

# Small-depth Multilinear Formula Lower Bounds for Iterated Matrix Multiplication, with Applications

**Suryajith Chillara**

Chennai Mathematical Institute, India  
suryajith@cmi.ac.in

**Nutan Limaye**

Department of CSE, IIT Bombay, India  
nutan@cse.iitb.ac.in

**Srikanth Srinivasan**

Department of Mathematics, IIT Bombay, India  
srikanth@math.iitb.ac.in

---

## Abstract

---

The complexity of Iterated Matrix Multiplication is a central theme in Computational Complexity theory, as the problem is closely related to the problem of separating various complexity classes within P. In this paper, we study the algebraic formula complexity of multiplying  $d$  many  $2 \times 2$  matrices, denoted  $\text{IMM}_d$ , and show that the well-known divide-and-conquer algorithm cannot be significantly improved at any depth, as long as the formulas are multilinear.

Formally, for each depth  $\Delta \leq \log d$ , we show that any product-depth  $\Delta$  multilinear formula for  $\text{IMM}_d$  must have size  $\exp(\Omega(\Delta d^{1/\Delta}))$ . It also follows from this that any multilinear circuit of product-depth  $\Delta$  for the same polynomial of the above form must have a size of  $\exp(\Omega(d^{1/\Delta}))$ . In particular, any polynomial-sized multilinear formula for  $\text{IMM}_d$  must have depth  $\Omega(\log d)$ , and any polynomial-sized multilinear circuit for  $\text{IMM}_d$  must have depth  $\Omega(\log d / \log \log d)$ . Both these bounds are tight up to constant factors.

Our lower bound has the following consequences for multilinear formula complexity.

- 1. Depth-reduction:** A well-known result of Brent (JACM 1974) implies that any formula of size  $s$  can be converted to one of size  $s^{O(1)}$  and depth  $O(\log s)$ ; further, this reduction continues to hold for multilinear formulas. On the other hand, our lower bound implies that any depth-reduction in the multilinear setting cannot reduce the depth to  $o(\log s)$  without a superpolynomial blow-up in size.
- 2. Separations from general formulas:** Shpilka and Yehudayoff (FnTCS 2010) asked whether general formulas can be more efficient than multilinear formulas for computing multilinear polynomials. Our result, along with a non-trivial upper bound for  $\text{IMM}_d$  implied by a result of Gupta, Kamath, Kayal and Saptharishi (SICOMP 2016), shows that for any size  $s$  and product-depth  $\Delta = o(\log s)$ , general formulas of size  $s$  and product-depth  $\Delta$  cannot be converted to multilinear formulas of size  $s^{O(1)}$  and product-depth  $\Delta$ , when the underlying field has characteristic zero.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Algebraic complexity theory

**Keywords and phrases** Algebraic Circuit Complexity, Multilinear Formulas, Lower Bounds

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2018.21

**Related Version** A full version of the paper is available at [4], <https://eccc.weizmann.ac.il/report/2017/156/>.



© Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan;  
licensed under Creative Commons License CC-BY

35th Symposium on Theoretical Aspects of Computer Science (STACS 2018).

Editors: Rolf Niedermeier and Brigitte Vallée; Article No. 21; pp. 21:1–21:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



## 1 Introduction

Algebraic Complexity theory is the study of the complexity of those computational problems that can be phrased as computing a multivariate polynomial  $f(x_1, \dots, x_N) \in \mathbb{F}[x_1, \dots, x_N]$  over elements  $x_1, \dots, x_N \in \mathbb{F}$ . Many central algorithmic problems such as the Determinant, Permanent, Matrix product etc. can be cast in this framework. The natural computational models that we consider in this setting are models such as *Algebraic circuits*, *Algebraic Branching Programs* (ABPs), and *Algebraic formulas* (or just formulas), all of which use the natural algebraic operations of  $\mathbb{F}[x_1, \dots, x_N]$  to compute the polynomial  $f$ . These models have by now been the subject of a large body of work with many interesting upper bounds (i.e. circuit constructions) as well as lower bounds (i.e. impossibility results). (See, e.g. the surveys [23, 22] for an overview of many of these results.)

Despite this, many fundamental questions remain unresolved. An important example of such a question is that of proving lower bounds on the size of formulas for the *Iterated Matrix Multiplication* problem, which is defined as follows. Given  $d$   $n \times n$  matrices  $M_1, \dots, M_d$ , we are required to compute (an entry of) the product  $M_1 \cdots M_d$ ; we refer to this problem as  $\text{IMM}_{n,d}$ . Proving superpolynomial lower bounds on the size of formulas for this problem is equivalent to separating the power of polynomial-sized ABPs from polynomial-sized formulas, which is the algebraic analogue of separating the Boolean complexity classes NL and NC<sup>1</sup>.

A standard divide-and-conquer algorithm yields the best-known formulas for  $\text{IMM}_{n,d}$ . More precisely, for any  $\Delta \leq \log d$ , this approach yields a formula of product-depth<sup>1</sup>  $\Delta$  and size  $n^{O(\Delta d^{1/\Delta})}$  for  $\text{IMM}_{n,d}$  and choosing  $\Delta = \log d$  yields the current best formula upper bound of  $n^{O(\log d)}$ , which has not been improved in quite some time. On the other hand, separating the power of ABPs and formulas is equivalent to showing that  $\text{IMM}_{n,d}$  does not have formulas of size  $\text{poly}(nd)$ .

The Iterated Matrix Multiplication problem has many nice features that render its complexity an interesting object to study. For one, it is the algebraic analogue of the Boolean reachability problem, and thus any improved formula upper bounds for  $\text{IMM}_{n,d}$  could lead to improved Boolean circuit upper bounds for the reachability problem, which would resolve a long-standing open problem in that area. For another, this problem has strong self-reducibility properties, which imply that improving on the simple divide-and-conquer approach to obtain formulas of size  $n^{o(\log d)}$  for any  $d$  would lead to improved upper bounds for all  $D > d$ ; this implies that the lower-degree variant is no easier than the higher-degree version of the problem, which can be very useful (e.g. for homogenization [16]). Finally, the connection to the Reachability problem imbues  $\text{IMM}_{n,d}$  with a rich combinatorial structure via its graph theoretic interpretation, which has been used extensively in lower bounds for depth-4 arithmetic circuits [6, 9, 12, 10, 11].

We study the formula complexity of this problem in the *multilinear* setting, which restricts the underlying formulas to only compute multilinear polynomials at intermediate stages of computation. Starting with the breakthrough work of Raz [15], many lower bounds have been proved for multilinear models of computation [18, 19, 17, 5]. Further, it is known by a result of Dvir, Malod, Perifel and Yehudayoff [5] that multilinear ABPs are in fact superpolynomially more powerful than multilinear formulas. Unfortunately, however, this does not imply any non-trivial lower bound for Iterated Matrix Multiplication (see the Related Work section

---

<sup>1</sup> The *product-depth* of an arithmetic circuit or formula is the maximum number of product gates on a path from output to input. If the product-depth of a circuit or formula is  $\Delta$ , then its depth can be assumed to be at least  $2\Delta - 1$  and at most  $2\Delta + 1$ .

below), and as far as we know, it could well be the case that there are multilinear formulas that beat the divide-and-conquer approach in computing this polynomial.

Here, we are able to show that this is not the case for the problem of multiplying  $2 \times 2$  matrices (and by extension  $c \times c$  matrices for any constant  $c$ ) at any product-depth. Our main theorem is the following (stated more formally as Theorem 9 later).

► **Theorem 1.** *For  $\Delta \leq \log d$ , any product-depth  $\Delta$  multilinear formula that computes  $\text{IMM}_{2,d}$  must have size  $2^{\Omega(\Delta d^{1/\Delta})}$ .*

This lower bound strengthens a result of Nisan and Wigderson [13] who prove a similar lower bound in the more restricted *set-multilinear* setting.

