Electronic Colloquium on Computational Complexity, Report No. 44 (2005)

Locally Decodable Codes with 2 queries and Polynomial Identity Testing for depth 3 circuits

Zeev Dvir* Amir Shpilka†

Abstract

In this work we study two, seemingly unrelated, notions. *Locally Decodable Codes* (LDCs) are codes that allow the recovery of each message bit from a constant number of entries of the codeword. *Polynomial Identity Testing* (PIT) is one of the fundamental problems of algebraic complexity: we are given a circuit computing a multivariate polynomial and we have to determine whether the polynomial is identically zero. We improve known results on locally decodable codes and on polynomial identity testing and show a relation between the two notions. In particular we obtain the following results:

1. We show that if $E : \mathbb{F}^n \mapsto \mathbb{F}^m$ is a linear LDC with 2 queries then $m = exp(\Omega(n))$. Previously this was only known for fields of size $<< 2^n$ [GKST01].
2. We show that from every depth 3 arithmetic circuit ($\Sigma\Pi\Sigma$ circuit), $\mathcal{C}$, with a bounded (constant) top fan-in that computes the zero polynomial, one can construct a locally decodeable code. More formally: Assume that $\mathcal{C}$ is minimal (no subset of the multiplication gates sums to zero) and simple (no linear function appears in all the multiplication gates). Denote by d the degree of the polynomial computed by $\mathcal{C}$ and by r the rank of the linear functions appearing in $\mathcal{C}$. Then we can construct a linear LDC with 2 queries, that encodes messages of length $r/polylog(d)$ by codewords of length $O(d)$.
3. We prove a structural theorem for $\Sigma\Pi\Sigma$ circuits, with a bounded top fan-in, that compute the zero polynomial. In particular we show that if such a circuit is simple and minimal and of polynomial size then its rank, r, is only polylogarithmic in the number of variables (a priory it could have been linear).
4. We give new PIT algorithms for $\Sigma\Pi\Sigma$ circuits with a bounded top fan-in:
   (a) A deterministic algorithm that runs in quasipolynomial time.
   (b) A randomized algorithm that runs in polynomial time and uses only polylogarithmic number of random bits.

   Moreover, when the circuit is multilinear our deterministic algorithm runs in polynomial time. Previously deterministic subexponential time algorithms for PIT in bounded depth circuits were known only for depth 2 circuits (in the black box model) [GKS90, BOT88, KS01]. In particular, for the special case of depth 3 circuits with 3 multiplication gates our result resolves an open question asked by Klivans and Spielman [KS01].

*Department of Computer Science, Weizmann institute of science, Rehovot, Israel.
Email: zeev.dvir@weizmann.ac.il.

†Department of Computer Science, Weizmann institute of science, Rehovot, Israel.
Email: amir.shpilka@weizmann.ac.il. This work was supported by the Koshland fellowship.

ISSN 1433-8092

1 Introduction

Locally Decodable Codes (LDCs) are error correcting codes that allow the recovery of each symbol of the message from a constant number of entries of the codeword. Polynomial Identity Testing (PIT) is one of the fundamental problems of algebraic complexity: we are given a circuit computing a multivariate polynomial and we have to determine whether the polynomial is identically zero. In this paper we show a relation between this two notions - roughly, from every depth 3 circuit which is identically zero, one can construct a locally decodable code. Using this relation and a new lower bound on LDCs, we devise new PIT algorithms for depth 3 circuits.

1.1 Locally Decodable Codes

Locally decodable codes are error correcting codes that allow the recovery of each symbol of the message, from a corrupted codeword, by looking at only a constant number of entries of the corrupted word. Roughly, a (q, δ, ϵ)**-locally decodeable code** encodes $x \in \mathbb{F}^n$ to $E(x) \in \mathbb{F}^m$, such that for each index $i \in [n]$, x_i can be recovered from $E(x)$ with probability[1] $> \frac{1}{|\mathbb{F}|} + \epsilon$ by reading only q (random) entries, even if $E(x)$ was corrupted in δm positions.

Locally decodable codes have many applications - they are related to private information retrieval (PIR) schemes [CGKS95, KT00, GKST01], they can be used for amplification of hardness [GL89, GRS00, AGS03] and for the construction of hard-core predicates for one-way permutations [Lev87, FF93] (see [Tre04] for a survey on LDCs).

The notion of Locally decodable codes was explicitly discussed in [BFLS91] and explicitly defined in [KT00]. Implicit constructions of local decoders can be found in the context of random self reducibility and self correcting computations (see e.g. [Lip90, BF90, GLR$^+$91, GS92, FF93]). There are two main questions related to LDCs: Finding explicit constructions and proving limits of such constructions (i.e. proving lower bounds on the length of the encoding). Explicit constructions were given by [BFLS91, BI01, BIKR02]. The best current construction is due to Beimel et al [BIKR02] who gave an LDC with q queries of length $m = exp(n^{O(\log\log q/q \log q)})$.

The problem of proving lower bounds was first studied by Katz and Trevisan [KT00] who proved that for every LDC with q queries, the length of the codeword, m, is at least $n^{1+\frac{1}{q-1}}$. This is currently the best lower bound for general LDCs (see also [DJK$^+$02]). It is a very challenging open question to give tight lower bounds (or upper bounds) on the length of LDCs. Due to the difficulty of the problem many works focused on the case of codes with two queries ($q = 2$). Exponential lower bounds were first proved for linear codes [GKST01, Oba02] and then, by techniques from quantum computation, for non-linear codes over $GF(2)$ [KdW03]. The bound of Goldreich et al [GKST01] actually holds for linear LDCs with 2 queries over any finite field, namely that m is at least $2^{\Omega(n)-\log(|\mathbb{F}|)}$, where $\mathbb{F}$ is the underlined field. This result is (nearly) tight when the field is of constant size, however it gives no significant bound for infinite fields.

1.2 Polynomial Identity Testing

Polynomial Identity Testing (PIT) is a fundamental problem in algebraic complexity: We are given a multivariate polynomial (in some representation) over some field $\mathbb{F}$ and we have to determine

[1] If $\mathbb{F}$ is infinite then the probability of success is $> \epsilon$.

whether it is identically zero[2]. The importance of this problem follows from its many applications: Algorithms for primality testing [AB03, AKS02], for deciding if a graph contains a perfect matching [Lov79, MVV87, CRS95] and more, are based on reductions to the PIT problem (see the introduction of [LV98] for more applications).

Determining the complexity of PIT is one of the greatest challenges of theoretical computer science. It is one of a few problems (and in some sense PIT is the most general problem) for which we have *coRP* algorithms but no deterministic subexponential time algorithms. Recently Kabanets and Impagliazzo [KI03] suggested an explanation for the lack of algorithms. They showed that efficient deterministic algorithms for PIT imply that $NEXP$ does not have polynomial size arithmetic circuits. Specifically, if PIT has deterministic polynomial time algorithms then either the Permanent cannot be computed by polynomial size arithmetic circuits or $NEXP \not\subset P/poly$.

The first randomized algorithm for PIT was discovered independently by Schwartz [Sch80] and Zippel [Zip79]. Their well known algorithm simply evaluates the polynomial at a random point and accepts iff the polynomial vanishes at the point. If the polynomial is of degree d and each variable is randomly chosen from a domain S, then the error probability is bounded by $d/|S|$. Two kind of works followed the Schwartz-Zippel algorithm: Randomized algorithms that use fewer random bits [CK97, LV98, AB03] and algorithms for restricted models of arithmetic circuits. In [GKS90, BOT88, KS01] polynomial time deterministic PIT algorithms for depth 2 arithmetic circuits were given. More recently, [RS04] gave a polynomial time PIT algorithm for non-commutative formulas. All algorithms, with the exception of [AB03, RS04], are black box algorithms. That is, these algorithms do not have access to a circuit computing the polynomial and they can only evaluate it on different inputs (as in the Schwartz-Zippel algorithm).

A result of a different nature was proved by Kabanets and Impagliazzo [KI03]. They designed a deterministic quasipolynomial time algorithm based on unproved hardness assumptions.

1.3 Depth 3 arithmetic circuits

Proving lower bounds for general arithmetic circuits is the greatest challenge of algebraic complexity. Unfortunately, except for the lower bounds of Strassen [Str73] and Baur-Strassen [BS83], no lower bounds are known for general arithmetic circuits. Due to the difficulty of the problem research focused on restricted models such as monotone circuits and bounded depth circuits. Exponential lower bounds were proved on the size of monotone arithmetic circuits [SS77, JS80], and linear lower bounds were proved on their depth [SS80, TT94]. However, unlike the situation in the boolean case, only weak lower bounds were proved for bounded depth arithmetic circuits [Pud94, RS01]. Thus, a more restricted model was considered - the model of depth 3 arithmetic circuits (also known as $\Sigma\Pi\Sigma$ circuits). A $\Sigma\Pi\Sigma$ circuit computes a polynomial of the form

$$\mathcal{C} = \sum_{i=1}^{k} \prod_{j=1}^{d_i} L_{ij}(x), \tag{1}$$

where the L_{ij}'s are linear functions. Grigoriev and Karpinski [GK98] and Grigoriev and Razborov [GR98] proved exponential lower bounds on the size of $\Sigma\Pi\Sigma$ circuits computing the Permanent and Determinant over finite fields. Over infinite fields exponential lower bounds are known only for the

[2]Note that we want the polynomial to be identically zero and not just to be equal to the zero function. For example, $x^2 - x$ is the zero function over $GF(2)$ but not the zero polynomial.

restricted models of *multilinear*[3] $\Sigma\Pi\Sigma$ circuits and for *homogeneous* $\Sigma\Pi\Sigma$ circuits [Nis91, NW95]. For general $\Sigma\Pi\Sigma$ circuits over infinite fields only the quadratic lower bound of [SW99] is known. Thus, proving exponential lower bounds for $\Sigma\Pi\Sigma$ circuits over $\mathbb{C}$ is a major open problem in arithmetic circuit complexity.

In this work we are interested in the problem of polynomial identity testing for depth 3 circuits. As mentioned earlier there are no efficient PIT algorithms for arithmetic circuits, even if we just consider bounded depth circuits. Thus, finding efficient algorithms for PIT in $\Sigma\Pi\Sigma$ circuits seems like the first step towards proving more general results.

1.4 Our Results

Lower Bounds for Linear Locally Decodable Codes with 2 Queries
We study linear LDCs with 2 queries over arbitrary fields and prove lower bounds on their length. The first such lower bound was proved by Goldreich et al [GKST01]:

Theorem 1.1 (thm 1.4 of [GKST01]). *Let $\delta, \epsilon \in [0,1]$, $\mathbb{F}$ a field, and let $E : \mathbb{F}^n \to \mathbb{F}^m$ be a linear $(2, \delta, \epsilon)$-locally decodeable code. Then*

$$m \geq 2^{\frac{\epsilon\,\delta\,n}{16} - 1 - \log_2 |\mathbb{F}|}.$$

Note that this result only makes sense when $|\mathbb{F}|$ is finite. We prove the following theorem.

Theorem 1.2. *Let $\delta, \epsilon \in [0,1]$, $\mathbb{F}$ a field, and let $E : \mathbb{F}^n \to \mathbb{F}^m$ be a linear $(2, \delta, \epsilon)$-locally decodeable code. Then*

$$m \geq 2^{\frac{\epsilon\,\delta\,n}{4} - 1}.$$

Compared with Theorem 1.4 of [GKST01] our result removes the dependance on the size of the field in the exponent and works for every field size, finite and infinite. The idea of the proof is similar to the one in [GKST01] - we show that given a linear 2-LDC over an arbitrary field $\mathbb{F}$ we can construct from it a linear 2-LDC over $GF(2)$, with almost the same parameters, and then we use the lower bound of [GKST01] for codes over $GF(2)$.

Relation between Depth 3 circuits and Locally Decodeable Codes
The main result of the paper is that from every $\Sigma\Pi\Sigma$ circuit, that computes the zero polynomial, one can construct a linear LDC with 2 queries. Relations between arithmetic circuits and error correcting codes were known before [Bsh89, Shp03], however this is the first time that LDCs appear in the context of arithmetic circuits. More formally, let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit, as in equation 1, computing the zero polynomial. We say that $\mathcal{C}$ is minimal if no subset of the multiplication gates sums to zero. We say that $\mathcal{C}$ is simple if there is no linear function that appears in all the multiplication gates (up to a multiplicative constant). Denote with r the rank of the linear functions appearing in $\mathcal{C}$.

Theorem 1.3. *Let $k \geq 3$, $d \geq 2$, and let $\mathcal{C} \equiv 0$ be a simple and minimal $\Sigma\Pi\Sigma$ circuit of degree d, with k multiplication gates and n inputs. Then we can construct a linear $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable*

[3] More accurately for pure multilinear $\Sigma\Pi\Sigma$ circuits.

code $E : \mathbb{F}^{n_1} \to \mathbb{F}^{n_2}$*, with*

$$\frac{r}{2^{O(k^2)} \log(d)^{k-3}} \leq n_1 \quad \text{and} \quad n_2 \leq k \cdot d.$$

Thus, if k is a constant then we can construct a linear $(2, \frac{1}{12}, \frac{1}{4})$-LDC that encodes messages of length $r/polylog(d)$ by codewords of length $O(d)$. As a corollary of theorem 1.2 and theorem 1.3 we get:

Theorem 1.4. *Let $k \geq 3$, $d \geq 2$, and let $\mathcal{C} \equiv 0$ be a simple and minimal $\Sigma\Pi\Sigma$ circuit of degree d with k multiplication gates and n inputs, then $r \leq 2^{O(k^2)} \log(d)^{k-2}$.*

Notice that the bound on r depends only on the degree and the number of multiplication gates and not on the number of variables! If the degree is polynomial in n (i.e. the circuit is of polynomial size) then the rank is bounded by $polylog(n)$, where a priory the rank could have been n.

