On Problems of Hardness, Counting, and Factoring in Algebraic Complexity

A thesis submitted

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

by

Bhargav C S

to the

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

August, 2025

CERTIFICATE

It is certified that the work contained in the thesis titled **On Problems of Hardness, Counting, and Factoring in Algebraic Complexity**, by **Bhargav C S**, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

[illegible]

Prof Nitin Saxena
Department of Computer Science & Engineering
IIT Kanpur

August, 2025

DECLARATION

This is to certify that the thesis titled **On Problems of Hardness, Counting, and Factoring in Algebraic Complexity** has been authored by me. It presents the research conducted by me under the supervision of **Prof Nitin Saxena**.

To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted elsewhere, in part or in full, for a degree. Further, due credit has been attributed to the relevant state-of-the-art and collaborations (if any) with appropriate citations and acknowledgements, in line with established norms and practices.

[illegible]

Bhargav C S
Roll no.: 19111269
Program: Doctor of Philosophy
Department of Computer Science & Engineering
Indian Institute of Technology Kanpur

ABSTRACT

Name of student: **Bhargav C S** Roll no: **19111269**

Degree for which submitted: **Doctor of Philosophy**

Department: **Computer Science & Engineering**

Thesis title: **On Problems of Hardness, Counting, and Factoring in Algebraic Complexity**

Name of Thesis Supervisor: **Prof Nitin Saxena**

Month and year of thesis submission: **August, 2025**

"How best to compute a multivariate polynomial using arithmetic operations?" – this is the central question that has driven the results in this thesis. Polynomials are perhaps the most fundamental mathematical objects (after numbers). In theoretical computer science alone, they are a crucial ingredient to applications in cryptography, error-correcting codes, numerical analysis and approximation, computer vision, robotics, machine learning, and much more. It is very natural to study their complexity, and *algebraic circuits* – directed acyclic graphs with addition and multiplication gates, and variables as leaves, are the most natural models of computing polynomials (in a syntactic way). In the late '70s, Valiant (STOC, 1979) conjectured the algebraic analog of Cook's P vs. NP – that there are low-degree polynomials whose coefficients can be efficiently described (the class VNP), but the polynomial itself cannot be computed by low-degree small-sized circuits (the class VP).

Bounded-depth circuits. Unlike their Boolean counterparts, a surprising phenomenon of "efficient parallelization" holds true for algebraic circuits. In an influential line of studies beginning with the work of Agrawal and Vinay (FOCS

2008), improved by Koiran (Theoret. Comput. Sci. 2012) and Tavenas (Inform. and Comput. 2015), finally culminating in the work of Gupta *et al.* (SIAM J. Comput. 2016), it was shown that even algebraic circuits with just 3 or 4 layers of gates are *almost* as powerful as general ones! Proving sufficiently strong lower bounds for these *constant-depth* circuits can resolve Valiant's conjecture, and has been the focus of lower bounds research for the past decade in Algebraic Complexity.

Recently, in a breakthrough work, Limaye, Srinivasan, and Tavenas (FOCS 2021) proved *superpolynomial* lower bounds for all constant-depth circuits. As a natural next step, we push their techniques further and prove stronger bounds that are state of the art. We also exhibit a *barrier* to further improvement of the bounds using the same techniques.

Algebraic branching programs. Models different from circuits have also been studied. Algebraic *formulas* are circuits where the directed acyclic graph is a tree. Algebraic *branching programs* are a model of computation intermediate in power between formulas and circuits. Valiant's conjecture is generally considered the problem of showing superpolynomial lower bounds for any of these models. Much as in the Boolean world, the lower bounds we know for these general models are barely quadratic.

In this thesis, we show that proving strong enough lower bounds for the sum of a very restricted type of algebraic branching program that is *set-multilinear*, is enough to resolve Valiant's conjecture. An analogous statement for bounded-depth circuits was used by Limaye, Srinivasan and Tavenas (FOCS 2021) to prove their superpolynomial lower bounds. Nisan (STOC 1991) had already studied these restricted branching programs in the '90s and shown *exponential* lower bounds against them. Our work shows that their *sum* is surprisingly powerful. We prove exponential lower bounds for this 'sum' model (in fact, even the sum of *general* branching programs) when the size of each program is sub-polynomial.

Counting homomorphisms. In another seminal work, Valiant (Theoret. Comput. Sci. 1979) showed that counting the number of perfect matchings in bipartite graphs is #P–hard. In general, the question of detecting and counting some "pattern" graph in a "host" graph is very interesting and has many applications in combinatorics, computer science (theoretical and otherwise) and even biology. If the pattern graph is fixed, these problems can be solved in time polynomial in the size of the host graph, which is usually large. So we would really prefer much faster algorithms. An algebraic version of this problem is to consider the *Homomorphism* polynomial that enumerates 'adjacency-preserving' maps from the pattern graph to the host, and construct small algebraic circuits computing the polynomial. The best known constructions are in fact *monotone*, where the circuit never uses negative numbers. Recently, Komarath, Pandey and Rahul (Algorithmica 2023) showed that the *treewidth* of the pattern graph completely characterizes the size of the best monotone circuit computing the polynomial.

We generalize their results to *bounded-depth* monotone circuits computing the homomorphism polynomial. We introduce a bounded-depth version of treewidth (where tree decompositions have bounded depth), and show that this parameter characterizes the size of the bounded-depth monotone circuit computing the homomorphism polynomial. Proving these fine-grained bounds further leads to an *optimal* depth-hierarchy theorem for monotone circuits.

Polynomial factorization. In the 1980's, Kaltofen (SICOMP 1985, STOC 1986, STOC 1987, RANDOM 1989) showed a remarkable fact – factors of small multivariate circuits of low degree are small circuits themselves, i.e., VP is closed under taking factors. This is indeed a splendid demonstration of the robustness and naturalness of the algebraic circuit model. Via repeated squaring, small circuits can also compute polynomials of *exponential degree*. An earlier result of Lipton and Stockmeyer (J. Comput. System Sci. 1978) implies that such circuits are *not* closed under taking factors.

Kaltofen (STOC 1986, 1987) wondered, and Bürgisser in his monograph (2000)

conjectured, that at least the *low-degree* factors of small exponential-degree circuits can be computed by small circuits. Except for certain special cases, the above 'factor conjecture' is open. Interestingly, Bürgisser (Found. Comput. Math 2004) showed that these low-degree factors can be *approximated* by small circuits! Thus, the factor conjecture can be seen as saying that approximation is not necessary. These approximate circuits are in fact more structured, and when the underlying field is finite, this structure can be exploited to show that these factors have coefficients that are easy to describe. That is, the families defined by the factors are in VNP (the factor conjecture says they are in VP). In this thesis, we suggest natural conjectures to extend this result to infinite fields.

To the giants of yore, on whose shoulders I stand

Acknowledgements

I began my PhD wondering how I would ever publish a novel result. My advisor Nitin Saxena, nonetheless, wanted me to think about hard problems along with him right away. Horrified at first, I soon realized this to be the only sane path to research. Over the years, Nitin has shown by example how to pick problems worth solving, modify them when stuck, and most counter-intuitively to me, how to use conjectures that repeatedly fail, to deepen understanding, until they fail no more. I thank him for his generous patience in guiding me to be a better researcher, while giving me the freedom to explore my interests, which were rather tangential, at times. I have learnt a great deal from him, and am really fortunate to have had him as my advisor.

I learnt a tremendous amount from my collaborators. Working with Sagnik Dutta was so much fun, it sort of made up for our joint inexperience. Even with the endless distractions discussing music, chess and movies, we did manage to prove something! Most of the work in this thesis is joint with Prateek Dwivedi. In him, I found a dear friend and an excellent collaborator. I shared a large part of my PhD journey with him, and am very grateful for it. Our discussions ranged from the most recent papers all the way to debates about decades old definitions. I greatly enjoyed these, and learnt a lot, both about mathematics and collaboration. The first time I met Radu Curticapean and Shiteng Chen was when Radu hosted me and Prateek for a summer of counting and algebraic complexity. Working with them has been an absolute pleasure, and I am eagerly looking forward to more.

Any serious interest of mine in Mathematics came due to my uncle, Manjunath Krishnapur. Since the time he showed me Cantor's proof of the uncountability of reals, I was hooked. I thank him for introducing me to this beautiful world of Mathematics. I hope to some day have a fraction of his clarity of thinking and understanding. My interest in theory CS solidified after attending the amazing courses by Meena Mahajan at IMSc, and by Partha Mukhopadhyay and Prajakta Nimbhorkar at CMI. I thank them for their guidance and inspiration.

A PhD is, by nature, somewhat solitary. Akhil, Koustav, Mahesh, Muzafar, Prateek, and Vipin dampened its effects. Their support and camaraderie made the stay in Kanpur, far away from family, infinitely better than it would have otherwise been. So did the friendship and guidance of Abhranil, Amit, Ashish, Pranav, Pranjal, Priyanka, Rajendra and Sumanta. I hold fond memories of times spent with Anindya, Arun, Avideep, Charan, Dattatreya, Foram, Madhavan, Shweta, Tufan and Waseem. During the trips back home, I have been really grateful for the constant presence and friendship of all the three Akshays, Anuttam, Bharath, Darshan, Gourish, KK, and Vishwa. It has kept me anchored, and showed me the pleasures in life outside academia.

During my PhD, I was fortunate to visit many people and have enlightening discussions. I thank Nutan Limaye, Partha Mukhopadhyay, Radu Curticapean, and Shubhangi Saraf for hosting me during these visits. The organizers of WACT 2023 and WACT 2025 with their generous funding enabled me to meet researchers around the world in algebraic complexity. I am grateful to them for that. I also thank Nitin, IIT Kanpur and the department of CSE, Microsoft Research, and ACM IARCS for making many of these visits possible.

Sunil Simon has been a source of guidance and kindness since I began my PhD. I thank him and Rajat Mittal for being on my doctoral committee and guiding me through the years. Satyadev Nandakumar, and Raghunath Tewari taught great theory courses that broadened my horizon. I thank the office staff at CSE IIT Kanpur, especially Rajesh ji for helping me navigate the labyrinths of bureaucracy.

My deepest gratitude is to my family. I am thankful to my brother Sharat for always being there to offer support and laughter when I needed it the most. It carried me through many hard times. To my incredible wife Shanumugapriya, words cannot fully express how grateful I am for her endless love, support, and patience throughout this long and challenging journey. Her belief in me is bigger than my own. This would have been impossible without her. Finally, I owe an unpayable debt to *Amma* and *Anna*, who have sacrificed so much for me to be here. They have been steadfast in their love, support and trust, even during times of great uncertainty. I have been very lucky indeed.

Contents

List of Publications

The following is a list of publications and preprints on which the thesis is based.

Journal articles

[BDS24a] C. S. Bhargav, Sagnik Dutta, and Nitin Saxena. *Improved Lower Bound, and Proof Barrier, for Constant Depth Algebraic Circuits*. In: *ACM Transactions on Computation Theory* 16.4 (2024), Art. 23, 22. ISSN: 1942-3454,1942-3462. Conf. version - MFCS 2022.

[BDS25b] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *Lower Bounds for the Sum of Small-Size Algebraic Branching Programs*. In: *Theoretical Computer Science* 1041 (2025), Paper No. 115214, 11. ISSN: 0304-3975. Conf. version - TAMC 2024.

Conference articles

[BDS24b] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *Learning the Coefficients: A Presentable Version of Border Complexity and Applications to Circuit Factoring*. In: *STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. ACM, New York, 2024, pp. 130–140.

Manuscripts

[BDS25a] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *A Primer on the Closure of Algebraic Complexity Classes under Factoring*. 2025. arXiv: `2506.19604 [cs]`. The preprint is under review in the special issue of the workshop RTCA'23 Paris.

[Bha+25] C. S. Bhargav, Shiteng Chen, Radu Curticapean, and Prateek Dwivedi. *Monotone Bounded-Depth Complexity of Homomorphism Polynomials*. 2025. arXiv: `2505.22894 [cs]`. To appear in 50th International Symposium on Mathematical Foundations of Computer Science (MFCS) 2025.

List of Figures

Chapter 1

Introduction

Few ideas are as deeply pervasive in mathematics as the *polynomial*. The word itself is derived from the Greek *poly* (meaning 'many'), and the Latin *nomen* (meaning 'names' or in this case, 'terms'). True to its etymology, a polynomial such as $b^2 - 4ac$ in the variables a, b, and c is a very humble object – a sum of terms b^2 and $-4ac$, that we call *monomials*. Nevertheless, we learn in high school that for a quadratic univariate polynomial $f = ax^2 + bx + c$, the value $b^2 - 4ac$ tells us quite a bit about the roots of f (whether they are real, imaginary, distinct, etc). This sort of encoding of a problem as properties of a polynomial is extremely common. In the case above, the *discriminant* polynomial $b^2 - 4ac$ encodes key information about other quadratic polynomials! What makes polynomials fundamental is the vast number of problems whose encoding and solution they facilitate. Surely then, it is natural to study their various properties. In this thesis, we will mainly be concerned with questions of a *computational* nature.

1. Are all polynomials easy to compute? How does one show a polynomial *cannot* be computed efficiently?

2. Can polynomials help us *count*? Are such polynomials efficiently computable?

3. What structure can be inferred about *factors* of polynomials that can be computed efficiently?

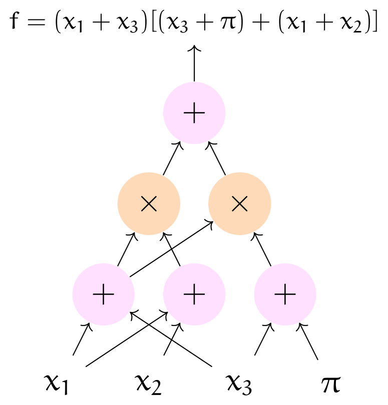


Figure 1.1: Example of an algebraic circuit

In order to study questions of algebraic computation and develop the algebraic analog of NP-completeness, Valiant [Val79a; Val82] introduced *arithmetic/algebraic circuits* (Definition 2.2.1), a natural model that has led to the development of a rich and varied theory of algebraic complexity (see [Bür24]).

An algebraic circuit is essentially a Boolean circuit where the logical operations AND and OR are replaced by addition and multiplication (Figure 1.1). It computes a polynomial bottom up inductively with variables and field elements at the leaves. If the underlying directed acyclic graph is a *tree*, we call the circuit a *formula*. The *size* of the circuit in Figure 1.1 is 6, given by the number of vertices/gates in the graph. Since the length of the longest path from the root to any leaf is 3, that is the *depth* of the circuit. Many results are better stated in terms of the *product-depth* (1 in our example) which counts the maximum number of multiplication gates along any root–leaf path.

In Boolean complexity, the notion of efficient computation is captured by the *complexity class* P of problems solvable in polynomial time. A complexity class in the algebraic world is made of families/sequences of polynomials $(f_n)_{n \in \mathbb{N}}$ where f_n is a multivariate polynomial over some field $\mathbb{F}$. We will mostly be interested in families where the number of variables in f_n grows as a polynomial function of n. The algebraic analog of P is called VP (Definition 2.2.15) and consists of all polynomial families where f_n is of $\mathrm{poly}(n)$-degree and the smallest circuit computing f_n has size $\mathrm{poly}(n)$. The underlying field of computation $\mathbb{F}$ is usually fixed beforehand, and we omit it from the notation. The notion of computation is *non-uniform* – the circuits of f_n for different n need not be related to one another. A prime example of a polynomial family in VP is (Det_n), defined by the *determinant* of the $n \times n$ symbolic matrix $(x_{ij})_{1\leq i,j\leq n}$:

$$\mathrm{Det}_n = \sum_{\sigma\in S_n} \left(\mathrm{sgn}(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)} \right).$$

The algebraic analog of the class NP is called VNP (Definition 2.2.18). Informally, it consists of polynomial families which are 'explicit', in the sense that given a monomial of f_n, we can compute the corresponding coefficient efficiently, say in polynomial time. Similar to the Boolean setting, it is not hard to show that a 'random' polynomial cannot be computed by small circuits [HY11]. The long-standing conjecture of Valiant [Val79a] is that there are *explicit* polynomial families that cannot be computed efficiently, i.e., $\mathrm{VP} \subsetneq \mathrm{VNP}$. A prominent candidate for this separation is the family of *permanents*,

$$\mathrm{Per}_n = \sum_{\sigma\in S_n} \left(\prod_{i=1}^{n} x_{i,\sigma(i)} \right).$$

The determinant and permanent families essentially characterize the classes VP and VNP, respectively. Hence, Valiant's conjecture is also sometimes called the Permanent versus Determinant problem [Agr06]. It is the algebraic version of Cook's hypothesis [Coo71] – also known as the P vs. NP problem. There is a formal sense in which the VP vs. VNP problem is a 'stepping stone' towards the P vs. NP problem [Bür00b]. For details on the connection between Valiant's and Cook's hypotheses, we encourage readers to consult [BCS97; Bür99; Bür00a]. Though the best known lower bounds of $\Omega(n \log n)$ for general arithmetic circuits by Baur and Strassen [BS83] and $\Omega(n^2)$ for formulas by Kalorkoti [Kal85a] fall far short of the superpolynomial lower bounds that we hope to prove, such bounds are known for various *restricted* classes [NW96; Raz06; Raz09] (see [SY09; CKW10; Mah14; Sap21] for excellent survey).

1.1 Low depth circuits

One of the most interesting restrictions is that of bounding the *depth* of circuits and formulas. When the depth is a constant, circuits and formulas are equivalent up to polynomial blow up in their size and hence we use them interchangeably. Unlike the Boolean world, a very curious phenomenon of *depth reduction* occurs in arithmetic circuits [Val+83; AV08; Koi12; Tav15; Gup+16]. Essentially, circuits of depth 3 and 4 are almost as powerful as general ones. More formally, any degree d polynomial f that has a size s circuit can also be computed by a depth 4 *homogeneous* circuit or a depth 3 (possibly non-homogeneous) circuit of size $s^{O(\sqrt{d})}$. Hence, proving an $n^{\omega(\sqrt{d})}$ lower bound against these special circuits is enough to separate VP from VNP. The extreme importance of bounded depth circuits has led to a large body of work proving lower bounds for these models and their variants [SS96; SW01; Raz10; Kay12; Gup+14; KSS14; Fou+15; KS15; KST16; Kay+17; KST18; GST20].

1.1.1 The LST breakthrough

In a remarkable recent work, Limaye, Srinivasan and Tavenas [LST21] proved the first superpolynomial lower bound for general constant-depth circuits. More precisely, they showed that there is a polynomial $P_{n,d}$ on $O(dn^2)$ variables of degree $d = o(\log n)$ which has no product-depth Δ circuits of size $n^{d^{\exp(-O(\Delta))}}$.

The variable set X is partitioned into d sets $(X_1, \ldots, X_d)$ of n^2 variables each (viewed as $n \times n$ matrices). The polynomial $P_{n,d}$ is the *Iterated Matrix Multiplication* polynomial, obtained by picking a particular (say the $(1,1)$-th) entry of the matrix product $X_1 \cdot X_2 \cdots X_d$:

$$\mathrm{IMM}_{n,d} = \left(\begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{1,n^2-n+1} & \dots & x_{1,n^2} \end{bmatrix} \cdots \cdots \begin{bmatrix} x_{d,1} & \dots & x_{d,n} \\ \vdots & \ddots & \vdots \\ x_{d,n^2-n+1} & \dots & x_{d,n^2} \end{bmatrix} \right)_{(1,1)} .$$

As all monomials are of the same degree d, the polynomial is *homogeneous*. It is also *multilinear* since every variable has individual degree at most 1. Additionally, every monomial has exactly one variable from each of the d sets of the partition. Such polynomials are called *set-multilinear*. An algebraic circuit is said to be *set-multilinear* if every node in the circuit computes a set-multilinear polynomial with respect to (a subset of) the variable sets.

For any $\Delta \leq \log d$, $\mathrm{IMM}_{n,d}$ has a set-multilinear circuit of product-depth Δ and size $n^{O(d^{1/\Delta})}$, obtained via basic recursion. No significantly better upper bounds are known, even if we allow general circuits. It makes sense to conjecture that this upper bound is tight (see [CLS19] for limitations to improving the upper bound when the matrices X_i are 2×2). The lower bound of [LST21] proceeds by first transforming size s, product-depth Δ, general circuits computing a set-multilinear polynomial of degree d to *set-multilinear* algebraic circuits of product-depth 2Δ and size $\mathrm{poly}(s)d^{O(d)}$ (which is not huge if d is small). Hence, in the low-degree regime, lower bounds on bounded depth set-multilinear circuits translate to bounded depth general circuit lower bounds, albeit with some loss[1]. Finally, they use a *measure* which is high for $\mathrm{IMM}_{n,d}$ and which they show is low for small-size, bounded-depth set-multilinear circuits.

1.1.2 Our contribution: improved lower bound

In Chapter 3, we investigate the limits of the techniques introduced by Limaye, Srinivasan, and Tavenas. Our first result (Theorem 3.0.1) is an improvement of

[1]Interestingly, set-multilinear lower bounds against arbitrary depth were known before [NW96; Raz09; RY09], but degenerated to trivial bounds when the degree was small.

their lower bound.

> We show that in the low-degree regime (when $d = O(\log n / \log \log n)$), any circuit of product-depth Δ for $\mathrm{IMM}_{n,d}$ must be of size at least $n^{\Omega\left(d^{1/(\varphi^2)^\Delta}\right)}$, where $\varphi = 1.618\ldots$ is the golden ratio.

This is an improvement over the $n^{\Omega\left(d^{1/4^\Delta}\right)}$ lower bound previously shown, since $\varphi^2 < 4$. Our lower bound, much like that of [LST21], proceeds by transforming a general bounded-depth circuit to a set-multilinear one without increasing the depth by much, or blowing up the size (when the degree is low). This transformation originally required fields of characteristic zero. In a very nice work, Forbes [For24] removed this restriction. Hence, the lower bound now holds over all fields.

A crucial idea of the [LST21] result was to use set-multilinear polynomials where each set in the variables' underlying partition are of different sizes and using this discrepancy. By picking the set sizes more carefully (depending on the depth we are working with), we improve their $n^{\Omega\left(d^{1/2^\Delta}\right)}$ set-multilinear lower bound.

> We show that any product-depth Δ *set-multilinear* circuit for $\mathrm{IMM}_{n,d}$ (when $d = O(\log n)$) needs size at least $n^{\Omega\left(d^{1/\varphi^\Delta}\right)}$.

In a further recent work, Tavenas, Limaye and Srinivasan [TLS22] proved a product-depth Δ *set-multilinear* formula lower bound of $(\log n)^{\Omega(\Delta d^{1/\Delta})}$ for $\mathrm{IMM}_{n,d}$. There is no degree restriction, but in the small degree regime, the bound is much weaker than [LST21] and cannot be used for escalation. Improving on it, Kush and Saraf [KS22] showed a lower bound of $n^{\Omega(n^{1/\Delta}/\Delta)}$ for the size of product-depth Δ set-multilinear formulas computing an n^2-variate, degree n polynomial family in VNP from the Nisan-Wigderson design-based polynomial family. Later [KS23], they showed the same bound for polynomial families in VP (and even lower, though not $\mathrm{IMM}_{n,d}$). Unlike all these works, we are interested in the low-degree regime where set-multilinear lower bounds can be lifted, making these works incomparable to ours.

1.1.3 Our contribution: proof barrier

We also show that these techniques cannot improve our lower bound significantly. Since we could use two different (but dintict from [LST21]) set sizes and improve the lower bound, it might seem plausible that using many more set sizes (out of a total possible d) could improve the bound further. We show that this is false for most cases (Theorem 3.4.1).

> For any choice of $O(1)$ set sizes, we show that there is a set-multilinear product-depth Δ circuit with size matching our lower bound. When the number of sets is $d^{o(1)}$, there exists a different set-multilinear circuit that has product-depth Δ and size *almost* matching our lower bound. In both cases, the value of the measure used to prove the lower bound is maximized.

This results in a barrier to further improvement since this particular measure cannot prove a better lower bound, as there are small (matching our lower bound) circuits that maximize the measure. Such a barrier was known for the two *specific* variable set sizes considered in [LST21]. In a further work (that was almost parallel to ours), Limaye, Srinivasan and Tavenas also showed similar barrier results [LST22]. They simplified the proof framework of [LST21] and characterized the lower bounds that can be proved via this technique using a combinatorial property, which they termed *Tree Bias*. Their result works for any d set sizes, but the size upper bound on the measure-maximizing circuits they obtain is weaker. These barrier results suggest that new measures and techniques might be necessary if we are to prove significantly better constant-depth lower bounds.