Our result is also qualitatively different from the previous lower bounds for multilinear formulas since  $\text{IMM}_{2,d}$  does in fact have polynomial-sized formulas of product-depth  $O(\log d)$  (via the divide-and-conquer approach), whereas we show a superpolynomial lower bound for product-depth  $o(\log d)$ . This observation leads to interesting consequences for multilinear formula complexity in general, which we now describe.

- **Depth Reduction:** An important theme in Circuit complexity is the interplay between the size of a formula or circuit and its depth [3, 24, 26, 1, 25]. In the context of algebraic formulas, a result of Brent [3] says that any formula of size  $s$  can be converted into another of size  $s^{O(1)}$  and depth  $O(\log s)$ . Further, the proof of this result also yields the same statement for multilinear formulas.

Can the result of Brent be improved? Theorem 1 implies that the answer is no in the multilinear setting (Corollary 12). More precisely, since the  $\text{IMM}_{2,d}$  polynomial (over  $O(d)$  variables) has formulas of size  $\text{poly}(d)$  and depth  $O(\log d)$  but no formulas of size  $d^{O(1)}$  and depth  $o(\log d)$  (by Theorem 1), we see that any multilinear depth-reduction procedure that reduces the depth of a size- $s$  formula to  $o(\log s)$  must incur a superpolynomial blow-up in size. This strengthens a result of Raz and Yehudayoff [19], whose results imply that any depth-reduction of multilinear formulas to depth  $o(\sqrt{\log s}/\log \log s)$  should incur a superpolynomial blow-up in size. It is also an analogue in the algebraic setting of some recent results proved for Boolean circuits [20, 21].

- **Multilinear vs. general formulas:** Shpilka and Yehudayoff [23] ask the question of whether general formulas can be more efficient at computing multilinear polynomials than multilinear formulas. This is an important question, since we have techniques for proving lower bounds for multilinear formulas, whereas the same question for general formulas (or even depth-3 formulas over large fields) remains wide open.

We are able to make progress towards this question here by showing a separation between the two models for small depths when the underlying field has characteristic zero (Corollary 13). We do this by using Theorem 1 in conjunction with a (non-multilinear) formula *upper bound* for  $\text{IMM}_{2,d}$  over fields of characteristic zero due to Gupta et al. [7]. In particular, the result of Gupta et al. [7] implies that for any depth  $\Delta$ , the polynomial  $\text{IMM}_{2,d}$  has formulas of product depth  $\Delta$  and size  $2^{O(\Delta d^{1/2\Delta})}$ , which is considerably smaller than our lower bound in the multilinear case for small  $\Delta$ . From this, it follows that for any size parameter  $s$  and product-depth  $\Delta = o(\log s)$ , general formulas of size  $s$  and product-depth  $\Delta$  cannot be converted to multilinear formulas of size  $s^{O(1)}$  and product-depth  $\Delta$ . Improving our result to allow for  $\Delta = O(\log s)$  would resolve the question entirely.

**Related Work.** The multilinear formula model has been the focus of a large body of work on Algebraic circuit lower bounds. Nisan and Wigderson [13] proved some of the early

results in this model by showing size lower bounds for small-depth *set-multilinear*<sup>2</sup> circuits computing  $\text{IMM}_{2,d}$ . They showed that any product-depth  $\Delta$  circuit for  $\text{IMM}_{2,d}$  must have a size of  $2^{\Omega(d^{1/\Delta})}$  matching the upper bound from the divide-and-conquer algorithm for  $\Delta = o(\log d / \log \log d)$ . Our lower bounds for multilinear formulas imply similar lower bounds for multilinear circuits of product-depth  $\Delta$ .

Raz [15] proved the first superpolynomial lower bound for multilinear formulas by showing an  $n^{\Omega(\log n)}$  lower bound for the  $n \times n$  Determinant and Permanent polynomials. This was further strengthened by the results of Raz [14] and Raz and Yehudayoff [18] to a similar lower bound for an explicit polynomial family that has polynomial-sized multilinear *circuits*. In particular, these results show the tightness of the depth-reduction procedure for algebraic *circuits* in the multilinear setting [26, 18].

Similar polynomial families were also used in the work of Raz and Yehudayoff [19] to prove *exponential* lower bounds for multilinear constant-depth circuits. By proving a tight lower bound for depth- $\Delta$  circuits computing an explicit polynomial (similar to the construction of Raz [14]), Raz and Yehudayoff [19] showed superpolynomial separations between multilinear circuits of different depths.

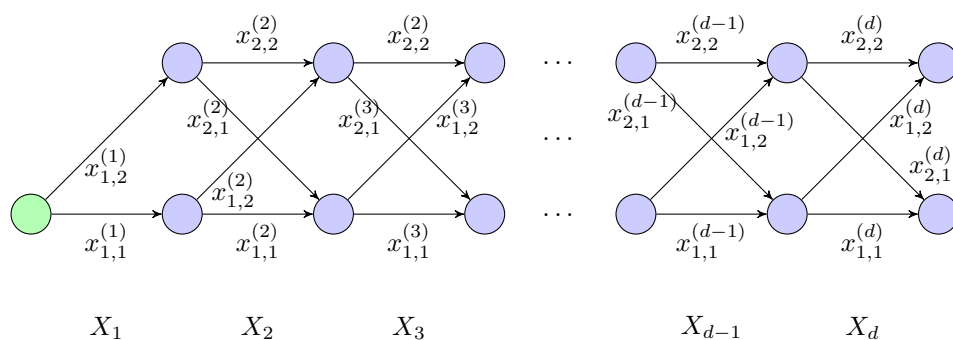
In particular, the result of Raz and Yehudayoff [19] implies that the polynomial families of [14, 18], which have formulas of size  $n^{O(\log n)}$ , cannot be computed by formulas of size less than some  $s(n) = n^{\omega(\log n)}$  if the product-depth  $\Delta = o(\log n / \log \log n)$ . This yields the superpolynomial separation between formulas of size  $s$  and depth  $o(\sqrt{\log s} / \log \log s)$  alluded to above. Unfortunately, these polynomials also have nearly optimal formulas of depth just  $O(\log n) = O(\sqrt{\log s})$ , so they cannot be used to obtain the optimal size  $s$  vs depth  $o(\log s)$  separation we obtain here.

Dvir et al. [5] showed that there is an explicit polynomial on  $n$  variables that has multilinear ABPs of size  $\text{poly}(n)$  but no multilinear formulas of size less than  $n^{\Omega(\log n)}$ . One might hope that this yields a superpolynomial lower bound for multilinear formulas computing  $\text{IMM}_{N,d}$  for some  $N, d$  but this unfortunately does not seem to be the case. The reason for this is that while any polynomial  $f$  on  $n$  variables that has an ABP of size  $\text{poly}(n)$  can be reduced via variable substitutions to  $\text{IMM}_{N,d}$  for  $N, d = n^{O(1)}$ , this reduction might substitute different variables in the  $\text{IMM}_{N,d}$  polynomial by the same variable  $x$  of  $f$  and in the process destroy multilinearity.

Gupta et al. [7] showed the surprising result that general (i.e. non-multilinear) formulas of depth-3 can beat the divide-and-conquer approach for computing  $\text{IMM}_{n,d}$ , when the underlying field has characteristic zero. Their result implies that, in this setting,  $\text{IMM}_{n,d}$  has product-depth 1 formulas of size  $n^{O(\sqrt{d})}$ , as opposed to the  $n^{O(d)}$ -sized formula that is obtained from the traditional divide-and-conquer approach. Using the self-reduction properties of  $\text{IMM}_{n,d}$ , this can be easily seen to imply the existence of  $n^{O(\Delta d^{1/2\Delta})}$ -sized formulas of product-depth  $\Delta$ . This construction uses the fact that the formulas are allowed to be non-multilinear. Our result shows that this cannot be avoided.