PIT algorithms for depth 3 circuits

We design algorithms for PIT of depth 3 circuits with a constant number of multiplication gates. In particular we get a deterministic quasipolynomial time algorithm, and a randomized polynomial time algorithm that uses only polylog random bits. If the circuit is multilinear, i.e. every multiplication gate computes a multilinear polynomial, then we give a deterministic polynomial time algorithm for PIT. Our algorithms are non black-box - all of them use the circuit computing the polynomial. We prove the following result.

Theorem 1.5. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit of degree d, with k multiplication gates and n inputs. Then we can check if $\mathcal{C} \equiv 0$:*

1. *Deterministically, in time $\exp\left(2^{O(k^2)} \log^{k-1}(d)\right)$. Thus, for a constant k the running time is $exp(polylog(d))$.*
2. *Probabilistically, in time $2^{O(k)} poly(d, \frac{1}{\epsilon})$, using $2^{O(k^2)} log^{k-2}(d) \log(1/\epsilon)$ random bits, with error probability ϵ. For constant k the running time is $poly(d, \frac{1}{\epsilon})$ and the number of random bits is $polylog(d) \log(1/\epsilon)$.*
3. *If $\mathcal{C}$ is also multilinear then we can check if $\mathcal{C}$ is identically zero deterministically in time $\exp(2^{O(k^2)}) \cdot poly(d)$. For constant k the running time is $poly(d)$.*

Prior to our work the only algorithms that were designed for bounded depth circuits were the deterministic algorithm of [RS04] for pure multilinear depth 3 circuits, and the black box algorithms of [GKS90, BOT88, KS01] for polynomials computed by depth 2 circuits (also known as sparse polynomials). None of the algorithms for sparse polynomials work in the case of depth 3 circuits, as such circuits can compute polynomials with exponentially many monomials. In fact, Klivans and Spielman [KS01] ask whether one could derandomize PIT for $\Sigma\Pi\Sigma$ circuits with only 3 multiplication gates (k=3 in our notations). We give a deterministic algorithm that runs in quasipolynomial time for this case, thus resolving the question of [KS01].

1.5 Organization

In Section 2 we analyze linear locally decodable codes, and derive Theorem 1.2. Section 3 is devoted to $\Sigma\Pi\Sigma$ circuits and their properties, and serves as an introduction to the main part of the paper. In Section 4 we give the proof of Theorem 1.3, and discuss the relation between $\Sigma\Pi\Sigma$ circuits and locally decodable codes. Finally, in Sections 5 and 6 we use our results to prove a structural theorem for zero $\Sigma\Pi\Sigma$ circuits, and devise PIT algorithms based on this theorem.

2 Locally Decodable Codes

In this section we prove Theorem 1.2. We start by formally defining locally decodable codes.

For a natural number n, let $[n] \triangleq \{1, \ldots, n\}$. Let $\mathbb{F}$ be a field. For a vector $x \in \mathbb{F}^n$ we write x_i for the i'th coordinate of x. We denote by e_i the i'th unit vector. For two vectors $y, z \in \mathbb{F}^m$, denote by $\Delta(y, z)$ the number of coordinates in which y and z differ.

Definition 2.1. *Let $\delta, \epsilon \in [0,1]$, q an integer. We say that $E : \mathbb{F}^n \to \mathbb{F}^m$ is a (q, δ, ϵ)-**locally decodeable code** if there exists a probabilistic oracle machine A such that:*

- *In every invocation, A makes at most q queries (non-adaptively).*
- *For every $x \in \mathbb{F}^n$, for every $y \in \mathbb{F}^m$ with $\Delta(y, E(x)) < \delta m$, and for every $i \in [n]$, we have*

$$\begin{aligned} |\mathbb{F}| < \infty: &\quad \boldsymbol{Pr}[A^y(i) = x_i] \geq \tfrac{1}{|\mathbb{F}|} + \epsilon \\ |\mathbb{F}| = \infty: &\quad \boldsymbol{Pr}[A^y(i) = x_i] \geq \epsilon \end{aligned}$$

  where the probability is taken over the internal coin tosses of A.

*We say the the code E is a **linear** code, if E is a linear transformation between $\mathbb{F}^n$ and $\mathbb{F}^m$.*

We are now ready to prove Theorem 1.2. To ease the reading of the paper we repeat its formulation here:

Theorem 1.2 *Let $\delta, \epsilon \in [0,1]$, $\mathbb{F}$ a field, and let $E : \mathbb{F}^n \to \mathbb{F}^m$ be a linear $(2, \delta, \epsilon)$-locally decodable code. Then*

$$m \geq 2^{\frac{\epsilon\,\delta\,n}{4} - 1}.$$

Our proof will build upon the methods of [GKST01], together with a novel reduction from LDCs over arbitrary fields to LDCs over $GF(2)$. We start by reviewing the results of [GKST01]. The first step of their proof, given by lemma 2.2, is a reduction from the problem of proving lower bounds for LDCs, to a graph-theoretic problem. The first such reduction was given in [KT00], where it was used to prove lower bounds on general LDCs. We note that in [GKST01] the lemma was proved only over finite fields, however, it is easy to modify the proof to work for infinite fields as well.

Lemma 2.2. (Implicit in [GKST01]) *Let $E : \mathbb{F}^n \to \mathbb{F}^m$ be a linear $(2, \delta, \epsilon)$-locally decodable code, and let $a_1, \ldots, a_m \in \mathbb{F}^m$ be vectors such that:*

$$E(x) = (\langle a_1, x \rangle, \ldots, \langle a_m, x \rangle)$$

($\langle\cdot,\cdot\rangle$ denotes the standard inner product). Then, for every $i \in [n]$, there exists a set $M_i \subset [m]\times[m]$, of at least $\frac{\epsilon\delta m}{4}$ disjoint pairs, such that for every $(j_1, j_2) \in M_i$, $e_i \in Span\{a_{j_1}, a_{j_2}\}$.

From Lemma 2.2 we see that in order to prove lower bounds for 2-query locally decodable codes, it is sufficient to deal with the more combinatorial setting in which a given multiset of vectors contain many disjoint pairs spanning each unit vector.

The next step in the proof of [GKST01] is a reduction from arbitrary finite fields to $GF(2)$. The next lemma summarizes the reduction given by [GKST01].

Lemma 2.3. (implicit in [GKST01]) *Let $\mathbb{F}$ be a finite field, and let $a_1, \ldots, a_m \in \mathbb{F}^n$. For every $i \in [n]$ let $M_i \subset [m]\times[m]$ be a set of disjoint pairs of indices, such that $e_i \in Span\{a_{j_1}, a_{j_2}\}$ for every $(j_1, j_2) \in M_i$. Then, there exist m' vectors $b_1, \ldots, b_{m'} \in \{0,1\}^n$, and n sets $M'_1, \ldots, M'_n \subset [m']\times[m']$ of disjoint pairs, such that:*

1. *for every $(j_1, j_2) \in M'_i$, $b_{j_1} \oplus b_{j_2} = e_i$.*
2. *$m' = (|\mathbb{F}| - 1)m$.*
3. *$\sum_{i=1}^n |M_i| \leq 2m + \frac{2}{|\mathbb{F}|-1}\sum_{i=1}^n |M'_i|$.*

The third and final step in the proof of [GKST01] is a lemma which bounds the size of the matchings M_i, when the underlying field is $GF(2)$.

Lemma 2.4. ([GKST01]) *Let $a_1, \ldots, a_m$ be elements of $\{0,1\}^n$. For every $i \in [n]$ let $M_i \subset [m]\times[m]$ be a set of disjoint pairs of indices, such that $e_i = a_{j_1} \oplus a_{j_2}$ for every $(j_1, j_2) \in M_i$. Then*

$$\sum_{i=1}^{n} |M_i| \leq \frac{1}{2} m \log(m).$$

Our proof differs from that of [GKST01] only in its second part - the reduction from $\mathbb{F}$ to $GF(2)$. Our reduction holds for any field, in particular for infinite $\mathbb{F}$, and does not involve the field size as a parameter.

Lemma 2.5. *Let $\mathbb{F}$ be any field, and let $a_1, \ldots, a_m \in \mathbb{F}^n$. For every $i \in [n]$ let $M_i \subset [m]\times[m]$ be a set of disjoint pairs of indices, such that $e_i \in Span\{a_{j_1}, a_{j_2}\}$ for every $(j_1, j_2) \in M_i$. Then, there exist m vectors $b_1, \ldots, b_m \in \{0,1\}^n$, and n sets $M''_1, \ldots, M''_n \subset [m]\times[m]$ of disjoint pairs, such that:*

1. *for every $(j_1, j_2) \in M''_i$, $b_{j_1} \oplus b_{j_2} = e_i$.*
2. *$\sum_{i=1}^n |M_i| \leq 2\sum_{i=1}^n |M''_i| + m$.*

Before giving the proof of the Lemma we combine Lemma 2.2, Lemma 2.5, and Lemma 2.4, to prove Theorem 1.2.

Proof of Theorem 1.2. Let $a_1, \ldots, a_m \in \mathbb{F}^n$ be vectors such that

$$E(x) = (\langle a_1, x\rangle, \ldots, \langle a_m, x\rangle).$$

From Lemma 2.2, we know that there exist n sets, $M_1, \ldots, M_n \subset [m] \times [m]$, of disjoint pairs of indices, such that for every $(j_1, j_2) \in M_i$, we have $e_i \in Span\{a_{j_1}, a_{j_2}\}$. We also know that

$$\forall\, i \in [n]\ , |M_i| \geq \frac{\epsilon\, \delta\, m}{4}.$$

Now, let $b_1, \ldots, b_m \in \{0,1\}^n$ and $M_1'', \ldots, M_n'' \subset [m] \times [m]$ be as in Lemma 2.5. That is:

1. for every $(j_1, j_2) \in M_i''$, $b_{j_1} \oplus b_{j_2} = e_i$.
2. $\sum_{i=1}^n |M_i| \leq 2 \sum_{i=1}^n |M_i''| + m$.

Using Lemma 2.4 we now have

$$\sum_{i=1}^{n} |M_i''| \leq \frac{1}{2} m \log(m).$$

This implies

$$n \cdot \frac{\epsilon\, \delta\, m}{4} \leq \sum_{i=1}^{n} |M_i| \leq 2 \sum_{i=1}^{n} |M_i''| + m \leq m \log(m) + m.$$

Which, after division by m, gives the bound stated by the theorem. □

We now give the proof of Lemma 2.5.

Proof of Lemma 2.5: The proof will consist of two stages. First, we will remove a relatively small number of "bad" pairs from the given matchings M_i, then we will transform the vectors $a_1, \ldots, a_m$ to vectors in $\{0,1\}^n$, while preserving a large portion of the pairs spanning the unit vectors.

Let (j_1, j_2) be a pair in M_i for some i, such that either a_{j_1} or a_{j_2} are proportional to the unit vector e_i. w.l.o.g assume $a_{j_1} = c \cdot e_i$. We replace this pair with the pair (j_1, j_1). We do the same for all pairs containing a vector proportional to the unit vector spanned by this pair. This change does not affect the parameters of the lemma, and is done only to simplify the analysis.

Next, we define a function $\theta : \mathbb{F}^n \backslash \{0\} \to [n]$ by

$$\theta(v) = \min\{i : v_i \neq 0\}.$$

For the rest of the proof we assume w.l.o.g that in each pair (j_1, j_2) we have $\theta(a_{j_1}) \leq \theta(a_{j_2})$ (note that we can assume w.l.o.g that the vectors $a_1, \ldots, a_m$ are all different from zero). We remove from each matching M_i all the pairs (j_1, j_2) in which $\theta(a_{j_1}) = i$ (this includes all pairs (j_1, j_1) described in the previous paragraph, and more). Denote the resulting matchings by M_i'. We claim that the total number of pairs removed in this stage is at most m:

Claim 2.6.

$$\sum_{i=1}^{n} |M_i| \leq \sum_{i=1}^{n} |M_i'| + m. \tag{2}$$

Proof. let $p_1 = (j_1, j_2)$ and $p_2 = (k_1, k_2)$ be two removed pairs . If p_1 and p_2 were in the same matching M_i , then they are disjoint, and so $j_1 \neq k_1$. If the pairs belonged to two different

matchings , say M_{i_1} and M_{i_2}, then $\theta(a_{j_1}) = i_1$ and $\theta(a_{k_1}) = i_2$, and again we get that $j_1 \neq k_1$. It follows that every removed pair has a distinct first element in the set $[m]$. Therefore, the total number of removed pairs cannot exceed m. □

In the following we assume w.l.o.g that the first non-zero coordinate of each a_j is one (we can assume that because we are allowed to use arbitrary linear combinations of the a_j's when spanning the e_i's). The next claim asserts an important property of the matchings M'_i.