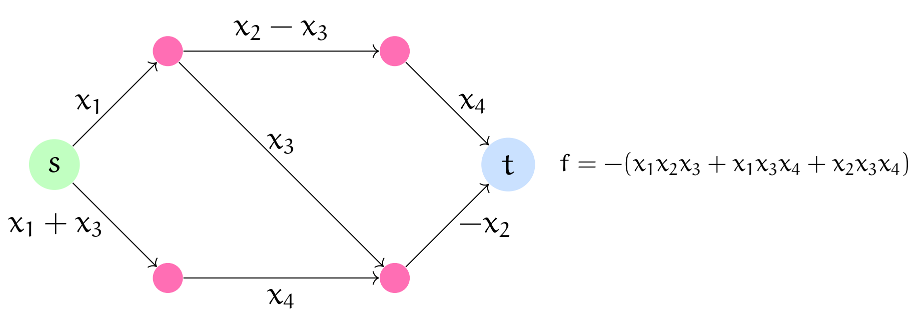


Figure 1.2: Example of an ABP

1.2 Algebraic branching programs

Intermediate in power, and in between circuits and formulas lie Algebraic Branching Programs (Definition 2.2.5). The computation of an ABP is slightly different, and we explain it with an example (Figure 1.2). Each of the three paths from a designated *source* vertex s to the *sink* t computes a polynomial that is the product of the edge labels in that path: $x_1(x_2-x_3)x_4$, $x_1x_3(-x_2)$, and $(x_1+x_3)x_4(-x_2)$. Edges go from one *layer* to the next and are labeled by linear polynomials. Summing up the polynomials computed by all $s \rightsquigarrow t$ paths gives the polynomial computed by the ABP. The *size* of the ABP is 6, the total number of vertices in the graph. The *width* of the ABP is 2, the maximum number of vertices in a layer. We can also view the computation of the ABP in Figure 1.2 as a product of matrices with entries coming from the edge labels:

$$f = \begin{bmatrix} x_1 & x_1 + x_3 \end{bmatrix} \begin{bmatrix} x_2 - x_3 & x_3 \\ 0 & x_4 \end{bmatrix} \begin{bmatrix} x_4 \\ -x_2 \end{bmatrix}.$$

Similar to circuits and formulas, the families of polynomials (f_n) that have $\mathrm{poly}(n)$-sized ABPs make up the class VBP (Definition 2.2.17). It is known that VF $\subseteq$ VBP $\subseteq$ VP, and conjectured that all the inclusions are strict. Valiant's hypothesis is considered more generally as the problem of separating any of the classes VF, VBP or VP from VNP. Unfortunately (although probably not surprisingly), general lower bounds in any of these models is hard to come by. In a recent

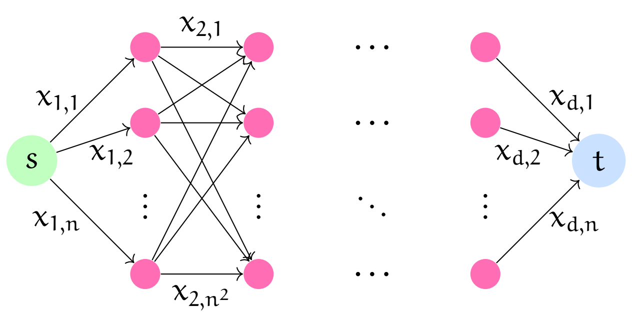


Figure 1.3: An $O(nd)$ sized ABP for $\text{IMM}_{n,d}$

work, Chatterjee, Kumar, She and Volk [Cha+22] proved a lower bound of $\Omega(n^2)$ for ABPs. Evidently, the state of affairs is quite similar to that of circuits. In fact, the polynomial $\sum_{i=1}^{n} x_i^n$ used in the ABP lower bound is the same one that Baur and Strassen [BS83] used for their circuit lower bound.

1.2.1 Our contribution: lower bounds for the sum of small ABPs

In Chapter 4, we prove a lower bound against the sum of general *sub-polynomial* sized algebraic branching programs (Theorem 4.2.1).

> We show that $\text{IMM}_{n,d}$ *cannot* be computed by the sum of $\text{poly}(n, d)$ ABPs, where each ABP is of size $(nd)^{o(1)}$.

Note that naive ABP for $\text{IMM}_{n,d}$ is of size $O(nd)$ (Figure 1.3). Our result shows that this is almost optimal: we cannot reduce the size significantly, even by using a sum of polynomially many ABPs. When $d = n^{\Omega(1)}$ is large, ABPs of size $(nd)^{o(1)}$ cannot even produce monomials of degree d. Hence, the lower bound is obtained trivially (in general, a lower bound of d is trivial for ABPs). But when $d = n^{o(1)}$ is small, the model is quite powerful. In fact, for $d = n^{o(1)}$, the power sum polynomial $\sum_{i=1}^{n} x_i^d$, that was used in previous ABP lower bounds *can* be computed efficiently in our model: as a sum of n ABPs, each of size $(nd)^{o(1)}$.

Remark 1.2.1. A lower bound of n is *not* trivial for ABPs (unlike circuits and formulas). Moreover, each edge label can be a general affine linear form, allowing a single path to generate exponentially many monomials. Notwithstanding that, ABPs of size $(nd)^{o(1)}$ are still an incomplete model of computation. Nevertheless, the *sum* of such ABPs *is* a complete model – every polynomial of degree $d = n^{o(1)}$ can be written as a sum of $n^{O(d)}$ width-1 ABPs (monomials). Our lower bound also holds if we replace $\mathsf{IMM}_{n,d}$ with an appropriate polynomial from the family of Nisan-Wigderson design-based polynomials (see Section 4.2.1).

1.2.2 Our contribution: hardness bootstrapping for ABPs

Our next result (Theorem 4.1.1) is a reformulation of Valiant's conjecture in terms of a different model: the sum of *set-multilinear* ABPs (smABPs) on the set of variables $X = X_1 \sqcup \ldots \sqcup X_d$.

In an (ordered) smABP, edge labels from a layer to the next are linear forms in a single set X_i from the partition, and all the sets occur (once) in some order (called the order of the smABP). The most natural ABP for $\mathsf{IMM}_{n,d}$ is also set-multilinear: each layer (other than the first and the last) has n nodes and the edge connecting the p-th node in layer i to the q-th node in layer $i+1$ is labeled by $x_{i,(p-1)n+q}$ (Figure 1.3). We denote by $\sum \mathsf{smABP}$ the sum of set-multilinear ABPs, each in a possibly different order. The *width* of a $\sum \mathsf{smABP}$ is the sum of the widths of the constituent smABPs.

> We show that a superpolynomial lower bound against $\sum \mathsf{smABP}$ for a set-multilinear polynomial of very low degree $d = O(\log n / \log \log n)$ is enough to separate VNP from VBP.

The above result shows that the sum of set-multilinear ABPs, which looks quite restrictive, is surprisingly powerful. This is a recurring theme in algebraic complexity, as already seen in the context of depth-reduction. Interestingly, analogous reductions to the set-multilinear case were known for formulas [Raz13, Theorem 3.1] and circuits [NW96, Lemma 2.11]. The above result is in a similar vein. It also shows that the low-degree regime that was useful in proving lower bounds for constant-depth circuits [LST21] is also useful for ABP lower bounds. The model of $\sum \mathsf{smABP}$ is particularly appealing to study since a single smABP is one of the most well-understood objects in algebraic complexity. If the matrices defining the smABP were commutative, we can treat $\sum \mathsf{smABP}$ as a *single* smABP, against which we know how to prove exponential lower bounds (Section 1.2.3). So in order to lift the lower bound to VNP, it is essential that we understand the sum of smABPs with non-commuting matrices (see Section 1.2.4 for a detailed discussion).

Recently, Kush and Saraf [KS23] proved near-optimal lower bounds against set-multilinear formulas for a polynomial family in VBP. Surprisingly, if their hard polynomial were computable by an (ordered) smABP, we would obtain *general* formula lower bounds! This further illustrates the need to study smABPs.

Remark 1.2.2. In Section 4.1, we also give a smooth generalization of the above result using more general versions of both set-multilinear polynomials and smABPs that are called *set-multi-*k*-ic* (Definition 2.2.8).

1.2.3 The sum of ROABPs perspective

One can also view the above bootstrapping result through the lens of another well-studied model in the literature, first defined by Forbes and Shpilka [FS13]. In a Read-once Oblivious Algebraic Branching Program (ROABP) (Definition 2.2.9) over the variables $x_1, \dots, x_n$, the edge labels from a layer to the next are *univariate polynomials* in some variable x_i, and the variables occur (once) in some order (called the order of the ROABP).

The computation that an ROABP performs is essentially non-commutative since the variables along a path get multiplied in the same sequence as the order of the ROABP. Nisan [Nis91] introduced the powerful technique of using spaces of partial derivatives to study lower bound questions in non-commutative models. This technique can be used to calculate the exact width of the ROABP computing a polynomial. Following our notation for smABPs, we denote by $\sum \mathsf{RO}$ the sum of ROABPs, each possibly in a different order. The width of a $\sum \mathsf{RO}$ is the sum of the widths of the constituent ROABPs. In fact, ROABPs and smABPs are essentially the same (up to an invertible map), and a version of the reduction result to $\sum \mathsf{smABP}$ model can also be stated for $\sum \mathsf{RO}$ (Corollary 4.1.3). In Section 4.1, we also show a generalization for the read-k versions of ROABP (Definition 2.2.10). In contrast to the case of smABPs, we are interested in the dual *low-variate* $n = O(\log d/\log\log d)$ regime.

The low-variate regime[2] has also recently been shown to be extremely important. The Polynomial Identity Testing (PIT) problem asks to efficiently test whether a polynomial (given as an algebraic circuit, for example) is identically *zero*. In the black-box setting, we are only allowed to evaluate the polynomial (circuit) at various points. Hence, PIT algorithms are equivalent to the construction of *hitting sets* – a collection of points that witness the (non)zeroness of the polynomial computed by the circuit (see [Sax09; Sax14; DG24] for a survey of PIT and techniques used). Several surprising results [AGS19; KST23; Guo+22] essentially conclude that hitting sets for circuits computing extremely low-variate polynomials can be "bootstrapped" to obtain hitting sets for general circuits. The survey of Kumar and Saptharishi [KS19] gives a lucid exposition of the ideas involved.

1.2.4 Proof techniques and previous work

Our proof of the first result on $\sum \mathsf{ABP}$ lower bound uses the implicit reduction from an ABP to the $\sum \mathsf{smABP}$ model as shown in our second result. We show how to simulate an ABP using a sum of $d^{O(d)}$ smABPs, which is not large when the degree $d = O(\log n/\log\log n)$ is small. Nisan's characterization [Nis91] can be used to prove exponential lower bounds against single smABPs (ROABPs), but it does not extend to their sums.

There has been some exciting progress in handling the sums in recent years. Arvind and Raja [AR16] proved a superpolynomial lower bound for the Per-

[2] In general for VNP, we require the degree to be polynomially related to the number of variables. One way to make sense of the low variate case is to assume that the polynomial is defined on d variables but only depends on the first $n = O(\log d/\log\log d)$ many of them.

manent polynomial against the sum of sub-linear many ROABPs (the bound is exponential if the number of ROABPs is bounded by a constant). Ramya and Rao [RR20] showed that a sum of sub-exponential size ROABPs computing the multilinear polynomial defined by Raz and Yehudayoff [RY08] needs exponentially many summands. Ghosal and Rao [GR21] showed an exponential lower bound for the sum of ROABPs computing the multilinear polynomial defined by Dvir, Malod, Perifel and Yehudayoff [Dvi+12], provided each of the constituent ABPs is polynomial in size.

Unfortunately, these results do not imply general ABP lower bounds using our hardness escalation theorems, as they only work in regimes where the degree and number of variables are comparable. Viewed differently, they cannot handle a sum of $d!$ smABPs (or $n!$ ROABPs) which is necessary to prove lower bounds in our low-degree (or low-variate) regime. In a very recent work Chatterjee, Kush, Saraf and Shpilka [Cha+24] improve the bounds in the above works and also prove superpolynomial lower bounds against the sum of smABPs when the degree is $d = \omega(\log n)$. Improving this to work for $d = O(\log n / \log\log n)$ would have dramatic consequences.

We demonstrate a way to handle our low-degree regime in certain cases. To prove lower bounds for the sum of smABPs, we use the *partial derivative method*, introduced in the highly influential work of Nisan and Wigderson [NW96]. We show that the partial derivative measure $\mu(\cdot)$ is large for our hard polynomial but small for the model. In fact, a majority of the lower bounds in algebraic complexity (including the results described earlier) use modifications and extensions of this measure [CKW10; Sap21].

We work with the polynomial $\mathrm{IMM}_{n,d}$, which gives us more flexibility in independently choosing n and d. Unfortunately, this choice creates a two-fold problem. The fundamental one is that $\mathrm{IMM}_{n,d}$ has a small smABP, as we saw before. So we can never prove a superpolynomial lower bound for even a single $\mathrm{poly}(n, d)$ sized smABP (let alone their sum). One might try to avoid this by choosing a different hard polynomial that gives similar flexibility, perhaps something from the family of Nisan-Wigderson design-based polynomials. But in fact, the complexity measure μ is also *maximal* for $\mathrm{IMM}_{n,d}$. Hence, the usual partial derivative method cannot be used to prove lower bounds against any model that efficiently computes $\mathrm{IMM}_{n,d}$. Be that as it may, it might still be possible to use the same technique to prove lower bounds for restrictions of the model. We are able to do this when the smABPs are sub-polynomial in size. It also enables us to handle extremely large sums of smABPs (including those that occur from considering sums of multiple ABPs). This approach works in the low-degree regime, since our reductions are efficient if the degree is very small. To handle higher degrees, we note that $\mathrm{IMM}_{n,d'}$ with d' small can be obtained as a set-multilinear restriction of $\mathrm{IMM}_{n,d}$. Therefore, our lower bounds translate to higher degrees.

1.3 Counting Homomorphisms

Counting and deciding the existence of patterns in graphs play an important role in computer science. In theoretical computer science, pattern counting was among the first problems to be investigated in Valiant's seminal paper on the class #P of counting problems [Val79b], which showed #P-hardness for the permanent of zero-one matrices, a problem that can equivalently be viewed as counting perfect matchings in bipartite graphs.

1.3.1 Counting small patterns

In many applications, the pattern is smaller in comparison to the target graph. Curticapean and Marx [CM14] modeled this setting by classifying the complexity of counting subgraphs from fixed pattern classes: Given any fixed class of graphs $\mathcal{H}$, they defined a problem $\#\mathrm{Sub}(\mathcal{H})$ that asks, given a graph $H \in \mathcal{H}$ and a general graph G, to count the H-subgraphs in G. The parameter is $|V(H)|$. This problem is known to be polynomial-time solvable when the graphs in $\mathcal{H}$ do not contain arbitrarily large matchings—if they do contain arbitrarily large matchings, then $\#\mathrm{Sub}(\mathcal{H})$ with parameter $|V(H)|$ is complete for the parameterized complexity class $\#W[1]$, i.e., the analogue of #P in parameterized complexity.

An analogously defined problem $\#\mathrm{Hom}(\mathcal{H})$ of counting *homomorphisms* with patterns drawn from $\mathcal{H}$ was also classified by Dalmau and Jonsson [DJ04]. Here, the tractability criterion is a constant bound on the *treewidth* (Definition 2.3.1) of graphs in $\mathcal{H}$, a measure of the "tree-likeness" of H: The problem $\#\mathrm{Hom}(\mathcal{H})$ is polynomial-time solvable when all graphs in $\mathcal{H}$ admit a constant upper bound on their treewidth, and the problem is $\#W[1]$-hard otherwise with respect to the parameter $|V(H)|$. Here, a homomorphism from H to G is a function $h : V(H) \to V(G)$ such that $uv \in E(H)$ implies $h(u)h(v) \in E(G)$. Homomorphism counts from small patterns find direct applications in database theory, where they capture answer counts to so-called conjunctive queries [CM16]. It was also shown that $\#\mathrm{Hom}(\mathcal{H})$ captures the complexity of other pattern counting problems, including that of counting subgraphs [CDM17]: In a nutshell, (i) many pattern counting problems can be expressed as unique linear combinations of homomorphism counts from graphs H, and (ii) in many models of computation, such linear combinations turn out to be precisely as hard as their hardest terms. This strongly motivates understanding the complexity of these individual terms, i.e., of homomorphism counts.

Following the classification of $\#\mathrm{Sub}(\mathcal{H})$ and $\#\mathrm{Hom}(\mathcal{H})$ under parameterized complexity assumptions, almost-tight quantitative bounds were obtained under the exponential-time hypothesis [IP01; LMS11]: For any graph H, there is an $O(n^{\mathrm{tw}(H)+1})$ time algorithm for counting homomorphisms from H into n-vertex target graphs, and assuming the exponential-time hypothesis, Marx ruled out $n^{o(\mathrm{tw}(H)/\log \mathrm{tw}(H))}$ time algorithms [Mar10], recently revisited in [Kar+24; CM14]. Through connections between homomorphism counts and other pattern counts [CDM17], these bounds translate directly to other counting problems. For example, there is an $O(n^{\mathrm{vc}(H)})$ time algorithm for counting H-subgraphs in an n-

vertex graph G, where $vc(H)$ is the vertex-cover number of H. As a consequence of the lower bound on counting homomorphisms, the exponential-time hypothesis rules out $n^{o(vc(H)/\log vc(H))}$ time algorithms [CDM17].

Thus, some slack remains between the known upper and conditional lower bounds on the exponents of pattern counting problems, even asymptotically: It would be desirable to settle the log-factor in the exponent of the running time. Moreover, one might also dare to ask for the precise exponent for concrete finite graphs H, such as K_3 (which amounts to triangle counting) or K_4 or C_6. Finally, let us stress that the lower bounds on the exponent are conditioned on the exponential-time hypothesis, an assumption that is *a priori* stronger than $P \neq NP$.

1.3.2 From counting problems to polynomials

Valiant's seminal papers [Val79a; Val80] studied the problem of counting perfect matchings both via counting and algebraic complexity. Following the work of Komarath, Pandey, and Rahul [KPR23], we consider an algebraic version of pattern counting problems. For undirected graphs H and $n \in \mathbb{N}$, we consider the homomorphism polynomial $\mathrm{Hom}_{H,n}$ on variables $x_{i,j}$ for $i, j \in [n]$ and its set-multilinear version, the colorful subgraph polynomial $\mathrm{ColSub}_{H,n}$ on variables $x_{i,j}^{(e)}$ for $i, j \in [n]$ and $e \in E(H)$. The latter can often be handled more easily in proofs, while complexity results can be transferred between these two polynomials.[3]

The polynomial $\mathrm{Hom}_{H,n}$ can be viewed as the weighted homomorphism count from H into a complete n-vertex graph with generic indeterminates as edge-weights. Similarly, $\mathrm{ColSub}_{H,n}$ can be viewed as counting the color-preserving homomorphism count from a colorful graph H into a complete graph with indeterminate edge-weights and n vertices per color class. Formally, these multivariate

[3]Here, we deviate slightly from the notation used by Komarath, Pandey, and Rahul [KPR23], who defined a polynomial $\mathsf{ColIso}_{\mathcal{H},n}$ with variable indices that differ from ours. Our polynomial $\mathrm{ColSub}_{H,n}$ and their polynomial $\mathsf{ColIso}_{\mathcal{H},n}$ can be obtained from each other by renaming variables. We consider our notation more intuitive, as it can be obtained from the homomorphism polynomial more directly and also highlights the set-multilinearity of the polynomial.

polynomials are defined as

$$\mathsf{Hom}_{H,n} = \sum_{f:V(H)\to[n]} \prod_{uv\in E(H)} x_{f(u),f(v)},$$
$$\mathsf{ColSub}_{H,n} = \sum_{f:V(H)\to[n]} \prod_{uv\in E(H)} x^{(uv)}_{f(u),f(v)}.$$

In a recent line of work [CLV21; Dur+16; MS18], homomorphism polynomials have been used to obtain natural polynomials which are complete for several well-studied algebraic circuit classes.

Monotone circuits over a field like $\mathbb{Q}$ are circuits that do not use negative constants, and hence computations performed by them cannot feature cancellations (Definition 2.2.11). Several important techniques for proving upper bounds on the complexity of polynomials (e.g., dynamic programming) directly yield monotone circuits. Compared to general computational models, lower bounds for monotone computation are much better understood, and many exponential lower bounds [Sch76; Val80; JS82; RY11; GS12; CKR22] and strong algebraic complexity class separations [Sni80; HY16; Yeh19; Sri20] are known. As a striking example, monotone variants of the algebraic complexity classes VP and VNP are proven to be different [Sri20; Yeh19].

In a fascinating recent work, Komarath, Pandey and Rahul [KPR23] studied the monotone arithmetic circuit complexity of the polynomials $\mathsf{Hom}_{H,n}$ and discovered that this complexity is completely determined by the treewidth of the pattern graph H. More precisely, they show that the smallest monotone circuit computing $\mathsf{Hom}_{H,n}$ is of size $\Theta(n^{tw(H)+1})$. Similarly, they show that algebraic branching programs for $\mathsf{Hom}_{H,n}$ are of size $\Theta(n^{pw(H)+1})$, where $pw(H)$ is the *pathwidth* of H, a linear version of treewidth. Moreover, they also consider the monotone formula complexity of $\mathsf{Hom}_{H,n}$ and show that it is $\Theta(n^{td(H)+1})$, where $td(H)$ is the *treedepth* of H, the minimum height of a tree on vertex set $V(H)$ that contains all edges of H in its tree-order. These results together show that, when considering homomorphism polynomials for fixed patterns H, the power of monotone computation is precisely characterized by graph-theoretic quantities of H: For natural and well-studied monotone computational models, the precise exponent c_H in the complexity $\Theta(n^{c_H})$ is the value of a natural and well-studied graph parameter of H.

1.3.3 Our contribution: monotone bounded-depth complexity

In Chapter 5, we investigate whether the correspondence between monotone circuit complexity and graph parameters can also be established for bounded-depth monotone circuits: Are there natural graph parameters that dictate the *bounded-depth* monotone complexity of $\mathrm{Hom}_{H,n}$? Studying this question naturally leads us to define bounded-depth versions of treewidth, the Δ-*treewidth* $tw_\Delta(H)$ for any fixed $\Delta \in \mathbb{N}$. These graph parameters ask to minimize the maximum bag size over all tree-decompositions of H, however with the twist that only tree-decompositions with an underlying tree of height at most Δ are admissible. Their values interpolate between $|V(H)| - 1$ (when only height 1 is allowed) and $tw(H)$ (when no height restrictions are imposed), and they are connected to the vertex-cover number in the special case $\Delta = 2$ (see Section 5.1).

Bounded-depth variants of treewidth implicitly appear in balancing techniques for tree-decompositions [CIP16; BH98], and the Δ-treewidth of paths also appears implicitly in divide-and-conquer schemes for iterated matrix multiplication in a bounded-depth setting [LST21]. A recent work of Adler and Fluck [AF24] studied a notion that bounds the width and depth simultaneously, which they call bounded depth treewidth. Our notion of $tw_\Delta(H)$ only bounds the height of the tree decomposition to Δ. In particular, for a fixed Δ, there is always a tree decomposition of height Δ for any graph H, but with a possibly large treewidth.

We show that the Δ-treewidth of graphs completely characterizes the complexity of $\mathrm{Hom}_{H,n}$ for monotone circuits of product-depth at most Δ (Theorem 5.1.7). For technical reasons described later, it is however not the Δ-treewidth of H itself that governs the complexity, but rather the Δ-treewidth of the graph $H^\dagger$ obtained by removing all vertices of degree at most 1. We call this the *pruned* Δ-treewidth ptw_Δ of a graph H.

> Our main result is that the monotone, product-depth Δ circuit complexity of $\mathrm{Hom}_{H,n}$ and $\mathrm{ColSub}_{H,n}$ is $\Theta(n^{\mathrm{ptw}_\Delta(H)+1})$.

Note that any fixed pattern graph H on k vertices gives a homomorphism polynomial on n^k monomials, which has a trivial $\mathrm{poly}(n)$ sized monotone circuit of depth two. We stress that we wish to determine the precise *exponent* in this polynomial size: In general, k could be much larger than the pruned Δ-treewidth of H.

For a graph H, we also define a new graph parameter called Δ-*pathwidth* $\mathrm{pw}_\Delta(H)$ which asks to minimize the bag size over all *path decompositions* of H where the underlying path is of *length* Δ. We defer the formal definitions of these graph parameters to Section 5.1. Similar to the case of circuits, we show that the Δ-pathwidth of the pruned graph $H^\dagger$ (obtained by removing degree 1 vertices from H), which we call the *pruned* Δ-pathwidth ppw_Δ of H characterizes the monotone ABP of length Δ computing $\mathrm{Hom}_{H,n}$.

For a fixed pattern graph, it was shown in [KPR23] that $\mathrm{Hom}_{H,n}$ and $\mathrm{ColSub}_{H,n}$ have the 'same' monotone complexity. We observe that the reduction holds even in the bounded-depth (bounded-length) case (Lemma 5.3.1). So, we prove our results only for $\mathrm{ColSub}_{H,n}$. The upper bounds are shown in Section 5.2, while the lower bounds are shown in Section 5.3.

1.3.4 Our contribution: monotone depth hierarchy

Finally, by turning our attention to pattern graphs of non-constant size, we can prove a depth hierarchy theorem for monotone circuits (Theorem 5.4.2).

> Using our tight characterization results and the properties of pruned Δ-treewidth, we are able to obtain a polynomial family (f_n) where f_n can be computed by a product-depth $(\Delta + 1)$ circuit of size $\mathrm{poly}(n)$,

but any monotone circuit of product-depth Δ computing f_n must be of size $n^{\Omega(n^{1/\Delta})}$.

By general depth-reduction results, any monotone circuit of size $\mathrm{poly}(n)$ computing a polynomial of degree n can be flattened to a monotone circuit of product-depth Δ and size $n^{O(n^{1/\Delta})}$, showing that our theorem is optimal. We also note that a similar *near-optimal* statement with a lower bound of $\exp(n^{\Omega(1/\Delta)})$ can be obtained from earlier results provided the product-depth $\Delta = o(\log n/\log\log n)$ is small (see [Chi+18]). Our results improve upon this in two ways: Firstly, our Ω appears in the first rather than second exponent, thus yielding a stronger and optimal lower bound. Secondly, our results hold for any product-depth Δ.