---

<sup>2</sup> Set-multilinear circuits are further restrictions of multilinear circuits. A set-multilinear circuit for  $\text{IMM}_{n,d}$  is defined by the property that each intermediate polynomial computed must be a linear combination of monomials that contain exactly one variable from each matrix  $M_i$  ( $i \in S$ ), for some choice of  $S \subseteq [d]$ .



■ **Figure 1** The directed acyclic graph  $G_d$  that defines the polynomial  $\text{IMM}_d$  with its labeling.

## 2 Preliminaries

### 2.1 Basic setup

Unless otherwise stated, let  $\mathbb{F}$  be an arbitrary field. Let  $d \in \mathbb{N}$  be a growing integer parameter. We define  $X^{(1)}, \dots, X^{(d)}$  to be disjoint sets of variables where each  $X^{(i)} = \{x_{j,k}^{(i)} \mid j, k \in [2]\}$  is a set of four variables that we think of forming a  $2 \times 2$  matrix. Let  $X = \bigcup_{i \in [d]} X^{(i)}$ .

A polynomial  $P \in \mathbb{F}[X]$  is called *multilinear* if the degree of  $P$  in each variable  $x \in X$  is at most 1. We define the multilinear polynomial  $\text{IMM}_d \in \mathbb{F}[X]$  as follows. Consider the matrices  $M^{(1)}, \dots, M^{(d)}$  where the entries of  $M^{(i)}$  are the variables of  $X^{(i)}$  arranged in the obvious way. Define the matrix  $M = M^{(1)} \cdots M^{(d)}$ ; the entries of  $M$  are multilinear polynomials over the variables in  $X$ . We define  $\text{IMM}_d = M(1, 1) + M(1, 2)$ , i.e. the sum of the  $(1, 1)$ th and  $(1, 2)$ th entries of  $M$ . Note, in particular, that the polynomial  $\text{IMM}_d$  does not depend on the variables  $x_{2,1}^{(1)}$  and  $x_{2,2}^{(1)}$ .

This is a slight variant of the Iterated Matrix Multiplication polynomial seen in the literature, as it is usually defined to be either the matrix entry  $M(1, 1)$  or the trace  $M(1, 1) + M(2, 2)$ . Our results can easily be seen to hold for these variants, but we deal with the definition above for some technical simplicity.

Another standard way of defining the polynomial  $\text{IMM}_d$  is via graphs. Define the edge-labelled directed acyclic graph  $G_d = (V, E, \lambda)$  as follows: the vertex set  $V$  is defined to be the disjoint union of vertex sets  $V^{(0)}, \dots, V^{(d)}$  where  $V^{(i)} = \{v_1^{(i)}, v_2^{(i)}\}$ . The edge set  $E$  is the set of all possible edges from some set  $V^{(i)}$  to  $V^{(i+1)}$  (for all  $i < d$ ). The labelling function  $\lambda : E \rightarrow X$  is defined by  $\lambda((v_j^{(i)}, v_k^{(i+1)})) = x_{j,k}^{(i+1)}$ . See Figure 1 for a depiction of this graph.

Given a path  $\pi$  in the graph  $G_d$ ,  $\lambda(\pi)$  is defined to be the product of all labels of edges in  $\pi$ . In this notation,  $\text{IMM}_d$  can be seen to be the following.

$$\text{IMM}_d = \sum_{\substack{\text{paths } \pi \text{ from } v_1^{(0)} \\ \text{to } v_1^{(d)} \text{ or } v_2^{(d)}}} \lambda(\pi) = \sum_{\pi_1, \dots, \pi_d \in \{1, 2\}} x_{1, \pi_1}^{(1)} x_{\pi_1, \pi_2}^{(2)} \cdots x_{\pi_{d-1}, \pi_d}^{(d)} \quad (1)$$

### 2.2 Multilinear formulas and circuits

We refer the reader to the standard resources (e.g. [23, 22]) for basic definitions related to algebraic circuits and formulas. Having said that, we do make a few remarks.

- All the gates in our formulas and circuits will be allowed to have *unbounded* fan-in.

- The size of a formula or circuit will refer to the number of gates (including input gates) in it, and depth of the formula or circuit will refer to the number of gates on the longest path from an input gate to output gate.
- Further, the *product-depth* of the formula or circuit (as in [18]) will refer to the maximum number of product gates on a path from an input gate to the output gate. Note that the product depth of a formula or circuit can be assumed to be within a factor of two of the overall depth (by collapsing sum gates if necessary).

**Multilinear circuits and formulas.** An algebraic formula  $F$  (resp. circuit  $C$ ) computing a polynomial from  $\mathbb{F}[X]$  is said to be *multilinear* if each gate in the formula (resp. circuit) computes a multilinear polynomial<sup>3</sup>. Moreover, a formula  $F$  is said to be *syntactic multilinear* if for each multiplication gate  $\Phi$  of  $F$  with children  $\Psi_1, \dots, \Psi_t$ , we have  $\text{Supp}(\Psi_i) \cap \text{Supp}(\Psi_j) = \emptyset$  for each  $i \neq j$ , where  $\text{Supp}(\Phi)$  denotes the set of variables that appear in the subformula rooted at  $\Phi$ . Finally, for  $\Delta \geq 1$ , we say that a multilinear formula (resp. circuit) is a  $(\Sigma\Pi)^\Delta\Sigma$  formula (resp. circuit) if the output gate is a sum gate and along any path, the sum and product gates alternate, with each product gate appearing exactly  $\Delta$  times and the bottom gate being a sum gate. We can define  $(\Sigma\Pi)^\Delta, \Sigma\Pi\Sigma, \Sigma\Pi\Sigma\Pi$  formulas and circuits similarly.

For a gate  $\Phi$  in a syntactically multilinear formula, we define a set of variables  $\text{Vars}(\Phi)$  in a top-down fashion as follows.

► **Definition 2.** Let  $C$  be a syntactically multilinear formula computing a polynomial on the variable set  $X$ . For the output gate  $\Phi$ , we define  $\text{Vars}(\Phi) = X$ . If  $\Phi$  is a sum gate with children  $\Psi_1, \dots, \Psi_k$  and  $\text{Vars}(\Phi) = S \subseteq X$ , then for each  $1 \leq i \leq k$ ,  $\text{Vars}(\Psi_i) = S$ . If  $\Phi$  is a product gate with children  $\Psi_1, \dots, \Psi_k$  and  $\text{Vars}(\Phi) = S \subseteq X$ , then  $\text{Vars}(\Psi_i) = \text{Supp}(\Psi_i)$  for  $1 \leq i \leq k-1$  and  $\text{Vars}(\Psi_k) = S \setminus (\cup_{i=1}^{k-1} \text{Vars}(\Psi_i))$ .

It is easy to see that  $\text{Vars}(\cdot)$  satisfies the properties listed in the following proposition.

► **Proposition 3.** For each gate  $\Phi$  in a syntactically multilinear formula  $C$ , let  $\text{Vars}(\Phi)$  be defined as in Definition 2 above.

1. For any gate  $\Phi$  in  $C$ ,  $\text{Supp}(\Phi) \subseteq \text{Vars}(\Phi)$ .
2. If  $\Phi$  is a sum gate, with children  $\Psi_1, \Psi_2, \dots, \Psi_k$ , then  $\forall i \in [k]$ ,  $\text{Vars}(\Psi_i) = \text{Vars}(\Phi)$ .
3. If  $\Phi$  is a product gate, with children  $\Psi_1, \Psi_2, \dots, \Psi_k$ , then  $\text{Vars}(\Phi) = \cup_{i=1}^k \text{Vars}(\Psi_i)$  and the sets  $\text{Vars}(\Psi_i)$  ( $i \in [k]$ ) are pairwise disjoint.