Claim 2.7. *For every $i \in [n]$,and $(j_1, j_2) \in M'_i$:*

$$e_i \in Span\{a_{j_1} - a_{j_2}\}.$$

Proof. Let $u = a_{j_1}$, $v = a_{j_2}$. We know that there exist two non-zero coefficients $\alpha, \beta \in \mathbb{F}$ such that $\alpha u + \beta v = e_i$ (both coefficients are non-zero because we removed from M_i all pairs containing a vector proportional to e_i). From this property it is clear that $\theta(u) \leq i$ (remember that $\theta(u) \leq \theta(v)$). As we removed all pairs in which $\theta(a_{j_1}) = i$ we conclude that $\theta(u) < i$. This in turn implies that $\theta(u) = \theta(v) < i$, because if $\theta(v) > \theta(u)$, then the vector $\alpha u + \beta v = e_i$ would have a non-zero coordinate in position $\theta(u) < i$. Now, since $v_{\theta(v)} = u_{\theta(u)} = 1$ we have that $\alpha + \beta = (\alpha u + \beta v)_{\theta(u)} = (e_i)_{\theta(u)} = 0$. Hence $e_i \in Span\{a_{j_1} - a_{j_2}\}$. □

Let us now proceed to the second stage of the proof of Lemma 2.5, in which we move from the field $\mathbb{F}$ to $GF(2)$. We will use a probabilistic argument to show the existence of a transformation that maps $\mathbb{F}$ to $GF(2)$, while preserving a large portion of the pairs that span a given unit vector.

For each $i \in [n]$, let a_{ji} denote the i'th coordinate of the vector a_j. Let $V = \{a_{ji}\}_{j\in[m],i\in[n]}$ be the set of all field elements appearing in one of the vectors $a_1, \ldots, a_m$. We pick a random function $f : V \to \{0, 1\}$, and apply f to all the coordinates in all the vectors. Let

$$b_j = (f(a_{j1}), \ldots, f(a_{jn}))$$

be the vector in $\{0, 1\}^n$ obtained from a_j after the transformation. We say that a pair $(j_1, j_2) \in M'_i$ "survived" the transformation if $e_i = b_{j_1} \oplus b_{j_2}$.

Claim 2.8. *The expected number of surviving pairs is $\frac{1}{2}\sum_{i=1}^{n} |M'_i|$.*

Proof. Consider a pair $(j_1, j_2) \in M'_i$. Since $e_i \in Span\{a_{j_1} - a_{j_2}\}$ we know that the vectors a_{j_1}, a_{j_2} are identical in all coordinates different from i. Hence, the vectors b_{j_1}, b_{j_2} will also be identical in those coordinates. From this we see that $e_i = b_{j_1} \oplus b_{j_2}$ iff b_{j_1} and b_{j_2} differ in their i'th coordinate. This happens with probability of one half. By linearity of expectation we can conclude that the expected number of surviving pairs is at least half the number of original pairs, which was $\sum_{i=1}^{n} |M'_i|$. □

From the above claim we can assert that there exist a function f for which the number of surviving pairs is at least $\frac{1}{2}\sum_{i=1}^{n} |M'_i|$. Thus, we have shown that there exist a set of vectors $b_1, \ldots, b_m \in \{0, 1\}^n$ and matchings $M''_i \subset [m] \times [m]$ such that for every $(j_1, j_2) \in M''_i$, we have $e_i = b_{j_1} \oplus b_{j_2}$. Furthermore, we can assume that

$$\sum_{i=1}^{n} |M'_i| \leq 2 \sum_{i=1}^{n} |M''_i|, \tag{3}$$

which completes the proof of the lemma, since now

$$\sum_{i=1}^{n} |M_i| \leq \sum_{i=1}^{n} |M'_i| + m \leq 2 \sum_{i=1}^{n} |M''_i| + m.$$

□

The next Corollary combines the results of Lemma 2.5 and Lemma 2.4 in a compact form. This Corollary will be used in the proof given in Section 4.

Corollary 2.9. *Let $\mathbb{F}$ be any field, and let $a_1, \ldots, a_m \in \mathbb{F}^n$. For every $i \in [n]$ let $M_i \subset [m] \times [m]$ be a set of disjoint pairs of indices (j_1, j_2) such that $e_i \in Span\{a_{j_1}, a_{j_2}\}$. Then*

$$\sum_{i=1}^{n} |M_i| \leq m \log(m) + m.$$

□

3 $\Sigma\Pi\Sigma$ Circuits

In this section we give some definitions related to $\Sigma\Pi\Sigma$ circuits, and describe some elementary operations that can be preformed on them. These definitions and operations will be used in the following sections.

3.1 Definitions

In the following we treat vectors in $\mathbb{F}^n$ also as linear forms in $\mathbb{F}[x_1, \ldots, x_n]$.

Definition 3.1. *Let $u \in \mathbb{F}^n$, $u = (u_1, \ldots, u_n)$. Then:*

$$u(x) = u_1 x_1 + u_2 x_2 + \cdots + u_n x_n.$$

Definition 3.2. *Let $v, u \in \mathbb{F}^n \backslash \{0\}$. We write $u \sim v$ if there exists $c \in \mathbb{F}$ such that $u = c \cdot v$.*

We proceed to the main definition of this section:

Definition 3.3. *Let $\mathbb{F}$ be a field. A $\Sigma\Pi\Sigma$ circuit, $\mathcal{C}$, over $\mathbb{F}$, with n inputs, and k multiplication gates (i.e. top fan-in is k) is the formal expression*

$$\mathcal{C}(x) = \sum_{i=1}^{k} c_i \prod_{j=1}^{d_i} L_{ij}(x),$$

where for each $i \in [k]$, $j \in [d_i]$, L_{ij} is a non constant linear function:

$$L_{ij}(x) = L_{ij}^0 + L_{ij}^1 \cdot x_1 + \cdots + L_{ij}^n \cdot x_n,$$

and $c_i, L_{ij}^t \in \mathbb{F}$ for all i, j, t.

For every $i \in [k]$ define N_i to be the i'th multiplication gate of $\mathcal{C}$:

$$N_i(x) \triangleq \prod_{j=1}^{d_i} L_{ij}(x).$$

For each $i \in [k]$, d_i is the degree of N_i. The number k denotes the number of different multiplication gates, and is referred to as the top fan-in of the circuit. The total degree of $\mathcal{C}$ is $\max\{d_i\}$, and the size of $\mathcal{C}$ is $\sum_{i=1}^{k} d_i$. We denote with $rank(\mathcal{C})$ the rank of $\mathcal{C}$:

$$rank(\mathcal{C}) \triangleq dim(Span\{L_{ij} : i \in [k], j \in [d_i]\}).$$

Comment: when dealing with $\Sigma\Pi\Sigma$ circuits, we will always assume that all the linear functions appearing in the circuit are different from zero.

We are interested in $\Sigma\Pi\Sigma$ circuits that compute the zero polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. If $\mathcal{C}$ is such a circuit, we write $\mathcal{C} \equiv 0$. When dealing with circuits of this kind, it is sufficient to consider circuits of limited structure. This notion is made precise by the following definition and the lemma that follows.

Definition 3.4. *Let $k, d > 0$ be integers. A $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$ is called a $\Sigma\Pi\Sigma(k, d)$ circuit if the following three conditions hold:*

- *the top fan-in of $\mathcal{C}$ is k.*
- *$d_1 = d_2 = \cdots = d_k = d$.*
- *for every $i \in [k]$ and $j \in [d]$, L_{ij} is a homogenous linear form, that is: $L_{ij}(x) = L_{ij}^1 \cdot x_1 + \cdots + L_{ij}^n \cdot x_n$ (the free coefficient in each linear function is zero).*

When dealing with $\Sigma\Pi\Sigma(k, d)$ circuits we will treat the linear functions L_{ij} also as vectors in $\mathbb{F}^n$, that is: $L_{ij} = (L_{ij}^1, \ldots, L_{ij}^n)$.

Lemma 3.5. *There exists a polynomial time algorithm, such that given as input a $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$, with top fan-in k, and total degree $d > 0$, outputs a $\Sigma\Pi\Sigma(k, d)$ circuit $\mathcal{C}'$, such that: $\mathcal{C} \equiv 0$ iff $\mathcal{C}' \equiv 0$. The circuit $\mathcal{C}'$ is called the corresponding $\Sigma\Pi\Sigma(k, d)$ circuit of $\mathcal{C}$.*

Proof. We introduce a new variable y, and define $\mathcal{C}'$ to be a circuit with input variables $x_1, \ldots, x_n, y$. Let

$$L_{ij}(x) = L_{ij}^0 + \sum_{t=1}^{n} L_{ij}^t \cdot x_t$$

be a linear function appearing in $\mathcal{C}$. Define

$$L'_{ij}(x, y) = L_{ij}^0 \cdot y + \sum_{t=1}^{n} L_{ij}^t \cdot x_t,$$

and define $\mathcal{C}'$ to be

$$\mathcal{C}'(x, y) = \sum_{i=1}^{k} c_i \, y^{d-d_i} \prod_{j=1}^{d_i} L_{ij}(x, y).$$

Clearly, $\mathcal{C}'$ is a $\Sigma\Pi\Sigma(k,d)$ circuit, and can be computed from $\mathcal{C}$ in time polynomial in the size of $\mathcal{C}$. Note that if we write

$$\mathcal{C}(x) = \sum_{i=0}^{d} P_i(x),$$

where $P_i(x)$ denotes the homogeneous part of degree i of $\mathcal{C}(x)$, then

$$\mathcal{C}'(x,y) = \sum_{i=0}^{d} P_i(x) y^{d-i}.$$

Therefore $\mathcal{C} \equiv 0$ iff $\mathcal{C}' \equiv 0$. □

Lemma 3.5 shows that in order to achieve our final goal, which is to derive PIT algorithms for $\Sigma\Pi\Sigma$ circuits, it is sufficient to consider $\Sigma\Pi\Sigma(k,d)$ circuits. For the rest of the paper we will deal only with $\Sigma\Pi\Sigma(k,d)$ circuits, and we shall sometimes refer to them simply as $\Sigma\Pi\Sigma$ circuits, omitting the suffix (k,d) where it is not needed.

3.2 Identically zero $\Sigma\Pi\Sigma$ Circuits

Simple Circuits:

It might be the case that there exist a linear function, L, that appears (up to a constant) in all multiplication gates of $\mathcal{C}$. In this case, we can divide each multiplication gate by L, and get a simpler circuit $\mathcal{C}'$, whose degree is smaller then that of $\mathcal{C}$ by one. Clearly $\mathcal{C} \equiv 0$ iff $\mathcal{C}' \equiv 0$. The next two definitions deal with this case in a more general way.

Definition 3.6. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit, and let $N_1, \ldots, N_k$ be its multiplication gates. Define*[4]

$$gcd(\mathcal{C}) \triangleq g.c.d.(N_1(x), \ldots, N_k(x)).$$

Since each multiplication gate is a product of linear forms: $N_i(x) = \prod_{j=1}^{d_i} L_{ij}(x)$, we get that $gcd(\mathcal{C})$ is the product of all the linear forms that appear in all the multiplication gates (up to multiplication by constants). Note also that $gcd(\mathcal{C})$ can be easily computed from $\mathcal{C}$.

It is clear that $\mathcal{C} \equiv 0$ iff $\frac{\mathcal{C}}{gcd(\mathcal{C})} \equiv 0$. This fact motivates the following definition.

Definition 3.7. *A $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$ is called **simple** if $gcd(\mathcal{C}) = 1$. Let us also define $\mathbf{sim}(\mathcal{C})$ to be the simple circuit obtained from $\mathcal{C}$ by dividing each multiplication gate by $gcd(\mathcal{C})$. It is clear that $sim(\mathcal{C})$ is always simple, and that*

$$\mathcal{C}(x) = sim(\mathcal{C})(x) \cdot gcd(\mathcal{C})(x).$$

Example 3.8. *Let*

$$\begin{aligned} \mathcal{C}(x) &= (\mathbf{x_1 + 2x_2 + x_3 + 1})(2x_1 + 4x_2 + 5x_3 + 2)(2x_1 + 4x_2 + 2x_3) \\ &+ (\mathbf{x_1 + 2x_2 + x_3 + 1})(6x_1 + 4x_2 + 5x_3)(1x_1 + 1x_2 + 2x_3 + 4) \\ &+ (\mathbf{2x_1 + 4x_2 + 2x_3 + 2})(4x_2 + 1x_3)(7x_1 + 4x_2 + 2x_3). \end{aligned}$$

[4] g.c.d. stands for greatest common divisor.

Then

$$gcd(\mathcal{C}) = x_1 + 2x_2 + x_3 + 1,$$

and

$$\begin{aligned} sim(\mathcal{C})(x) &= (2x_1 + 4x_2 + 5x_3 + 2)(2x_1 + 4x_2 + 2x_3) \\ &+ (6x_1 + 4x_2 + 5x_3)(1x_1 + 1x_2 + 2x_3 + 4) \\ &+ 2 \cdot (4x_2 + 1x_3)(7x_1 + 4x_2 + 2x_3). \end{aligned}$$

Minimal Circuits:

Suppose we have two $\Sigma\Pi\Sigma$ circuits $\mathcal{C}_1$ and $\mathcal{C}_2$, both of them equal to zero. Let k_1, k_2 denote the top fan-in of $\mathcal{C}_1$ and of $\mathcal{C}_2$ respectively. We can add $\mathcal{C}_1$ to $\mathcal{C}_2$ to create a new circuit $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$, with top fan-in $k_1 + k_2$, that will also be equal to zero. This new circuit $\mathcal{C}$ however, can be 'broken down' into two smaller subcircuits that are zero. In the following we will be interested in circuits that **cannot** be broken down into smaller subcircuit that are equal to zero. The next two definitions deal with circuits of this type.