1.4 Polynomial Factorization

The problem of finding a nontrivial factor of a polynomial is a classical and fundamental one, with a rich history spanning centuries [Gat06]. Comprehensive treatments of the problem can be found in [Kal82; Kal90; Kal92; GP01; Sho09; GG13]. Erich Kaltofen, in a series of influential papers in the 80's [Kal85b; Kal86; Kal87; Kal89], showed that over fields of characteristic zero, the class VP is closed under taking factors. This was a testament to the robustness and naturalness of the algebraic circuit model. The attentive reader might have noticed that the families of polynomials we considered till now had relatively low degree, bounded by a polynomial in the number of variables. This was true for VP, VNP and all the other classes we encountered, and there are good reasons for doing so [Gro13]. However, a small size circuit can generate polynomials of high degree by repeated squaring. It is natural to ask if small high-degree circuits are also robust with respect to factoring.

1.4.1 High degree circuits

In the context of algebraic complexity, Malod [Mal03; Mal07] first considered the class VP_{nb} that consists of all polynomial families where the smallest circuit computing f_n is of size $\mathrm{poly}(n)$ (Definition 2.2.21). Note that the degree of a polynomial family in VP_{nb} can be exponential in the size of the circuit. The results of Kaltofen [Kal87] mentioned before actually shows that for an n-variate polynomial $f = g^e h$ where g and h are coprime and g has multiplicity e,

$$\mathrm{size}(g) = \mathrm{poly}(\mathrm{size}(f), \deg(g), e, n). \tag{1.4.1}$$

Hence, the best size upper bound one can deduce for (exponential–degree) factors of polynomials in VP_{nb} from Kaltofen's result is exponential. This is unavoidable in general: the polynomial $x^{2^n} - 1 = \prod_{i=1}^{2^n}(x - \xi^i)$ where ξ is a 2^n–th root of unity, can be computed by a circuit of size $O(n)$. Lipton and Stockmeyer [LS78] showed that a random exponential–degree factor $\prod_{i\in S}(x - \omega^i)$ where $S \subset [2^n]$ and $|S| = \exp(n)$ requires $\exp(n)$–size circuits. So VP_{nb} is *not* closed under taking factors. Nevertheless, Bürgisser's *Factor Conjecture* [Bür00a, Conj. 8.3] is that the size of the factor g in Equation (1.4.1) should be independent of its multiplicity e. In particular, $\mathrm{poly}(n)$–degree factors of a family in VP_{nb} should be in VP. The conjecture also has applications to decision complexity [Bür04, Section 4].

Over the years, there has been some partial progress on the problem. In the case when f is a power of g, i.e. $f = g^e$, Kaltofen [Kal87] (cf. [Bür04, Proposition 6.1]) already showed that $\mathrm{size}(g)$ is independent of e. As another special case, Dutta, Saxena and Sinhababu [DSS22] showed that the conjecture also holds if the *squarefree* part of f is of low degree. Suppose $f = \prod_{i=1}^m f_i^{e_i}$ is the complete factorization of f with f_i's irreducible. Then, they showed that the size of any factor g of f is polynomially bounded in $\mathrm{size}(f), n$ and the degree of the *radical* $\mathrm{rad}(f) := \prod_{i=1}^m f_i$. Shortly after proposing the factor conjecture, Bürgisser [Bür04] showed its plausibility: low-degree factors can be *approximated* by small circuits!

1.4.2 Algebraic approximation

A natural notion of approximation for algebraic computation was defined by Bürgisser [Bür04]. A polynomial $f \in \mathbb{F}[\mathbf{x}]$ is approximated by a polynomial $F \in \mathbb{F}[\varepsilon][\mathbf{x}]$ to an *order of approximation* M if

$$F(\mathbf{x}, \varepsilon) = \varepsilon^M f(\mathbf{x}) + \varepsilon^{M+1}\, Q(\mathbf{x}, \varepsilon), \tag{1.4.2}$$

for some polynomial $Q(\mathbf{x}, \varepsilon) \in \mathbb{F}[\mathbf{x}, \varepsilon]$. The approximate/border size of f, denoted $\overline{\mathsf{size}}(f)$, is defined as the size of the smallest circuit *over the ring of constants* $\mathbb{F}[\varepsilon]$ computing a polynomial $F \in \mathbb{F}[\varepsilon][\mathbf{x}]$ that approximates f.

Over fields like $\mathbb{R}$ or $\mathbb{C}$, one can think of the above as an approximation in the sense $\lim_{\varepsilon\to 0} \varepsilon^{-M} F = f$. Another way of formulating the notion of approximation is to consider $F_\varepsilon \in \mathbb{F}(\varepsilon)[\mathbf{x}]$ over the rational function field $\mathbb{F}(\varepsilon)$ instead, and require an approximation of the form

$$F_\varepsilon(\mathbf{x}) = f(\mathbf{x}) + \varepsilon Q_\varepsilon(\mathbf{x}),$$

where $Q_\varepsilon \in \mathbb{F}[\varepsilon][\mathbf{x}]$ is a polynomial. The border complexity of f is defined as before, but with the circuit size now calculated over $\mathbb{F}(\varepsilon)$. In this case, although $F_{\varepsilon=0} = f$ is defined at $\varepsilon = 0$, the intermediate computations in the circuit for F (over $\mathbb{F}(\varepsilon)$) might not be. Scaling arguments show that these two notions of approximation are equivalent [Bür04, Lemma 5.6]. For a detailed discussion on other natural definitions of approximation (both topological and algebraic), and their equivalence, we point the reader to [Bür04, Section 5] and [BIZ18, Section 2]. The notion of border complexity naturally suggests an approximation version of the class VP called $\overline{\mathsf{VP}}$, that consists of families (f_n) where f_n can be *approximated* by $\mathsf{poly}(n)$-sized circuits (Definition 2.2.22).

It is clear that $\mathsf{VP} \subseteq \overline{\mathsf{VP}}$. In an attempt to utilize sophisticated tools from algebraic geometry and representation theory to study Valiant's conjecture, Mulmuley and Sohoni [MS01; MS08] proposed $\overline{\mathsf{VP}} \not\subseteq \mathsf{VNP}$ as the mathematically nicer (but possibly harder) conjecture. Further details on the *Geometric Complexity Theory* (GCT) program for proving lower bounds can be found in [Reg02; Mul11; Bür+11; Mul12; Gro12; Lan17; BI25].

Returning to the factorization problem, Bürgisser showed that $\mathrm{poly}(n)$-degree factors of families in VP_{nb} are in $\overline{\mathsf{VP}}$. In fact, the approximate circuit constructed for the low-degree factor has additional structure. Note that a priori, the order of approximation M for a polynomial f can be arbitrarily large. However, Bürgisser [Bür04; Bür20, Theorem 5.7] showed that over algebraically closed fields, $M = \exp(\overline{size}(f))$. Therefore, the free constants from $\mathbb{F}[\varepsilon]$ used for approximation can be assumed to be 'only' exponential in degree and hence, size. The circuit for the factor g above is constructed by approximating the power series roots of g (up to lower-order terms) efficiently via Newton Iteration. It can then be observed that the constants in $\mathbb{F}[\varepsilon]$ are in fact circuits of size $\mathrm{poly}(size(f))$ (but exponential in degree).

Given the above fact, formally imposing a restriction on the size of the univariate polynomials in ε is quite natural. We define the new class $\overline{\mathsf{VP}}_\varepsilon$, called *presentable* $\overline{\mathsf{VP}}$ (Definition 2.2.23), which is the same as $\overline{\mathsf{VP}}$, but with the additional restriction that the circuit size of the approximating polynomial over the base field $\mathbb{F}$ is $\mathrm{poly}(n)$, thus also incorporating the size of the constants in $\mathbb{F}[\varepsilon]$. Over finite fields, we show [BDS24a] that $\overline{\mathsf{VP}}_\varepsilon$ is in fact in VNP, something that is not known about $\overline{\mathsf{VP}}$. Combining this with our earlier observation about the (truncated) roots being 'presentable', we can conclude that over finite fields, the low-degree truncation of *separable* roots (whose multiplicity is coprime to the field characteristic) can be computed by a hypercube sum of small circuits. The factors can be recovered by multiplying appropriate truncated roots. Thus, families of low-degree factors (that are generated by few roots), of high-degree circuits are in VNP [BDS24a, Corollary 1.3]. It is an open problem to extend these results to all fields.

1.4.3 Our contribution: the reverse Kronecker conjecture

For a polynomial $F(z_1, \ldots, z_n)$ with individual degree r, the Kronecker map replaces the variable z_i with a power of a new variable: $\varepsilon^{(r+1)^{i-1}}$. This ensures that each monomial of F is mapped to a unique and distinct power of ε [Kro82]. In order to extend the above factoring result to all fields, we propose a 'reverse' Kronecker conjecture (Conjecture 6.1.1).

> A version of our conjecture is that over any field, a *univariate* circuit of size s (and degree $\exp(s)$, possibly) can be obtained via a Kronecker substitution to a $\mathsf{poly}(s)$-variate circuit of size $\mathsf{poly}(s)$.

We will show in Chapter 6 that the above conjecture implies $\overline{\mathsf{VP}}_\varepsilon \subseteq \mathsf{VNP}$ over all fields. Weaker versions of the conjecture also suffice for our applications. Using our observations about being able to approximate roots of small circuits of high degree in a presentable way, this in turn implies that these roots are computable by hypercube sums of small circuits, over all fields.

Chapter 2

Preliminaries

2.1 Notation

We record some standard notation used in the thesis.

- We will denote by $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, the set of natural numbers, integers, rationals, reals, and complex numbers, respectively.
- A field will usually be denoted by $\mathbb{F}$. We will use $\mathbb{F}_q$ to denote a finite field of order q.
- For a natural number $n \in \mathbb{N}$, we will use $[n]$ to refer to the set $\{1, \ldots, n\}$.
- For any real number r, we will denote the nearest integer by $\lfloor r \rceil$.
- A list of objects will be denoted by bold letters. For e.g., $\mathbf{x}$ will usually denote the set of variables $(x_1, \ldots, x_n)$, and $|\mathbf{x}|$ will denote its size, n.
- For a vector $\mathbf{e} \in \mathbb{Z}^n$, we denote by $\mathbf{x}^{\mathbf{e}}$ the monomial $x_1^{e_1} \cdots x_n^{e_n}$.

We now define some algebraic complexity and graph-theoretic preliminaries. Readers comfortable with these notions can safely skip these sections.

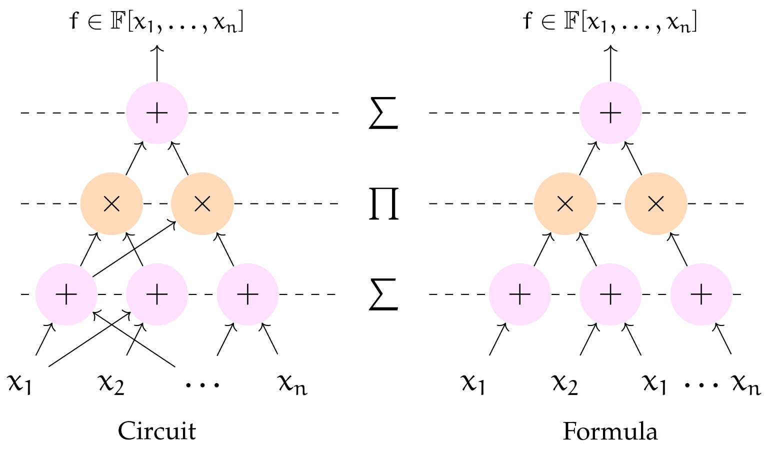


Figure 2.1: An algebraic circuit and a formula of depth 3.

2.2 Algebraic complexity theory

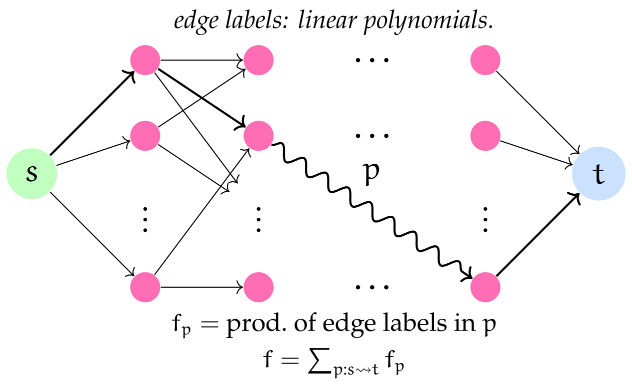
Algebraic circuits are analogous to Boolean circuits, where logical operators are replaced with underlying field operations. See Figure 2.1 for an illustration.

Definition 2.2.1 (Algebraic Circuits and Formulas)**.** An *algebraic circuit*, defined over a field $\mathbb{F}$, is a *layered directed acyclic graph* with alternating layers of '$+$' and '$\times$' gates, and a single root, called the 'output' gate. The 'input' *leaf* gates are labeled by either a variable from $x_1, \ldots, x_n$ or a constant from $\mathbb{F}$. If the graph is a *tree*, then we call it a *formula*.

A circuit computes a polynomial $f \in \mathbb{F}[\mathbf{x}]$ in the natural way: a '$+$' gate sums up the polynomials from its children, whereas a '$\times$' gate computes their product, with the root finally computing f. The *size* of a circuit is the total number of vertices in the graph. The *depth* δ of the circuit is the number of layers in the circuit, or equivalently, the length of the longest path from the root to a leaf. The number of layers of multiplication gates (usually denoted by Δ) is called the *product-depth* (depth is roughly twice the product-depth).

A polynomial is called *homogeneous* if all its monomials have the same degree.

Definition 2.2.2 (Homogenous Circuits)**.** A circuit is called *homogeneous* if every gate in the circuit computes a homogeneous polynomial.



edge labels: linear polynomials.
s
t
p
f_p = prod. of edge labels in p
$f = \sum_{p:s\rightsquigarrow t} f_p$

Figure 2.2: Algebraic Branching Program (ABP)

Consider a partition of the variable set $X = X_1 \sqcup X_2 \sqcup \ldots \sqcup X_d$. A polynomial f is called *set-multilinear* with respect to X if for every monomial m (seen as a multiset), it holds that $|m \cap X_i| = 1$ for all $i \in [d]$.

Definition 2.2.3. A circuit is called *set-multilinear* if every gate in the circuit computes a set-multilinear polynomial with respect to (a subset of) the variable sets.

We will also encounter circuits that are lopsided.

Definition 2.2.4 (Skew Circuits)**.** An algebraic circuit is called *skew* if, for every multiplication gate, *at most one* of its children is an internal (non-input) gate.

Algebraic Branching Programs (Figure 2.2) are a model of computation different from (but related to) circuits and formulas.

Definition 2.2.5 (Algebraic Branching Programs)**.** An *Algebraic Branching Program* (ABP) is a directed acyclic graph with edges labeled by a variable or a field constant. It has a designated *source* node s (of in-degree 0) and a *sink* node t (of out-degree 0). A path from s to t computes the product of all edge labels along the path. The polynomial computed by the ABP is the sum of the terms computed along all the paths from s to t. The *size* of an ABP is the total number of vertices in the graph, and the length of the longest path from s to t is the *length* of the ABP.

An ABP of length ℓ with n_i vertices in the i-th layer can be written as a product of $\ell - 1$ matrices $\prod_{i=1}^{\ell-1} M_i$ in a natural way: the matrix M_i is of dimension $n_i \times n_{i+1}$

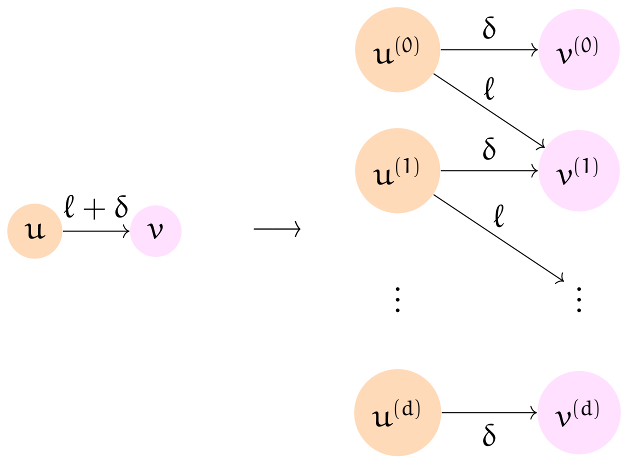
and contains the edge labels between layers i and $i+1$ as entries. The size of an ABP is equivalently the sum of the number of rows of the matrices in matrix representation.

Any vertex v in an ABP can be thought of as computing a polynomial corresponding to the 'sub-ABP' between the source s and the vertex v. An ABP is *homogenous* if the polynomial computed at every vertex is homogenous.

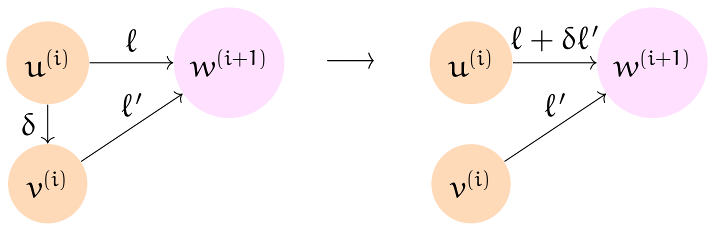
Lemma 2.2.6 (ABP homogenization). *Let $f(x_1, \ldots, x_n)$ be a degree d polynomial. Suppose that f can be computed by an* ABP *of size s. Then there is a* homogeneous ABP *of width s and length d that can compute the same polynomial. Furthermore, all the edge labels are* linear forms.

The above lemma is "folklore" with the proof idea already present in [Nis91]. We provide a proof for completeness, based on the exposition of [IL17].

Proof of Lemma 2.2.6. For every vertex v (other than the start vertex), we replace it with $d+1$ vertices $v^{(0)}, v^{(1)}, \ldots, v^{(d)}$. Each $v^{(i)}$ corresponds to the homogeneous degree i component of the polynomial computed at v. In the original ABP, say an edge from vertex u to v is labelled $\ell + \delta$ (ℓ is a linear form and δ is a constant). We replace it with $2d+1$ edges. We add edges from $u^{(i)}$ to $v^{(i)}$ with label δ for $0 \leq i \leq d$. And we add edges from $u^{(i)}$ to $v^{(i+1)}$ with the label ℓ for $0 \leq i \leq d-1$. This ABP now computes the same polynomial as before and is *homogeneous*.



To make the length d, we modify it so that all vertices computing degree i polynomials are in the layer i (this makes the width s). If some of these vertices have no incoming edges from layer $i-1$, we can safely remove them. Note that the edges between layers will be linear forms. But we may have edges labeled with constants between two vertices in the i-th layer due to our reorganisation.



So for every vertex u in the i-th layer, and vertex w in the $(i+1)$-th layer, we add an edge with a linear form obtained by the *sub*-ABP between u and w. Then we drop all the in-layer edges. This gives a homogeneous ABP of d layers with all edges being *linear forms*. Indeed, the edges we added initially were already linear forms, and the sub-ABPs all compute linear forms as well since every path is of length 2 with one edge label being a constant and the other being a linear form. Note that there are multiple output vertices now. In layer i for example, the sum of the polynomials computed at vertices with no outgoing edges is the degree i homogeneous component of f. $\square$

There are also set-multilinear versions of ABPs where the variable set is partitioned as before: $X = X_1 \sqcup \ldots \sqcup X_d$.

Definition 2.2.7 (Set-multilinear ABPs)**.** An smABP in the *natural order* is a $(d+1)$ layered ABP with edges between layers i and $i+1$ labeled by *linear forms* only in X_i. More generally, for a permutation $\pi \in S_d$ of the variable sets, we say that an smABP is in the order π if the edges between i-th and $(i+1)$-th layer are labeled by *linear forms* in $X_{\pi(i)}$.

Remark. The above definition differs slightly from that of Forbes [For14] as it does not allow *affine* linear forms as edge labels. We use this definition as the ABPs we encounter are of this more restricted form and proving lower bounds for them is sufficient. Our definition is more in line with the earlier work of Klivans and Shpilka [KS06].

We can also define more general versions of set-multilinear polynomials and ABPs. A polynomial g is called *set-multi-k-ic* with respect to X if every monomial of g has exactly k variables (with multiplicity) from each of the d sets. That is, for a monomial m (seen as a multiset) in the support of g, $|m \cap X_i| = k$. When $k = 1$, the polynomial g is set-multilinear.

Definition 2.2.8 (Set-multi-k-ic ABPs)**.** We call an ABP of length kd a *set-multi-k-ic* ABP (denoted $\mathsf{sm}(k)\mathsf{ABP}$) if every layer has edges labeled by linear forms from exactly one of the sets X_i, and there are exactly k layers corresponding to each X_i. As a special case, an sm(1)ABP is just a set-multilinear ABP as defined before.

The *set-multi-k-ic* ABP is inspired from the well-studied *multi-k-ic* depth-restricted circuits and formulas, initiated by Kayal and Saha [KS17]. We encourage readers to consult [Sap21, Chapter 14] and references therein for a comprehensive discussion.

There is also a model closely related (and essentially equivalent) to smABPs. An algebraic branching program over the variables $(x_1, \dots, x_n)$ is said to be *oblivious* if, for any layer, all the edge labels are univariate polynomials in a single variable. It is further called a *read-once* oblivious ABP (or an ROABP) if every variable appears in at most one layer.

Definition 2.2.9 (ROABP)**.** An ROABP in the *natural order* is an $(n+1)$-layered ABP where the edges between layers i and $i+1$ are labeled by univariate polynomials in x_i of degree d. If, instead, the labels were univariate polynomials in $x_{\pi(i)}$ for some permutation $\pi \in S_d$ of the variables, then we say that the ROABP is in the order π.

We also define a more general version of ROABPs.

Definition 2.2.10 (Read-k Oblivious ABPs)**.** An *oblivious* ABP is said to be *read*-k (denoted $\mathsf{R}(k)\mathsf{O}$) if every layer has edges labeled by univariate polynomials in a single variable x_i, and each x_i appears in k layers. As a special case, an $\mathsf{R}(1)\mathsf{O}$ is just an ROABP.

We denote the sum of *read*-k oblivious ABPs as $\sum \mathrm{R}(k)\mathrm{O}$. Once again, the width of a $\sum \mathrm{R}(k)\mathrm{O}$ is the sum of the widths of the constituent branching programs.

Monotone algebraic computation, which is free of cancellations, can be simulated by circuits (branching programs) by restricting the choice of field constants.

Definition 2.2.11 (Monotone Circuits and ABPs)**.** A *monotone* circuit (ABP) is an algebraic circuit (ABP) where all the field constants are non-negative. The circuit (ABP) computes a *monotone polynomial*, where coefficients of all the monomials are non-negative.

Remark 2.2.12. It is not hard to show that skew circuits and ABPs are essentially the same model, up to constant factors (for e.g., see the discussion in [Mah14]). In particular, an ABP of size s and length ℓ can be converted to a skew circuit of size $O(s)$ and product-depth ℓ. If the original ABP is monotone, the skew circuit is monotone as well.

Parse trees model the computation of each *monomial* in a circuit. The notion has been quite useful in algebraic complexity [All+98; JS82; KPR23; MP08; Ven92].

Definition 2.2.13 (Parse Trees)**.** A *parse tree* $\mathcal{T}$ of an algebraic circuit C is obtained as follows:

- We include the root gate of C in $\mathcal{T}$.
- For every $+$ gate in $\mathcal{T}$, we arbitrarily include *any one* of its children in $\mathcal{T}$.
- For every $\times$ gate in $\mathcal{T}$, we include *all* of its children in $\mathcal{T}$.

We call a parse tree *reduced* if we ignore every $+$ gate, and its parent and (only) child are directly connected by an edge.

Remark 2.2.14. It is easy to see that every (reduced) parse tree is associated with a monomial of the polynomial computed by the circuit. For a (reduced) parse tree $\mathcal{T}$, let $\mathrm{val}(\mathcal{T})$ be its output. Then the polynomial computed by the circuit C is $\sum_{\mathcal{T}} \mathrm{val}(\mathcal{T})$, where the sum is over all the parse trees of C. From here on, whenever we use the term parse tree, we mean the reduced parse tree.

2.2.1 Algebraic complexity classes

A number of complexity classes have been studied based on the models described earlier. We provide a non-exhaustive list here. All the complexity classes are defined with respect to an underlying field $\mathbb{F}$, where the computation occurs. We will not make this dependence explicit in our notation, unless necessary.

We begin by defining the classes that encapsulate efficient algebraic computation. The algebraic version of the Boolean class P is defined by considering polynomial families with small circuit size and low degree.

Definition 2.2.15 (VP)**.** A polynomial family $f = (f_n)$ is in the class VP if both the number of variables and degree of f_n are bounded by $\mathrm{poly}(n)$ and moreover, the size of the smallest circuit computing f_n, denoted $\mathit{size}(f_n)$ is bounded by $\mathrm{poly}(n)$.

As a formula of size s can only compute polynomials of degree $O(s)$, we do not need any explicit degree restriction.

Definition 2.2.16 (VF)**.** A family of polynomials $f = (f_n)$ is said to be in VF if the minimum size of the *formula* computing f_n is bounded by $\mathrm{poly}(n)$.

The degree restriction is also not necessary for algebraic branching programs.

