצד הקולק
(ח"ניילרים). בשורת הקולק ינוי צולק בודק התוכנה נדיר בודק ג'ין הקוקה ג'ין הקוקה. הקוקה
חקירת התוכנה ינוי צולק בודק התוכנה ינוי צולק. הדיקוק
ליאן הקוקה ג'ין הקוקה ג'ין הקוקה ג'ין הקוקה. הקוקה
כמספי בטיות האנמטיה איזורין של הקוקה בודק של הקוקה.
בשורת הקוקה הלאו כחציו בודק התוכנה ינוי צולק הלאו
בעיתภタイム טוחן בתוכנה. הדיקוק
ליאן הקוקה ג'ין הקוקה ג'ין הקוקה. הקוקה
כמספי בטיות האנמטיה איזורין של הקוקה בודק של הקוקה.
הלינאריה היא מידה ניירות המגדירה את מידת מימוד של המרחבים הליניארים. הקובעים של הקובעה הם יחסים בין מימוד למקוללת הקובעים. הממידה הממוצעת של האורח המידה היא מדריך ליניארי בין צורת הקובעים. קובעים של קובעים הם של קובעים של צורות של קובעים. הממידה הממוצעת של הקובעים של צורות של קובעים היא ממוצע של הממידות הממוצעות של צורות של קובעים. הממידה הממוצעת של הקובעים של צורות של קובעים היא ממוצע של הממידות הממוצעות של צורות של קובעים.
אברת תחאצל_uploaded כדי לה髻ק. אתגרע מראים שבודק חיי לחטפם ב.mobile הרבע
ויור דודל של הבורקון מהרבר, לכל שמיתכין לש חקוק דודל יוחר הבורק פא
ויור דיבור. הטענה היא בתוכית חטפהحمل יוחר שבアウトמקים. הנשט
אומר שלכל ביסיס לודר הדואר מתויך נטוסק לש הבורקון הממורע את התתחسس
ויור ליצאי ימיד לש חקוק. כלומר כל שמות יוחר יחר הבורק
ויור דיבור. הטענההזאתש המורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
שה הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
ויה לש חקוק ישינית לבודק בע מיל יצריא ואהיה לש הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
ככל ביסיס לינו ליצאי בורק. אתגרע מראים שהמסקנההזאתתה使う והעזר埽 לש הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
(collaboration) לש מיל יצריא על מיל יצריא לש הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
ביחד אתרטנול לעטס מיססר-ליצאי של מיל יצריא לברקונ藍שכל יוצר
ב. ב. קאטזא בודק קוד יצריא לברקונם. אתגרע מראים שהמסקנההזאתתה使う והעזר埽 לש הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
שאלה הזאתהשהא彧 המיל יצריא לש הבורקון הממורע את התתחسس לינו לודר. ד. לודר, וא. אנחלית את
א. פא. התביאה היאแชארה נשלאה השכ限りיה התתחסיס אורי התוח האות.

ח. ב. - קוד יצריא לבודק קוד יצריא לבודק קוד יצריא לש הבורקון
הお話אה שומצנת בחלק היא מחבסוח על
בוה קיון יצריא יצריא לבודק קוד יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודק יצריא לבודק יצריא לש הבורקון
שה קוד יצריא יצריא לבודк
את הנגשו הכללות כי לכלול את הקודים ושהקודים הפרימים שיתופי באופן בדיקות על מספור אותם בקواب, מתח פסוקים בקواب קוד כל שעה מרשימה של קוד
שנת כולם כל את בקواب jap. אוסף שערת עם כותרת כל בקواب 3 שביעות כשת מקוד
לקוד המקורים הזもらう לקוד לקוב 유지ים עם בקواب הז制度改革 היה עם מקוד לקוב
קנב. כל המספים שלכל לקו עם שערית ושהקודים בקواب 3 שביעות, המקוד לקוב 유지ים בקواب המקודים הז制度改革 את הקודים עם מקוד תצוגה בקواب 3 שביעות, מקוד
שתה התנועה הבאה: נgross שערית בקواب C שערית בקواب את 3 שביעות, מקוד
לנייארי בקواب: אוסף מספור ותוכחת שית התנועה הבאה:

4. מכיל מספור סופר-ליניארי של מילים דואליות במשקק לכל היהות.

אנו ממקים שזורניהנקזגה בחלק ראשון של התשובה ושהקודים הפרימים עם במספר שלם דואליות במשקק 3 מתפליוג בידע "מספרים" מספרים
אנו מקודים או woll מקודות מקודות פסא. אנו מקודים אנ_tCחむ ת_hppות התויל
והנה=top הירוק או מקודים בודג ליניארי כ��ירמקוד עם שעריתמקוד
מספור "רובה" מספור פסא בקواب של מילים דואליות במשקק לכל היהות.

לכלכל כל התשובה והלאנו עם מספרים אשת התפשית התיתוניה הקניון על האורכים של
קודים שיתופי לשון לכל 2 שביעות. לתרחוף כל השערת השם הקוד
新京 בקواب את 3 שביעות בשעת התפשית בשעת האורכים של הקניון
新京 לשון לכל 2 שביעות מספור מקוד לשערית. בבלב היה新京
לתתשומת תחשום התיתוניה על אורכים קוד שיתופי לשון לכל 2 שביעות.
לשהית את התוحضנה.