Definition 3.9. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit , and let $\emptyset \neq T \subseteq [k]$. Then $\mathcal{C}_T$ is defined to be the subcircuit of $\mathcal{C}$ composed of the multiplication gates whose indices appear in T:*

$$\mathcal{C}_T(x) \triangleq \sum_{i \in T} c_i \prod_{j=1}^{d_i} L_{ij}(x) = \sum_{i \in T} c_i N_i(x).$$

Definition 3.10. *Let $\mathcal{C} \equiv 0$ be a $\Sigma\Pi\Sigma$ circuit. We say that $\mathcal{C}$ is **minimal** if for every non-empty subset $T \subset [k]$,apart from $[k]$ itself, we have $\mathcal{C}_T \not\equiv 0$.*

The following easy claim shows that most properties of a $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$ remain when we move to the corresponding $\Sigma\Pi\Sigma(k, d)$ circuit. The proof is immediate from the proof of lemma 3.5.

Claim 3.11. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit, and let $\mathcal{C}'$ be the corresponding $\Sigma\Pi\Sigma(k, d)$ circuit (as defined in Lemma 3.5). Then we have the following:*

- *$rank(\mathcal{C}) \leq rank(\mathcal{C}') \leq rank(\mathcal{C}) + 1$.*
- *$\mathcal{C}$ is simple iff $\mathcal{C}'$ is simple.*
- *$\mathcal{C}$ is minimal iff $\mathcal{C}'$ is minimal.*

□

Taking a Linear Transformation:

We start with a simple operation of setting one of the variables to zero. This operation can be looked at as projecting all the linear functions in the circuit on a sub-space of co-dimension 1.

Definition 3.12. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma$ circuit, and let $t \in [n]$. Define $\mathcal{C}|_{x_t=0}$ to be the circuit obtained from $\mathcal{C}$ by setting the variable x_t to zero (this is the same as changing the t'th coordinate in each linear form L_{ij} to zero). The polynomial computed by $\mathcal{C}|_{x_t=0}$ is therefore*

$$(\mathcal{C}|_{x_t=0})(x) = \mathcal{C}(x_1, \ldots, x_{t-1}, 0, x_{t+1}, \ldots, x_n).$$

We can generalize the operation just defined, by applying a general linear transformation on the linear functions of the circuit.

Definition 3.13. *Let*

$$\mathcal{C}(x) = \sum_{i=1}^{k} c_i \prod_{j=1}^{d} L_{ij}(x)$$

be a $\Sigma\Pi\Sigma(k,d)$ circuit on n variables, and let $\pi : \mathbb{F}^n \to \mathbb{F}^n$ be a linear transformation. Define $\pi(\mathcal{C})$ to be the circuit obtained from $\mathcal{C}$ by applying π on all linear forms appearing in the circuit[5]*. That is*

$$\pi(\mathcal{C})(x) = \sum_{i=1}^{k} c_i \prod_{j=1}^{d} \pi(L_{ij})(x).$$

The following claim is easy to verify.

Claim 3.14. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k,d)$ circuit, and let $\pi : \mathbb{F}^n \to \mathbb{F}^n$ be an invertible linear transformation. Then*

- *$\mathcal{C} \equiv 0$ iff $\pi(\mathcal{C}) \equiv 0$.*
- *$\mathcal{C}$ is simple iff $\pi(\mathcal{C})$ is simple.*
- *$\mathcal{C}$ is minimal iff $\pi(\mathcal{C})$ is minimal.*
- *$rank(\mathcal{C}) = rank(\pi(\mathcal{C}))$.*

□

4 $\Sigma\Pi\Sigma$ Circuits and Locally Decodable Codes

In this section we prove Theorem 1.3, which is the main result of the paper. This theorem shows the relation between $\Sigma\Pi\Sigma$ circuits and linear locally decodable codes. It is more convenient to us to prove the theorem for $\Sigma\Pi\Sigma(k,d)$ circuits instead of general $\Sigma\Pi\Sigma$ circuits. From claim 3.11, we know that moving from $\mathcal{C}$ to its corresponding $\Sigma\Pi\Sigma(k,d)$ circuit does not affect any of the relevant properties of $\mathcal{C}$, so the following theorem is equivalent to Theorem 1.3.

Theorem 4.1. *Let $k \geq 3$, $d \geq 2$, and let $\mathcal{C} \equiv 0$ be a simple and minimal $\Sigma\Pi\Sigma(k,d)$ circuit, on n inputs, over a field $\mathbb{F}$. Then, there exists a linear $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable code $E : \mathbb{F}^{n_1} \to \mathbb{F}^{n_2}$, with*

$$\frac{rank(\mathcal{C})}{P(k)\log(d)^{k-3}} \leq n_1 \quad \text{and} \quad n_2 \leq k \cdot d, \quad \text{where} \quad P(k) = 2^{O(k^2)}.$$

We prove Theorem 4.1 by induction on k. We devote Section 4.1 to the base case of $k = 3$, and give the proof of the inductive step in Section 4.2.

Before moving on to the proof of Theorem 4.1 we should explain why we are only dealing with circuits whose top fan-in is at least 3. The reason for this is that the structure of a zero $\Sigma\Pi\Sigma(k,d)$

[5]Remember that we identify linear forms with vector in $\mathbb{F}^n$.

circuit with $k = 1, 2$ is trivial. If $\mathcal{C}$ has only one multiplication gate ($k = 1$), then it is zero iff one of the linear functions appearing in it is the zero function. The case of $k = 2$ is equally trivial, as seen by the next claim.

Claim 4.2. *Let $\mathcal{C} = c_1 N_1(x) + c_2 N_2(x)$, be a $\Sigma\Pi\Sigma(2, d)$ circuit. Suppose $\mathcal{C} \equiv 0$. Then, the linear functions, appearing in the two multiplication gates N_1 and N_2, are the same, up to an ordering and multiplication by constants.*

Proof. Since $\mathcal{C} \equiv 0$, we have that $c_1 N_1(x) \equiv -c_2 N_2(x)$. Each multiplication gate N_i is a product of linear functions. Since every polynomial can be written, in a unique way, as a product of irreducible polynomials, and since every linear function is irreducible, we have that the linear functions in the two gates must be the same (up to an ordering and multiplication by constants). □

4.1 Proof of Theorem 4.1 for k=3

Let $r = rank(\mathcal{C})$. Then there exists r linearly independent functions $L_1, \ldots, L_r$ in $\mathcal{C}$. Using Claim 3.14 we can assume w.l.o.g that for every $t \in [r], L_t(x) = x_t$ (or in other words: $L_t = e_t$). Consider the circuit $\mathcal{C}|_{x_t=0}$ for some $t \in [r]$. Clearly $\mathcal{C}|_{x_t=0} \equiv 0$. From the fact that the function $L_t = e_t$ appears in one of the multiplication gates, we know that this gate will become zero in $\mathcal{C}|_{x_t=0}$. The following claim assures us that neither of the other two multiplication gates will become zero in $\mathcal{C}|_{x_t=0}$.

Claim 4.3. *Let L and L' be two linear functions appearing in two different multiplication gates of $\mathcal{C}$. Then $L \not\sim L'$.*

Proof. Assume for a contradiction that L divides both N_1 and N_2. As $c_3 N_3(x) = -c_1 N_1(x) - c_2 N_2(x)$ we get that $N_3(x)$ is also divisible by L. But, $\mathcal{C}$ is simple so this is a contradiction. □

How can a circuit with two non-zero multiplication gates be zero ? From Claim 4.2, this is only possible if the two gates contain the same linear functions, up to an ordering and multiplication by constants.

We thus get that every variable x_t, $t \in [r]$, induces a matching on the linear functions of the circuit. This matching contains d pairs of linear functions, such that for every pair (L, L') in the matching, we have that L and L' belong to two different multiplication gates, and that $L|_{x_t=0} \sim L'|_{x_t=0}$. Denote with M_t the matching induced by x_t. The next claim gives us more information about the pairs appearing in those matchings.

Claim 4.4. *Let $t \in [n]$, and let $L, L' \in \mathbb{F}^n$ such that: $L \not\sim L'$, and $L|_{x_t=0} \sim L'_{x_t=0}$. Then*

$$e_t \in Span\{L, L'\}.$$

Proof. Let $L = (a_1, \ldots, a_n), L' = (b_1, \ldots, b_n)$. Since $L|_{x_t=0} \sim L'|_{x_t=0}$, we know that there exists a constant $c \in \mathbb{F}$, such that for all $j \neq t$ we have $a_j = c \cdot b_j$. The fact that $L \not\sim L'$ implies that $a_t \neq c \cdot b_t$. It follows that $e_t \sim L - c \cdot L'$. In particular we get that $e_t \in Span\{L, L'\}$. □

From Claim 4.4 we see that every pair $(L, L') \in M_t$ span the vector e_t. We also have that all the matchings $\{M_t\}_{t \in r}$ are contained in a set of $3d$ linear functions, and that each matching contains

d pairs. We can now construct a linear locally decodable code in the following way: For each $i \in [3], j \in [d]$, let $l_{ij} \in \mathbb{F}^r$ be the projection of L_{ij} on the first r coordinates. define $E : \mathbb{F}^r \to \mathbb{F}^{3d}$ by

$$E_{ij}(x) = l_{ij}(x).$$

In order to show that E is a $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable code, we need to show a decoding algorithm for it. For each $t \in [r]$ we know that there are d disjoint pairs of code positions that span e_t (note that taking the projection on the first r coordinates doesn't affect this property). In order to decode x_t we simply pick a random pair, uniformly, among these d pairs, and compute the linear combination giving e_t. Suppose we picked $l_{ij}(x)$ and $l_{i'j'}(x)$. We know that there exist constants $a, b \in \mathbb{F}$ such that

$$a \cdot l_{ij} + b \cdot l_{i'j'} = e_t.$$

Therefore

$$a \cdot E_{i,j}(x) + b \cdot E_{i',j'}(x) = a \cdot l_{ij}(x) + b \cdot l_{i'j'}(x) = e_t(x) = x_t.$$

If our codeword has at most $\frac{1}{12}(3d) = \frac{d}{4}$ corrupted positions, then at least $\frac{3}{4}$ of the d pairs are uncorrupted, and our algorithm will succeed with probability greater then $\frac{3}{4}$.

In the notation of the theorem, we have $n_1 = r$, and $n_2 = 3d = kd$. Let $P(3) = 1$, then

$$n_1 = r \geq \frac{r}{P(k)\log(d)^{k-3}},$$

and the theorem follows for $k = 3$. □

4.2 Proof of Theorem 4.1 for $k \geq 4$

The proof is by induction on k. The idea behind the proof is the following: Assume that x_1 appears as a linear function in the circuit. A natural thing to do is to consider $\mathcal{C}|_{x_1=0}$. This circuit contains less multiplication gates and so we would like to find an LDC in it by induction. A possible problem is that the rank of every minimal subcircuit is low. We can overcome this problem by showing that there are many variables $x_1, ..., x_m$ ($m \geq r/2^k$) such that there exists $I \subset [k]$ for which $\mathcal{C}_I \not\equiv 0$, but for every $t \in [m]$, $(\mathcal{C}_I)|_{x_t=0}$ is identically zero and minimal. In particular we show that this implies that the rank of $\mathcal{C}_I$ is at least m. We would like to construct a code from $\mathcal{C}_I$, so we consider, say, $(\mathcal{C}_I)|_{x_1=0}$. This circuit is identically zero and minimal, but it is not necessarily simple. Therefore we take $sim((\mathcal{C}_I)|_{x_t=0})$. However, it might be the case that the rank of this circuit is very small, i.e. that we lost a lot of rank when we removed the g.c.d. We overcome this difficulty, by proving that there are relatively few ($\approx \log d$) variables, say $x_1, ..., x_{\log d}$, such that the span of the linear functions in $sim((\mathcal{C}_I)|_{x_t=0})_{t=1,...,\log d}$ contains almost all the functions of $\mathcal{C}_I$. In particular, for some t, the rank of $sim((\mathcal{C}_I)|_{x_t=0})$ is relatively high, so we can apply the induction hypothesis on this circuit. Proving the existence of such t is the main technical difficulty of the proof (claim 4.8). We now give the formal proof.

Let $k \geq 4$, and assume the correctness of Theorem 4.1 for all $3 \leq k' < k$. Let

$$\mathcal{C}(x) = \sum_{i=1}^{k} c_i \prod_{j=1}^{d} L_{ij}(x),$$

be a $\Sigma\Pi\Sigma(k,d)$ circuit satisfying the conditions of the theorem. As in the proof for $k=3$, let $r = rank(\mathcal{C})$, and w.l.o.g assume that the circuit contains the first r unit vectors $e_1, \ldots, e_r$. We can also assume that

$$r \geq P(k)\log(d)^{k-3}, \tag{4}$$

for otherwise the theorem is trivially true, since we can always construct a 2-query locally decodable code whose message size is 1, satisfying the requirements of the theorem.