Definition 2.2.17 (VBP)**.** A polynomial family $f = (f_n)$ is said to be in the class VBP if the number of variables, and the size of the smallest *algebraic branching program* computing f_n are bounded by $\mathrm{poly}(n)$.

The Boolean class NP (or rather, its non-uniform version) can be thought of as a logical OR of small Boolean circuits. The algebraic analog is defined by instead considering a hypercube *sum* of small algebraic circuits.

Definition 2.2.18 (VNP)**.** A family of polynomials $f = (f_n)$ is said to be in VNP if there exist functions $k, \ell, m : \mathbb{N} \to \mathbb{N}$ all polynomially bounded, and a polynomial family $g = (g_n) \in \mathrm{VP}$ with $g_n \in \mathbb{F}[x_1, \ldots, x_{k(n)}, y_1, \ldots, y_{m(n)}]$ such that for all n,

$$f_n(x_1, \ldots, x_{k(n)}) = \sum_{\mathbf{a} \in \{0,1\}^{m(n)}} g_{\ell(n)}(x_1, \ldots, x_{k(n)}, a_1, \ldots, a_{m(n)}).$$

The following criterion by Valiant [Val79a] provides a way to test if a polynomial family is in VNP, and was the basis of the informal definition for VNP we gave in Chapter 1. We denote by $\langle a \rangle$ the boolean encoding of any mathematical object a. For more on Boolean complexity, refer [AB09].

Proposition 2.2.19 (Valiant's Criterion)**.** *Consider a polynomial family* $f = (f_n)$ *with the number of variables and degree of* f_n *bounded by* $\mathrm{poly}(n)$. *Suppose* $f_n = \sum_{\mathbf{e}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}$. *If for every* n, *there exists a function* $\phi_n \in \#\mathrm{P}/\mathrm{poly}$ *such that given any exponent vector* $\mathbf{e}$, *we have* $\phi_n(\langle \mathbf{e} \rangle) = \langle c_{\mathbf{e}} \rangle$, *then* $f \in \mathrm{VNP}$.

The converse is only true over finite fields. We also state few useful closure properties of VNP [Val82].

Proposition 2.2.20 (Closure properties of VNP)**.** *Suppose* $f = (f_n), g = (g_n)$ *are two polynomial families in* VNP. *Then,*

1. *(Sum and product)* $f + g$ *and* $(f \cdot g)$ *are also in* VNP.
2. *(Coefficient) If* h_n *is the coefficient of some monomial in* f_n *with respect to a subset of the variables, then* $h = (h_n)$ *is also in* VNP.

We will also consider classes where the degree of the circuits is large. If we only require the size of the circuits to be small, we obtain a version of VP where the degree in *not bounded*.

Definition 2.2.21 ($\mathrm{VP}_{\mathrm{nb}}$)**.** A family of polynomials $f = (f_n)$ is said to be in $\mathrm{VP}_{\mathrm{nb}}$ if the number of variables and the size of the smallest circuit computing f_n are bounded by $\mathrm{poly}(n)$.

If we allow the polynomials to be *approximated* (as in Equation (1.4.2)) by small-size circuits over $\mathbb{F}[\varepsilon]$, we obtain the *closure* of VP.

Definition 2.2.22 ($\overline{\mathrm{VP}}$)**.** A polynomial family $f = (f_n)$ is in the class $\overline{\mathrm{VP}}$ if the number of variables of f_n, its degree, and the size of the smallest circuit (over $\mathbb{F}[\varepsilon]$) *approximating* f_n are bounded by $\mathrm{poly}(n)$.

If we further impose the restriction that even the constants in $\mathbb{F}[\varepsilon]$ have small circuits, we get a presentable version of $\overline{\mathsf{VP}}$.

Definition 2.2.23 ($\overline{\mathsf{VP}}_\varepsilon$)**.** A polynomial family $f = (f_n)$ is in the class $\overline{\mathsf{VP}}_\varepsilon$ if the number of variables of f_n, its degree, and the size of the smallest circuit *over* $\mathbb{F}$ approximating f_n are bounded by $\mathrm{poly}(n)$.

2.3 Graph theory

In the following, let H be a graph. A vertex cover of H is a subset $C \subseteq V(H)$ such that for every edge $e \in E(H)$, some vertex $v \in C$ is an endpoint of e. The vertex-cover number of H is the minimum size of a vertex-cover in H.

Definition 2.3.1 (Tree-decomposition and treewidth)**.** A tree-decomposition of H is a tree T whose vertices are annotated with *bags* $\{X_t\}_{t\in V(T)}$, subject to the following conditions:

1. Every vertex v of H is in at least one bag.
2. For every edge (u, v) in H there is a bag in T that contains both u and v.
3. For any vertex v of H, the subgraph of T induced by the bags containing v is a subtree.

The *width* of a tree decomposition T is $\max_{t\in V(T)} |X_t| - 1$. The *treewidth* of the graph H, denoted $tw(H)$, is the minimum width over all tree decompositions of H.

Definition 2.3.2 (Path-decomposition and pathwidth)**.** A path-decomposition of a graph H is a tree-decomposition where the underlying tree is a path. The *width* of a path decomposition P is $\max_{t\in V(P)} |X_t| - 1$. The *pathwidth* of the graph H, denoted $pw(H)$, is the minimum width over all path decompositions of H.

We usually consider trees with a designated root vertex. The height of such a tree is the number of vertices on a longest root-to-leaf path. We assume trees in tree-decompositions to be rooted so as to minimize their height.

Chapter 3

Constant Depth Circuits

In this chapter, we improve the lower bound for IMM against constant-depth circuits shown by Limaye, Srinivasan and Tavenas [LST21]. We also exhibit barriers to further improving the bound using these techniques, which is important as this is the only known approach to achieve super polynomial lower bounds for general circuits of low depth. The results in this chapter appeared in [BDS24a]. The lower bound result based on the conference version was also reported in [Dut23]. We simplify and fix remaining issues here, in addition to proving the barrier results.

For the rest of this chapter, let $\mu(\Delta) = 1/(F(\Delta) - 1)$ where $F(n) = \Theta(\varphi^n)$ is the n-th Fibonacci number (starting with $F(0) = 1$, $F(1) = 2$) and $\varphi = (1 + \sqrt{5})/2 = 1.618\ldots$ is the golden ratio.

Theorem 3.0.1. *(General circuit lower bound) Suppose* N, d, Δ *are integers such that* $d = O(\log N/\log\log N)$. *Then, any product-depth* Δ *circuit computing* $\mathrm{IMM}_{n,d}$ *on* $N = dn^2$ *variables must have size at least* $N^{\Omega\left(d^{\mu(2\Delta)}/\Delta\right)}$.

Remark 3.0.2. Theorem 3.0.1 improves on the lower bound of $N^{\Omega\left(d^{1/(2^{2\Delta}-1)}/\Delta\right)}$ of [LST21] since $F(2\Delta) = \Theta(\varphi^{2\Delta}) \ll 2^{2\Delta}$.

To prove Theorem 3.0.1, we use the hardness escalation given by [LST21, Proposition 9], which allows for the conversion of general circuits to set-multilinear ones without significant blow-up (provided the degree is small). We will use the version from the result of Forbes [For24] that works over all fields.

Lemma 3.0.3. *[For24, Corollary 27] Let s, N, d, Δ be growing parameters with $s \geq Nd$. If C is a circuit of size at most s and product-depth at most Δ computing a set-multilinear polynomial P over the sets of variables $(X_1, \ldots, X_d)$ (with $|X_i| \leq N$), then there is a* set-multilinear circuit $\tilde{C}$ *of size $d^{O(d)}\mathrm{poly}(s)$ and product-depth at most 2Δ computing the same polynomial P.*

The actual lower bound is for set-multilinear circuits.

Theorem 3.0.4. *(Set-multilinear circuit lower bound) Let $d \leq (\log n)/4$. Any product-depth Δ* set-multilinear *circuit computing $\mathrm{IMM}_{n,d}$ must have size at least $n^{\Omega\left(d^{\mu(\Delta)}/\Delta\right)}$.*

Remark 3.0.5. Theorem 3.0.4 is an improvement over the $n^{\Omega\left(d^{1/(2^{\Delta}-1)}/\Delta\right)}$ bound of [LST21, Lemma 15]. The result holds over any field $\mathbb{F}$, same as [LST21]. The difference between $\mu(2\Delta)$ in Theorem 3.0.1 and $\mu(\Delta)$ in Theorem 3.0.4 is due to the doubling of product-depth during the conversion from a general circuit to a set-multilinear one (Lemma 3.0.3).

We now prove Theorem 3.0.1 à la [LST21, Corollary 4]:

Proof of Theorem 3.0.1. From Lemma 3.0.3 and Theorem 3.0.4, for a circuit of product-depth Δ and size s computing $\mathrm{IMM}_{n,d}$ we get that

$$d^{O(d)}\mathrm{poly}(s) \geq N^{\Omega\left(d^{\mu(2\Delta)}/2\Delta\right)}.$$

Since $d = O(\log N/\log\log N)$, it follows that $d^{O(d)} = N^{O(1)}$. Therefore,

$$\mathrm{poly}(s) \geq N^{\Omega\left(d^{\mu(2\Delta)}/2\Delta\right)}/d^{O(d)} \geq N^{\Omega\left(d^{\mu(2\Delta)}/4\Delta\right)}$$

implying the required lower bound on s and, thus, the theorem. $\square$

Remark 3.0.6. Theorem 3.0.1 also holds when $d = O(\log N)$ and $\Delta \leq 1/4 \log_{\varphi} \log d$. This is because the above bound on Δ implies that $d^{\mu(2\Delta)}/\Delta \geq d^{\Omega(1/\varphi^{2\Delta})}/\Delta \geq d^{\Omega(1/\sqrt{\log d})}/\log\log d = \omega(\log d)$. Using this inequality together with the assumption $d = O(\log N)$, we get $d^{O(d)} = 2^{O(d\log d)} \leq 2^{o(\log N \cdot d^{\mu(2\Delta)}/\Delta)} = N^{o(d^{\mu(2\Delta)}/\Delta)}$ we can then proceed similarly to the proof of Theorem 3.0.1.

3.1 Preliminaries

For any positive integer n, we denote by $F(n)$ the n-th Fibonacci number with $F(0) = 1$, $F(1) = 2$ and $F(n) = F(n-1) + F(n-2)$. The nearest integer to any real number r is denoted by $\lfloor r \rceil$. We follow the notation of [LST21] as much as possible for better readability.

We consider words that are tuples $(w_1, \ldots, w_d)$ of length d where $2^{|w_i|}$ are integers. These words define the set sizes of the set-multilinear polynomials we will work with. Given a word w, let $\overline{X}(w)$ denote the tuple of sets of variables $(X_1(w), \ldots, X_d(w))$ where the size of each $X_i(w)$ is $2^{|w_i|}$. We denote the space of set-multilinear polynomials over $\overline{X}(w)$ by $\mathbb{F}_{sm}[\overline{X}(w)]$.

For a word w and any subset $S \subseteq [d]$, the sum of elements of w indexed by S is denoted by $w_S = \sum_{i \in S} w_i$. For all $t \leq d$, if it holds that $|w_{[t]}| \leq b$, then we call w 'b-unbiased'. Denote by $w_{|S}$ the sub-word indexed by S. The positive and negative indices of w are denoted $\mathcal{P}_w = \{i \mid w_i \geq 0\}$ and $\mathcal{N}_w = \{i \mid w_i < 0\}$ respectively with the corresponding collections $\{X_i(w)\}_{i \in \mathcal{P}_w}$ and $\{X_i(w)\}_{i \in \mathcal{N}_w}$ being the positive and negative variable sets. We denote by $\mathcal{M}_w^{\mathcal{P}}$ (resp. $\mathcal{M}_w^{\mathcal{N}}$) the set of all set-multilinear monomials over the positive (resp. negative) variable sets.

The *partial derivative matrix* $\mathcal{M}_w(f)$ of f has rows indexed by $\mathcal{M}_w^{\mathcal{P}}$ and columns by $\mathcal{M}_w^{\mathcal{N}}$. The entry corresponding to row $m_+ \in \mathcal{M}_w^{\mathcal{P}}$ and $m_- \in \mathcal{M}_w^{\mathcal{N}}$ is the coefficient of the monomial $m_+ m_-$ in f. The complexity measure we use is the *relative rank*, same as [LST21]:

$$\mathrm{relrk}_w(f) := \frac{\mathrm{rank}(\mathcal{M}_w(f))}{\sqrt{|\mathcal{M}_w^{\mathcal{P}}| \cdot |\mathcal{M}_w^{\mathcal{N}}|}} = \frac{\mathrm{rank}(\mathcal{M}_w(f))}{2^{\frac{1}{2}\sum_{i \in [d]} |w_i|}} \leq 1\,.$$

The following properties of relrk_w will be useful (for proofs, see [LST21]).

1. (Imbalance) For any $f \in \mathbb{F}_{sm}[\overline{X}(w)]$, $\mathrm{relrk}_w(f) \leq 2^{-|w_{[d]}|/2}$.

2. (Sub-additivity) For any $f, g \in \mathbb{F}_{sm}[\overline{X}(w)]$, $\mathrm{relrk}_w(f + g) \leq \mathrm{relrk}_w(f) + \mathrm{relrk}_w(g)$.

3. (Multiplicativity) Suppose we have $f = f_1 f_2 \cdots f_t$ where $f_i \in \mathbb{F}_{sm}[\overline{X}(w_{|S_i})]$, and $(S_1, \ldots, S_t)$ is a partition of $[d]$. Then, $\mathrm{relrk}_w(f) = \mathrm{relrk}_w(f_1 f_2 \cdots f_t) = \prod_{i \in [t]} \mathrm{relrk}_{w_{|S_i}}(f_i)$.

We now define the hard polynomials for which we prove lower bounds. For any monomial $m \in \mathbb{F}_{sm}[\overline{X}(w)]$, let $m_+ \in \mathcal{M}_w^{\mathcal{P}}$ and $m_- \in \mathcal{M}_w^{\mathcal{N}}$ be its "positive" and "negative" parts. As $|X_i| = 2^{|w_i|}$, the variables of X_i can be indexed using Boolean strings of length $|w_i|$. This gives a way to associate a Boolean string with any monomial. Let $\sigma(m_+)$ and $\sigma(m_-)$ be the strings associated with m_+ and m_- respectively. We write $\sigma(m_+) \sim \sigma(m_-)$ if one is a prefix of the other.

Definition 3.1.1. [LST21, Word polynomials] Let w be any word. The polynomial P_w is defined as the sum of all monomials $m \in \mathbb{F}_{sm}[\overline{X}(w)]$ such that $\sigma(m_+) \sim \sigma(m_-)$.

The matrices $M_w(P_w)$ have full rank (equal to either the number of rows or columns, whichever is smaller), and hence $\mathrm{relrk}_w(P_w) = 2^{-|w_{[d]}|/2}$. We note (without proof) that these polynomials can be obtained as *set-multilinear* restrictions of $\mathrm{IMM}_{n,d}$.

Lemma 3.1.2. *[LST21, Lemma 8] Let w be any b-unbiased word. If there is a set-multilinear circuit computing $\mathrm{IMM}_{2^b,d}$ of size s and product-depth Δ, then there is also a set-multilinear circuit of size s and product-depth Δ computing the polynomial $P_w \in \mathbb{F}_{sm}[\overline{X}(w)]$. Moreover, $\mathrm{relrk}_w(P_w) \geq 2^{-b/2}$.*

3.2 Proof outline

From the discussion in Section 1.1 and Lemma 3.1.2 and Lemma 3.0.3, to prove general circuit lower bounds, it suffices to prove that there is a word polynomial of high rank that needs large set-multilinear formulas. For a word (and hence set sizes) of our choice, we show that relrk_w is small for set-multilinear formulas of a certain size.

Let k be an integer close to $\log_2 n$. In [LST21], the authors choose the positive entries of the word w as an integer close to $k/\sqrt{2}$ and the negative entries as $-k$.

Evidently, these entries are independent of the product-depth Δ. We instead take the positive entries as $(1-p/q)k$ and the negative entries as $-k$ where p and q are suitable integers dependent on Δ. This *depth-dependent* construction of the word enables us to improve the lower bound. We demonstrate the high-level proof strategy of the lower bound for the case of product-depth 3.

3.2.1 Proof overview of Theorem 3.0.4 for $\Delta = 3$

Define $G(i) = 1/\mu(i) = F(i) - 1$ for all i and let $\lambda = \lfloor d^{1/G(3)} \rfloor$. Consider a set-multilinear formula C of product-depth 3 and let ν be a gate in it. Suppose that the subformula $C^{(\nu)}$ rooted at ν has product-depth $\delta \leq 3$, size s and degree $\geq \lambda^{G(\delta)}/2$. We will prove that $\mathsf{relrk}_w(C^{(\nu)}) \leq s2^{-k\lambda/48}$ by induction on δ. This will give us the desired upper bound of the form $s2^{-k\lambda/48} = sn^{-\Omega(d^{\mu(3)})}$ on the relative rank of the whole formula when ν is taken to be the output gate. Write $C^{(\nu)} = C_1 + \cdots + C_t$ where each C_i is a subformula of size s_i rooted at a product gate. Because of the subadditivity of relrk_w, it suffices to show that $\mathsf{relrk}_w(C_i) \leq s_i 2^{-k\lambda/48}$ for all i.

Base case: If $\delta = 1$, then C_i is a product of linear forms. Thus, it has a rank of 1 and a low relative rank.

Induction step: $\delta \in \{2, 3\}$. Write $C_i = C_{i,1} \ldots C_{i,t_i}$ where each $C_{i,j}$ is a subformula of product-depth $\delta - 1$. If any $C_{i,j}$ has degree $\geq \lambda^{G(\delta-1)}/2$, then by the induction hypothesis, the relative rank of $C_{i,j}$ and hence C_i will have the desired upper bound, and we are done.

Otherwise each $C_{i,j}$ has degree $D_{ij} < \lambda^{G(\delta-1)}/2$. As the formula is set-multilinear, there is a collection of variable sets $(X_l)_{l \in S_j}$ with respect to which $C_{i,j}$ is set-multilinear. For $j \in [t_i]$, let a_{ij} be the number of positive indices in S_j i.e. the number of positive sets in the collection $(X_l)_{l \in S_j}$. Then the number of negative indices is $(D_{ij} - a_{ij})$.

We consider two cases: if $a_{ij} \leq D_{ij}/3$, then $w_{S_j} \leq (D_{ij}/3) \cdot (1-p/q)k + (2D_{ij}/3) \cdot (-k)$
$\leq -D_{ij}k/3$. Otherwise, $a_{ij} > D_{ij}/3$ and if we can prove that $|w_{S_j}| \geq a_{ij}k/(4\lambda^{G(\delta)-1})$,

then in both of the above cases, we would have $|w_{S_j}| \geq D_{ij}k/(12\lambda^{G(\delta)-1})$. By the multiplicativity and imbalance property of relrk_w, it would follow that $\mathsf{relrk}_w(C_i) \leq 2^{\sum_{j=1}^{t_i} -\frac{1}{2}|w_{S_j}|} \leq 2^{-k\lambda/48}$ and we would be done. Thus, we now only have to show that $|w_{S_j}| \geq a_{ij}k/(4\lambda^{G(\delta)-1})$. We have

$$|w_{S_j}| = |a_{ij}(1 - p/q) - (D_{ij} - a_{ij})|\, k\,.$$

Notice that $|w_{S_j}|/k$ is the distance of $a_{ij}p/q$ from some integer, so it must be at least the minimum of $\{a_{ij}p/q\}$ and $1 - \{a_{ij}p/q\}$ where $\{.\}$ denotes the fractional part. The number $a_{ij}p/q$ being rational, has a fractional part $\zeta = (a_{ij}p \bmod q)/q$, and hence it comes down to solving the following system of inequalities:

$$\min(\zeta,\ 1-\zeta) \geq a_{ij}/(4\lambda^{G(\delta)-1}) \text{ for } \delta \in \{2,3\} \text{ when } a_{ij} \leq D_{ij} < \lambda^{G(\delta-1)}/2\,.$$

Assign $p = \lambda$, $q = \lambda^2+1$. The $\delta = 2$ case is clearly satisfied as $(a_{ij}\lambda \mod (\lambda^2+1)) = a_{ij}\lambda$ when $0 \leq a_{ij} \leq \lambda/2$.

Consider the case of $\delta = 3$ and $a_{ij} < \lambda^2/2$. Write $a_{ij} = y_1\lambda + y_0$ for integers $y_1 = \lfloor a_{ij}/\lambda \rfloor < \lambda/2$ and $y_0 \leq \lambda - 1$. Thus, $a_{ij}\lambda \equiv -y_1 + y_0\lambda \mod (\lambda^2+1)$. Through some case analysis, one can show that $\min\Big(|y_0\lambda - y_1|,\ \lambda^2 + 1 - |y_0\lambda - y_1|\Big) \geq y_1$ which immediately implies the inequality for the $\delta = 3$ case as $y_1 = \lfloor a_{ij}/\lambda \rfloor \geq a_{ij}/(2\lambda)$.

We can attempt to extend this proof technique to product-depth 4 as follows: We would similarly want to express a_{ij} as $a_{ij} = y_2\lambda^2 + y_1\lambda + y_0$ for integers $y_2 = \lfloor a_{ij}/\lambda^2 \rfloor, y_0 \leq \lambda - 1$ and $y_1 \leq \lambda - 1$. Ideally, we would want that for some $q \approx \lambda^4$,

$$p\lambda^2 \equiv 1 \bmod q,\ p\lambda \equiv -\lambda^2 \bmod q \text{ and } p \equiv \lambda^3 \bmod q$$

so that $a_{ij}p \equiv y_2 - y_1\lambda^2 + y_0\lambda^3 \bmod q$ and then we can carry out a similar analysis as in the $\Delta = 3$ case. But this is impossible since multiplying the second congruence equation by λ gives $p\lambda^2 \equiv -\lambda^3 \bmod q$, which contradicts the first congruence equation. So we decide to express a_{ij} as $a_{ij} = y_2b_2 + y_1b_1 + y_0b_0$ where b_2, b_1, b_0

are close to $\lambda^2, \lambda, 1$ respectively, instead of being precisely equal to these powers of λ. Then we choose $c_2 \approx 1, c_1 \approx -\lambda^2, c_0 \approx \lambda^3$ and we assign values to p and q such that

$$pb_2 \equiv c_2 \bmod q, \ pb_1 \equiv c_1 \bmod q \text{ and } pb_0 \equiv c_0 \bmod q.$$

It is easy to verify that all these conditions are satisfied if we define
$b_0 = 1, b_1 = \lambda, b_2 = b_1(\lambda - 1) + b_0; \qquad c_2 = 1, c_1 = -\lambda^2, c_0 = c_2 - c_1(\lambda - 1);$
$p = c_0$ and $q = pb_1 - c_1$.
This inspired our construction of the sequences $\{b_m\}$ and $\{c_m\}$ for general product-depth Δ.

3.2.2 Proof overview of Theorem 3.4.1

As previously mentioned, we would like to find a family of polynomials for which our lower bound is tight. All the same, we want to maintain a high relative rank of these polynomials. If we can achieve this and find the appropriate small-sized formulas for the said polynomials, we will have that the lower bound cannot be improved using the relative rank measure.

The polynomial P we define will be a close variant of the word polynomials from before. This will ensure that the partial derivative matrix has the maximum possible rank for a matrix of its dimension. From the Imbalance property, the relative rank we obtain is $2^{-|w_{[d]}|/2}$ where we have ensured that $w_{[d]}$ is small. We want to construct the formula F for P with a nice inductive structure. We also want the polynomials computed by the subformulas of F to have a high relative rank. This will help us construct a formula from its sub-formulas while maintaining a high relative rank.

Suppose a subformula F' of F is set multilinear with respect to a subtuple $\mathcal{T}$ of the sets of variables $\overline{X}(w)$. Let these sets in $\mathcal{T}$ be indexed by a set $S_{\mathcal{T}} \subseteq [d]$. As we would like high relative rank of F', the Imbalance property again suggests that

$|w_{S_{\mathcal{T}}}|$ be small. And we desire this of every subformula, their subformulas, and so on. So roughly, we want a way to partition our initial index set $[d]$ into some number of index sets $S_1, \dots, S_r$ such that each $|w_{S_i}|$ is small. Suppose we can then create subformulas of rank $2^{-|w_{S_i}|/2}$. We will have to roughly add $2^{\sum_i |w_{S_i}|}$ many of them to get a polynomial of high relative rank. So, to control the size of the formula, we would like $\sum_i |w_{S_i}|$ to be small as well.

In their Depth Hierarchy section, [LST21] use Dirichlet's approximation principle [Sch91] to pick these nice index sets $\{S_i\}$. Their procedure only works for the particular two variable-set sizes they choose. We extend this to any two set sizes in Claim 3.4.8. Interestingly, we do not use Dirichlet's approximation to pick the index sets but rather to obtain a lower bound on the size of the sets we eventually pick. We think of picking sets as an investment process: when we pick a set S, we buy the $|S|$ elements in it for a cost of $|w_S|$. Hence, the cost per element is $|w_S|/|S|$. At each product-depth, we are only allowed to pick sets of size under a certain threshold, and we pick the ones with the lowest cost per element. It turns out that this lowest cost decreases exponentially as the depth increases, which helps us build a small formula. The decrease is captured by the Fibonacci numbers and is the reason why they emerge in our lower bound and upper bound.

Making these ideas precise requires substantial notation, and we postpone further discussion to Section 3.4.