We will use the following structural results that convert general multilinear circuits (resp. formulas) to  $(\Sigma\Pi)^\Delta\Sigma$  circuits (resp. formulas).

► **Lemma 4** (Raz and Yehudayoff [19], Claims 2.3 and 2.4). For any multilinear formula  $F$  of product depth at most  $\Delta$  and size at most  $s$ , there is a syntactic multilinear  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F'$  of size at most  $(\Delta + 1)^2 \cdot s$  computing the same polynomial as  $F$ .

► **Lemma 5** (Raz and Yehudayoff [19], Lemma 2.1). For any multilinear circuit  $C$  of product depth at most  $\Delta$  and size at most  $s$ , there is a syntactic multilinear  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F$  of size at most  $(\Delta + 1)^2 \cdot s^{2\Delta+1}$  computing the same polynomial as  $C$ .

We will also need the following structural result.

<sup>3</sup> It is important to note that any multilinear polynomial can be computed by a non-multilinear formula (resp. circuit) as well.

► **Lemma 6** (Raz, Shpilka and Yehudayoff [17], Claim 5.6). *Let  $F$  be a syntactic multilinear formula computing a polynomial  $f$  and let  $\Phi$  be any gate in  $F$  computing a polynomial  $g$ . Then  $f$  can be written as  $f = Ag + B$ , where  $A \in \mathbb{F}[X \setminus \text{Vars}(\Phi)]$ ,  $B \in \mathbb{F}[X]$  and  $B$  is computed by replacing  $\Phi$  with a 0 in  $F$ .*

A standard divide-and-conquer approach (see [2, Proposition 3.10]) yields the best-known multilinear formulas/circuits for  $\text{IMM}_d$  for all depths. (A proof sketch is presented in [4].)

► **Lemma 7**. *For each  $\Delta \leq \log d$ ,<sup>4</sup>  $\text{IMM}_d$  is computed by a syntactic multilinear  $(\Sigma\Pi)^\Delta$  circuit  $C_\Delta$  of size at most  $d^{O(1)} \cdot 2^{O(d^{1/\Delta})}$  and a syntactic multilinear  $(\Sigma\Pi)^\Delta$  formula  $F_\Delta$  of size at most  $2^{O(\Delta d^{1/\Delta})}$ .*

We will show that the above bounds are nearly tight in the multilinear setting. If we remove the multilinear restriction on  $(\Sigma\Pi)^\Delta\Sigma$  formulas computing  $\text{IMM}_d$ , we can get better upper bounds, as long as the underlying field has characteristic zero.

► **Lemma 8** (follows from [7]). *Let  $\mathbb{F}$  be a field of characteristic zero. For each  $\Delta \leq \log d$ ,  $\text{IMM}_d$  has a  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F_\Delta$  of size at most  $2^{O(\Delta d^{1/(2\Delta)})}$ .*

A proof sketch for the above lemma is presented in the full version [4].

### 3 Lower bounds for multilinear formulas and circuits computing $\text{IMM}_d$

The main theorem of this section is the following lower bound.

► **Theorem 9**. *Let  $d \geq 1$  be a growing parameter and fix any  $\Delta \leq \log d$ . Any syntactic multilinear  $(\Sigma\Pi)^\Delta\Sigma$  formula for  $\text{IMM}_d$  must have a size of  $2^{\Omega(\Delta d^{1/\Delta})}$ .*

By applying Theorem 9 and Lemmas 4, 5, we get the following (immediate) corollaries.

► **Corollary 10**. *Let  $d \geq 1$  be a growing parameter and fix any  $\Delta \leq \log d / \log \log d$ . Any multilinear circuit of product-depth at most  $\Delta$  for  $\text{IMM}_d$  must have a size of  $2^{\Omega(d^{1/\Delta})}$ . In particular, any polynomial-sized multilinear circuit for  $\text{IMM}_d$  must have product-depth  $\Omega(\log d / \log \log d)$ .*

► **Corollary 11**. *Let  $d \geq 1$  be a growing parameter and fix any  $\Delta \leq \log d$ . Any multilinear formula of product depth at most  $\Delta$  for  $\text{IMM}_d$  must have size  $2^{\Omega(\Delta d^{1/\Delta})}$ . In particular, any polynomial-sized multilinear formula for  $\text{IMM}_d$  must have product-depth  $\Omega(\log d)$ .*

As the product-depth of a formula is at most its depth, Lemma 7, Corollary 11 imply:

► **Corollary 12** (Tightness of Brent's depth-reduction for multilinear formulas). *For each  $d \geq 1$ , there is an explicit polynomial  $F_d$  on  $O(d)$  variables such that  $F_d$  has a multilinear formula of size  $d^{O(1)}$ , but any multilinear formula of depth  $o(\log d)$  for  $F_d$  must have a size of  $d^{\omega(1)}$ .*

Choosing parameters carefully, we also obtain the following.

► **Corollary 13** (Separation of multilinear formulas and general formulas over zero characteristic). *Let  $\mathbb{F}$  be a field of characteristic zero. Let  $s \in \mathbb{N}$  be any growing parameter and  $\Delta \in \mathbb{N}$  be such that  $\Delta \leq o(\log s)$ . There is an explicit multilinear polynomial  $F_{s,\Delta}$  that has a  $(\Sigma\Pi)^\Delta\Sigma$  formula of size  $s$ , but any  $(\Sigma\Pi)^\Delta\Sigma$  multilinear formula for  $F_{s,\Delta}$  must have a size of  $s^{\omega(1)}$ .*

<sup>4</sup> All our logarithms will be to base 2.

**Proof.** We choose the polynomial  $F_{s,\Delta}$  to be  $\text{IMM}_d$  for suitable  $d$  and then simply apply Corollary 11 and Lemma 8 to obtain the result. Details follow.

Say  $\Delta = \frac{\log s}{f(s)}$  for some  $f(s) = \omega(1)$ . By Lemma 8, for any  $d$ ,  $\text{IMM}_d$  has a product-depth  $\Delta$  formula of size  $s(d, \Delta) = 2^{O(\Delta d^{1/2\Delta})}$ ; we choose  $d$  so that  $s(d, \Delta) = s$ . It can be checked that for  $d = \Theta(f(s))^{2\Delta}$ , this is indeed the case.

Having chosen  $d$  as above, we define  $F_{s,\Delta} = \text{IMM}_d$ . Clearly,  $F_{s,\Delta}$  has a (non-multilinear) formula of product-depth  $\Delta$  and size at most  $s$ . On the other hand, by Theorem 9, any multilinear product-depth  $\Delta$  formula for  $\text{IMM}_d$  must have size at least  $2^{\Omega(\Delta d^{1/\Delta})} = s^{\Omega(d^{1/2\Delta})} = s^{\Omega(f(s))} = s^{\omega(1)}$ , which proves the claim.

It can also be proved similarly that for  $d$  as chosen above,  $\text{IMM}_d$  in fact has no multilinear formulas of size  $s^{O(1)}$  and product-depth up to  $(2 - \varepsilon)\Delta$  for any absolute constant  $\varepsilon$ . ◀

## 4 Proof of Theorem 9

Our proof follows a two-step argument as in [15, 19] (see the exposition in [23, Section 3.6]).