Claim 4.5. *For every $t \in [r]$ there exists a set $I_t \subset [k]$ such that:*

1. $2 \leq |I_t| \leq k-1$
2. $(\mathcal{C}|_{x_t=0})_{I_t}$ *is identically zero and minimal.*

Proof. Let $t \in [r]$. Clearly $\mathcal{C}|_{x_t=0} \equiv 0$. Denote with k' the number of multiplication gates in $\mathcal{C}$ that become zero when $x_t = 0$ (these are exactly those multiplication gates that contain a linear function proportional to e_t). Since we assumed that $\mathcal{C}$ contains e_t, we know that $k' \geq 1$. It is also easy to verify that $k' \leq k-2$ (if $k' = k$ then $\mathcal{C}$ is not simple, and if $k' = k-1$ then $\mathcal{C}$ is not divisible by x_t - as in Claim 4.3). Therefore, the circuit $\mathcal{C}|_{x_t=0}$ is identically zero, and contains at least two (and at most $k-1$) non-zero multiplication gates. Hence, we can decompose $\mathcal{C}|_{x_t=0}$ into minimal subcircuits, each of top fan-in at least two and at most $k-1$. Take I_t to be the index set of any one of these minimal subcircuits. □

From Claim 4.5 we can conclude that there are $m \geq \frac{r}{2^k}$ variables (w.l.o.g: $x_1, \ldots, x_m$) that have the same set I_t. Let $I = I_1 = \cdots = I_m$, and define

$$\hat{\mathcal{C}} = sim(\mathcal{C}_I).$$

The next claim summarizes several facts we know about the circuit $\hat{\mathcal{C}}$.

Claim 4.6.

1. $\hat{\mathcal{C}}$ *is a* $\Sigma\Pi\Sigma(\hat{k},\hat{d})$ *circuit with* $2 \leq \hat{k} \leq k-1$, $0 < \hat{d} \leq d$.
2. $\hat{\mathcal{C}}$ *is simple.*
3. $\hat{\mathcal{C}} \not\equiv 0$.
4. *For all* $t \in [m]$, $\hat{\mathcal{C}}|_{x_t=0} \equiv 0$ *and is minimal.*
5. *For all* $t \in [m]$, e_t *does not appear in* $\hat{\mathcal{C}}$.

Proof. Parts (1) and (2) follow from the definition of $\hat{\mathcal{C}}$ (the fact that $0 < \hat{d}$ follows from (3) and (4)). (3) is true because we assumed that $\mathcal{C}$ is minimal. (4) follows from the fact that $\hat{\mathcal{C}} = sim(\mathcal{C}_I)$ and that $(\mathcal{C}_I)|_{x_t=0} \equiv 0$ is minimal for all $t \in [m]$. Finally, (5) is a direct consequence of (4). □

Let $\hat{r} \triangleq rank(\hat{\mathcal{C}})$. The next claim shows that the rank of our chosen subcircuit $\hat{\mathcal{C}}$, is not considerably smaller then the rank of $\mathcal{C}$.

Claim 4.7. $\hat{r} \geq m \left(\geq \frac{r}{2^k}\right)$.

Proof. In order to prove the claim, we will show that the linear functions of $\hat{\mathcal{C}}$ span the unit vectors $e_1, \ldots, e_m$. Suppose, on the contrary, that there exists an index $t \in [m]$ for which e_t is not spanned by the linear functions of $\hat{\mathcal{C}}$. Assume w.l.o.g that $t = 1$. There exists an invertible linear transformation $\pi : \mathbb{F}^n \to \mathbb{F}^n$, that satisfies the following two constraints:

- $\pi(e_1) = e_1$.
- The variable x_1 does not appear in the circuit $\pi(\hat{\mathcal{C}})$ (equivalently, all the linear functions in $\pi(\hat{\mathcal{C}})$ are orthogonal to e_1).

From Claim 4.6 we know that $\hat{\mathcal{C}} \not\equiv 0$, and that $\hat{\mathcal{C}}|_{x_1=0} \equiv 0$. Hence $\hat{\mathcal{C}}(x)$ can be written as

$$\hat{\mathcal{C}}(x) \equiv x_1 \cdot g(x),$$

where $g(x)$ is a nonzero polynomial. We can look at the transformation π as a linear change of variables, and denote with $\pi(g)$ the polynomial obtained from $g(x)$ after this change. Thus,

$$\pi(\hat{\mathcal{C}})(x) \equiv \pi(x_1) \cdot \pi(g)(x) \equiv x_1 \cdot \pi(g)(x). \tag{5}$$

Now, since $g(x) \not\equiv 0$, and since π is invertible, claim 3.14 implies[6] that $\pi(g)(x) \not\equiv 0$. From this, and from Equation 5 we see that $\pi(\hat{\mathcal{C}})(x)$ is a nonzero polynomial divisible by x_1. This is a contradiction, since we assumed that x_1 does not appear in $\pi(\hat{\mathcal{C}})$. □

We would like to use the inductive hypothesis on a well chosen circuit among $\hat{\mathcal{C}}|_{x_1=0}, \ldots, \hat{\mathcal{C}}|_{x_m=0}$. However, there are two obstacles in the way. The first is that the top fan-in of $\hat{\mathcal{C}}$ might be equal to 2 (the theorem only holds for $k \geq 3$). This case is rather simple, since we can use the analysis given in Section 4.1 to construct a locally decodable code satisfying the conditions of the theorem (a detailed analysis of this special case is deferred to the end of this section). From now on we assume that $\hat{k} \geq 3$. The second obstacle is that these circuits are not necessarily simple. We overcome this obstacle by using the inductive hypothesis on $sim(\hat{\mathcal{C}}|_{x_t=0})$ instead. The next claim, whose proof is deferred to Section 4.3, tells us which of these circuits we should pick.

For each $t \in [m]$, let $r_t \triangleq rank(sim(\hat{\mathcal{C}}|_{x_t=0}))$.

Claim 4.8. *There exists $t \in [m]$, such that*

$$r_t \geq \frac{\hat{r}}{2^{k+1}\log(d)}.$$

□

Claim 4.8 assures us that one of the r_t's is large (we assume w.l.o.g. that $t = 1$). We get that

$$r_1 \geq \frac{\hat{r}}{2^{k+1}\log(d)}. \tag{6}$$

Our next step is to apply the induction hypothesis to the circuit $sim(\hat{\mathcal{C}}|_{x_1=0})$. However, In order to use Theorem 4.1, we require that the degree of the given circuit is at least two. The next claim shows that the degree of $sim(\hat{\mathcal{C}}|_{x_1=0})$ is indeed at least two.

[6] It is easy to see that this part of Claim 3.14 holds also for general polynomials, and not just $\Sigma\Pi\Sigma$ circuits.

Claim 4.9. *Let d_1 denote the degree of $sim(\hat{\mathcal{C}}|_{x_1=0})$. Then $d_1 \geq 2$.*

Proof. If $d_1 < 2$, then $r_1 < k$ (the number of linear functions is at most $\hat{k} < k$). By Equation 6 we get that

$$\hat{r} \leq k2^{k+1}\log(d).$$

Now, using the fact that $\hat{r} \geq m \geq \frac{r}{2^k}$ (Claim 4.7), we conclude that

$$r \leq 2^k\hat{r} \leq k2^{2k+1}\log(d),$$

contradicting Equation 4, for an appropriate choice of $P(k) = 2^{O(k^2)}$. □

Therefore $sim(\hat{\mathcal{C}}|_{x_1=0})$ satisfies all the conditions of Theorem 4.1. The induction hypothesis, applied on $sim(\hat{\mathcal{C}}|_{x_1=0})$, asserts that there exists a $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable code, $E : \mathbb{F}^{n_1} \to \mathbb{F}^{n_2}$, with

$$n_1 \geq \frac{r_1}{P(\hat{k})\log(d_1)^{\hat{k}-3}} \quad \text{and} \quad n_2 \leq \hat{k} \cdot d_1 (\leq k \cdot d).$$

Using Equation 6, and the fact that $\hat{k} \leq k-1$ and $\hat{r} \geq m \geq \frac{r}{2^k}$, we derive the following inequalities

$$\begin{aligned}
n_1 &\geq \frac{r_1}{P(\hat{k})\log(d_1)^{\hat{k}-3}} \\
&\geq \frac{r_1}{P(k-1)\log(d)^{k-4}} \\
&\geq \frac{\hat{r}}{2^{k+1}P(k-1)\log(d)^{k-3}} \\
&\geq \frac{r}{2^{2k+1}P(k-1)\log(d)^{k-3}} \\
&\geq \frac{r}{P(k)\log(d)^{k-3}}
\end{aligned}$$

(for an appropriate choice of $P(k) = 2^{O(k^2)}$). This completes the proof of the inductive step, and of Theorem 4.1. □

4.2.1 A special case : $\hat{k} = 2$

In this subsection we analyze a special case of the proof of Theorem 4.1. This case is when $\hat{k}$ (the top-fanin of the circuit $\hat{C}$) is equal to 2. The analysis of this case differs from the analysis of the general ($\hat{k} \geq 3$) case because we cannot apply the inductive hypothesis on $\hat{C}$ (or more precisely, on the circuits $C|_{x_t=0}$). We now show how to complete the proof of the theorem (that is, to construct an LDC satisfying the requirements of the thorem) in this case.

Denote by $\hat{N}_1$ and $\hat{N}_2$ the two multiplication gates of $\hat{\mathcal{C}}$. We can write

$$\hat{\mathcal{C}}(x) \equiv c_1\hat{N}_1(x) + c_2\hat{N}_2(x).$$

Now, since $\hat{\mathcal{C}}$ is simple and non-zero, we have

$$gcd(\hat{\mathcal{C}}) \equiv gcd(\hat{N}_1(x), \hat{N}_2(x)) \equiv 1.$$

Next, let $t \in [m]$, and consider what happens to $\hat{\mathcal{C}}$ after we set x_t to zero. We know that $\hat{\mathcal{C}}|_{x_t=0} \equiv 0$, and so

$$c_1 \hat{N}_1|_{x_t=0} \equiv -c_2 \hat{N}_2|_{x_t=0}.$$

Now, since $\hat{N}_1|_{x_t=0}$ and $\hat{N}_2|_{x_t=0}$ are both non-zero ($e_1, \ldots, e_m$ do not appear in $\hat{\mathcal{C}}$), we can deduce, as we did in Section 4.1, that there exists m matchings M_t, $t \in [m]$, of size $|M_t| = \hat{d}$, of linear functions appearing in $\hat{\mathcal{C}}$, such that for every pair $(L, L') \in M_t$, $e_t \in Span\{L, L'\}$. Projecting each linear function in $\hat{\mathcal{C}}$ on the first m coordinates, and using the construction from Section 4.1, we see that there exists a $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable code [7] $E : \mathbb{F}^m \to \mathbb{F}^{2\hat{d}}$. In the notation of the theorem, we have

$$n_2 = 2\hat{d} \leq kd,$$

and

$$n_1 = m \geq \frac{r}{2^k} \geq \frac{r}{P(k)\log(d)^{k-3}},$$

as required by the theorem.

4.3 Proof of Claim 4.8

In this section we prove Claim 4.8. We start by defining some new notations, required for the proof.

4.3.1 Notations

Let $\hat{N}_1, \ldots, \hat{N}_{\hat{k}}$ denote the multiplication gates of $\hat{\mathcal{C}}$. We will treat $\hat{\mathcal{C}}, \hat{N}_1, \ldots, \hat{N}_{\hat{k}}$ also as sets of indices. We shall abuse notations and write

$$\hat{\mathcal{C}} = \{(i,j) \,|\, i \in [\hat{k}], j \in [\hat{d}]\},$$

$$\hat{N}_i = \{(i,j) \,|\, j \in [\hat{d}]\}.$$

For a set $H \subset \hat{\mathcal{C}}$, we denote with $rank(H)$ the dimension of the vector space spanned by the linear functions whose indices appear in H. That is

$$rank(H) \triangleq dim\left(Span\{L_{ij} : (i,j) \in H\}\right).$$

For the rest of the proof we will treat subsets of $\hat{\mathcal{C}}$ interchangeably as sets of indices and as (multi)sets of linear functions.

We would next like to define, for each $t \in [m]$, certain subsets of $\hat{\mathcal{C}}$ that capture the structure of $\hat{\mathcal{C}}|_{x_t=0}$. Fix some $t \in [m]$, and consider what happens to $\hat{\mathcal{C}}$ when we set x_t to be zero. The resulting circuit $\hat{\mathcal{C}}|_{x_t=0}$ is generally not simple, and can therefore be partitioned (see Definition 3.7) into two disjoint sets: A set containing the indices of the linear functions appearing in $gcd(\hat{\mathcal{C}}|_{x_t=0})$, and a set containing the indices of the remaining linear functions (these are the linear funcitons appearing in $sim(\hat{\mathcal{C}}|_{x_t=0})$). To be more precise, denote by δ_t the degree of $gcd(\hat{\mathcal{C}}|_{x_t=0})$. In every multiplication gate $\hat{N}_i$, there are δ_t linear functions such that the restriction of their product to $x_t = 0$ is equal to $gcd(\hat{\mathcal{C}}|_{x_t=0})$. Denote the set of indices of these functions by G_t^i, and let $R_t^i \triangleq \hat{N}_i \backslash G_t^i$ be the set of

[7] We could have actually taken δ to be $\frac{1}{8}$ instead of $\frac{1}{12}$, because the number of multiplication gates is 2 and not 3.

indices of the remaining linear functions of this multiplication gate. We thus have (for some choice of constants $\{c_i\}$)

$$sim(\hat{\mathcal{C}}|_{x_t=0}) = \sum_{i=1}^{\hat{k}} c_i \prod_{(i,j)\in R_t^i} (L_{ij}|_{x_t=0}),$$

and,

$$\forall i \in [\hat{k}] \quad , \quad gcd(\hat{\mathcal{C}}|_{x_t=0}) = \prod_{(i,j)\in G_t^i} (L_{ij}|_{x_t=0}).$$

We now define, for each $t \in [m]$, the sets $R_t \triangleq \bigcup_{i=1}^{\hat{k}} R_t^i$, and $G_t \triangleq \bigcup_{i=1}^{\hat{k}} G_t^i$. The following claim summarizes some facts that we will later need.