3.3 The lower bound: Proof of Theorem 3.0.4

Fix the product-depth Δ for which we want to prove the set-multilinear formula lower bound. Define $G(i) := F(i) - 1$ for all i and $\lambda = \lfloor d^{1/G(\Delta)} \rfloor$. We can assume that $\lambda \geq 3$ because otherwise $d^{\mu(\Delta)} < 3$, and in that case, the lower bound is trivial. The lower bound we aim to prove is $n^{\Omega(d^{1/G(\Delta)})}$.

3.3.1 The sequences $\{b_m\}$, $\{c_m\}$ and the integers p, q

We first define the sequences of integers $\{b_m\}$ and $\{c_m\}$ mentioned in the proof overview:

Let $r_m := \lambda^{G(m+1)-G(m)} - 1$ for $0 \le m \le \Delta - 2$.

Define

$$b_0 := 1, \quad b_1 := \lambda \text{ and } b_m := b_{m-2} + r_{m-1} b_{m-1} \text{ for } 2 \le m \le \Delta - 2\,.$$

Define

$$c_{\Delta-2} := (-1)^{\Delta-2}, \quad c_{\Delta-3} := (-1)^{\Delta-3}\lambda^{G(\Delta-1)-G(\Delta-2)} \text{ and}$$
$$c_m := (-1)^m(|c_{m+2}| + r_{m+1}|c_{m+1}|) \text{ for } \Delta - 4 \ge m \ge 0\,.$$

Note that the sign parity of c_m is $(-1)^m$ for all m.

Thus, $c_{m-2} = (-1)^{m-2}(|c_m| + r_{m-1}|c_{m-1}|) = c_m - r_{m-1}c_{m-1}$ which implies

$$c_m = c_{m-2} + r_{m-1}c_{m-1} \text{ for } 2 \le m \le \Delta - 2\,.$$

Observe that as m increases, b_m increases and $|c_m|$ decreases.

Define

$$p := c_0 \text{ and } q := pb_1 - c_1 = c_0(r_0 + 1) - c_1\,.$$

By defining the integers p and q this way, we have ensured that $pb_0 \equiv c_0 \bmod q$ and $pb_1 \equiv c_1 \bmod q$. Hence from the relations $b_m = b_{m-2} + r_{m-1}b_{m-1}$ and $c_m = c_{m-2} + r_{m-1}c_{m-1}$, it inductively follows that

$$pb_m \equiv c_m \bmod q \quad \text{for } 0 \le m \le \Delta - 2\,. \tag{3.3.1}$$

3.3.2 Bounds on the values of b_m and $|c_m|$

Each b_m is close to $\lambda^{G(m)}$ and each $|c_m|$ is close to $\lambda^{G(\Delta-1)-G(m+1)}$:

Lemma 3.3.1. *Let $\lambda \geq 3$ be as defined in Section 5.3. Then for $0 \leq m \leq \Delta - 2$, $\frac{\lambda^{G(m)}}{2} \leq b_m \leq \lambda^{G(m)}$, and $\frac{\lambda^{G(\Delta-1)-G(m+1)}}{2} \leq |c_m| \leq \lambda^{G(\Delta-1)-G(m+1)}$.*

To prove these bounds, we use a generalized version of the well-known Bernoulli's inequality [Mit70, Section 2.4]:

Claim 3.3.2 (Bernoulli's inequality). *Let $x_1, \dots, x_r$ be real numbers all greater than -1 and all with the same sign. Then,*

$$(1 + x_1)(1 + x_2) \dots (1 + x_r) \geq 1 + x_1 + \dots + x_r \,.$$

Proof of Lemma 3.3.1. Clearly, b_m satisfies the bounds when $m = 0$ or 1. For $m \geq 2$,

$$\begin{aligned} b_m &= (\lambda^{G(m)-G(m-1)} - 1)b_{m-1} + b_{m-2} \\ &\leq \lambda^{G(m)-G(m-1)} b_{m-1} \\ &\leq \lambda^{G(m)-G(m-1)} . \lambda^{G(m-1)-G(m-2)} \dots \lambda^{G(2)-G(1)} b_1 \\ &= \lambda^{G(m)}. \end{aligned}$$

$$\begin{aligned} b_m &= (\lambda^{G(m)-G(m-1)} - 1)b_{m-1} + b_{m-2} \\ &\geq (\lambda^{G(m)-G(m-1)} - 1)b_{m-1} \\ &\geq (\lambda^{G(m)-G(m-1)} - 1).(\lambda^{G(m-1)-G(m-2)} - 1) \dots (\lambda^{G(2)-G(1)} - 1)b_1 \\ &= \lambda^{G(m)-G(1)} b_1 . \left(1 - \frac{1}{\lambda^{G(m)-G(m-1)}}\right)\left(1 - \frac{1}{\lambda^{G(m-1)-G(m-2)}}\right) \dots \left(1 - \frac{1}{\lambda^{G(2)-G(1)}}\right) \\ &\geq \lambda^{G(m)} . \left(1 - \frac{1}{\lambda^{G(m)-G(m-1)}} - \dots - \frac{1}{\lambda^{G(2)-G(1)}}\right) \text{ [by Claim 3.3.2]} \\ &\geq \lambda^{G(m)} . \left(1 - \frac{1}{\lambda^{m-1}} - \frac{1}{\lambda^{m-2}} - \dots - \frac{1}{\lambda}\right) \text{[since } G(i) \geq i\text{]} \\ &= \lambda^{G(m)} . \left(1 - \frac{1}{\lambda - 1}\left(1 - \frac{1}{\lambda^{m-1}}\right)\right) \geq \frac{\lambda^{G(m)}}{2}. \end{aligned}$$

Clearly, $|c_m|$ satisfies the bounds when $m = \Delta - 2$ or $\Delta - 3$. For $m \leq \Delta - 4$,

$$\begin{aligned}
|c_m| &= (\lambda^{G(m+2)-G(m+1)} - 1)|c_{m+1}| + |c_{m+2}| \\
&\leq \lambda^{G(m+2)-G(m+1)}|c_{m+1}| \\
&\leq \lambda^{G(m+2)-G(m+1)} \cdot \lambda^{G(m+3)-G(m+2)} \dots \lambda^{G(\Delta-2)-G(\Delta-3)}|c_{\Delta-3}| \\
&= \lambda^{G(\Delta-2)-G(m+1)} \cdot \lambda^{G(\Delta-1)-G(\Delta-2)} = \lambda^{G(\Delta-1)-G(m+1)}.
\end{aligned}$$

$$\begin{aligned}
|c_m| &= (\lambda^{G(m+2)-G(m+1)} - 1)|c_{m+1}| + |c_{m+2}| \\
&\geq (\lambda^{G(m+2)-G(m+1)} - 1)|c_{m+1}| \\
&\geq (\lambda^{G(m+2)-G(m+1)} - 1) \cdot (\lambda^{G(m+3)-G(m+2)} - 1) \dots (\lambda^{G(\Delta-2)-G(\Delta-3)} - 1)|c_{\Delta-3}| \\
&= \lambda^{G(\Delta-2)-G(m+1)}|c_{\Delta-3}| \cdot \left(1 - \frac{1}{\lambda^{G(m+2)-G(m+1)}}\right)\left(1 - \frac{1}{\lambda^{G(m+3)-G(m+2)}}\right)\dots \\
&\qquad \dots\left(1 - \frac{1}{\lambda^{G(\Delta-2)-G(\Delta-3)}}\right) \\
&\geq \lambda^{G(\Delta-2)-G(m+1)}|c_{\Delta-3}|.\left(1 - \frac{1}{\lambda^{G(m+2)-G(m+1)}} - \dots - \frac{1}{\lambda^{G(\Delta-2)-G(\Delta-3)}}\right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{[by Claim 3.3.2]} \\
&\geq \lambda^{G(\Delta-2)-G(m+1)}|c_{\Delta-3}|.\left(1 - \frac{1}{\lambda^{m+1}} - \frac{1}{\lambda^{m+2}} - \dots - \frac{1}{\lambda^{\Delta-3}}\right) \\
&= \lambda^{G(\Delta-1)-G(m+1)}.\left(1 - \frac{1}{\lambda^m(\lambda-1)}\left(1 - \frac{1}{\lambda^{\Delta-3-m}}\right)\right) \geq \frac{\lambda^{G(\Delta-1)-G(m+1)}}{2}.
\end{aligned}$$

□

3.3.3 Constructing the word and proving the lower bound

Define $\alpha = 1-p/q$. As $\dfrac{p}{q} \leq \dfrac{c_0}{c_0(r_0+1)} = 1/\lambda$, we have $\alpha \geq 1/2$. Since $q = c_0\lambda - c_1$, it implies that

$$q \leq |c_0|\lambda + |c_1| \leq 2\lambda^{G(\Delta-1)} \leq d < \lfloor \log_2 n \rfloor/2$$

where the second inequality follows from the upper bound on each $|c_m|$ in Lemma 3.3.1. Therefore, there exists a multiple of q in the interval $\left[\frac{\lfloor \log_2 n \rfloor}{2}, \lfloor \log_2 n \rfloor\right]$. Let k be this multiple of q.

Then αk is an integer. We can construct a word w over the alphabet $\{\alpha k, -k\}$ such that w is k-unbiased. This can be done using induction: if $|w_{[i]}| \leq 0$, set $w_{i+1} = \alpha k$, otherwise set $w_{i+1} = -k$.

With these definitions in place, we can prove Theorem 3.0.4. Assume the following lemma:

Lemma 3.3.3. *Let $\delta \leq \Delta$ be an integer and α, k be as defined above. Let w be any word of length d over the alphabet $\{\alpha k, -k\}$. Then any set-multilinear formula C of product-depth δ, degree $D \geq \lambda^{G(\delta)}/8$ and size at most s satisfies*

$$\mathrm{relrk}_w(C) \leq s2^{-k\lambda/256}.$$

Proof of Theorem 3.0.4. Note that by Lemma 3.1.2, there exists a set-multilinear projection P_w of $\mathrm{IMM}_{2^k,d}$ such that $\mathrm{relrk}_w(P_w) \geq 2^{-k}$. Consider a set-multilinear circuit of size s and product-depth Δ computing $\mathrm{IMM}_{n,d}$. We can expand it to a set-multilinear formula of size at most $s^{2\Delta}$ which computes the same polynomial. Hence, we will also have a set-multilinear formula of size at most $s^{2\Delta}$ computing P_w. As $d \geq \lambda^{G(\Delta)}/8$, taking the particular case of $\delta = \Delta$ in Lemma 3.3.3, we obtain $\mathrm{relrk}_w(P_w) \leq s^{2\Delta}2^{-k\lambda/256}$. This gives the desired lower bound

$$s^{2\Delta} \geq 2^{-k}2^{k\lambda/256} \geq \left(\sqrt{\frac{n}{2}}\right)^{\lfloor d^{1/G(\Delta)} \rfloor/256} / n = n^{\Omega(d^{\mu(\Delta)})}.$$

□

Proof of Lemma 3.3.3. We proceed by induction on δ. We can write $C = C_1 + \cdots + C_t$ where each C_i is a subformula of size s_i rooted at a product gate. Because of the subadditivity of relrk_w, it suffices to show that

$$\mathrm{relrk}_w(C_i) \leq s_i 2^{-k\lambda/256} \text{ for all } i.$$

Base case: C has product-depth $\delta = 1$ and degree $D \geq \lambda/8$. Then C_i is a product of linear forms. If L is linear form on some variable set $X(w_j)$, then

$\mathrm{relrk}_w(L) \leq 2^{-|w_j|/2} \leq 2^{-k/4}$. Therefore, by the multiplicativity of relrk_w,

$$\mathrm{relrk}_w(C_i) \leq 2^{-kD/4} \leq 2^{-k\lambda/32}\,.$$

Induction hypothesis: Assume that the lemma is true for all product-depths $\leq \delta - 1$.

Induction step: Let C be a formula of product-depth δ and degree $D \geq \lambda^{G(\delta)}/8$. We can write $C_i = C_{i,1} \dots C_{i,t_i}$ where each $C_{i,j}$ is a subformula of product-depth $\delta - 1$.

If C_i has a factor, say $C_{i,1}$, of degree $\geq \lambda^{G(\delta-1)}/8$, then by induction hypothesis,

$$\mathrm{relrk}_w(C_i) \leq \mathrm{relrk}_w(C_{i,1}) \leq s_i 2^{-k\lambda/256}\,.$$

Otherwise every factor of C_i has degree $< \lambda^{G(\delta-1)}/8$. Let $C_i = C_{i,1} \dots C_{i,t_i}$ where each $C_{i,j}$ has degree $D_{ij} < \lambda^{G(\delta-1)}/8$. If C_i is set-multilinear with respect to $(X_l)_{l\in S}$, then let $(S_1, \dots, S_{t_i})$ be the partition of S such that each $C_{i,j}$ is set-multilinear with respect to $(X_l)_{l\in S_j}$.

For $j \in [t_i]$, let a_{ij} be the number of positive indices in S_j. We have two cases: If $a_{ij} \leq D_{ij}/2$, then

$$w_{S_j} \leq \frac{D_{ij}}{2} \cdot \alpha k + \frac{D_{ij}}{2} \cdot (-k) = -\frac{D_{ij} p}{2q} k \leq -\frac{D_{ij} k}{4\lambda},$$

where the last inequality follows from $\frac{p}{q} \geq \frac{c_0}{2c_0(r_0+1)} = \frac{1}{2\lambda}$. The other case is $a_{ij} > D_{ij}/2$. If we can prove that $|w_{S_j}| \geq a_{ij} k/(8\lambda^{G(\delta)-1})$, then in both of the above cases, we would have $|w_{S_j}| \geq D_{ij} k/(16\lambda^{G(\delta)-1})$. By the multiplicativity and imbalance property of relrk_w and the assumption $D \geq \lambda^{G(\delta)}/8$, it would follow that

$$\mathrm{relrk}_w(C_i) \leq \prod_{j=1}^{t_i} 2^{-\frac{1}{2}|w_{S_j}|} \leq 2^{-\sum_{j=1}^{t_i} D_{ij} k/(32\lambda^{G(\delta)-1})} = 2^{-Dk/(32\lambda^{G(\delta)-1})} \leq 2^{-k\lambda/256}$$

and we would be done. Thus, we now only have to show that $|w_{S_j}| \geq a_{ij}k/(8\lambda^{G(\delta)-1})$.

$$\begin{aligned}|w_{S_j}| = |a_{ij} \cdot \alpha k + (D_{ij} - a_{ij}) \cdot (-k)| &= \left|a_{ij}\frac{p}{q} - (2a_{ij} - D_{ij})\right| k \qquad \text{as } \alpha = 1 - p/q \\ &\geq \left|\frac{a_{ij}p}{q} - \left\lfloor \frac{a_{ij}p}{q} \right\rceil\right| k \qquad \text{where } \lfloor . \rceil \text{ denotes the nearest integer.}\end{aligned}$$

The fractional part of $\frac{a_{ij}p}{q}$ is $\frac{a_{ij}p \bmod q}{q}$. Hence to prove that $|w_{S_j}| \geq a_{ij}k/(8\lambda^{G(\delta)-1})$, it is enough to verify that the following inequality is satisfied:

$$\min\left(\frac{a_{ij}p \bmod q}{q}, 1 - \frac{a_{ij}p \bmod q}{q}\right) \geq \frac{a_{ij}}{8\lambda^{G(\delta)-1}}. \tag{3.3.2}$$

Showing that the $p,\ q$ we defined satisfy the inequality in Equation (3.3.2):

We will first find what we call the base $(b_0, \ldots, b_{\Delta-2})$ representation of the number a_{ij}. For $0 \leq m \leq \Delta - 2$, inductively define y_m to be the integer quotient when $\left(a_{ij} - \sum_{m'=m+1}^{\Delta-2} b_{m'}y_{m'}\right)$ is divided by b_m. Then we can express a_{ij} as $a_{ij} = \sum_{m=0}^{\Delta-2} b_m y_m$. Since $b_m \geq \lambda^{G(m)}/2$ for all m (Lemma 3.3.1) and $a_{ij} \leq D_{ij} < \lambda^{G(\delta-1)}/8$, we have the following bounds on the values of y_m:

$$y_m = 0 \text{ for } m \geq \delta - 1, \tag{3.3.3}$$

$$y_{\delta-2} = \left\lfloor \frac{a_{ij}}{b_{\delta-2}} \right\rfloor < \frac{\frac{\lambda^{G(\delta-1)}}{8}}{\frac{\lambda^{G(\delta-2)}}{2}} \leq \frac{\lambda^{G(\delta-1)-G(\delta-2)} - 1}{2} = \frac{r_{\delta-2}}{2}, \tag{3.3.4}$$

$$y_m \leq \left\lfloor \frac{b_{m+1} - 1}{b_m} \right\rfloor = r_m \text{ for } m < \delta - 2\,. \tag{3.3.5}$$

By Equation (3.3.1), $a_{ij}p \equiv \sum_{m=0}^{\Delta-2} c_m y_m \bmod q$.

Define f to be the highest index such that $y_f \geq 1$ (by Equation (3.3.3), $f \leq \delta - 2$) and e to be the smallest index such that $y_e \geq 1$. Then $a_{ij}p \equiv \sum_{m=e}^{f} c_m y_m \bmod q$. Therefore,

$$\min\left(\frac{a_{ij}p \bmod q}{q}, 1 - \frac{a_{ij}p \bmod q}{q}\right) = \min\left(\left|\sum_{m=e}^{f} c_m y_m\right|/q,\ 1 - \left|\sum_{m=e}^{f} c_m y_m\right|/q\right) \tag{3.3.6}$$

if $\left|\sum_{m=e}^{f} c_m y_m\right|/q \le 1$, which is true by the upper bound in the following claim (See the end of this section for the proof):

Claim 3.3.4. *If $0 \le y_m \le r_m$ for all m and $y_e \ge 1$, then $|c_{2\lfloor (f-e+1)/2\rfloor+e}| \le \left|\sum\limits_{m=e}^{f} c_m y_m\right| < q - c_0$ and the sign parity of $\sum\limits_{m=e}^{f} c_m y_m$ is $(-1)^e$.*

Now, we prove a lower bound on the RHS of Equation (3.3.6). We have three cases:

- If $f < \delta - 2$, then $y_{\delta-2} = 0$ i.e. $a_{ij} < b_{\delta-2}$. By Claim 3.3.4, we have

$$\begin{aligned}\min\left(\left|\sum_{m=e}^{f} c_m y_m\right|/q,\ 1 - \left|\sum_{m=e}^{f} c_m y_m\right|/q\right) &\ge \frac{1}{q}\min(|c_{2\lfloor (f-e+1)/2\rfloor+e}|, c_0)\\ &\ge \frac{|c_{\delta-2}|}{q} > \left|\frac{c_{\delta-2}a_{ij}}{b_{\delta-2}q}\right|\end{aligned}$$

  where the second inequality follows from $2\lfloor(f-e+1)/2\rfloor + e \le f+1 \le \delta-2$.

- If $e = f = \delta - 2$, then $a_{ij} = b_{\delta-2}y_{\delta-2}$. Hence,

$$\begin{aligned}\min\left(\left|\sum_{m=e}^{f} c_m y_m\right|/q,\ 1 - \left|\sum_{m=e}^{f} c_m y_m\right|/q\right) &= \frac{|c_{\delta-2}|y_{\delta-2}}{q}\\ &= \left|\frac{c_{\delta-2}a_{ij}}{b_{\delta-2}q}\right|,\end{aligned}$$

  where the first equality is true since $|c_{\delta-2}|y_{\delta-2} \le \frac{|c_{\delta-2}|r_{\delta-2}}{2} \le \frac{c_0 r_0}{2} < q/2$.

- If $e < f = \delta - 2$, then use the bounds of Lemma 3.3.1 to see that

$$|c_{\delta-3}| > \frac{\lambda^{G(\Delta-1)-G(\delta-2)}}{2} > \frac{\lambda^{G(\Delta-1)-G(\delta-1)}}{\lambda^{G(\delta-2)}/2}\cdot\frac{\lambda^{G(\delta-1)}}{8} > \frac{|c_{\delta-2}|}{b_{\delta-2}}a_{ij}\ .$$

  Thus, Claim 3.3.4 implies that

$$1 - \left|\sum_{m=e}^{f} c_m y_m\right|/q > \frac{c_0}{q} \ge \frac{c_{\delta-3}}{q} > \left|\frac{c_{\delta-2}a_{ij}}{b_{\delta-2}q}\right|.$$

If e and f have the same parity, then

$$\left|\sum_{m=e}^{f} c_m y_m\right| / q \geq \frac{1}{q}|c_f y_f| \\ = \frac{1}{q}\left|c_{\delta-2}\left\lfloor\frac{a_{ij}}{b_{\delta-2}}\right\rfloor\right| \geq \left|\frac{c_{\delta-2}a_{ij}}{2b_{\delta-2}q}\right| ,$$

where the first inequality holds since $\sum_{m=e}^{f-1} c_m y_m$ has same sign parity as $c_f y_f$ by Claim 3.3.4. If e and f have different parity, then

$$\begin{aligned}\left|\sum_{m=e}^{f} c_m y_m\right| / q &\geq \frac{1}{q}\left(\left|\sum_{m=e}^{f-1} c_m y_m\right| - |c_f y_f|\right) \\ &\geq \frac{1}{q}\left(|c_{2\lfloor(f-e)/2\rfloor+e}| - \frac{|c_f| r_f}{2}\right) && \text{by Claim 3.3.4} \\ &> \frac{|c_{f-1}|}{2q} && \text{since } |c_{2\lfloor(f-e)/2\rfloor+e}| = |c_{f-1}| > |c_f| r_f \\ &> \left|\frac{c_{\delta-2}a_{ij}}{2b_{\delta-2}q}\right| && \text{since } f-1=\delta-3\,.\end{aligned}$$

Thus in all three cases,

$$\min\left(\left|\sum_{m=e}^{f} c_m y_m\right|/q,\ 1-\left|\sum_{m=e}^{f} c_m y_m\right|/q\right) \geq \left|\frac{c_{\delta-2}a_{ij}}{2b_{\delta-2}q}\right|.$$

By Lemma 3.3.1, we have

$$|c_{\delta-2}| \geq \lambda^{G(\Delta-1)-G(\delta-1)}/2, \quad b_{\delta-2} \leq \lambda^{G(\delta-2)}, \quad q \leq |c_0|\lambda + |c_1| \leq 2\lambda^{G(\Delta-1)}\,.$$

Hence,

$$\min\left(\left|\sum_{m=e}^{f} c_m y_m\right|/q,\ 1-\left|\sum_{m=e}^{f} c_m y_m\right|/q\right) \geq \frac{a_{ij}}{8\lambda^{G(\delta-1)+G(\delta-2)}} = \frac{a_{ij}}{8\lambda^{G(\delta)-1}},$$

which together with Equation (3.3.6) implies Equation (3.3.2). □

We now show the technical Claim 3.3.4 used in the above proof.

Proof of Claim 3.3.4. Define $c_{-1} = |c_0|r_0 + |c_1|$. Then from the definition of c_m, for all $m \geq 0$, $|c_m| r_m = |c_{m-1}| - |c_{m+1}|$. Hence, by a telescopic sum,

$$\left|\sum_{i=0}^{t} c_{a+2i} y_{a+2i}\right| \leq \sum_{i=0}^{t} |c_{a+2i}| r_{a+2i} = |c_{a-1}| - |c_{a+2t+1}| \,.$$

Consequently,

$$\begin{aligned}(-1)^e \sum_{m=e}^{f} c_m y_m &\geq |c_e| - \sum_{i=0}^{\lfloor (f-e-1)/2 \rfloor} c_{e+1+2i} y_{e+1+2i} \\ &\geq |c_e| - (|c_e| - |c_{2\lfloor (f-e-1)/2 \rfloor + e+2}|) \\ &= |c_{2\lfloor (f-e+1)/2 \rfloor + e}|\end{aligned}$$

where the first inequality holds since $y_e \geq 1$ and the sign parity of c_m is $(-1)^m$. This proves the second part and the lower bound on $\left|\sum_{m=e}^{f} c_m y_m\right|$ in the first part of the claim. As the sign parity of $\sum_{m=e}^{f} c_m y_m$ is $(-1)^e$, we also have

$$\left|\sum_{m=e}^{f} c_m y_m\right| \leq \left|\sum_{i=0}^{\lfloor (f-e)/2 \rfloor} c_{e+2i} y_{e+2i}\right| \leq |c_{e-1}| - |c_{2\lfloor (f-e)/2 \rfloor + e+1}| \leq q - c_0 \,.$$

□

3.4 Limitations on improving the bounds: Proof of Theorem 3.4.1

Theorem 3.4.1. *(Barrier) Let $s_1, \ldots, s_\gamma$ be positive integers. Fix sets $X_1, \ldots, X_d$ where for all i, $|X_i| \in \{s_1, \ldots, s_\gamma\}$. For any fixed positive integer Δ, there exist polynomials P_Δ and Q_Δ that are set-multilinear with respect to $X_1, \ldots, X_d$ such that P_Δ can be computed by product-depth Δ circuits of size $n^{O\left(\Delta \gamma d^{\mu(\Delta)}\right)}$ and Q_Δ can be computed by product-depth Δ circuits of size $n^{O\left(\Delta d^{\mu(\Delta-1)} + \gamma\right)}$. Moreover, P_Δ and Q_Δ maximize the measure used to prove lower bounds.*

Remark 3.4.2. The two different polynomials with slightly different sizes will imply barriers to improving the lower bound in different regimes of γ. Suppose that $\Delta = O(1)$ is small. When $\gamma = O(1)$, the size of P_Δ matches our lower bound, essentially implying its tightness. When γ is $d^{o(1)}$, the size of Q_Δ is only slightly larger than our lower bound (note $\mu(\Delta-1)$ vs $\mu(\Delta)$). Hence, even when multiple set sizes are used, the scope for improvement is tiny.