**Step 1 – The product lemma.** The first step is a “product-lemma” for multilinear formulas. Formally, define a polynomial  $f \in \mathbb{F}[X]$  to be a *t-product polynomial* if we can write  $f$  as  $f_1 \cdots f_t$ , where we can find a partition of  $X$  into non-empty sets  $X_1^f, \dots, X_t^f$  such that  $f_i$  is a multilinear polynomial from  $\mathbb{F}[X_i^f]$ .<sup>5</sup> We say that  $X_i^f$  is the set *ascribed* to  $f_i$  in the *t-product polynomial*  $f$ . We use  $\text{Vars}(f_i)$  (with a slight abuse of notation)<sup>6</sup> to denote  $X_i^f$ . We drop  $f$  from the superscript if  $f$  is clear from the context.

We define  $f \in \mathbb{F}[X]$  to be *r-simple* if  $f = L_1 \cdots L_{r'} \cdot G$ , where  $r' \leq r$ , is an  $(r' + 1)$ -product polynomial where  $L_1, \dots, L_{r'}$  are polynomials of degree at most 1, the sets  $X_1^f, \dots, X_{r'}^f$  ascribed to these linear polynomials satisfy  $|\bigcup_{i \leq r'} X_i^f| \geq 400r$ . We prove the following.

► **Lemma 14.** *Let  $\Delta \leq \log d$ . Assume that  $f \in \mathbb{F}[X]$  can be computed by a syntactic multilinear  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F$  of size at most  $s$ . Then,  $f$  is the sum of at most  $s$  many *t-product polynomials* and at most  $s$  many *t-simple polynomials* for  $t = \Omega(\Delta d^{1/\Delta})$ .*

While our proof (presented in the full version [4]) of this lemma is motivated by earlier work [23, 8, 19], we give slightly better parameters, which is crucial for proving tight lower bounds for formulas. In particular, [19, Claim 5.5] yields the above but with  $t = \Omega(d^{1/\Delta})$ .

**Step 2 – Rank measure and the hard polynomial.** The second step is to show that any such decomposition for  $\text{IMM}_d$  must have many terms. Our proof of this step is inspired by the proof of the multilinear formula lower bound of Raz [15] for the determinant and also the slightly weaker lower bound of Nisan and Wigderson [13] for  $\text{IMM}_d$  in the *set-multilinear* case. Following [15], we define a suitable *random restriction*  $\rho$ , of the  $\text{IMM}_d$  polynomial by assigning variables from the set  $X$  to  $Y \cup Z \cup \{0, 1\}$ , where  $Y$  and  $Z$  are disjoint sets of new variables. As the restriction sets distinct variables in  $X$  to distinct variables in  $Y \cup Z$  or constants, it preserves multilinearity.

Having performed the restriction, we consider the *partial derivative matrix* of the restricted polynomial, which is defined as follows. Let  $g \in \mathbb{F}[Y \cup Z]$  be a multilinear polynomial. Define

<sup>5</sup> Note that we do not need  $f_i$  to depend non-trivially on all (or any) of the variables in  $X_i^f$ .

<sup>6</sup>  $\text{Vars}(\cdot)$  is used to describe variables ascribed to gates in a circuit as well as to denote variables ascribed to polynomials.



**Algorithm 1** Sampling algorithm  $\mathcal{S}$ .

- 
- 1: Choose  $\pi$  uniformly at random from  $\{1, 2\}^d$ . Define  $\pi(0) = 1$ .
  - 2: Choose  $a$  uniformly at random from  $\{0, 1\}^d$ . Let  $A = \{i \mid a_i = 1\}$ .
  - 3: **for**  $i \in [d]$  **do**
  - 4: Let  $b_i = 0$  if  $\pi(i-1) = \pi(i)$  and 1 if  $\pi(i-1) \neq \pi(i)$ .
  - 5: **end for**
  - 6: **for**  $i = 1$  to  $d$  **do**
  - 7: **if**  $i \notin A$  **then**
  - 8: Choose  $\rho|_{X^{(i)}}$  such that  $M^{(i)}$  is  $I$  if  $b_i = 0$  and  $E$  if  $b_i = 1$ . (In particular, all variables are set to constants from  $\{0, 1\}$ .)
  - 9: **else if**  $i \in A$  and  $i$  is the  $j$ th smallest element of  $A$  for odd  $j$  **then**
  - 10: Fix

$$\rho(x_{u,v}^{(i)}) = \begin{cases} y_{\lceil j/2 \rceil} & \text{if } u = \pi(i-1) \text{ and } v = \pi(i), \\ 1 & \text{if } u = \pi(i-1) \text{ and } v = \overline{\pi(i)}, \\ 0 & \text{otherwise.} \end{cases}$$

- 11: **else**
- 12: Now,  $i \in A$  and  $i$  is the  $j$ th smallest element of  $A$  for even  $j$ . We fix

$$\rho(x_{u,v}^{(i)}) = \begin{cases} z_{j/2} & \text{if } u = \pi(i-1) \text{ and } v = \pi(i), \\ 1 & \text{if } u = \overline{\pi(i-1)} \text{ and } v = \pi(i), \\ 0 & \text{otherwise.} \end{cases}$$

- 13: **end if**
  - 14: **end for**
- 

the  $2^{|Y|} \times 2^{|Z|}$  matrix  $M_{(Y,Z)}(g)$  such that rows and columns are labelled by distinct multilinear monomials in  $Y$  and  $Z$  respectively and the  $(m_1, m_2)$ th entry of  $M_{(Y,Z)}(g)$  is the coefficient of the monomial  $m_1 \cdot m_2$  in  $g$ .

Our restriction is defined to have the following two properties.

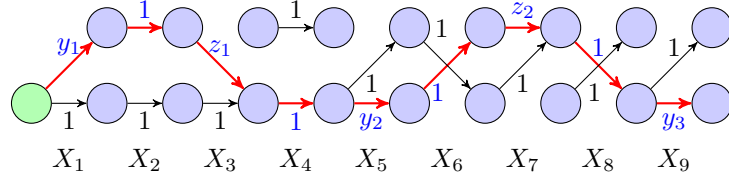
1. The rank of  $M_{(Y,Z)}(g)$  is equal to its maximum possible value (i.e.  $\min\{2^{|Y|}, 2^{|Z|}\}$ ) with probability 1 where  $g$  is the restricted version of  $\text{IMM}_d$ .
2. On the other hand, let  $f$  be either a  $t$ -product or a  $t$ -simple polynomial, and let  $f'$  denote its restriction under  $\rho$ . Then, the rank of  $M_{(Y,Z)}(f')$  is small with high probability.

Now, if  $\text{IMM}_d$  has a  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F$  of small size, then it is a sum of a small number of  $t$ -product and  $t$ -simple polynomials by Lemma 14 and hence by a union bound, we will be able to find a restriction under which the partial derivative matrices of each of these polynomials have a small rank. By the subadditivity of rank, this will imply that  $M_{(Y,Z)}(g)$  will itself have low rank, contradicting the first property of our restriction.

To make the above precise, we first define our restrictions. Let  $\tilde{Y} = \{y_1, \dots, y_d\}$  and  $\tilde{Z} = \{z_1, \dots, z_d\}$  be two disjoint sets of variables. A restriction  $\rho$  is a function mapping the set  $X$  to  $\tilde{Y} \cup \tilde{Z} \cup \{0, 1\}$ . We consider the following process for sampling a random restriction.

Recall that  $M^{(i)}$  is the  $2 \times 2$  matrix whose  $(u, v)$ th entry is  $x_{u,v}^{(i)}$ . Let  $I$  and  $E$  denote the standard  $2 \times 2$  identity matrix and the  $2 \times 2$  flip permutation matrix respectively. For  $a \in \{1, 2\}$ , we use  $\bar{a}$  to denote the other element of the set.