Claim 4.10. *For every $t \in [m]$:*

1. $R_t \cap G_t = \emptyset$.
2. $\hat{\mathcal{C}} = R_t \cup G_t$.
3. $|G_t| = \hat{k} \cdot deg(\, gcd(\hat{\mathcal{C}}|_{x_t=0})\,) = \hat{k} \cdot \delta_t$.
4. R_t *contains the indices of the linear functions appearing in* $sim(\hat{\mathcal{C}}|_{x_t=0})$.
5. $r_t = rank(sim(\hat{\mathcal{C}}|_{x_t=0})) = rank(R_t)$.

Proof. Follows directly from the previous paragraph and from Definition 3.7. □

4.3.2 The Proof

We will start by assuming that the claim is false. In other words we assume that for every $t \in [m]$

$$r_t < \frac{\hat{r}}{2^{k+1}\log(d)}. \tag{7}$$

Having defined, for each $t \in [m]$, the sets R_t and G_t, we would now like to show that there exist a 'small' ($\sim \log(d)$) number of sets R_t, such that their union covers almost all of $\hat{\mathcal{C}}$. As $rank(\hat{\mathcal{C}})$ is relatively high, and for each t, $r_t = rank(R_t)$ is (assumed to be) relatively small, we will get a contradiction. We construct the cover step by step, in each step we will find an index $t \in [m]$ such that the set R_t covers at least half of the linear functions not yet covered. This idea is made precise by the following claim.

Claim 4.11. *For every integer $1 \leq q \leq \log(\hat{d})$ there exist q indices $t_1, \ldots, t_q \in [m]$ for which*

$$\left| \bigcup_{s=1}^{q} R_{t_s} \right| \geq \hat{k}\hat{d}(1 - 2^{-q}).$$

Proof. by induction on q:

Base case $q = 1$**:** In order to prove the claim for $q = 1$, it is sufficient to show that there exists $t \in [m]$ for which $|R_t| \geq \frac{1}{2}\hat{k}\hat{d}$. Suppose, on the contrary, that for all $t \in [m]$, $|R_t| < \frac{1}{2}\hat{k}\hat{d}$. Claim 4.10 implies that for all $t \in [m]$, $|G_t| \geq \frac{1}{2}\hat{k}\hat{d}$. This in turn implies that for all $t \in [m]$

$$|G_t^1| \geq \frac{1}{2}\hat{d}. \tag{8}$$

The next lemma shows that, under the conditions just described, the linear functions of $\hat{\mathcal{C}}$ 'contain' a 2-query locally decodable code. We will then apply our results on LDCs from Section 2 (namely Corollary 2.9) to derive a contradiction. Lemma 4.12 is more general than what is required at this point, however, we will need it in its full generality when we handle $q > 1$.

Lemma 4.12. *Let $\mathcal{C}$ be a simple $\Sigma\Pi\Sigma(k, d)$ circuit with n inputs. Let $t \in [n]$, $i_t \in [k]$. Denote $\delta_t = deg(gcd(\mathcal{C}|_{x_t=0}))$. Suppose that the linear functions in N_{i_t} are ordered such that*

$$gcd(\mathcal{C}|_{x_t=0}) = (L_{i_t1}|_{x_t=0})(x) \cdot (L_{i_t2}|_{x_t=0})(x) \cdot \ldots \cdot (L_{i_t\delta_t}|_{x_t=0})(x).$$

Then, there exist a matching, $M = \{\mathcal{P}_1, \ldots, \mathcal{P}_g\} \subseteq \mathcal{C} \times \mathcal{C}$, consisting of δ_t disjoint pairs of linear functions, such that for each $j \in [\delta_t]$:

- *The two linear functions in $\mathcal{P}_j$ span e_t.*
- *The first element of $\mathcal{P}_j$ is L_{i_tj}.*

Proof. We can reorder the linear functions in each gate N_i, $i \neq i_t$, such that

$$\forall j \in [\delta_t] \quad : \quad L_{1j}|_{x_t=0} \sim L_{2j}|_{x_t=0} \sim \ldots \sim L_{i_tj}|_{x_t=0} \sim \ldots \sim L_{kj}|_{x_t=0}.$$

As $\mathcal{C}$ is simple, it cannot be the case that, for some j, L_{i_tj} divides all the multiplication gates. Therefore, for every $j \in [\delta_t]$ there exists an index $\alpha(j) \in [k]$, such that $L_{i_tj} \not\sim L_{\alpha(j)j}$. From Claim 4.4 it follows that,

$$\forall j \in [\delta_t] \quad : \quad e_t \in Span\{L_{i_tj}, L_{\alpha(j)j}\}.$$

For each $j \in [\delta_t]$ let $\mathcal{P}_j = (L_{i_tj}, L_{\alpha(j)j})$. Set $M = \{\mathcal{P}_1, \ldots, \mathcal{P}_{\delta_t}\}$. It is clear that each $\mathcal{P}_j$ satisfies the two conditions of the lemma, and that the $\mathcal{P}_j$'s are disjoint. □

We continue with the proof of Claim 4.11. From Equation 8 and Lemma 4.12 we conclude that for each $t \in [m]$, there exists a matching $M_t \subset \mathcal{C} \times \mathcal{C}$, containing at least $\frac{1}{2}\hat{d}$ disjoint pairs of linear functions, such that every pair in M_t spans e_t. Corollary 2.9 implies that

$$\frac{1}{2}\hat{d}m \leq \sum_{t=1}^{m} |M_t| \leq \hat{k}\hat{d}\log(\hat{k}\hat{d}) + \hat{k}\hat{d}$$

which gives

$$m \leq 2\hat{k}\log(\hat{k}\hat{d}) + 2\hat{k} < \log(d)^{k-3}P(k)2^{-k}$$

(for an appropriate choice of $P(k) = 2^{O(k^2)}$). Now, since $m \geq \frac{r}{2^k}$, we have that

$$r < \log(d)^{k-3}P(k),$$

contradicting Equation 4. Therefore our initial assumption was wrong and we conclude that there exists t_1 with $|R_{t_1}| \geq \frac{1}{2}\hat{k}\hat{d}$. This completes the proof of Claim 4.11 for the case of $q = 1$.

Induction step: Let us now assume that we have found $q-1$ indices $t_1, \ldots, t_{q-1} \in [m]$ for which

$$\left|\bigcup_{s=1}^{q-1} R_{t_s}\right| \geq \hat{k}\hat{d}(1 - 2^{-(q-1)}).$$

Let

$$R \triangleq \bigcup_{s=1}^{q-1} R_{t_s},$$

$$S \triangleq \hat{\mathcal{C}} \backslash R.$$

Then, by our assumption

$$|S| \leq \hat{k}\hat{d}2^{-(q-1)}. \tag{9}$$

The proof goes along the same lines as the proof for $q = 1$: we show that there exists an index $t \in [m]$, such that R_t covers at least half of S. We will argue that if such an index does not exists, then a contradiction to Equation 4 can be derived. Our main tools in doing so are Lemma 4.12 and Corollary 2.9.

Claim 4.13. *There exists $t \in [m]$, such that for all $i \in [\hat{k}]$,*

$$|G_t^i \cap S| < \hat{d}2^{-q}.$$

Roughly, the lemma states that there exists some variable, x_t, such that most of the linear functions in S do not belong to $gcd(\hat{\mathcal{C}}|_{x_t=0})$. In particular it implies that R_t covers a large fraction of S, as needed.

Proof. Assume, on the contrary, that for every $t \in [m]$, there exists $i_t \in [\hat{k}]$, for which

$$|G_t^{i_t} \cap S| \geq \hat{d}2^{-q}.$$

From Lemma 4.12 we get that, for every $t \in [m]$, there exists a matching M_t, consisting of $\hat{d}2^{-q}$ disjoint pairs of linear functions, such that each pair spans e_t, and that the first element in each pair is in $G_t^{i_t} \cap S$ (from the lemma we actually get that M_t contains $deg(gcd(\hat{\mathcal{C}}|_{x_t=0}))$ number of pairs, but we are only interested in the pairs whose first element is in $G_t^{i_t} \cap S$).

We would now like to apply Corollary 2.9 on the matchings $\{M_t\}_{t\in[m]}$, however, for our needs, we would also like that all the linear functions in all the matchings will belong to S. We achieve this by projecting all functions in R to zero.

Claim 4.14. *There exists a subset $A \subset [m]$, of size $|A| \geq \frac{m}{2}$, and a linear transformation $\pi : \mathbb{F}^n \to \mathbb{F}^n$ such that*

- $\ker(\pi) = Span(R)$.
- $\forall t \in A \quad , \quad \pi(e_t) = e_t$.

Proof. Calculating, we get that

$$\begin{aligned} rank(R) &= rank(\bigcup_{s=1}^{q-1} R_{t_s}) \leq \sum_{s=1}^{q-1} rank(R_{t_s}) = \sum_{s=1}^{q-1} r_{t_s} \\ &\leq (q-1)\frac{\hat{r}}{\log(d)2^{k+1}} \leq \frac{r}{2^{k+1}} \leq \frac{m}{2}, \end{aligned} \tag{10}$$

where the second inequality follows from Equation 7, the third inequality follows from the fact that $q \leq \log \hat{d} \leq \log d$ and $\hat{r} \leq r$, and the last inequality follows from the fact that $\frac{r}{2^k} \leq m$. Let $m' = m - rank(R)$. From Equation 10 we get that $m' \geq m/2$. In particular, there exists a subset $A \subset [m]$, of size $|A| = m'$, such that $Span(\{x_t | t \in A\}) \cap Span(R) = \{0\}$. Hence, there exists a linear transformation $\pi : \mathbb{F}^n \to \mathbb{F}^n$ such that

- $\ker(\pi) = Span(R)$.
- $\forall t \in A \quad , \quad \pi(e_t) = e_t$.

This completes the proof of the claim. □

Let A be the set obtained from the above claim, and π the corresponding linear transformation. We assume, w.l.o.g., that $A = [m']$. From here on, we only consider variables x_t such that $t \in [m']$ (i.e. $t \in A$). Fix such $t \in [m']$, and let $M'_t = \pi(M_t)$. In other words, $M'_t = \{(\pi(L), \pi(L'))\}_{(L,L') \in M_t}$. Clearly,

$$|M'_t| = |M_t| \geq \hat{d}2^{-q}. \tag{11}$$

Note that the pairs in M'_t still span e_t, as for any pair $(L.L') \in M_t$, with $e_t = \alpha L + \beta L'$, we have that

$$e_t = \pi(e_t) = \pi(\alpha L + \beta L') = \alpha\pi(L) + \beta\pi(L').$$

Since all the linear functions appearing in R were projected to zero, we know that all the pairs in each M'_t are contained in the multiset[8] $S' \triangleq \{\pi(L) : L \in S\}$.

After this long preparation we apply Corollary 2.9 to the matchings M'_t, and derive the following inequality:

$$\sum_{t=1}^{m'} |M'_t| \leq |S'| \log(|S'|) + |S'|. \tag{12}$$

As $|S'| = |S|$ (remember that S' is a multiset), we get by Equation 9 that

$$|S'| \leq \hat{k}\hat{d}2^{-(q-1)}. \tag{13}$$

By Equations 11, 12 and 13, it follows that

$$\begin{aligned} m' \cdot (\hat{d}2^{-q}) &\leq \sum_{t=1}^{m'} |M'_t| \leq |S'| \log(|S'|) + |S'| \\ &\leq \hat{k}\hat{d}2^{-(q-1)} \log(\hat{k}\hat{d}2^{-(q-1)}) + \hat{k}\hat{d}2^{-(q-1)}. \end{aligned}$$

[8] Note that, as in the proof of Lemma 2.5, we can replace each pair in M'_t, that contains the zero vector, with a singleton.