We will show here that the techniques of [LST21] cannot hope to prove much stronger lower bounds. We do this by constructing polynomials for which the lower bound we proved earlier is tight. We begin by showing this in the case of two different set sizes. We can normalize with respect to the bigger set size to assume that the weights are $-k$ and αk ($\alpha \in [0,1]$) without loss of generality. Clearly, $k \leq \log n$.

Lemma 3.4.3. *Let n, d, Δ be such that $d \leq n$. For any $\alpha \in [0,1]$ let $w \in \{-k, \alpha k\}^d$ be a word with $|w_{[d]}| \leq k$. There is a polynomial $P_\Delta \in \mathbb{F}_{sm}[\overline{X}(w)]$ which is computable by a set-multilinear formula of product-depth at most Δ, size at most $n^{O\left(\Delta d^{\mu(\Delta)}\right)}$ and has the maximum possible relative rank.*

Remark 3.4.4. We can replace αk with $\lfloor \alpha k \rfloor$ and assume that the weights in w are integers. It can be shown that this will not change the arguments in any significant way (see Claim 3.4.10).

We will need the extensive notation from [LST21], which we restate here.

3.4.1 Notation

- As in Section 3.1 and from the remark above, we assume $|X(w_i)| = 2^{|w_i|}$ and that the variables are indexed by binary strings $\{0,1\}^{|w_i|}$.
- Given any subset $S \subseteq [d]$, we denote by $S_+ = \{i \in S \mid w_i > 0\}$ the positive indices of S and similarly by S_-, the negative indices.
- We let $K = \sum_{i\in[d]} |w_i|$, $k_+ = \sum_{i\in S_+} |w_i|$ and $k_- = \sum_{i \in S_-} |w_i|$. We say S is $\mathcal{P}$-heavy if $k_+ \geq k_-$ and $\mathcal{N}$-heavy otherwise.

- Setting $I = [K]$, we partition the set $I = I_1 \cup \cdots \cup I_d$ where I_j is an interval of length $|w_j|$ that starts at $\left(\sum_{i<j} |w_j|\right) + 1$. Given a $T \subseteq [d]$, we let $I(T) = \bigcup_{j \in T} I_j$.
- Let $m = m_+ m_- \in \mathcal{M}_w^S$ be a monomial supported on variable sets indexed by S, with $m_+ \in \mathcal{M}_w^{S_+}$ and $m_- \in \mathcal{M}_w^{S_-}$. The Boolean string $\sigma(m_+)$ associated with the positive monomial (as defined in Section 3.1) can be thought of as a labeling of the elements of $I(S_+)$ in the natural way - $\sigma(m_+) : I(S_+) \to \{0, 1\}$. Similarly, for $\sigma(m_-)$.

Given a set S, we define a sequence of polynomials that we will later show to have set multilinear formulas of small size but large rank.

Fix $J_+ \subseteq I(S_+)$ and $J_- \subseteq I(S_-)$ such that $|J_+| = |J_-| = \min\{k_+, k_-\}$. Let π be a bijection from J_+ to J_-. Such a tuple (S, J_+, J_-, π) is called valid. Fix a valid (S, J_+, J_-, π).

A string $\tau \in \{0, 1\}^{|k_+ - k_-|}$ defines a map $I(S_+) \setminus J_+ \to \{0, 1\}$ if S is $\mathcal{P}$-heavy and a map $I(S_-) \setminus J_- \to \{0, 1\}$ if S is $\mathcal{N}$-heavy.

The polynomial $P_{(S, J_+, J_-, \pi, \tau)}$ is the sum of all monomials m such that

1. $\sigma(m_+)(j) = \sigma(m_-)(\pi(j))$ for all $j \in J_+$, and
2. $\sigma(m_+)(j) = \tau(j)$ for all $j \in I(S_+) \setminus J_+$ if S is $\mathcal{P}$-heavy or $\sigma(m_-)(j) = \tau(j)$ for all $j \in I(S_-) \setminus J_-$ if S is $\mathcal{N}$-heavy.

As observed in [LST21], these polynomials have desirable properties that help build formulas for them inductively.

(P1) For any valid (S, J_+, J_-, π) and any $\tau \in \{0, 1\}^{|k_+ - k_-|}$ the matrix $M_{w_{|S}}(P_{(S, J_+, J_-, \pi, \tau)})$ has the maximum possible rank for a matrix with its dimensions:

$$\mathrm{rank}(M_{w_{|S}}(P_{(S, J_+, J_-, \pi, \tau)})) = \min\{| \mathcal{M}_w^{S_+} |, | \mathcal{M}_w^{S_-} |\} = 2^{\min\{k_+, k_-\}}$$

(P2) Let $(S_i, J_{i,+}, J_{i,-}, \pi_i)$ $(i \in [r])$ be valid tuples with $S_i (i \in [r])$ being all $\mathcal{P}$-heavy and pairwise disjoint. Also assume that we have $\tau_i \in \{0, 1\}^{k_{i,+} - k_{i,-}}$ where

$k_{i,+} = \sum_{j \in I(S_{i,+})} w_j$. We can construct a new polynomial using these. Let $S = \bigcup_i S_i$ (also $\mathcal{P}$-heavy by definition), $J_+ = \bigcup_i J_{i,+}$, $J_- = \bigcup_i J_{i,-}$, $\pi = \bigcup_i \pi_i$ and $\tau = \bigcup_i \tau_i$. Then, (S, J_+, J_-, π) is a valid tuple and moreover

$$P_{(S,J_+,J_-,\pi,\tau)} = \prod_{i=1}^{r} P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)}$$

If each S_i is $\mathcal{N}$-heavy, an analogous fact can be shown to hold.

(P3) Say S', S'' are disjoint sets where S' is $\mathcal{P}$-heavy and S'' is $\mathcal{N}$-heavy. Also fix any valid (S', J'_+, J'_-, π') and $(S'', J''_+, J''_-, \pi'')$.

Assume that $S = S' \cup S''$ is $\mathcal{P}$-heavy. Let $J_- = I(S_-)$ and $J_+ = J'_+ \cup J''_+ \cup J'''$ where $J''' \subseteq I(S'_+)$ is any set of size $|I(S''_-)| - |I(S''_+)|$ disjoint from $J'_+ \cup J''_+$ (As S is $\mathcal{P}$-heavy, a set like this exists). Fix any bijection $\pi''' : J''' \to I(S''_-) \setminus J''_-$ Assume $\pi : J_+ \to J_-$ is defined to be $(\pi \cup \pi'' \cup \pi''')(j)$ for $j \in J'_+ \cup J''_+ \cup J'''$

Also, fix any $\tau : I(S_+) \setminus J_+ \to \{0,1\}$. Any $\tau' : I(S'_+) \setminus J'_+ \to \{0,1\}$ is said to extend τ if τ' restricts to τ on the set $I(S_+) \setminus J_+$ (note that J_+ contains $J''_+ = I(S''_+)$ and hence $I(S_+) \setminus J_+ \subseteq I(S'_+) \setminus J'_+$, so this definition makes sense). We denote by $\tau' \setminus \tau$ the restriction of τ' to the set J'''. We thus obtain

$$P_{(S,J_+,J_-,\pi,\tau)} = \sum_{\tau' \text{ extends } \tau} P_{(S',J'_+,J'_-,\pi',\tau')} \cdot P_{(S'',J''_+,J''_-,\pi'',(\tau'\setminus\tau)\circ\pi'''^{-1})}$$

The size of this sum is $2^{|J'''|} = 2^{k''_- - k''_+}$. An analogous identity holds when S is $\mathcal{N}$-heavy.

3.4.2 Cost of building formulas

To proceed, we introduce a few notions that help make the ideas in the proof overview (of Section 3.2.2) precise. We will only consider the case where $|w_{[d]}| \leq k$, i.e., $||\mathcal{P}_w|\alpha - |\mathcal{N}_w|| \leq 1$. Fix Δ as in Lemma 3.4.3, and recall that $\lfloor r \rceil$ denotes the nearest integer to the real number r.

Definition 3.4.5 (Fractional Cost)**.** Set $\mathrm{fc}(0) = 1$ and for $1 \le \delta \le \Delta - 1$,

$$\mathrm{fc}(\delta) := \min_{q < d^{\mu(\Delta)}/\mathrm{fc}(\delta-1)} |q\alpha - \lfloor q\alpha \rceil|/q,$$

where $q \in \mathbb{N}$ is a natural number. In case $\mathrm{fc}(\delta') = 0$ for some $\delta' \le \Delta - 1$, we set $\mathrm{fc}(\delta) = 0$ for all $\delta > \delta'$ as well. Let $\hat{\Delta} \le \Delta - 1$ be the largest integer such that $\mathrm{fc}(\hat{\Delta}) \ne 0$.

For $1 \le \delta \le \hat{\Delta}$, we denote by p_δ the (least) value of q for which the expression for $\mathrm{fc}(\delta)$ attains the minimum. Note that, by definition,

$$p_\delta \le d^{\mu(\Delta)}/\mathrm{fc}(\delta - 1). \tag{3.4.1}$$

We also denote by $n_\delta := \lfloor p_\delta \alpha \rceil$ the nearest integer to $p_\delta\alpha$. We first observe that the fractional cost falls exponentially with depth.

Claim 3.4.6 (Exponential decline)**.** *For* $0 \le \delta \le \Delta - 1$,

$$\mathrm{fc}(\delta) \le 1/(d^{\mu(\Delta)})^{F(\delta+1)-2}.$$

Proof of Claim 3.4.6. Note that when $\delta = 0$, the claim holds since $1 = \mathrm{fc}(0) \le 1/(d^{\mu(\Delta)})^{F(1)-2} = 1$. The claim also holds trivially if $\mathrm{fc}(\delta) = 0$. To prove the claim when $1 \le \delta \le \hat{\Delta}$, we will use Dirichlet's approximation principle ([Sch91, Theorem 1A]), which essentially implies that for any real numbers α, and $N \ge 1$, there exists an integer $q \le N$ such that the distance from $q\alpha$ to the nearest integer is bounded by $1/N$. Consequently, we get that there exists an integer $q' \le d^{\mu(\Delta)}/\mathrm{fc}(\delta - 1)$ such that

$$|q'\alpha - \lfloor q'\alpha \rceil| < \mathrm{fc}(\delta - 1)/d^{\mu(\Delta)}\,. \tag{3.4.2}$$

When $\delta = 1$, the claim now follows from the definition since

$$\mathrm{fc}(1) = \min_{q < d^{\mu(\Delta)}/\mathrm{fc}(0)} |q\alpha - \lfloor q\alpha \rceil|/q \le |q'\alpha - \lfloor q'\alpha \rceil|/q' < 1/d^{\mu(\Delta)},$$

where we used the fact that $q' \geq 1$.

When $\delta \geq 2$, we claim that the q' obtained in Equation (3.4.2) isn't too small:

$$q' \geq d^{\mu(\Delta)}/fc(\delta-2)\,. \tag{3.4.3}$$

Indeed, if not, then

$$\begin{aligned} fc(\delta-1) &= \min_{q<d^{\mu(\Delta)}/fc(\delta-2)} \frac{|q\alpha - \lfloor q\alpha \rceil|}{q} \\ &\leq fc(\delta-1)/d^{\mu(\Delta)}. \qquad \text{from Equation (3.4.2), and } q' \text{ is now a candidate} \end{aligned}$$

This leads to a contradiction since $d^{\mu(\Delta)} > 1$. Hence, Equation (3.4.3) holds, and we obtain the following bound on $fc(\delta)$ using Equation (3.4.2) and Equation (3.4.3):

$$fc(\delta) \leq \frac{|q'\alpha - \lfloor q'\alpha \rceil|}{q'} \leq \frac{fc(\delta-1)}{d^{\mu(\Delta)}} \cdot \frac{fc(\delta-2)}{d^{\mu(\Delta)}}\,. \tag{3.4.4}$$

Solving Equation (3.4.4) readily gives

$$fc(\delta) \leq \frac{1}{d^{f(\delta)\mu(\Delta)}} \qquad \text{where } f(i) \geq f(i-1) + f(i-2) + 2\,. \tag{3.4.5}$$

Rearranging, we have, $f(i) + 1 \geq (f(i-1)+1) + (f(i-2)+1) + 1$ whence we see that setting $f(i-1)+1 := F(i) - 1$ satisfies the required constraints and proves the claim. □

We make an additional useful observation. Recall that $|\mathcal{P}_w|$ is the total number of positive sets of variables, and $|\mathcal{N}_w|$ is the total number of negative sets. Now, $\mu(\Delta) = \frac{1}{F(\Delta)-1}$ implies $\mu(\Delta)f(\Delta) = 1 + \mu(\Delta)$. Combining this with Claim 3.4.6, we get $fc(\Delta-1) \leq 1/d^{1-\mu(\Delta)} = d^{\mu(\Delta)}/d$. Further noting that $d \geq |\mathcal{P}_w|$, we find

$$fc(\Delta-1) \leq d^{\mu(\Delta)}/|\mathcal{P}_w|. \tag{3.4.6}$$

Let Δ' be the smallest integer for which $fc(\Delta') \leq d^{\mu(\Delta)}/|\mathcal{P}_w|$ holds (such a Δ' exists and is bounded above by $\Delta - 1$ from Equation (3.4.6)). In fact, note that $\Delta' \leq \hat{\Delta} + 1$ since $\mathrm{fc}(\hat{\Delta} + 1) = 0$. We will now (re)define $p_{\Delta'+1} := |\mathcal{P}_w|$ and $n_{\Delta'+1} := |\mathcal{N}_w|$.

With the notation in place, we can now state the following central claim that constructs the polynomial needed for Lemma 3.4.3:

Claim 3.4.7. *Let Δ, Δ' be as fixed above. For any integer $\delta \leq \Delta' + 1$, if $S \subseteq [d]$ satisfies $|w_S| \leq k$, $|S_+| \leq p_\delta$ and $|S_-| \leq n_\delta$, then there exist J_+, J_-, π such that (S, J_+, J_-, π) is valid and for all $\tau \in \{0,1\}^{|k_+ - k_-|}$, the polynomial $P_{(S,J_+,J_-,\pi,\tau)}$ can be computed by a set-multilinear formula of product-depth δ and size at most $|S|^\delta 2^{5k\delta d^{\mu(\Delta)}}$.*

We finish the proof of Lemma 3.4.3 assuming the above claim:

Proof of Lemma 3.4.3. We know that $|w_{[d]}| \leq k$. Recall that $p_{\Delta'+1} = |\mathcal{P}_w|$ and $n_{\Delta'+1} = |\mathcal{N}_w|$. We can now apply Claim 3.4.7 with $S = [d]$ and $\delta = \Delta' + 1$. This gives a polynomial $P_{\Delta'+1} \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$ with $\mathrm{relrk}_w(P_{\Delta'+1}) = 2^{-|w_{[d]}|/2}$. The polynomial $P_{\Delta'+1}$ is computable by a set-multilinear formula of product-depth at most $\Delta'+1 \leq \Delta$, and size at most $d^\Delta 2^{5k\Delta d^{\mu(\Delta)}} \leq n^{O\left(\Delta d^{\mu(\Delta)}\right)}$. $\square$

The following claim is the main technical result that helps prove Claim 3.4.7. It is in the same spirit as [LST21, Claim 28], but we show the existence of a better partition with a more careful analysis. Furthermore, our analysis holds for any $\alpha \in [0,1]$.

Claim 3.4.8. *Fix $1 \leq \delta \leq \Delta' + 1$. Let $S \subseteq [d]$ with $|w_S| \leq k$ such that $|S_+| \leq p_\delta$ and $|S_-| \leq n_\delta$. Then there exists a partition of S as $S_1 \cup S_2 \cup \ldots S_r$ where the following conditions hold:*

1. *$|S_{i,+}| \leq p_{\delta-1}$ and $|S_{i,-}| \leq n_{\delta-1}$ for all $i \in [r]$.*
2. *$\sum_{i=1}^r |w_{S_i}| \leq 5kd^{\mu(\Delta)}$.*
3. *$|w_{S_i}| \leq k$ for all $i \in [r]$.*

Proof of Claim 3.4.8. As long as possible, pick sets S_i with $|S_{i,+}| = p_{\delta-1}$ positive indices and $|S_{i,-}| = n_{\delta-1}$ negative indices. For all such sets picked, we have

$$|w_{S_i}| = \left|\sum\nolimits_{j\in S_i} w_j\right| = k \cdot |p_{\delta-1}\alpha - n_{\delta-1}| \leq k\,. \tag{3.4.7}$$

Suppose that the sets chosen after the procedure are $S_1, \dots, S_m$, where $m = \min\left\{\left\lfloor \frac{|S_+|}{p_{\delta-1}}\right\rfloor, \left\lfloor \frac{|S_-|}{n_{\delta-1}}\right\rfloor\right\}$ and we are left with the set S'. Since we cannot pick the sets anymore, we must have that $|S'_+| < p_{\delta-1}$ or $|S'_-| < n_{\delta-1}$ (or both). We will have to deal with two cases to pick the next sets.

Case 1: $\alpha|S'_+| \leq |S'_-|$.

We pick a set S_{m+1} with $|S'_+|$ positive indices and $b = \lfloor \alpha|S'_+|\rceil$ negative indices (notice that $b \leq |S'_-|$) so that

$$|w_{S_{m+1}}| = k\,|\alpha|S'_+| - b| \leq k. \tag{3.4.8}$$

Note that if $|S'_+| > p_{\delta-1}$, then $|S'_-| \geq \lfloor \alpha|S'_+|\rceil \geq \lfloor \alpha p_{\delta-1}\rceil = n_{\delta-1}$ which contradicts with the fact that either $|S'_+| < p_{\delta-1}$ or $|S'_-| < n_{\delta-1}$. Therefore, we have $|S'_+| \leq p_{\delta-1}$ and $b = \lfloor \alpha|S'_+|\rceil \leq \lfloor \alpha p_{\delta-1}\rceil = n_{\delta-1}$ which ensures that condition (1) is satisfied for $i = m+1$.

The remaining set $T = S' \setminus S_{m+1}$ has only negative values which we split into singletons $S_{m+2}, \dots, S_r$ (there are $(|S'_-| - b)$ of these sets). As these are singletons, for $m+2 \leq j \leq r$, we trivially have $|w_{S_j}| \leq k$.

We also note that

$$\begin{aligned}
|S'_-| - b &= (|S'_-| - \alpha|S'_+|) + (\alpha|S'_+| - b)\\
&= (|S_-| - m\cdot n_{\delta-1} - \alpha|S_+| + \alpha m\cdot p_{\delta-1}) + (\alpha|S'_+| - b)\\
&\leq ||S_-| - \alpha|S_+|| + m|p_{\delta-1}\alpha - n_{\delta-1}| + |\alpha|S'_+| - b|
\end{aligned}$$

where the first term is at most 1 since $|w_S| \leq k$ and the last term is at most 1 as well. Putting it all together,

$$\sum_{i=m+2}^{r} |w_{S_i}| = (|S'_-| - b)k \leq (m|p_{\delta-1}\alpha - n_{\delta-1}| + 2)k.$$

Case 2: $\alpha|S'_+| > |S'_-|$.

Observe that if $|S'_-| > n_{\delta-1}$, then we must have $|S'_+| < p_{\delta-1}$ implying $|S'_-| \leq \lfloor \alpha|S'_+| \rceil \leq \lfloor \alpha p_{\delta-1} \rceil = n_{\delta-1}$. This is a contradiction. Therefore, we have $|S'_-| \leq n_{\delta-1}$.

Since $|S'_-| \leq n_{\delta-1} = \lfloor \alpha p_{\delta-1} \rceil$ and $|S'_-| \leq \lfloor \alpha|S'_+| \rceil$, there exists $c \leq \min(p_{\delta-1}, |S'_+|)$ such that $\lfloor \alpha c \rceil = |S'_-|$. Pick a set S_{m+1} with $|S'_-|$ negative indices and c positive indices so that

$$|w_{S_{m+1}}| = k|\alpha c - |S'_-|| \leq k\,.$$

Condition (1) is clearly satisfied for $i = m+1$.

The remaining set $T = S' \setminus S_{m+1}$ has only positive values which we split into singletons $S_{m+2}, \ldots, S_r$ (there are $(|S'_+| - c)$ of these sets). As these are singletons, for $m+2 \leq j \leq r$, we trivially have $|w_{S_j}| \leq k$.

Similar to the earlier case,

$$\begin{aligned}\sum_{i=m+2}^{r} |w_{S_i}| &= (|S'_+| - c)\alpha k \\ &= ((\alpha|S'_+| - |S'_-|) + (|S'_-| - \alpha c))\,k \\ &\leq (|\alpha|S_+| - |S_-|| + m|p_{\delta-1}\alpha - n_{\delta-1}| + ||S'_-| - \alpha c|)\,k \\ &\leq (m|p_{\delta-1}\alpha - n_{\delta-1}| + 2)k\,.\end{aligned}$$

Therefore, in both of the above cases,

$$\begin{aligned}\sum_{i=1}^{r} |w_{S_i}| &= \sum_{i=1}^{m} |w_{S_i}| + |w_{S_{m+1}}| + \sum_{i=m+2}^{r} |w_{S_i}| \\ &\leq km|p_{\delta-1}\alpha - n_{\delta-1}| + k + (m|p_{\delta-1}\alpha - n_{\delta-1}| + 2)k \\ &\leq k\left(2\left\lfloor \frac{|S_+|}{p_{\delta-1}} \right\rfloor |p_{\delta-1}\alpha - n_{\delta-1}| + 3\right) \leq k\left(2|S_+|\frac{|p_{\delta-1}\alpha - n_{\delta-1}|}{p_{\delta-1}} + 3\right) \\ &\leq k\,(2p_\delta \cdot fc(\delta-1) + 3) \qquad \text{(By definition of fc)} \\ &\leq 5kd^{\mu(\Delta)}\end{aligned}$$

where the last inequality is true because $fc(\delta - 1) \leq d^{\mu(\Delta)}/p_\delta$ holds for $\delta \leq \Delta'$ from Equation (3.4.1); it also holds for $\delta = \Delta' + 1$ by the definition of Δ'. $\square$

Armed with all this, the proof of Claim 3.4.7 becomes quite similar to the proof of Claim 27 in [LST21].

Proof of Claim 3.4.7. The proof is by induction on the product-depth δ for all $1 \le \delta \le \Delta' + 1$ where $\Delta' + 1$ is as defined above.

- **Base case**: When $\delta = 1$, we use the trivial expression for $P_{(S,J_+,J_-,\pi,\tau)}$ as a sum of monomials. This is a product-depth one $\sum\prod$ set-multilinear formula of size at most $2^{k|S|} \le 2^{k(p_1+n_1)}$. Note that since $|w_S| \le k$, $|p_1\alpha k - n_1 k| \le k$. This gives $n_1 \le p_1\alpha + 1 \le p_1 + 1$. Using the bound $p_1 \le d^{\mu(\Delta)}$ from Equation (3.4.1), we obtain a size bound of $2^{2k(d^{\mu(\Delta)}+1)} \le |S| 2^{5kd^{\mu(\Delta)}}$, as required.

- **Induction step**: Consider some $\delta > 1$. Let $k_+ := |I(S_+)|$ and $k_- := |I(S_-)|$. Without loss of generality, we can assume S is $\mathcal{P}$-heavy. We know that $|w_S| \le k$, $|S_+| \le p_\delta$ and $|S_-| \le n_\delta$. Thus, using Claim 3.4.8, we obtain a partition of $S = S_1 \cup \ldots \cup S_r$ where for all $i \in [r]$, we have $|w_{S_i}| \le k$, $|S_{i,+}| \le p_{\delta-1}$, $|S_{i,-}| \le n_{\delta-1}$ and
$$\sum_{i=1}^{r} |w_{S_i}| \le 5kd^{\mu(\Delta)} . \tag{3.4.9}$$
By induction hypothesis, for every $i \in [r]$, there exist $J_{i,+}, J_{i,-}, \pi_i$ such that $(S_i, J_{i,+}, J_{i,-}, \pi_i)$ is valid and for each $\tau_i \in \{0,1\}^{|k_{i,+} - k_{i,-}|}$, the polynomial $P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)}$ has a set-multilinear formula F_{i,τ_i} of product-depth $\delta - 1$ and size $s_i \le |S_i|^{\delta-1} 2^{5k(\delta-1)d^{\mu(\Delta)}}$.