We give a procedure  $\mathcal{S}$  for sampling a random restriction  $\rho : X \rightarrow \tilde{Y} \cup \tilde{Z} \cup \{0, 1\}$  in Algorithm 1. Based on the output  $\rho$  of  $\mathcal{S}$ , we define the (random) sets  $Y = \tilde{Y} \cap \text{Img}(\rho)$  and  $Z = \tilde{Z} \cap \text{Img}(\rho)$ . Let  $m = m(\rho) = \min\{|Y|, |Z|\}$ .



■ **Figure 2** Effect of  $\rho$  on  $\text{IMM}_9$  when the sampling algorithm  $\mathcal{S}$  yields  $\pi = (2, 2, 1, 1, 1, 2, 2, 1, 1)$  and  $a = (1, 0, 1, 0, 1, 0, 1, 0, 1)$ . Thus,  $\text{IMM}_9|_\rho$  in this case yields us  $(1 + y_1 z_1)(1 + y_2 z_2)(1 + y_3)$ .

► **Observation 15.** *The restriction  $\rho$  satisfies the following.*

1.  $|Y| = \lceil |A|/2 \rceil$  and  $|Z| = \lfloor |A|/2 \rfloor$ . Hence,  $|Z| \leq |Y| \leq |Z| + 1$  and  $m = |Z|$ .
2. Distinct variables in  $X$  cannot be mapped to the same variable in  $Y \cup Z$ .
3. Only the variables of the form  $x_{\pi(i-1), \pi(i)}^{(i)}$  can be set to variables in  $Y \cup Z$  by  $\rho$ . The rest are set to constants.

Note that  $b$  is distributed uniformly over  $\{0, 1\}^d$ . Given a polynomial  $f \in \mathbb{F}[X]$ , the restriction  $\rho$  yields a natural polynomial  $f|_\rho \in \mathbb{F}[Y \cup Z]$  by substitution. Note, moreover, that if  $f$  is multilinear then so is  $f|_\rho$  since distinct variables in  $X$  cannot be mapped to the same variable in  $Y \cup Z$  (Observation 15).

► **Lemma 16.** *Let us assume that  $\rho$  is sampled as above. Then we have the following:*

1.  $\text{rank}(M_{(Y,Z)}(\text{IMM}_d|_\rho)) = 2^m$  with probability 1.
2. If  $f \in \mathbb{F}[X]$  is any  $t$ -product polynomial, then for some absolute constant  $\varepsilon > 0$ ,  $\Pr[\text{rank}(M_{(Y,Z)}(f|_\rho)) \geq 2^{m-\varepsilon t}] \leq \frac{1}{2^{\Omega(t)}}$ .
3. If  $f \in \mathbb{F}[X]$  is any  $r$ -simple polynomial, then for some absolute constant  $\delta > 0$ ,  $\Pr[\text{rank}(M_{(Y,Z)}(f|_\rho)) \geq 2^{m-\delta r}] \leq \frac{1}{2^{\Omega(r)}}$ .

Given Lemmas 14 and 16, we can finish the proof of Theorem 9 as follows.

**Proof of Theorem 9 assuming Lemma 16.** Assume that  $\text{IMM}_d$  is computed by a syntactic multilinear  $(\Sigma\Pi)^\Delta\Sigma$  formula  $F$  of size at most  $s$ . By Lemma 14, we get that  $f$  can be expressed as a sum of at most  $2s$  many summands, say  $f_1, f_2, \dots, f_s$  and  $g_1, g_2, \dots, g_s$ , where each summand  $f_i$  is a  $t$ -product polynomial and each summand  $g_j$  is a  $t$ -simple polynomial for  $t = \Omega(\Delta d^{1/\Delta})$ .

For each  $i \in [s]$ , Lemma 16 implies that  $\Pr[\text{rank}(M_{(Y,Z)}(f_i|_\rho)) \geq 2^{m-\varepsilon t}] \leq \frac{1}{2^{\Omega(t)}}$  and  $\Pr[\text{rank}(M_{(Y,Z)}(g_i|_\rho)) \geq 2^{m-\delta t}] \leq \frac{1}{2^{\Omega(t)}}$ , where  $\varepsilon$  and  $\delta$  are absolute constants.

Thus, unless  $s \geq 2^{\Omega(t)}$ , we see by a union bound that there exists a  $\rho$  such that for each  $i \in [s]$ ,  $\text{rank}(M_{(Y,Z)}(f_i|_\rho)) \leq 2^{m-\varepsilon t}$  and  $\text{rank}(M_{(Y,Z)}(g_i|_\rho)) \leq 2^{m-\delta t}$ . For such a  $\rho$ , we have  $\text{rank}(M_{(Y,Z)}(F|_\rho)) \leq 2^m \cdot \left(\frac{s}{2^{\varepsilon t}} + \frac{s}{2^{\delta t}}\right) < 2^m$  unless  $s \geq 2^{\Omega(t)}$ .

From Lemma 16, we also know that for *any* choice of  $\rho$  in the sampling algorithm  $\mathcal{S}$ , we have  $\text{rank}(M_{(Y,Z)}(\text{IMM}_d|_\rho)) \geq 2^m$ . In particular, since  $F$  computes  $\text{IMM}_d$ , we must have  $s \geq 2^{\Omega(t)} = 2^{\Omega(\Delta d^{1/\Delta})}$ . ◀

**Proof of Lemma 16.**

**Part 1:  $\text{IMM}_d$  has high rank.** Let  $\pi \in \{1, 2\}^d$  and  $a \in \{0, 1\}^d$  be arbitrary. Note that in our sampling algorithm,  $\rho, A, b$  are completely determined given  $\pi$  and  $a$ .

Let us now examine the effect of  $\rho$  on  $\text{IMM}_d$ . We take the graph theoretic view of the polynomial  $\text{IMM}_d$  as given in Section 2.1.

Figure 2 illustrates how this restriction affects the variables labelling the edges of the graph  $G_d$  defined in Section 2.1. By substituting according to  $\rho$  in (1), we get that  $\text{IMM}_d(X)|_\rho = \prod_{i=1}^m (1 + y_i z_i)$  if  $|A| = 2m$  and  $\prod_{i=1}^m (1 + y_i z_i) \cdot (1 + y_{m+1})$  if  $|A| = 2m + 1$ , where  $m = |Z|$ . For any  $S \subseteq [m]$ , let  $Z_S$  (resp.,  $Y_S$ ) denote the monomial  $\prod_{i \in S} z_i$  (resp.,  $\prod_{i \in S} y_i$ ). Now consider the matrix  $M_{(Y,Z)}(\text{IMM}_d|_\rho)$ . We will simply use  $\mathcal{M}$  to denote this matrix. For the sake of simplicity let us assume that  $|A| = 2m$ . (The case when  $|A| = 2m + 1$  is similar.) Let the rows and columns of  $\mathcal{M}$  be labelled by the subsets of  $[m]$  and let  $\mathcal{M}(S, T)$  be the coefficient of  $Y_S \cdot Z_T$  in  $\text{IMM}_d|_\rho$ . It is easy to see that  $\mathcal{M}(S, T) = 0$  if  $S \neq T$  and 1 otherwise. That is,  $\mathcal{M}$  is the Identity matrix of size  $2^m \times 2^m$  and hence it has full rank.<sup>7</sup>  $\triangleleft$

**Part 2:  $t$ -product polynomials have low rank.** We now prove that for a  $t$ -product polynomial  $f$ ,  $\text{rank}(M_{(Y,Z)}(f|_\rho))$  is small w.h.p.