From the fact that $k \geq 4$ and $m' \geq m/2$ (and some simple manipulations), we see that for an appropriate choice of $P(k) = 2^{O(k^2)}$

$$m < 2^{-k} P(k) \log(d)^{k-3}.$$

As $m \geq \frac{r}{2^k}$, we get that

$$r < P(k) \log(d)^{k-3},$$

contradicting Equation 4. This completes the proof of Claim 4.13. □

Let us now proceed with the proof of Claim 4.11. Take t_q to be the index described by Claim 4.13, that is:

$$\forall i \in [\hat{k}] \quad : \quad |G^i_{t_q} \cap S| < \hat{d} 2^{-q}.$$

In particular

$$|G_{t_q} \cap S| < \hat{k}\hat{d} 2^{-q}.$$

As the complement of $\bigcup_{s=1}^{q} R_{t_s}$ is exactly $G_{t_q} \cap S$, we get that adding R_{t_q} to R gives

$$\left| \bigcup_{s=1}^{q} R_{t_s} \right| \geq \hat{k}\hat{d}(1 - 2^{-q}).$$

This completes the proof of the Claim 4.11. □

Having proved Claim 4.11, we are now just steps away from completing the proof of Claim 4.8. Taking q to be $\lfloor \log(\hat{d}) \rfloor$ in Claim 4.11, we get that there exist indices $t_1, \ldots, t_{\lfloor \log(\hat{d}) \rfloor} \in [m]$, such that

$$\left| \bigcup_{s=1}^{\lfloor \log(\hat{d}) \rfloor} R_{t_s} \right| \geq \hat{k}\hat{d} - 2\hat{k}...$$

Thus

$$\hat{r} - 2\hat{k} \leq rank \left(\bigcup_{s=1}^{\lfloor \log(\hat{d}) \rfloor} R_{t_s} \right) \leq \sum_{s=1}^{\lfloor \log(\hat{d}) \rfloor} r_{t_s}.$$

The last inequality tells us that there exists some $t \in [m]$ for which

$$r_t \geq \frac{\hat{r} - 2\hat{k}}{\lfloor \log(\hat{d}) \rfloor} \geq \frac{\hat{r} - 2\hat{k}}{\log(d)}. \tag{14}$$

In order to finish the proof of Claim 4.8 we prove the following inequality

Claim 4.15.

$$\hat{r} - 2\hat{k} \geq \frac{\hat{r}}{2^{k+1}}.$$

Proof. Using Equation 4 we get

$$\hat{r} \geq m \geq 2^{-k} r \geq 2^{-k} P(k) \log(d)^{k-3}.$$

Therefore we can choose $P(k) = 2^{O(k^2)}$ such that

$$\hat{r} > 2\hat{k}\frac{2^{k+1}}{2^{k+1}-1}.$$

This implies the inequality in the claim. □

Combining Claim 4.15 with Equation 14, we conclude that there exists $t \in [m]$ for which

$$r_t \geq \frac{\hat{r}}{\log(d)2^{k+1}},$$

which contradicts our initial assumption (Eq. 7). This completes the proof of Claim 4.8. □

5 A Structural Theorem For Zero ΣΠΣ Circuits

The main result of this section is a structural theorem for ΣΠΣ circuits which are identically zero. The proof is based on the results of Section 4. To ease the notations we will prove our results only for $\Sigma\Pi\Sigma(k,d)$ circuits, however from Claim 3.11 it will follow that all the results also hold for ΣΠΣ circuits with k multiplication gates of degree d.

Theorem 5.1. (Structural Theorem) *Let $\mathcal{C} \equiv 0$ be a $\Sigma\Pi\Sigma(k,d)$ circuit. Then, there exist a partition of $[k]$: $T_1, \ldots, T_s \subset [k]$, with the following properties.*

- $\mathcal{C} = \sum_{i=1}^{s} \mathcal{C}_{T_i} = \sum_{i=1}^{s} gcd(\mathcal{C}_{T_i}) \cdot sim(\mathcal{C}_{T_i})$.
- $\forall i \in [s] \quad sim(\mathcal{C}_{T_i}) \equiv 0$, *and is simple and minimal.*
- $\forall i \in [s] \quad rank(sim(\mathcal{C}_{T_i})) \leq 2^{O(k^2)} \log(d)^{k-2}$.

In other words the theorem says that every zero ΣΠΣ circuit, can be 'broken down' into zero subcircuits of low rank (ignoring the g.c.d.). This fact will be used in the next section, in which we devise PIT algorithms for ΣΠΣ circuits.

Before giving the proof of the theorem we prove a lemma that bounds the rank of a zero, simple and minimal ΣΠΣ circuit. Note that Theorem 1.4 follows from the lemma and Claim 3.11.

Lemma 5.2. *Let $k \geq 3$, $d \geq 2$, and let $\mathcal{C} \equiv 0$ be a simple and minimal $\Sigma\Pi\Sigma(k,d)$ circuit. Then*

$$rank(\mathcal{C}) \leq 2^{O(k^2)} \log(d)^{k-2}.$$

Proof. From Theorem 4.1 we know that there exists a linear $(2, \frac{1}{12}, \frac{1}{4})$-locally decodable code $E : \mathbb{F}^{n_1} \to \mathbb{F}^{n_2}$, with

$$\frac{rank(\mathcal{C})}{P(k)\log(d)^{k-3}} \leq n_1 \quad \text{and} \quad n_2 \leq k \cdot d, \quad \text{where} \quad P(k) = 2^{O(k^2)}.$$

Theorem 1.2 now tells us that

$$n_2 \geq 2^{\frac{1}{96}n_1 - 1}.$$

Combining the above inequalities we get the required bound on $rank(\mathcal{C})$. □

We now use Lemma 5.2 to prove Theorem 5.1:

Proof of Theorem 5.1. Since $\mathcal{C}$ is equal to zero, we can find a partition $T_1, \ldots, T_s \subset [k]$, such that the circuits $\mathcal{C}_{T_1}, \ldots, \mathcal{C}_{T_s}$ are all zero and minimal. Thus, the circuits $sim(\mathcal{C}_{T_1}), \ldots, sim(\mathcal{C}_{T_s})$ are all zero, simple and minimal. By Lemma 5.2 we get that if $|T_i| \geq 3$ and $deg(sim(\mathcal{C}_{T_i})) \geq 2$ then

$$rank(sim(\mathcal{C}_{T_i})) \leq 2^{O(k^2)} \log(d)^{k-2}.$$

If $|T_i| = 2$ then by Claim 4.2 we get that $deg(sim(\mathcal{C}_{T_i})) = 0$ and so its rank is 1. If $deg(sim(\mathcal{C}_{T_i})) \leq 1$ then its rank is at most k. Thus, we have covered all the possible cases and the lemma follows. □

6 PIT Algorithms

In this section we use the structural theorem (Theorem 5.1), proved in the previous section, to devise the PIT algorithms of Theorem 1.5. Again, to simplify the notations, we give algorithms for $\Sigma\Pi\Sigma(k, d)$ circuits, that work in the same manner also for $\Sigma\Pi\Sigma$ circuits with k multiplication gates of degree d. We state our results for a general k, however, our algorithms will be most applicable when k is a constant.[9]

From Theorem 5.1 we know that every zero $\Sigma\Pi\Sigma$ circuit can be broken down into zero sub-circuits whose ranks are small. The next two lemmas show that checking whether these low-rank circuits are zero or not, can be done efficiently.

Lemma 6.1. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k, d)$ circuit, with $rank(\mathcal{C}) = r$. Then, there exists a polynomial time algorithm, transforming $\mathcal{C}$ into a $\Sigma\Pi\Sigma(k, d)$ circuit $\mathcal{C}'$. Such that*

- *$\mathcal{C} \equiv 0$ iff $\mathcal{C}' \equiv 0$.*
- *$\mathcal{C}'$ contains only r variables.*

Proof. This is a direct consequence of Claim 3.14: we apply an invertible linear transformation on $\mathcal{C}$, taking a set of r linearly-independent vectors to $e_1, \ldots, e_r$. The transformed circuit will contain only the first r variables, and will be zero iff $\mathcal{C}$ was zero. □

Lemma 6.2. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k, d)$ circuit, and let $r = rank(\mathcal{C})$, $s = size(\mathcal{C})$. Then we can check if $\mathcal{C} \equiv 0$:*

1. *Deterministically, in time $poly(s) \cdot (r + d)^r$.*
2. *Probabilistically, in time $poly(s + \frac{1}{\epsilon})$, using $r \cdot \left(\log(d) + \log(\frac{1}{\epsilon})\right)$ random bits, with error probability ϵ.*

Proof. Using Lemma 6.1, we can transform $\mathcal{C}$ into a circuit $\mathcal{C}'$ with at most r variables, such that $\mathcal{C} \equiv 0$ iff $\mathcal{C}' \equiv 0$. Since $\mathcal{C}'$ contains only r variables, the number of different monomials in $\mathcal{C}'(x)$ is bounded by $\binom{r+d-1}{r-1} < (r + d)^r$. We can thus check if $\mathcal{C}' \equiv 0$ by computing the coefficients of all the monomials and seeing if they are all zero. This can be done in time $poly(s) \cdot (r + d)^r$. For the second part of the corollary, note that we can also check if $\mathcal{C}' \equiv 0$ probabilistically using the well known Schwartz-Zippel algorithm [Sch80] [Zip79] . □

[9] Our methods give sub-exponential time ($2^{o(n)}$) algorithms also if $k = o(\sqrt{\log n})$.

We are now ready to describe our PIT algorithm for $\Sigma\Pi\Sigma(k,d)$ circuits.

Algorithm 1 - Deterministic Algorithm

input: A $\Sigma\Pi\Sigma(k,d)$ circuit $\mathcal{C}$.

(1) For every subset $T \subset [k]$ do the following:
- (1.1) Compute $r_T = rank(sim(\mathcal{C}_T))$.
- (1.2) If $r_T \leq 2^{O(k^2)} \log(d)^{k-2}$, then:
- -check if $sim(\mathcal{C}_T) \equiv 0$ using part 1 of Lemma 6.2.

(2) If there exists a partition of $[k]$, such that for every set $T \subset [k]$ in the partition $sim(\mathcal{C}_T) \equiv 0$, then **accept**. Otherwise **reject**.

Theorem 6.3. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k,d)$ circuit, $s = size(\mathcal{C})$. Then, Algorithm 1 will check if $\mathcal{C} \equiv 0$. Further, the algorithm will run in time $poly(s) \cdot exp\left(2^{O(k^2)} \log(d)^{k-1}\right)$.*

Proof. First, note that if $\mathcal{C}$ is non-zero, then the algorithm will never accept (the algorithm accepts only when a partition of $\mathcal{C}$ into zero sub-circuits was found). Assume that $\mathcal{C}$ is zero. Then, by Theorem 5.1, there exists be a partition, $T_1, \ldots, T_s \subset [k]$, of $[k]$, such that the circuits $sim(\mathcal{C}_{T_1}), \ldots, sim(\mathcal{C}_{T_s})$ are all zero, and that for all $i \in [s]$, the rank of $sim(\mathcal{C}_{T_i})$ is bounded by $2^{O(k^2)} \log(d)^{k-2}$. Therefore, for every $\mathcal{C}_{T_i}$ we will check whether $sim(\mathcal{C}_{T_i}) \equiv 0$ in step (1.2) of the algorithm. Since we go over all subset of $[k]$, we are bound to find the above partition, and accept.

As for the running time of the algorithm, notice that we only apply the algorithm from Lemma 6.2 on circuits whose rank is smaller then $2^{O(k^2)} \log(d)^{k-2}$. Therefore, by Lemma 6.2, the time spent in each invocation of step (1.2) is at most

$$poly(s) \cdot exp\left(2^{O(k^2)} \log(d)^{k-1}\right).$$

Step (1.2) is run at most 2^k times, and so the total running time is also

$$poly(s) \cdot exp\left(2^{O(k^2)} \log(d)^{k-1}\right)$$

(the running times of all the other steps of the algorithm are "swallowed up" by the running time of step (1.2)). □

We can modify Algorithm 1 so that it will use a probabilistic check in step (1.2). This will result in a probabilistic PIT algorithm for $\Sigma\Pi\Sigma$ circuits, that uses fewer random bits than previous algorithms.

Theorem 6.4. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k,d)$ circuit, $s = size(\mathcal{C})$. Then, Algorithm 2 will check if $\mathcal{C} \equiv 0$. Further, the algorithm will run in time poly $\left(s + \frac{2^k}{\epsilon}\right)$, use $2^{O(k^2)} \log(d)^{k-1} \log(\frac{1}{\epsilon})$ random bits, and will make an error with probability less then ϵ.*

Proof. Using the same reasoning as in the proof of Theorem 6.3, we see that the algorithm can make an error only if one of the checks in step (1.2) fails. By the union bound, this happens with probability of at most ϵ.

Algorithm 2 - Probabilistic Algorithm

input: A $\Sigma\Pi\Sigma(k,d)$ circuit $\mathcal{C}$. An error probability ϵ.

(1) For every subset $T \subset [k]$ do the following:
- (1.1) Compute $r_T = rank(sim(\mathcal{C}_T))$.
- (1.2) If $r_T \leq 2^{O(k^2)} \log(d)^{k-2}$, then: check if $sim(\mathcal{C}_T) \equiv 0$ probabilistically,
- using part 2 of Lemma 6.2, with error probability $\epsilon 2^{-k}$.

(2) If there exists a partition of $[k]$, such that for every set $T \subset [k]$ in the partition $sim(\mathcal{C}_T) \equiv 0$, then **accept**. Otherwise **reject**.

Each check in step (1.2) takes time $poly\left(s + \frac{2^k}{\epsilon}\right)$. And so the total running time is

$$2^k \cdot poly\left(s + \frac{2^k}{\epsilon}\right) = poly\left(s + \frac{2^k}{\epsilon}\right).$$

By part 2 of Lemma 6.2, the number of random bits used in step (1.2) is at most $r_T \cdot \left(\log(d) + \log(\frac{1}{\epsilon})\right)$. Since we run the probabilistic check only when $r_T \leq 2^{O(k^2)} \log(d)^{k-2}$, it follows that the number of random bits used in each invocation of step (1.2) is bounded by $2^{O(k^2)} \log(d)^{k-1} \log(\frac{1}{\epsilon})$. As we can use the same random bits in all tests, this is also the total number of random bits needed. □

We restate the last two theorems for the case when k is a constant.

Theorem 6.5. *Let $\mathcal{C}$ be a $\Sigma\Pi\Sigma(k,d)$ circuit, k a constant, $s = size(\mathcal{C})$. Then we can check if $\mathcal{C} \equiv 0$:*

1. *Deterministically, in quasi-polynomial time.*
2. *Probabilistically, in time $poly(s + \frac{1}{\epsilon})$, using $O\left(\log(d)^{k-1} \log(\frac{1}{\epsilon})\right)$ random bits, with error probability ϵ.*

Note that Theorems 6.3,6.4 and 6.5 imply the first 2 claims of Theorem 1.5.