  We can assume that $S_1, \ldots, S_\gamma$ are $\mathcal{P}$-heavy and $S_{\gamma+1}, \ldots, S_r$ are $\mathcal{N}$-heavy where $\gamma \in [r]$. Using (P2) above, we get that
$$P_{(S',J'_+,J'_-,\pi',\tau')} = \prod_{i=1}^{\gamma} P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)} \ , \ P_{(S'',J''_+,J''_-,\pi'',\tau'')} = \prod_{i=\gamma+1}^{r} P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)} \tag{3.4.10}$$

where

$$(S', J'_+, J'_-, \pi') = \left(\bigcup_{i \in [\gamma]} S_i, \bigcup_{i \in [\gamma]} J_{i,+}, \bigcup_{i \in [\gamma]} J_{i,-}, \bigcup_{i \in [\gamma]} \pi_i \right),$$
$$(S'', J''_+, J''_-, \pi'') = \left(\bigcup_{i=\gamma+1}^{r} S_i, \bigcup_{i=\gamma+1}^{r} J_{i,+}, \bigcup_{i=\gamma+1}^{r} J_{i,-}, \bigcup_{i=\gamma+1}^{r} \pi_i \right)$$

and for $i \in [\gamma]$, each τ_i is a restriction of τ' to $I(S_{i,+}) \setminus J_{i,+}$ whereas for $i \in \{\gamma+1, \ldots, r\}$, each τ_i is a restriction of τ'' to $I(S_{i,-}) \setminus J_{i,+}$.

Note that both these tuples are valid and S' is $\mathcal{P}$-heavy and S'' is $\mathcal{N}$-heavy. Then, using (P3), we construct the polynomial

$$\begin{aligned} P_{(S,J_+,J_-,\pi,\tau)} &= \sum_{\tau' \text{ extends } \tau} P_{(S',J'_+,J'_-,\pi',\tau')} \cdot P_{(S'',J''_+,J''_-,\pi'',\tau'')} \\ &= \sum_{\tau' \text{ extends } \tau} \prod_{i=1}^{r} P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)} \end{aligned} \tag{3.4.11}$$

where (S', J'_+, J'_-, π') and $(S'', J''_+, J''_-, \pi'')$ are constructed as in (P3). We can now use the formulas F_{i,τ_i} we had before from induction and construct a set-multilinear product-depth δ formula for $P_{(S,J_+,J_-,\pi,\tau)}$ of size at most

$$\begin{aligned} r \cdot 2^{|k''_- - k''_+|} \cdot \max_{i \in [r]} s_i &\le |S| \cdot 2^{\sum_i |w_{S_i}|} \cdot |S_i|^{\delta-1} 2^{5k(\delta-1)d^{\mu(\Delta)}} \\ &\le |S| \cdot 2^{5kd^{\mu(\Delta)}} \cdot |S|^{\delta-1} 2^{5k(\delta-1)d^{\mu(\Delta)}} \\ &\le |S|^{\delta} 2^{5k\delta d^{\mu(\Delta)}} \end{aligned} \tag{3.4.12}$$

where the second inequality follows from Equation (3.4.9).

□

3.4.3 Handling more than two weights

With multiple weights, we partition the index set $[d]$ into sets $\{S_i\}$ such that the sub-word indexed by each S_i contains at most two distinct weights. This allows us to reduce the case of multiple weights to that of two weights, for which the machinery we built in the previous section can be used to prove upper bounds. We start by describing this reduction.

Lemma 3.4.9. *Let* $w \in \{\alpha_1, \ldots, \alpha_\gamma\}^d$ ($|\alpha_i| \leq k$ *for all* i) *be a word with* $\gamma \leq d$ *different weights and* $|w_{[d]}| \leq k$. *Then, the index set* $[d]$ *can be partitioned as* $S_1 \cup \ldots \cup S_\eta$ *with* $\eta \leq 6\gamma$ *such that for all* $i \in [\eta]$, *the sub-word* $w_{|S_i}$ *has at most two distinct weights and further,* $|w_{S_i}| \leq k$.

Proof. Let $\{T_1, \ldots, T_\gamma\}$ be a partition of $[d]$ where every set T_j in the partition corresponds to one weight (i.e., for every $I \in T_j$, $w_i = \alpha_j$). We give an algorithm to obtain the desired partition of $[d]$. The basic idea is to take two distinct weights and group as many buckets corresponding to these set sizes as possible while maintaining the constraint on the sum of weights.

1. Initialize $j = 1$ and $\pi := \{T_1, \ldots, T_\gamma\}$. Repeat the following steps until π is empty.

2. If possible, pick sets T_p and T_n from π such that α_p is positive and α_n is negative.

3. If $|T_p|\alpha_p + |T_n|\alpha_n \leq 0$, then it is easy to see that we can pick a subset $T'_n \subseteq T_n$ such that $\Big||T_p|\alpha_p + |T'_n|\alpha_n\Big| \leq k$ since $|\alpha_p|, |\alpha_n| \leq k$.

4. Set $S_j := T_p \cup T'_n$. We have $|w_{S_j}| = \Big||T_p|\alpha_p + |T'_n|\alpha_n\Big| \leq k$ as required. Set $T_n := T_n \setminus T'_n$. Drop T_p from π. If $|T_p|\alpha_p + |T_n|\alpha_n \geq 0$, proceed in a similar way.

5. If we can't pick two sets T_p and T_n as above, it means that for the remaining sets in π, either their corresponding weights are all positive or all negative. We consider the case when they are all positive (the other case can be dealt with analogously).

   (a) If there exists a set T_p such that $|T_p|\alpha_p \leq k$, then set $S_j := T_p$ and drop T_p from π.

(b) Otherwise, consider any remaining set T_p. We have $|T_p|\alpha_p > k$. Since $\alpha_p \leq k$, there exist $T'_p \subseteq T'_p \cup \{q\} \subseteq T_p$ such that $|T'_p|\alpha_p \leq k$ and $(|T'_p| + 1)\alpha_p > k$. Set $S_j := T'_p$, $S_{j+1} = \{q\}$ and $T_p := T_p \setminus (T'_p \cup \{q\})$. Increment $j = j + 1$.

6. Increment $j = j + 1$ and continue.

We have ensured that $|w_{S_i}| \leq k$ for all i. It suffices to show that the steps 2-6 are repeated at most 3γ times. Every time step 4 or step 5.a is executed, the size of π reduces by at least 1. Hence, they can be repeated at most γ times in total. When step 5.b is executed for the first time, we know that the remaining collection of sets is $\pi = \{T_1, \ldots, T_\beta\}$ where each T_j corresponds to a positive weight. Let us denote the weight of this collection by $w_\pi = \sum_{j=1}^{\beta} w_{T_j} = \sum_{j=1}^{\beta} |T_j|\alpha_j$. Suppose till now we have picked the sets $S_1, \ldots, S_{\beta'}$ for some $\beta' \leq \gamma$. Then $w_\pi = w_S - \sum_{i=1}^{\beta'} w_{S_i}$. Using the triangle inequality, $w_\pi \leq |w_S| + \sum_{i=1}^{\beta'} |w_{S_i}| \leq k + \gamma k$. Every time we remove two sets $S_j = T'_p$ and $S_{j+1} = \{q\}$ as in step 5.b, the value of w_π reduces by $(|T'_p| + 1)\alpha_p > k$. Hence, this can be repeated at most $\gamma + 1$ times. $\square$

We can now construct polynomials with small set-multilinear formula size but large rank, even when the number of distinct set sizes is not two. We construct two different polynomials that are useful in different regimes of the number of set sizes (see Remark 3.4.2).

Proof of Theorem 3.4.1. As $|w_{[d]}| \leq k$, by Lemma 3.4.9, we get a partition of the index set $[d]$ into sets $S_1, \ldots, S_\eta$ ($\eta \leq 6\gamma$) such that the sub-word corresponding to each $S_i \subseteq [d]$ contains at most two weights and $|w_{S_i}| \leq k$.

- **Constructing** P_Δ: Coresponding to each set S_i, we have a size parameter α_i and the corresponding fractional cost function fc_i. As in Section 3.4.2, we also have a $\Delta'_i \leq \Delta - 1$ and sequences $\{p^i_\delta\}_{\delta=0}^{\Delta-1}$ and $\{n^i_\delta\}_{\delta=0}^{\Delta-1}$ with $p_{\Delta'_i+1} = |S_{i,+}|$ and $n_{\Delta'_i+1} = |S_{i,-}|$.

  We apply Claim 3.4.8 to each S_i to get a partition $S_i = S_{i1} \cup \ldots \cup S_{ir_i}$, with $|S_{ij,+}| \leq p_{\Delta'_i}, |S_{ij,-}| \leq n_{\Delta'_i}$ and $|w_{S_{ij}}| \leq k$ for all $j \in [r_i]$. Moreover,

$\sum_{j\in[r_i]}|w_{S_{ij}}| \leq 5k|S_i|^{\mu(\Delta)}$. Applying Claim 3.4.7 to each of the S_{ij}s, we get that there exist $J_{ij,+}, J_{ij,-}, \pi_{ij}$ such that $(S_{ij}, J_{ij,+}, J_{ij,-}, \pi_{ij})$ is valid, and for all $\tau_{ij} \in \{0,1\}^{|k_+-k_-|}$, the polynomial $P_{(S_{ij},J_{ij,+},J_{ij,-},\pi_{ij},\tau_{ij})}$ can be computed by a set-multilinear formula of depth at most $\Delta-1$ (all the Δ_i's are at most $\Delta-1$) and size

$$s_{ij} \leq |S_{ij}|^{\Delta-1}2^{5k(\Delta-1)|S_i|^{\mu(\Delta)}} \leq |S_{ij}|^{\Delta-1}2^{5k(\Delta-1)d^{\mu(\Delta)}}.$$

We club all the $\mathcal{P}$-heavy sets together, and all the $\mathcal{N}$-heavy sets together across *all* the S_is. Now, using the exact same construction (which we skip) as in the induction part of the proof of Claim 3.4.7, we obtain a polynomial P_Δ of product-depth at most Δ and size at most

$$\begin{aligned}\sum_i r_i \cdot 2^{k_-''-k_+''} \cdot \max_{i\in[\eta],j\in[r_i]} s_{ij} &\leq d \cdot 2^{\sum_{i\in[\eta],j\in[r_i]}|w_{S_{i,j}}|} \cdot d^{\Delta-1}2^{5k(\Delta-1)d^{\mu(\Delta)}}\\ &\leq d^\Delta 2^{30k\gamma\Delta d^{\mu(\Delta)}} \leq n^{O\left(\gamma\Delta d^{\mu(\Delta)}\right)}.\end{aligned}$$

- **Constructing** Q_Δ: Once we have the sets $S_1\ldots,S_\eta$ with $|w_{S_i}| \leq k$, we can apply Lemma 3.4.3 directly to each of these S_is where we set the product-depth to $\Delta-1$. For all $i \in [\eta]$, we obtain polynomials $P_{(S_i,J_{i,+},J_{i,-},\pi_i,\tau_i)}$ with formulas of size

  $$|S_i|^{\Delta-1}2^{5k(\Delta-1)d^{\mu(\Delta-1)}},$$

  and product-depth $\Delta-1$.

  Again using the same construction as in the proof of Claim 3.4.7, we obtain the polynomial Q_Δ of product-depth Δ and size at most

  $$\begin{aligned}\eta \cdot 2^{k_-''-k_+''} \cdot \max_{i\in[\eta]} s_i &\leq d \cdot 2^{\sum_{i\in[\eta]}|w_{S_i}|} \cdot d^{\Delta-1}2^{5k(\Delta-1)d^{\mu(\Delta-1)}}\\ &\leq d^\Delta 2^{5k(\Delta-1)d^{\mu(\Delta-1)}+6\gamma k} \leq n^{O\left(\Delta d^{\mu(\Delta-1)}+\gamma\right)}.\end{aligned}$$

Note that by the properties described earlier, both these polynomials have the maximum possible relative rank $\mathrm{relrk}_w(P_\Delta) = \mathrm{relrk}_w(Q_\Delta) = 2^{-|w_{[d]}|/2}$. □

Finally, we show that in the above proofs, without loss of generality, it can be assumed that all entries of w are integers. We can always consider a word w' with integer entries such that the small-sized formula maximizing the relative rank for w' also nearly maximizes it for w, by which we mean that it differs from the maximum attainable relative rank by at most a factor of 2^d, which isn't much since $d = o(\log n)$. We formalize this now.

Claim 3.4.10. *Let* $S \subseteq [d]$ *and let* $w \in \{\alpha_1 k, \dots, \alpha_\gamma k, -\beta_1 k, \dots, -\beta_{\gamma'} k\}^d$,$(|\alpha_i|, |\beta_i| \leq 1$ *for all* i) *be a word with* $\gamma \leq d$ *different weights. Consider the word* w' *where every* $\alpha_i k$ *of* w *is replaced by* $\lfloor \alpha_i k \rfloor$ *and every* $-\beta_j k$ *of* w *is replaced by* $-\lfloor \beta_j k \rfloor$. *Let* P' *be the polynomial obtained in the proof of Theorem 3.4.1 for the word* w'. *Then,* $\mathrm{relrk}_w(P') \geq 2^{-d} 2^{-|w_{[d]}|/2}$.

Proof. From the definition of w', we have $|w'_i| \leq |w_i| \leq |w'_i| + 1$. Hence $\sum_i (|w_i| - |w'_i|) \leq d$. Using the definition of relative rank and noting that $\mathrm{rank}(\mathcal{M}_w(P')) = \mathrm{rank}(\mathcal{M}_{w'}(P'))$,

$$\mathrm{relrk}_w(P')/\mathrm{relrk}_{w'}(P') = \; = \frac{1}{2^{\sum_i (|w_i| - |w'_i|)/2}} \geq 2^{-d/2}.$$

As P' is the polynomial obtained in Theorem 3.4.1 for the word w', we have

$$\mathrm{relrk}_{w'}(P') = 2^{-|w'_{[d]}|/2}.$$

Thus it suffices to show that $|w'_{[d]}| \leq |w_{[d]}| + d$.

By the triangle inequality, $|\sum_i w'_i| \leq |\sum_i w_i| + |\sum_i w'_i - w_i|$ which implies

$$|w'_{[d]}| \leq |w_{[d]}| + \left| \sum_i w_i - w'_i \right| \leq |w_{[d]}| + \sum_i |w_i| - |w'_i| \leq |w_{[d]}| + d$$

where the second inequality holds because $|w_i| \geq |w'_i|$ for all i. □

Chapter 4

Algebraic Branching Programs

This chapter is focused on showing superpolynomial lower bounds for the polynomial $\mathrm{IMM}_{n,d}$ against the *sum* of sub-polynomial sized ABPs. The results appeared in [BDS25b]. As the main conceptual tool, we show that hardness for the seemingly weaker model of sum of set-multilinear ABPs can be escalated in the low-degree regime to general ABP hardness.

4.1 Hardness Bootstrapping Spectrum

In this section, we will show the following reformulation of Valiant's conjecture.

Theorem 4.1.1 (Hardness bootstrapping)**.** *Let n, d be integers such that $d = O(\log n / \log\log n)$. Let $P_{n,d}$ be a set-multilinear polynomial in VNP of degree d. If $P_{n,d}$ cannot be computed by a $\sum \mathsf{smABP}$ of width $\mathrm{poly}(n)$, then $\mathsf{VBP} \neq \mathsf{VNP}$.*

We begin by proving that in the low-degree regime, a small-sized ABP can be simulated by a $\sum \mathsf{smABP}$ of small width. This is very much in the spirit of the set-multilinearization result of Limaye, Srinivasan and Tavenas [LST21, Proposition 9] for small-depth circuits.

Lemma 4.1.2 (ABP set-multilinearization)**.** *Let $P_{n,d}$ be a polynomial of degree d, set-multilinear with respect to the partition $X = X_1 \sqcup \ldots \sqcup X_d$ where $|X_i| \leq n$ for all $i \in [d]$.*

If $P_{n,d}$ can be computed by an ABP of size s, then there is a $\sum$ smABP of width $d^{O(d)}s$ computing the same polynomial.

We immediately have

Proof of Theorem 4.1.1. Suppose that the polynomial $P_{n,d} \in \mathsf{VNP}$ can be computed by an ABP of size s. By Lemma 4.1.2, the polynomial can also be computed by a $\sum$ smABP of width $d^{O(d)}s$. The width of any $\sum$ smABP computing $P_{n,d}$ is, by assumption $n^{\omega(1)}$.

Consequently, our desired separation is obtained by first noting that the above discussion implies $d^{O(d)}s \geq n^{\omega(1)}$, whereby the degree bound $d = O(\log n/\log\log n)$ gives $d^{O(d)} = \mathrm{poly}(n)$ and hence $s \geq n^{\omega(1)}$. □

We also get a corollary for the low variate sum of ROABP model.

Corollary 4.1.3 (Low variate $\sum$ RO)**.** *Let n, d be integers such that $n = O(\log d/\log\log d)$. Let $f \in \mathsf{VNP}$ be a polynomial on n variables of individual degree d. If f cannot be computed by a $\sum$ RO of width $\mathrm{poly}(d)$, then $\mathsf{VBP} \neq \mathsf{VNP}$.*

Proof. Consider the *invertible* map $\phi : x_i^j \mapsto x_{ij}$ for the indices $i \in [n]$ and $j \in [d]$. This transforms an ROABP on n variables $(x_1, \ldots, x_n)$ of individual degree d and order π, to an smABP in the same order that is set-multilinear with respect to $X = X_1 \sqcup \ldots \sqcup X_n$ with $|X_i| \leq d$.

We apply the map ϕ to the $\sum$ RO computing f. This gives us a $\sum$ smABP of the same width that computes a set-multilinear polynomial $Q_{d,n}$ over $O(nd)$ variables with $n = O(\log d/\log\log d)$. Since f does not have a $\sum$ RO of width $\mathrm{poly}(d)$, $Q_{d,n}$ does not have $\sum$ smABP of width $\mathrm{poly}(d)$. Now Theorem 4.1.1 gives us our desired separation. □

In order to prove Lemma 4.1.2, we first homogenize the ABP (similar to the approach of Raz [Raz13] and Limaye, Srinivasan and Tavenas [LST21]). Since $P_{n,d}$ is a degree-d polynomial computed by an ABP of size s, using Lemma 2.2.6, we have that there is a *homogeneous* ABP of width s and length d that computes $P_{n,d}$

and has linear forms as edge labels. As our central argument, we show that this homogeneous ABP can be *efficiently* set-multilinearized.

Proposition 4.1.4. *Consider a set-multilinear polynomial* $P_{n,d}$ *over the variable set* $X = X_1 \sqcup \ldots \sqcup X_d$ *(with* $|X_i| \leq n$ *for all* $i \in [d]$*) computed by a* homogeneous ABP *of width* w *and length* d*. Then, there is a* $\sum$ smABP *of width* $d!w$ *computing* $P_{n,d}$.

Proof. We begin by writing the homogeneous ABP in its matrix form

$$P_{n,d} = \prod_{i=1}^{d} M_i, \tag{4.1.1}$$

where each M_i is a $w \times w$ matrix with entries that are *linear forms* in the variables X. We further write each M_i as a sum $\sum_{j=1}^{d} M_{ij}$, where for all j, M_{ij} is an $w \times w$ matrix with entries that are linear forms, but now in the X_j variables. Doing this for every M_i yields

$$P_{n,d} = \prod_{i=1}^{d} \sum_{j=1}^{d} M_{ij}. \tag{4.1.2}$$

Note that since $P_{n,d}$ is a homogeneous set-multilinear polynomial, the non-set-multilinear products in this expression can be ignored. The matrices only contain linear forms, and thus non-set-multilinear products in the above equation only produce non-set-multilinear monomials. We can ignore any product of the form $(\cdots M_{ij} \cdots M_{i'j} \cdots)$ for different i, i'. We can rearrange to obtain

$$P_{n,d} = \sum_{\pi \in S_d} \prod_{i=1}^{d} M_{i\pi(i)}. \tag{4.1.3}$$

This represents $P_{n,d}$ as the sum of $d!$ set-multilinear ABPs, each of width w. □

With this transformation in hand, we can complete the reduction and obtain Lemma 4.1.2.

Proof of Lemma 4.1.2. Suppose that the ABP for the polynomial $P_{n,d}$ has size s. Using Lemma 2.2.6, we can *homogenize* it to obtain a d-layered homogeneous ABP of *width* s. By Proposition 4.1.4, we obtain a $\sum \mathsf{smABP}$ of width $d!s = d^{O(d)}s$. $\square$

The bootstrapping result can be generalized to set-multi-k-ic models.

Theorem 4.1.5 (Hardness bootstrapping spectrum). *Let n, d, k be integers such that $\exp(kd \log d) = \mathsf{poly}(n)$, and let $P_{n,d,k}$ be a* set-multi-k-ic *polynomial in* VNP *of degree kd. If $P_{n,d,k}$ cannot be computed by a $\sum \mathsf{sm(k)ABP}$ of width $\mathsf{poly}(n)$, then* $\mathsf{VBP} \neq \mathsf{VNP}$.

Remark 4.1.6. We note that Theorem 4.1.1 is an immediate consequence of Theorem 4.1.5 when $k = 1$. An added advantage of this generalization is the flexibility with the degree of the hard polynomial.

The proof of Theorem 4.1.5 follows the template of Theorem 4.1.1. We begin with ABP homogenization, followed by a structural transformation to the sum of *set-multi-k-ic* ABP. The superpolynomial lower bound assumption on $\sum \mathsf{sm(k)ABP}$ gives the desired separation result. The following lemma is analogous to Lemma 4.1.2.

Lemma 4.1.7 (ABP to $\sum \mathsf{sm(k)ABP}$). *Let P be a set-multi-k-ic polynomial with respect to the partition $X = X_1 \sqcup \ldots \sqcup X_d$ where $|X_i| \leq n$ for all $i \in [d]$. If P can be computed by an ABP of size s, then there is a $\sum \mathsf{sm(k)ABP}$ of width $s \cdot (kd)!/(k!)^d$ computing the same polynomial.*

Proof. Using Lemma 2.2.6 on the ABP of size s computing the polynomial P of degree kd, we obtain a kd-layered homogeneous ABP of width s. Consider the homogeneous ABP in its matrix form:

$$P = \prod_{i=1}^{kd} M_i,$$

where each M_i is a $s \times s$ matrix with entries that are *linear forms* in the variable X. Express each M_i as a sum $\sum_{j=1}^{d} M_{ij}$, where for all j, M_{ij} is a $s \times s$ matrix with entries that are linear forms only in X_j variables. Doing this for every M_i yields

$$P = \prod_{i=1}^{kd} \sum_{j=1}^{d} M_{ij}.$$

Since P is a homogeneous *set-multi-k-ic* polynomial, products of the form $(\cdots M_{ij} \cdots M_{i'j} \cdots)$ for different i, i' are allowed in the expression, but not more than k. Formally, we say a tuple $\mathbf{j} := (j_1, \ldots, j_{kd}) \in [d]^{kd}$ is k-*unbiased* if all the elements in the tuple repeat exactly k times. Let S be the set of such k-unbiased tuples. We rearrange to obtain

$$P = \sum_{\mathbf{j} \in S} \prod_{i=1}^{kd} M_{ij_i}.$$

Noting that $|S| = (kd)!/(k!)^d$, the expression above represents P as sum of $(kd)!/(k!)^d$ *set-multi-k-ic* ABPs, each of width s. □

It is straightforward to prove Theorem 4.1.5 using the above lemma. The proof is similar to Theorem 4.1.1.

Proof of Theorem 4.1.5. Suppose that the polynomial $P_{n,d,k} \in \mathsf{VNP}$ can be computed by an ABP of size s. Using Lemma 4.1.7, it can also be computed by a $\sum \mathrm{sm}(k)\mathrm{ABP}$ of width $s \cdot (kd)!/(k!)^d$. By assumption, the width of any $\sum \mathrm{sm}(k)\mathrm{ABP}$ computing P is $n^{\omega(1)}$. We obtain the desired separation $s \geq n^{\omega(1)}$ by observing that:

$$s \cdot (kd)!/(k!)^d = s \cdot \exp(kd \log d) \geq n^{\omega(1)},$$

since $\exp(kd \log d) = \mathrm{poly}(n)$. □

We also obtain the corresponding corollary for $\sum \mathrm{R}(k)\mathrm{O}$, the sum of read-$k$ oblivious ABPs.

Corollary 4.1.8. *Let n, d, k be integers such that $\exp(kn \log n) = \mathrm{poly}(d)$. Let $f \in \mathsf{VNP}$ be a polynomial on n variables of individual degree d. If f cannot be computed by a $\sum \mathrm{R}(k)\mathrm{O}$ of width $\mathrm{poly}(d)$, then $\mathsf{VBP} \neq \mathsf{VNP}$.*

Proof. Consider the *invertible* map $\phi : x_i^j \mapsto x_{ij}$ for the indices $i \in [n]$ and $j \in [d]$. This transforms an $\mathrm{R}(k)\mathrm{OABP}$ on n variables $(x_1, \ldots, x_n)$ of individual degree d, to an $\mathsf{sm}(k)\mathsf{ABP}$ of width d and length kn wrt variable partitioning $X = X_1 \sqcup \ldots \sqcup X_n$ with $|X_i| \leq d$.

We apply the map ϕ to the $\sum \mathsf{R}(k)\mathsf{O}$ computing f. This gives us a $\sum \mathsf{sm}(k)\mathsf{ABP}$ of length kn that computes a *set-multi*-k-*ic* polynomial $Q_{d,n,k}$ over nd variables. Since f does not have a $\sum \mathsf{R}(k)\mathsf{O}$ of width $\mathsf{poly}(d)$, the transformation induced by the map implies that $Q_{d,n,k}$ does not have $\sum \mathsf{sm}(k)\mathsf{ABP}$ of width $\mathsf{poly}(d)$. Moreover, $\exp(kn \log n) = \mathsf{poly}(d)$. Then Theorem 4.1.5 gives us our desired separation. $\square$

Few lower bounds are known for *read*-k oblivious ABPs. They were studied by Anderson et al. [And+18] as a natural generalization of ROABPs and a lower bound of $\exp(n/k^{O(k)})$ for a single *read*-k oblivious ABP was shown. It remains open to improve this result to prove non-trivial lower bounds when k is large, as well as to prove lower bounds for sums of *read*-k oblivious ABPs. When k is small, the results of Ramya and Rao [RR20] extend to the sum of multilinear k-pass ABPs, a restriction of *read*-k oblivious ABPs in which the variables are read k times in sequence, each time in a possibly different order.