Let  $f$  be a  $t$ -product polynomial, i.e.  $f = f_1 f_2 \dots f_t$ . Let  $\chi : X \rightarrow [t]$  be a coloring function, which assigns colors to all the variables in  $X$ , so that  $\chi^{-1}(i) = X_i^f$ , where  $X_i^f$  is the variable set ascribed to  $f_i$ . That is, all the variables ascribed to  $f_i$  are assigned color  $i$  under the coloring function. To prove the lemma, we first show that, with high probability (over the choice of  $\pi$ ), a constant fraction of the  $t$  colors appear along the path defined by  $\pi$ , i.e. along  $(\pi(0), \pi(1)), (\pi(1), \pi(2)), \dots, (\pi(d-1), \pi(d))$ . Given such a *multi-colored path*, we then show that with a high probability, over the choice of  $a$ , many of the colors have an *imbalance*. A color is said to have an imbalance under  $\rho$  if more variables from  $X$  of that color are mapped to the  $Y$  variables than the  $Z$  variables or vice versa. We then appeal to arguments similar to those in [15, 19, 5] to conclude that the imbalance results in a low rank.

**Variable coloring,  $t$ -product polynomials and imbalance.** We start with some notation. Given a string  $\pi \in \{1, 2\}^d$ , let the path defined by  $\pi$  be the following sequence of pairs  $(\pi(0), \pi(1)), (\pi(1), \pi(2)), \dots, (\pi(d-1), \pi(d))$  (we call it a path since these pairs correspond naturally to the edges of a path in the graph  $G_d$  defined in Section 2.1). We say that a color  $\gamma \in [t]$  appears in layer  $\ell \in [d]$  if there exist  $u, v \in \{1, 2\}$  such that  $\gamma = \chi(x_{u,v}^{(\ell)})$ .

Let  $C^0 = \emptyset$  and let  $C^i = C^{i-1} \cup \{\chi(x_{u,v}^{(i)}) \mid u, v \in \{1, 2\}\}$  for  $i \in [d]$ , i.e.,  $C^i$  contains all the distinct colors appearing in layers  $\{1, 2, \dots, i\}$ . Therefore,  $|C^d| = t$ . We will also define  $O^{2i+1}$  to be all the colors appearing in odd numbered layers up to  $2i+1$ , i.e.  $O^{2i+1} = O^{2i-1} \cup \{\chi(x_{u,v}^{(2i+1)}) \mid u, v \in \{1, 2\}\}$ . Similarly, we define  $E^{2i} = E^{2i-2} \cup \{\chi(x_{u,v}^{(2i)}) \mid u, v \in \{1, 2\}\}$ .

Let  $C_\pi^0 = \emptyset$  and  $C_\pi^i = C_\pi^{i-1} \cup \{\chi(x_{(\pi(i-1), \pi(i))}^{(i)})\}$ , i.e.  $C_\pi^i$  contains all the distinct colors appearing along the path defined by  $\pi$  up to layer  $i$ . We first observe a property of  $C_\pi^d$  stated in the claim below.

► **Claim 17.** *If  $|C^d| = t$ , then  $\Pr_\pi[|C_\pi^d| \leq t/100] \leq 1/2^{\Omega(t)}$ .*

We will assume the claim and finish the proof of Part 2 of Lemma 16. The above claim shows that a lot of colors appear on the uniformly random path  $\pi$  with high probability. Using this, we will now show that a constant fraction of these colors also exhibit an imbalance with a high probability. Using the multiplicativity of the rank, we will then show that the imbalance for a large number of factors results in the low rank of the matrix  $M_{Y,Z}(f|_\rho)$ .

We will say that  $\pi$  is good if  $|C_\pi^d| > t/100$ . Let  $L = t/100$ . The above claim shows that a random  $\pi$  is good with high probability. In what follows, we condition on picking a good  $\pi$ . Let  $a \in \{0, 1\}^d$  be chosen uniformly at random as in the sampling algorithm. Let  $\rho$  be defined as in the sampling algorithm for  $\pi, a$ .

<sup>7</sup> If  $|A| = 2m + 1$  then  $\mathcal{M}$  has a  $2^m \times 2^m$  sized Identity matrix as a submatrix.

## 21:12 Small-depth Multilinear Formula Lower Bounds for IMM, with Applications

Let  $\gamma \in C_\pi^d$  be a color that appears along  $\pi$ . Let  $\pi_\gamma$  be the elements along the path defined by  $\pi$  with color  $\gamma$ , i.e.  $\pi_\gamma = \{(\pi(i-1), \pi(i)) \mid \chi(x_{(\pi(i-1), \pi(i))}^{(i)}) = \gamma\}$ . Let  $\rho(\pi_\gamma) = \{\rho(x_{(\pi(i-1), \pi(i))}^{(i)}) \mid (\pi(i-1), \pi(i)) \in \pi_\gamma\} \cap (Y \cup Z)$ . A color  $\gamma \in [t]$  is said to have an imbalance w.r.t.  $\rho$  if  $|\rho(\pi_\gamma) \cap Y| - |\rho(\pi_\gamma) \cap Z| \geq 1$ .

It is easy to see that if  $|\rho(\pi_\gamma)|$  is odd, then  $\gamma$  has an imbalance w.r.t.  $\rho$ . Note that the former event is equivalent to the event that  $\bigoplus_{i \in P_\gamma} a_i$  equals 1 where  $P_\gamma = \{i \mid (\pi(i-1), \pi(i)) \in \pi_\gamma\}$ . Hence for any  $\gamma \in C_\pi^d$ ,  $\Pr[\gamma \text{ has an imbalance with respect to } \rho \text{ along } \pi] = 1/2$ . Further, as  $|C_\pi^d| \geq L$  and the events corresponding to distinct  $\gamma \in C_\pi^d$  are mutually independent, the Chernoff bound gives  $\Pr[\leq L/4 \text{ colors have an imbalance w.r.t. } \rho \text{ along } \pi] \leq 1/2^{\Omega(L)}$ .  $\triangleleft$

Assuming Claim 17 we are now done. The proof of Claim 17 can be found in [4].

**Imbalance implies low rank.** Let us recall that  $f = f_1 f_2 \dots f_t$  is a  $t$ -product polynomial that is defined over the disjoint variable partition  $X = X_1 \cup X_2 \cup \dots \cup X_t$  such that  $|X_i| \geq 1$  for all  $i \in [t]$ . The following lemma (see, e.g., [19]) will be useful in bounding  $\text{rank}(M_{(Y,Z)}(f|_\rho))$ .

► **Lemma 18** ([19], Proposition 2.5). *Let  $g = g_1 g_2 \dots g_t$  be a  $t$ -product polynomial over the set of variables  $Y \cup Z$  where  $\text{Vars}(g_i) = Y_i \cup Z_i$ . Then  $\text{rank}(M_{(Y,Z)}(g)) = \prod_{i \in [t]} \text{rank}(M_{(Y_i, Z_i)}(g_i))$ .*

From Lemma 18, we get that  $\text{rank}(M_{(Y,Z)}(f|_\rho)) = \prod_{i=1}^t \text{rank}(M_{(Y_i, Z_i)}(f_i|_\rho))$  where  $Y_i = Y \cap \{\rho(x) \mid x \in X_i\}$  and  $Z_i = Z \cap \{\rho(x) \mid x \in X_i\}$ . For all  $i \in [t]$ , from the definition it is clear that the rank of the matrix  $M_{(Y_i, Z_i)}(f_i|_\rho)$  is upper bounded by  $2^{\min\{|Y_i|, |Z_i|\}} \leq 2^{(|Y_i| + |Z_i|)/2}$ . Let us note that these disjoint partitions in the  $t$ -product polynomial correspond to the colors in the coloring  $\chi$  with all variables in  $X_i$  colored  $i$ . Hence if color  $i$  has imbalance w.r.t.  $\rho$ , then  $\text{rank}(M_{(Y_i, Z_i)}(f_i|_\rho)) \leq 2^{\min\{|Y_i|, |Z_i|\}} \leq 2^{(|Y_i| + |Z_i| - 1)/2}$ . Thus,  $\text{rank}(M_{(Y,Z)}(f|_\rho)) \leq \prod_{i=1}^t 2^{(|Y_i| + |Z_i| - 1)/2} = 2^{((|Y| + |Z|) - \ell)/2} \leq 2^{m - (\ell - 1)/2}$  where  $\ell$  is the number of colors that have imbalance w.r.t.  $\rho$ . From the above discussion, we can infer that  $\Pr_\pi[\text{rank}(M_{Y,Z}(f|_\rho)) \geq 2^{m - (t/1000)}] \leq \Pr_\pi[\ell \leq t/400] \leq \frac{1}{2^{\Omega(t)}}$ .