6.1 Multilinear circuits

this section deals with a special kind of $\Sigma\Pi\Sigma$ circuits, described by the following definition.

Definition 6.6. *A $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$ is **multilinear**, if each of its multiplication gates computes a multilinear polynomial (a polynomial is multilinear if the degree of every variable is at most one).*

Let

$$\mathcal{C}(x) = \sum_{i=1}^{k} c_i \prod_{j=1}^{d_i} L_{ij}(x)$$

be a $\Sigma\Pi\Sigma$ circuit. Denote by $V_{ij} \subset [n]$, the set of variables appearing in the linear form L_{ij}. From Definition 6.6 we see that $\mathcal{C}$ is multilinear iff for every $i \in [k]$, and for every $j_1 \neq j_2$, we have

$$V_{ij_1} \cap V_{ij_2} = \emptyset.$$

This condition implies that, for every $i \in [k]$, the linear functions $\{L_{ij}\}_{j \in [d_i]}$ are linearly independent. This leads to the following observation.

Observation 6.7. *If $\mathcal{C}$ is a multilinear $\Sigma\Pi\Sigma$ circuit of degree d, then $rank(\mathcal{C}) \geq d$.*

Combining this observation and Theorem 1.4 we get the following theorem.

Theorem 6.8. *Let $\mathcal{C} \equiv 0$ be a multilinear $\Sigma\Pi\Sigma$ circuit with k multiplication gates ($k \geq 3$), which is simple and minimal. Let $d = deg(\mathcal{C})$, then*

$$d \leq 2^{O(k^2)} \log(d)^{k-2}. \tag{15}$$

□

Corollary 6.9. *There exists an integer function $D(k) = 2^{O(k^2)}$, such that every multilinear $\Sigma\Pi\Sigma$ circuit $\mathcal{C}$ with k multiplication gates, which is simple and equal to zero, and of degree $d = deg(\mathcal{C}) > D(k)$, is not minimal.*

Proof. Fix k, and consider Equation 15. This inequality holds only if $d \leq 2^{O(k^2)} = D(k)$. Thus, if $d > D(k)$, then the conditions of Theorem 6.8 are not satisfied. In particular, if $\mathcal{C} \equiv 0$ and is simple, then it is not minimal. □

We can use Corollary 6.9 in order to improve the algorithm given in Section 6, in the case that the given circuit is multilinear.

Theorem 6.10. *Let $\mathcal{C}$ be a multilinear $\Sigma\Pi\Sigma$ circuit, of size s, with k multiplication gates. We can check if $\mathcal{C} \equiv 0$ in time $poly(s) \cdot exp\left(2^{O(k^2)}\right)$. Thus, if k is constant, the algorithm runs in polynomial time.*

Proof sketch. The algorithm is the same as Algorithm 1 (it doesn't matter that our circuit is not a $\Sigma\Pi\Sigma(k, d)$ circuit). The only difference is that we only have to consider sub-circuits $\mathcal{C}_T$ such that the degree of $sim(\mathcal{C}_T)$ is less then $D(|T|) = 2^{O(k^2)}$. □

Theorem 6.10 implies the third claim of Theorem 1.5, thus completing the proof of the theorem.

7 Conclusions and open problems

Finding efficient deterministic PIT algorithms for general arithmetic circuits is a long standing open problem. We made the first step towards an efficient algorithm for PIT for depth 3 circuits by giving PIT algorithms for depth 3 circuits with bounded top fan-in, however the general case of depth 3 circuits is still open. In view of [KI03] it is natural to look for algorithms for PIT for restricted models of arithmetic circuits in which lower bounds are known. Recently Raz [Raz04] proved a quasipolynomial lower bound for *multilinear* arithmetic formulas computing the determinant and the permanent. Thus, giving PIT algorithms for multilinear formulas is a very interesting, and maybe even a solvable, problem.

The key to our result is the relation we have found between LDCs and depth 3 circuits. Previously, relations between circuits and error correcting codes were known only for bilinear circuits

over finite fields [Bsh89, Shp03]. It should be very interesting to find new relations between codes and arithmetic circuits. Another interesting question is whether the relation that we have found is tight. In particular we believe that in theorem 1.3 one should be able to replace $r/2^{O(k^2)}log(d)^{k-3}$ with $O(r/k)$. A related question is to improve theorem 1.4: We believe that for minimal and simple circuits the rank should be $O(k)$. Currently we have found circuits, which are minimal and simple, with $r = 3k - 2$, and we think that it is an interesting task to come up with (minimal and simple) circuits that have larger rank.

We conclude this section with a geometrical problem related to depth 3 circuits with 3 multiplication gates. It is well known that every set of n points in the plane, that have the property that every line that contains two points from the set also contains a third point from the set, is contained in a line. Consider the following generalization of the problem (colored version in the projective plane): Instead of one set of points we have 3 different sets. Each set is of size n. The points in the sets correspond to vectors from the r-dimensional sphere, and every two such vectors are linearly independent. The condition on the sets is that every 2 dimensional subspace that contains points from two different sets, also contains a point from the 3rd set[10]. What can be said about r in this case? Clearly the r-dimensional sphere can be embedded into the $(r+1)$-dimensional sphere so we only consider "irreducible" arrangements in which the vectors corresponding to the points, span the whole space. Using our lower bound on LDCs we can show that r is at most $O(\log n)$, however we think that this can be improved. In particular we conjecture that r is bounded (maybe even $r = 2$). If our conjecture is true then it will serve as an evidence that for $k = 3$ the rank of every simple and minimal depth 3 circuit, which is identically zero, is bounded.

We now give an example that shows the relation of the problem to identically zero depth 3 circuits with 3 multiplication gates. Consider the following equality $x_1^n - x_2^n = \prod_{i=0}^{n-1}(x_1 - w^i x_2)$, where w is a primitive n'th root of unity. We get that

$$\sum_{i=1}^{k-1}\prod_{j=0}^{n-1}(x_i - w^j x_{i+1}) + \prod_{j=0}^{n-1}(x_k - w^j x_1) = 0.$$

Notice that this is an identically zero depth 3 circuit with k multiplication gates. For the special case of $k = 3$ we get that

$$\prod_{j=0}^{n-1}(x_1 - w^j x_2) + \prod_{j=0}^{n-1}(x_2 - w^j x_3) + \prod_{j=0}^{n-1}(x_3 - w^j x_1) = 0.$$

Each multiplication gate corresponds to a different set of points: We map each linear function $x_1 - w^j x_2$ from the first gate to the point $(\frac{1}{\sqrt{2}}, \frac{-w^j}{\sqrt{2}}, 0)$, similarly we map the functions of the 2nd multiplication gate to $\{(0, \frac{1}{\sqrt{2}}, \frac{-w^j}{\sqrt{2}})\}_{j=0\ldots n-1}$ etc. Clearly all the points belong to the 2 dimensional sphere in $\mathbb{C}^3$. It is easy to see that for each point from the first set (i.e. point coming from the first multiplication gate) and each point from the second set there is a unique point from the third set that belongs to the same 2 dimensional space (similarly if we pick the first and third sets etc.). Therefore this construction satisfies our requirements. Our question is, can such arrangements be found in higher dimensions.

[10] Alternatively, the points belong to the r-dimensional projective space and every line that contains points from two different sets also contains a point from the third set.

Acknowledgements

The authors would like to thank Ran Raz and Avi Wigderson for helpful discussions during various stages of this work. A. S. would like to thank Boaz Barak, Valentine Kabanets and Salil Vadhan for useful conversations on the topic of the work. We are grateful to Ran Raz for many valuable comments that improved the presentation of the results.

References

[AB03] Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. *J. ACM*, 50(4):429–443, 2003.

[AGS03] Adi Akavia, Shafi Goldwasser, and Muli Safra. A unifying approach for proving hardcore predicates using list decoding. In *Proceedings of the fourty fourth IEEE Symposium on Foundations of Computer Science*, pages 146–155, Cambridge, MA, USA, 2003.

[AKS02] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p, 2002.

[BF90] Donald Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *Proceedings of the seventh annual symposium on Theoretical aspects of computer science*, pages 37–48. Springer-Verlag New York, Inc., 1990.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 21–32. ACM Press, 1991.

[BI01] Amos Beimel and Yuval Ishai. Information-theoretic private information retrieval: A unified construction. *Lecture Notes in Computer Science*, 2076:912–926, 2001.

[BIKR02] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-Francois Raymond. Breaking the $O(n^{1/2k-1})$ barrier for information-theoretic private information retrieval. In *Proceedings of the fourty third Symposium on Foundations of Computer Science*, pages 261–270, 2002.

[BOT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 301–309. ACM Press, 1988.

[BS83] Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22:317 – 330, 1983.

[Bsh89] Nader H. Bshouty. A lower bound for matrix multiplication. *SIAM J. Comput.*, 18(4):759–765, 1989.

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *IEEE Symposium on Foundations of Computer Science*, pages 41–50, 1995.

[CK97] Zhi-Zhong Chen and Ming-Yang Kao. Reducing randomness via irrational numbers. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 200–209. ACM Press, 1997.

[CRS95] Suresh Chari, Pankaj Rohatgi, and Aravind Srinivasan. Randomness-optimal unique element isolation withapplications to perfect matching and related problems. *SIAM J. Comput.*, 24(5):1036–1050, 1995.

[DJK+02] Amit Deshpande, Rahul Jain, T Kavitha, Jaikumar Radhakrishnan, and Satyanarayana V. Lokam. Better lower bounds for locally decodable codes. In *Proceedings of the 17th IEEE Annual Conference on Computational Complexity*, page 184. IEEE Computer Society, 2002.

[FF93] Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993.

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 577–582. ACM Press, 1998.

[GKS90] Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. *SIAM J. Comput.*, 19(6):1059–1063, 1990.

[GKST01] Oded Goldreich, Howard J. Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Electronic Colloquium on Computational Complexity (ECCC)*, (080), 2001.

[GL89] Oded Goldreich and Leonid A. Levin. A hardcore predicate for all oneway functions. In *Proceedings of the twenty first ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, USA, 1989.

[GLR+91] Peter Gemmell, Richard J. Lipton, Ronitt Rubinfeld, Madhu Sudan, and Avi Wigderson. Self-testing/correcting for polynomials and for approximate functions. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 33–42. ACM Press, 1991.

[GR98] Dima Grigoriev and Alexander A. Razborov. Exponential complexity lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. In *Proc. 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 269–278, 1998.

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM Journal on Discrete Mathematics*, 13(4):535–570, 2000.

[GS92] Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43(4):169–174, 1992.

[JS80] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semi-rings. Technical Report CRS-58-80, Univ. of Edinburgh, 1980.

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 106–115. ACM Press, 2003.

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 355–364. ACM Press, 2003.

[KS01] Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223. ACM Press, 2001.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86. ACM Press, 2000.

[Lev87] Leonid A. Levin. One–way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[Lip90] Richard J. Lipton. Efficient checking of computations. In C. Choffrut and T. Lengauer, editors, *STACS 90: Proc. of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, pages 207–215. Springer, Berlin, Heidelberg, 1990.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Comptuation Theory : Proceedings of the Conference on Algebraic, Arithmetic, and Categorial Methods in Computation Theory*, volume 2, pages 565–574. Akademie-Verlag, 1979.

[LV98] Daniel Lewin and Salil Vadhan. Checking polynomial identities over any field: towards a derandomization? In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 438–447. ACM Press, 1998.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 345–354. ACM Press, 1987.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 410–418. ACM Press, 1991.

[NW95] Noam Nisan and Avi Wigderson. Lower bounds for arithmetic circuits via partial serivatives (preliminary version). In *Proc. 36th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 16–25, 1995.

[Oba02] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *Proceedings of the 6th International Workshop on Randomization and Approximation Techniques*, pages 39–50. Springer-Verlag, 2002.

[Pud94] Pavel Pudlak. Communication in bounded depth circuits. *Combinatorica*, 14(2):203–216, 1994.

[Raz04] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 633–641. ACM Press, 2004.

[RS01] Ran Raz and Amir Shpilka. Lower bounds for matrix product, in bounded depth circuits with arbitrary gates. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 409–418. ACM Press, 2001.

[RS04] Ran Raz and Aamir Shpilka. Deterministic polynomial identity testing in non commutative models. *Conference on Computational Complexity (to appear)*, 2004.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

[Shp03] Amir Shpilka. Lower bounds for matrix product. *SIAM J. Comput.*, 32(5):1185–1200, 2003.

[SS77] Eli Shamir and Marc Snir. Lower bounds on the number of multiplications and the number of additions in monotone computations. Research Report RC6757, IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., sep 1977.

[SS80] Eli Shamir and Marc Snir. On the depth complexity of formulas. *Mathematical Systems Theory*, 13:301–322, 1980.

[Str73] Volker Strassen. Die berechnungskomplexitat vo elementarsymmetrischen funktionen und von interpolationskoefizienten. *Numer. Math*, 20:238–251, 1973.

[SW99] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic formulae over fields of characteristic zero. In *Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, page 87. IEEE Computer Society, 1999.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity, 2004.

[TT94] Prasoon Tiwari and Martin Tompa. A direct version of shamir and snir's lower bounds on monotone circuit depth. *Inf. Process. Lett.*, 49(5):243–248, 1994.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, pages 216–226. Springer-Verlag, 1979.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'