**Part 3:  $r$ -simple polynomials have low rank.** Here we prove that if  $f \in \mathbb{F}[X]$  is any  $r$ -simple polynomial, then for some absolute constant  $\delta > 0$ ,  $\Pr[\text{rank}(M_{(Y,Z)}(f|_\rho)) \geq 2^{m - \delta r}] \leq \frac{1}{2^{\Omega(r)}}$ .

As  $f$  is an  $r$ -simple polynomial we know that  $f = \left(\prod_{i=1}^{r'} L_i\right) \cdot G$ , where  $r' \leq r$ ,  $L_i$ s are linear polynomials,  $\forall i \in [r']$   $X_i$  is the set of variables ascribed to  $L_i$  and  $X_{r'+1}$  is the set of variables ascribed to  $G$ . Moreover,  $|\cup_{i=1}^{r'} X_i| \geq 400r$ .

To prove the above statement we set up some notation. Let  $f|_\rho = \left(\prod_{i=1}^{r'} L_i|_\rho\right) \cdot G|_\rho$ . Let  $Y_i = \{\rho(x) \mid x \in X_i\} \cap Y$  and  $Z_i = \{\rho(x) \mid x \in X_i\} \cap Z$  for each  $i \in [r']$ . Let  $Y' = \cup_{i=1}^{r'} Y_i$  and  $Z' = \cup_{i=1}^{r'} Z_i$ . Also, let  $Y'' = Y \setminus Y'$  and  $Z'' = Z \setminus Z'$ . Let  $U$  denote  $\cup_{i=1}^{r'} X_i$  and let  $U|_\rho = (\cup_{i=1}^{r'} Y_i) \cup (\cup_{i=1}^{r'} Z_i)$ .

In the following claim we show that if  $U$  is a large set to begin with then with high probability (over the restriction  $\rho$  defined by the sampling algorithm),  $U|_\rho$  is also large.

► **Claim 19.** *If  $|U| \geq 400r$ , then  $\Pr[|U|_\rho] \leq 4r] \leq \frac{1}{2^{\Omega(r)}}$ .*

We first finish the proof of Part 3 of Lemma 16 assuming this claim.

We say that a restriction  $\rho$  is good if we get  $|U|_\rho \geq 4r$ . In what follows we will condition on the event that we have a good  $\rho$ . For a restriction  $\rho$ , for each  $i \in [r']$ , we can write  $L_i|_\rho(Y_i, Z_i)$  as  $L_i'|_\rho(Y_i) + L_i''|_\rho(Z_i)$  as  $L_i$ s are linear polynomials. Therefore we get  $\prod_{i=1}^{r'} L_i|_\rho(Y', Z') = \sum_{S \subseteq [r']} \prod_{i \in S} L_i'|_\rho(Y_i) \cdot \prod_{j \in [r'] \setminus S} L_j''|_\rho(Z_j)$ .

Let  $L_S$  denote the polynomial  $\prod_{i \in S} L_i|_{\rho}(Y_i) \cdot \prod_{j \in [r'] \setminus S} L_j''|_{\rho}(Z_j)$ . Note that for all  $S \subseteq [r']$ ,  $\text{rank}(M_{(Y', Z')} (L_S))$  is at most 1. Therefore, by the subadditivity of matrix rank, we get that  $\text{rank}\left(M_{(Y', Z')} \left(\prod_{i=1}^{r'} L_i|_{\rho}(Y', Z')\right)\right) \leq 2^{r'} \leq 2^r$ . We can now bound  $\text{rank}(M_{(Y, Z)}(f|_{\rho}))$ .

$$\begin{aligned} \frac{\text{rank}(M_{(Y, Z)}(f|_{\rho}))}{2^{(|Y|+|Z|)/2}} &= \frac{\text{rank}\left(M_{(Y, Z)}\left(\prod_{i=1}^{r'} L_i|_{\rho} \cdot G|_{\rho}\right)\right)}{2^{(|Y|+|Z|)/2}} \\ &= \frac{\text{rank}\left(M_{(Y', Z')} \left(\prod_{i=1}^{r'} L_i|_{\rho}\right)\right)}{2^{(|Y'|+|Z'|)/2}} \cdot \frac{\text{rank}(M_{(Y'', Z'')} (G|_{\rho}))}{2^{(|Y''|+|Z''|)/2}} \\ &\leq \frac{2^r}{2^{|\cup|_{\rho}|/2}} \cdot 1 \leq \frac{2^r}{2^{2r}} = \frac{1}{2^r}. \end{aligned}$$

where the second equality follows from Lemma 18. Therefore, we have  $\text{rank}(M_{(Y, Z)}(f|_{\rho})) \leq 2^{(|Y|+|Z|)/2} / 2^r \leq 2^{m+(1/2)-r}$  for any good  $\rho$ .

As Claim 19 tells us that  $\rho$  is good with probability  $1 - 1/2^{\Omega(r)}$ , we are done.  $\triangleleft$

Assuming Claim 19 (proof in [4]) we are done with the proof of Part 3 of Lemma 16.  $\blacktriangleleft$

---

## References

- 1 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25–28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.
- 2 Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–14:36, 2010. doi:10.1145/1706591.1706594.
- 3 Richard P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21(2):201–206, 1974. doi:10.1145/321812.321815.
- 4 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication, with applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:156, 2017. URL: <https://eccc.weizmann.ac.il/report/2017/156>.
- 5 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012. doi:10.1145/2213977.2214034.
- 6 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 128–135. ACM, 2014. doi:10.1145/2591796.2591824.
- 7 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. doi:10.1137/140957123.
- 8 Pavel Hrubeš and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011.
- 9 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18–21, 2014*, pages 61–70. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.15.

- 10 Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (roabps) and multilinear depth three circuits. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPICs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.STACS.2016.46.
- 11 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 626–632. ACM, 2016. doi:10.1145/2897518.2897550.
- 12 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 364–373. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.46.
- 13 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. doi:10.1007/BF01294256.
- 14 Ran Raz. Multilinear-nc neq multilinear-nc. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 344–351. IEEE Computer Society, 2004. doi:10.1109/FOCS.2004.42.
- 15 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.
- 16 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.
- 17 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. doi:10.1137/070707932.
- 18 Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Computational Complexity*, 17(4):515–535, 2008. doi:10.1007/s00037-008-0254-0.
- 19 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.
- 20 Benjamin Rossman. The average sensitivity of bounded-depth formulas. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 424–430. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.33.
- 21 Benjamin Rossman and Srikanth Srinivasan. Separation of  $ac^0$ [oplus] formulas and circuits. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPICs*, pages 50:1–50:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.ICALP.2017.50.
- 22 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015. URL: <https://github.com/dasarpmar/lowerbounds-survey/releases/>.
- 23 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. doi:10.1561/04000000039.
- 24 Philip M Spira. Computation times of arithmetic and boolean functions in  $(d, r)$  circuits. *IEEE Transactions on Computers*, 100(6):552–555, 1973.
- 25 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. doi:10.1016/j.ic.2014.09.004.

- 26 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. doi:10.1137/0212043.