4.2 Lower Bound for the sum of ABPs

We are now ready to show that in the low degree regime, the Iterated Matrix Multiplication polynomial $\mathsf{IMM}_{n,d}$ cannot be computed even by a polynomially large sum of ABPs, provided that each of the ABPs is small in size.

Theorem 4.2.1 ($\sum$ ABP lower bound). *Let* $d = n^{o(1)}$. *The polynomial* $\mathsf{IMM}_{n,d}$ *cannot be computed by the sum of* $\mathsf{poly}(n, d)$ *ABPs, each of size* $(nd)^{o(1)}$.

We begin by showing a lower bound for $\sum$ smABP in the *low-degree* regime. Note that in this regime, IMM has an smABP of width $O(nd)$. The lemma shows that even using the sum of multiple smABPs cannot help in reducing the width.

Lemma 4.2.2. *Any* $\sum$ smABP *computing the polynomial* $\mathsf{IMM}_{n,d}$ *with* $d = O(\log n/ \log\log n)$, *must have width at least* $n^{\Omega(1)}$.

Proof. Let the maximum width of any smABP in the sum be w. Every path in a particular set-multilinear ABP is of length d and computes a product of linear forms. Using the definition of ABP computation, we sum over all paths to obtain a depth-3 *set-multilinear circuit*[1] of top fanin w^d. Doing the same for all the smABPs, we get a depth-3 set-multilinear circuit of top fan-in at most $d!w^d$.

We now apply the partial derivative method. Split $X = X_1 \sqcup \ldots \sqcup X_d$ into 'even' and 'odd' parts. That is, we consider the partition $X = X^{(0)} \sqcup X^{(1)}$, with

$$X^{(0)} = X_2 \sqcup X_4 \sqcup \ldots \sqcup X_k, \text{ and } X^{(1)} = X_1 \sqcup X_3 \sqcup \ldots \sqcup X_{k'}, \tag{4.2.1}$$

where $k = 2\lfloor d/2 \rfloor$ and $k' = 2\lceil d/2 \rceil - 1$.

The partial derivative matrix $\mathcal{M}(f)$ for any polynomial f has rows indexed by set-multilinear monomials in $X^{(0)}$ and columns indexed by set-multilinear monomials in $X^{(1)}$. Consider now monomials m_0, m_1 that are set-multilinear in $X^{(0)}, X^{(1)}$ respectively. For any set-multilinear polynomial f, the (m_0, m_1) entry in $\mathcal{M}(f)$ is the coefficient of the monomial $m_0 \cdot m_1$ in f. It is straightforward to see that the partial derivative matrix of $\mathrm{IMM}_{n,d}$ is of full rank, that is, $\mathrm{rank}(\mathcal{M}(\mathrm{IMM}_{n,d})) = n^{d/2}$.

On the other hand, when we consider a set-multilinear $\sum\prod\sum$ circuit, the linear forms at the bottom have a rank of at most 1 with respect to any partition of X. Consequently, taking products of linear forms cannot result in a polynomial of rank greater than 1. Finally, *subadditivity* of matrix rank implies that the rank of the set-multilinear circuit is at most the top-fanin $d!w^d$, giving

$$n^{d/2} \leq d!w^d. \tag{4.2.2}$$

Using the fact that $d! = O(d^d) = \mathrm{poly}(n)$ for our degree regime, it now follows that $w = n^{\Omega(1)}$ and we obtain the $\sum \mathrm{smABP}$ lower bound. $\square$

Suppose we had to prove the lower bound of Theorem 4.2.1 for a single ABP

[1]Every vertex in a set-multilinear circuit computes a set-multilinear polynomial with respect to a subset of the variable sets.

computing IMM. We could then use Lemma 4.2.2 above in conjunction with Lemma 4.1.2 to conclude the result. But when we are dealing with a sum of ABPs, we need to be more careful in how we set-multilinearize since the ABPs no longer need to compute set-multilinear or even homogeneous polynomials.

Proof of Theorem 4.2.1. Suppose that $\mathrm{IMM}_{n,d}$ (with $d \leq n^{o(1)}$) can be written as the sum of m ABPs of size $s = n^{o(1)}$ each[2]. In the corresponding matrix form, we have

$$\mathrm{IMM}_{n,d} = \sum_{i=1}^{m} \prod_{j=1}^{\ell} M_{ij}, \tag{4.2.3}$$

where each M_{ij} is an $s \times s$ matrix and $\ell \leq s$.

Consider now the polynomial $\mathrm{IMM}_{n,d'}$ with $d' = O(\log n / \log\log n)$. This polynomial can be obtained as a restriction of $\mathrm{IMM}_{n,d}$ by setting all matrices other than the first d' in the definition of IMM to the identity matrix I_n. Correspondingly, Equation 4.2.3 now becomes

$$\mathrm{IMM}_{n,d'} = \sum_{i=1}^{m} \prod_{j=1}^{\ell} M'_{ij}, \tag{4.2.4}$$

where just like in (4.2.3), each M'_{ij} is an $s \times s$ matrix and $\ell \leq s$. Note that any lower bound on $\mathrm{IMM}_{n,d'}$ also holds for $\mathrm{IMM}_{n,d}$.

We would like to set-multilinearize Equation 4.2.4. But we cannot directly apply Lemma 4.1.2 since the ABPs in the sum need not compute a set-multilinear polynomial anymore. In fact, they need not even compute a homogeneous polynomial. Nevertheless, we are only interested in the homogeneous component of degree d' of the polynomials that these ABPs compute, the rest vanishing in the final sum.

Consider a single ABP A of size $s = n^{o(1)}$ from the sum of m ABPs above. Suppose that it computes a (possibly non-homogenous) polynomial of degree d_A. Using Lemma 2.2.6, we can homogenize A to obtain an ABP of length d_A and

[2]When $d > n^{o(1)}$, the lower bound trivially holds.

width s, with linear forms on the edges. Consider now the (possibly empty) set T of vertices in layer d' of this ABP that have no outgoing edges. For every $v \in T$, the sub-ABP between the start vertex s and the vertex v computes a homogeneous polynomial of degree d', monomials of which might occur in the final polynomial $\mathrm{IMM}_{n,d'}$. Vertices not in T can be safely ignored as they have outgoing edges with linear *forms* on them and hence will only contribute to monomials of degree greater than d' in the polynomial computed by A.

We now identify all the vertices in T with a single vertex t. Furthermore, we replace all the possible multi-edges generated between a vertex u in layer $d' - 1$ and the vertex t, with a single edge that has as its edge label the sum of all the multi-edge labels. This gives us a homogeneous ABP of width s and length d' computing the homogeneous component of degree d' of the polynomial computed by A. Performing this operation for each of the m ABPs, we can write

$$\mathrm{IMM}_{n,d'} = \sum_{i=1}^{m} \prod_{j=1}^{d'} M'_{ij}, \tag{4.2.5}$$

where the new matrices obtained after homogenization have been renamed to M' for brevity. As before, we split each M'_{ij} as a sum $\sum_{k=1}^{d'} M'_{ijk}$ where for all $k \in [d']$, M'_{ijk} is an $s \times s$ matrix with entries that are *linear forms* in the X_k variables[3].

$$\mathrm{IMM}_{n,d'} = \sum_{i=1}^{m} \prod_{j=1}^{d'} \sum_{k=1}^{d'} M'_{ijk}, \tag{4.2.6}$$

In the proof of Proposition 4.1.4, we were crucially using the fact that the polynomial computed by the ABP was set-multilinear in order to ignore non-set-multilinear products. Although this is not the case any longer, we can still ignore all the non-set-multilinear products since they *only* produce non-set-multilinear monomials and the sum of the ABPs is $\mathrm{IMM}_{n,d'}$, a set-multilinear polynomial.

[3]Alternately, we can directly convert each of the m ABPs to a *homogenous* depth-3 circuit and use the result of [NW96] to prove our result.

We obtain an expression similar to Equation 4.1.3:

$$\mathrm{IMM}_{n,d'} = \sum_{i=1}^{m} \sum_{\pi \in S_{d'}} \prod_{j=1}^{d'} M'_{ij\pi(j)}. \tag{4.2.7}$$

That is, $\mathrm{IMM}_{n,d'}$ can be written as the sum of $md'!$ smABPs, each of width s. We now analyze similarly to the proof of Lemma 4.2.2. We convert the $\sum$ smABP to a depth 3 set-multilinear circuit of top-fanin at most $md'!s^{d'}$. Using the exact same partition of X into $X^{(0)}$ and $X^{(1)}$ as in (4.2.1), we construct the partial derivative matrix $\mathcal{M}$ for $\mathrm{IMM}_{n,d'}$ and the set-multilinear $\sum\prod\sum$ circuit that we obtained. The rank calculation results in

$$n^{d'/2} \leq md'!s^{d'}, \tag{4.2.8}$$

which along with $s = n^{o(1)}$ and $d'! = \mathrm{poly}(n)$ gives $m = n^{\omega(1)}$. $\square$

4.2.1 Lower Bound for $\mathrm{NW}_{n,d}$

We show that the lower bound of Theorem 4.2.1 also holds for a polynomial from the family of Nisan-Wigderson design-based polynomials.

Let $\mathbb{F}_n$ be a field of size n (we assume that n is a power of a prime). We will work in the low-degree regime. For $d = O(\log n/\log\log n)$, consider the set of variables $X = X_1 \sqcup \ldots \sqcup X_d$ where $X_i = \{x_{ij} \mid j \in [n]\}$ for all $i \in [d]$. Let $\mathcal{F}$ be the set of all *univariate* polynomials $f(y) \in \mathbb{F}_n[y]$ of degree less than $d/2$. The polynomial $\mathrm{NW}_{n,d}$ on the above nd variables is defined as

$$\mathrm{NW}_{n,d}(X) = \sum_{f \in \mathcal{F}} \prod_{i \in [d]} x_{if(i)}.$$

Each monomial encodes a univariate polynomial of degree less than $d/2$. Consider the partition $X = X^{(0)} \sqcup X^{(1)}$ from (4.2.1). For a monomial $m_0 = x_{2j_2} \cdots x_{kj_k}$ (with all j indices in $[n]$) that is set-multilinear in $X^{(0)}$, there is a unique "extension monomial" m_1 (set-multilinear in $X^{(1)}$) such that $m_0 m_1$ is a monomial of $\mathrm{NW}_{n,d}$.

This is because m_0 encodes the evaluations of some univariate polynomial on points $\{2,\ldots,k\}$. As the length of m_0 is at least $d/2$, interpolating these values gives a unique polynomial f which then determines the corresponding m_1 – obtained by evaluating f on the remaining points $\{1,3,\ldots,k'\}$ in $[d]$.

This implies that the partial derivative matrix $\mathcal{M}(NW_{n,d})$ of size $n^{d/2} \times n^{d/2}$ has full rank. The same rank analysis as before on sums of ABPs gives us Theorem 4.2.1, but with $NW_{n,d}$ as the hard polynomial. Nevertheless, the techniques used seem to not be enough to get us any better lower bounds. In particular, the loss of information in the conversion of an smABP (an essentially non-commutative model) to a set-multilinear circuit seems to be too large.

4.3 Discussion and Open problems

In order to separate VBP from VNP, we need to prove super-polynomial lower bounds against $\sum \mathsf{smABP}$ for a polynomial in VNP that we expect to be hard. As noted above, the IMM polynomial is in VBP (in fact, it is a canonical way to define the class VBP) and cannot be used for such a separation. Since our Theorem 4.2.1 also holds for a polynomial from the Nisan-Wigderson family of design-based polynomials that is in VNP but not conjectured to be in VBP, it is a better candidate.

A first step toward proving ABP lower bounds would be to prove any non-trivial lower bounds against the sum of smABPs in the low degree regime, i.e. prove *some* lower bound for the sum of $d!$ smABPs. Another interesting direction is to show a reduction from ABPs to the sum of fewer than $d!$ smABPs, with a possibly super polynomial blow up in the smABP size. This would still lead to ABP lower bounds if we can prove strongly exponential lower bounds against the sum of (fewer) smABPs. This question remains open as well.

Chapter 5

Counting Homomorphisms

In this chapter, we study the montone bounded-depth complexity of the homomorphism and colored subgraph polynomials

$$\mathrm{Hom}_{H,n} = \sum_{f:V(H)\to[n]} \prod_{uv\in E(H)} x_{f(u),f(v)} \quad \text{and}$$
$$\mathrm{ColSub}_{H,n} = \sum_{f:V(H)\to[n]} \prod_{uv\in E(H)} x^{(uv)}_{f(u),f(v)},$$

where we fix our field to be the rational numbers $\mathbb{Q}$[1]. For simplicity, we will assume the pattern H to be a simple, connected graph.

5.1 Bounded versions of treewidth and pathwidth

We begin by defining the *bounded-depth* version of treewidth (Definition 2.3.1). We stress that a tree with a single node has height 1 according to our definition.

Definition 5.1.1. For fixed $\Delta \in \mathbb{N}$, the Δ-treewidth of a graph H, denoted by $tw_\Delta(H)$, is the minimum width over all tree decompositions of H with underlying tree T of height at most Δ.

While depth-restricted tree-decompositions did arise before in the literature [CIP16], their depth was not fixed to concrete *constants* in these contexts,

[1]Our results hold for any field by making appropriate changes to the definition of monotone computation so that cancellations are avoided.

but rather to, say, $O(\log |V(H)|)$. In particular, differences between Δ-treewidth and $(\Delta - 1)$-treewidth were not considered.

5.1.1 Connections to other graph parameters

The notion of Δ-treewidth of graphs is connected to other parameters that have been studied:

- The 1-treewidth of a graph H is merely its number of vertices $|V(H)|$, as the requirement on the height forces the tree-decomposition to consist of a single bag. On the other extreme, the $|V(H)|$-treewidth of H equals the treewidth of H.

- The 2-treewidth is already more curious: For any vertex-cover C, a tree-decomposition of height 2 for H can be obtained by placing C into a root bag X_r that is connected to bags X_t for $t \in V(H) \setminus C$, where X_t contains t and its neighbors, all of which are in C. This shows that the 2-treewidth of H is at most the vertex-cover number $vc(H)$ of H. In fact, the 2-treewidth of H equals the so-called *vertex-integrity* of H (minus 1). This graph parameter is defined as $\min_{S \subseteq V(H)}(|S| + \max_C |V(C)|)$, where C ranges over all connected components in the graph $H - S$, see [BES87; Gim+25].

- By balancing tree-decompositions [CIP16], a universal constant c can be identified such that, for all graphs H on k vertices, the $c \log k$-treewidth of H is bounded by $4tw(H) + 3$. That is, at the cost of increasing width by a constant multiplicative factor, tree-decompositions can be assumed to be of logarithmic height.

Our upper bound proof will show that vertices of degree 1 can be removed safely from H without changing the bounded-depth complexity of $\mathrm{Hom}_{H,n}$. This holds essentially because such vertices and their incident edges can be assumed to be present in the leaves of a tree-decomposition; these leaves then do not contribute to the product-depth of the constructed circuit. This naturally leads to the notion of *pruned* Δ-treewidth.

Definition 5.1.2. The *pruned* Δ-treewidth of a graph H, denoted by $\mathrm{ptw}_{\Delta}(H)$, is the Δ-treewidth of the graph H with all vertices of degree at most 1 removed.

We also define analogous *bounded-length* version of pathwidth (Definition 2.3.2).

Definition 5.1.3. For fixed $\Delta \in \mathbb{N}$, the Δ-pathwidth of a graph H, denoted by $\mathrm{pw}_{\Delta}(H)$, is the minimum width over all path decompositions of H with underlying path P of length at most Δ.

There is also a corresponding pruned version.

Definition 5.1.4. The *pruned* Δ-pathwidth of a graph H, denoted by $\mathrm{ppw}_{\Delta}(H)$, is the Δ-pathwidth of the graph H with all vertices of degree at most 1 removed.

5.1.2 Full d-ary trees

We conclude this section by exhibiting a pattern H whose Δ-treewidth shows a strong phase transition that we can exploit in our depth hierarchy theorem: Its Δ-treewidth is low, but even its $(\Delta-1)$-treewidth is high. As it turns out, H can be chosen to be the full d-ary tree.

Theorem 5.1.5. *Let Δ, d be positive integers and let T_{Δ} be the full d-ary tree of height Δ. Then $\mathrm{tw}_{\Delta}(T_{\Delta}) = 1$ whereas $\mathrm{tw}_{\Delta-1}(T_{\Delta}) \geq d-1$.*

In order to prove the theorem, we first prove the following useful lemma for inductively bounding the Δ-treewidth of a given graph.

Lemma 5.1.6. *For any integers d and Δ, if a graph G contains at least d disjoint connected subgraphs G_1, G_2,...,G_d, and the $(\Delta-1)$-treewidth of each of them is at least $d-1$, then the Δ-treewidth of G is at least $d-1$.*

Proof. Suppose T is a rooted tree-decomposition of graph G, and R is the root bag of T. We are going to prove that either the height of T is larger than Δ or the width of T is at least $d-1$. There are two cases to consider:

1. R contains at least one vertex from each of the subgraphs $G_1, G_2, \ldots, G_d$.

2. R does not contain any vertex from (at least) one of the subgraphs $G_1, G_2, \ldots, G_d$.

In the first case, the size of R is at least d, so the width of T is at least $d-1$. In the second case, we can assume without loss of generality that R does not contain any vertex from G_1. Let $T_1, T_2, \ldots, T_k$ be the subtrees obtained by removing R from T. Since R does not contain any vertex of G_1, at least one of $T_1, T_2, \ldots, T_k$ must contain some vertex from G_1. Suppose that subtree is T_1. For any vertex v contained in both T_1 and G_1, since v is not in the root bag R, it must be the case that v is not contained in any other $T_2, \ldots, T_k$ as well. Similarly, every neighbor u of v in G_1 is also contained in T_1 as u is not contained in R, and there must be a bag in T which contains both u and v. Proceeding this way, we get that T_1 contains the whole of G_1, and the vertices from G_1 appear nowhere else.

Removing vertices not in G_1 from each bag of T_1, we obtain a new tree T_1'. We claim that T_1' is a tree-decomposition of G_1. Indeed, all vertices of G_1 are in T_1 and the bags in T_1 that contain a vertex v form a connected component since T was a tree decomposition of G. So the same holds for T_1'. Moreover, for every edge (u, v) in G_1, there is a bag in T_1 that contain both u and v, so the same holds for T_1'.

Since the $(\Delta-1)$-treewidth of G_1 is at least $d-1$, either the height of T_1' is larger than $\Delta-1$ or the width of T_1' is at least $d-1$. Since T_1' was formed by removing vertices from T_1, the same holds for T_1. Consequently, either the height of T is larger than Δ or the width of T is at least $d-1$. In both cases, the lemma holds. □

We are now ready to prove Theorem 5.1.5.

Proof of Theorem 5.1.5. It is evident that $\mathrm{tw}_\Delta(T_\Delta) = 1$ for all Δ, since T_Δ is a tree. As for the lower bound, consider the base case $\Delta = 2$. The height-1 tree-decomposition of the height-2 full d-ary tree T_2 has only one bag, and this bag contains all the vertices from T_2. Hence, its treewidth is $d \geq d - 1$.

Assume by induction that the theorem holds for all $2 \leq \Delta \leq k$ for some natural number k. Then, for $\Delta = k + 1$, consider the height-$(k + 1)$ full d-ary tree T_{k+1}. Removing the root node of T_{k+1} yields d pairwise disjoint height-k full d-ary trees. By our inductive assumption, the $(k-1)$-treewidth of each of these trees is at least $d - 1$. Then by Lemma 5.1.6, the k-treewidth of T_{k+1} is at least $d - 1$. □

In the next two sections, we will prove our main characterization theorem for bounded-depth monotone circuits.

Theorem 5.1.7. *Let H be a fixed graph and let Δ and n be natural numbers. Then the polynomials $\mathrm{Hom}_{H,n}$ and $\mathrm{ColSub}_{H,n}$ have monotone circuits of size $O(n^{\mathrm{ptw}_\Delta(H)+1})$ and product-depth Δ. Moreover, any monotone circuit of product-depth Δ has size $\Omega(n^{\mathrm{ptw}_\Delta(H)+1})$.*

We also show a similar characterization for bounded-length monotone branching programs.

Theorem 5.1.8. *Let H be a fixed graph and let Δ and n be natural numbers such that $\Delta \geq |E(H)|$. Then the polynomials $\mathrm{Hom}_{H,n}$ and $\mathrm{ColSub}_{H,n}$ can be computed by monotone algebraic branching programs of size $O(n^{\mathrm{ppw}_\Delta(H)+1})$ and length Δ. Moreover, any monotone algebraic branching program of length Δ has size $\Omega(n^{\mathrm{ppw}_\Delta(H)+1})$.*

Note that for a length-Δ ABP to compute the polynomial $\mathrm{Hom}_{H,n}$ (or $\mathrm{ColSub}_{H,n}$), we need the length to be at least the degree of the polynomial (which is $|E(H)|$) since otherwise, we cannot even compute a single monomial. We note that the above theorem also implies a bound on the *width* of the ABP.

5.2 Upper bounds in Theorem 5.1.7 and Theorem 5.1.8

We prove the upper bound in Theorem 5.1.7. First, we require additional standard notation for tree-decompositions: We consider T to be rooted with a choice of root that minimizes its height. Given a tree-decomposition of H with underlying tree T and bags $\{X_t\}_{t \in V(T)}$, write $\gamma(t) := \bigcup_{s \geq t} X_s$ for the cone at t, where s ranges over all descendants of t in the tree T.

Our second definition is more technical and specific to the dynamic programming approach we use to compute homomorphism polynomials in a bottom-up manner: It allows us to track *where* in the tree-decomposition an edge contributes to a monomial of the final polynomial. We say that an *edge-representation* of H in T is a function $\mathsf{rep} : E(H) \to V(T)$ that assigns to each edge of H a node in T such that $\{u, v\} \subseteq X_{\mathsf{rep}(uv)}$ for all $uv \in E(H)$. Note that each edge $uv \in E(H)$ is already entirely contained in *at least one* bag by the definition of a tree-decomposition; the function rep simply chooses one such bag for each edge.

Given an edge-representation rep, we define the rep-height of T (which will be the product-depth of the constructed circuit) as the maximum number of "active" nodes t on a root-to-leaf path in T, where we call a node t active iff

- there are distinct $e, e' \in E(H)$ with $\mathsf{rep}(e) = \mathsf{rep}(e') = t$, or
- there is at least one $e \in E(H)$ with $\mathsf{rep}(e) = t$ and t has a child, or
- t has at least two children.

In our dynamic programming approach that proceeds bottom-up on a tree-decomposition, only active nodes require multiplication gates; the rep-height will thus amount to the overall product-depth of the circuit.

Lemma 5.2.1. *Let H be a graph with a tree-decomposition consisting of tree T and bags $\{X_t\}_{t \in V(T)}$, and let rep be an edge-representation of H in T. Then there are circuits for $\mathrm{Hom}_{H,n}$ and $\mathrm{ColIso}_{H,n}$ with product-depth equal to the rep-height of T and $O(|V(T)| \cdot n^{w})$ gates for $\max_{t \in V(T)} |X_t| = w$.*

Proof. We describe the circuit for $\mathsf{Hom}_{H,n}$ and remark that the circuit for $\mathsf{ColSub}_{H,n}$ can be constructed analogously. Considering T to be rooted, and proceeding from the leaves of T to the root, we inductively compute polynomials $\mathsf{Restr}_{t,h}$ for nodes $t \in V(T)$ and functions $h : X_t \to [n]$. The polynomials are defined as

$$\mathsf{Restr}_{t,h} = \sum_{\substack{f:\gamma(t)\to[n] \\ f \text{ extends } h}} \prod_{\substack{uv\in E(H) \\ \mathsf{rep}(uv)\geq t}} x_{f(u),f(v)}.$$

Here, we write $s \geq t$ to denote that s is a descendant of t in T. Note that $\mathsf{Restr}_{t,h}$ is the restriction of $\mathsf{Hom}_{H,n}$ to homomorphisms f that extend a given homomorphism h for the bag at t, such that only those edges feature in the monomials that are represented in the cone $\gamma(t)$. Then $\mathsf{Hom}_{H,n}$ is the sum of $\mathsf{Restr}_{r,h}$ over all $h : X_r \to [n]$ at the root r of T.

We show how to compute the polynomials $\mathsf{Restr}_{p,h}$ for nodes $p \in V(T)$. Let $p \in V(T)$ be a node with children $N \subseteq V(T)$, possibly with $N = \emptyset$ if p is a leaf. Assume that $\mathsf{Restr}_{t,h'}$ is known for all $t \in N$ and functions $h' : X_t \to [n]$. Then we have

$$\mathsf{Restr}_{p,h} = \left(\prod_{\substack{uv\in E(H) \\ \mathsf{rep}(uv)=p}} x_{h(u),h(v)} \right) \cdot \prod_{t\in N} \sum_{\substack{h':X_t\to[n] \\ \text{agreeing with } h \\ \text{on } X_t\cap X_p}} \mathsf{Restr}_{t,h'}. \tag{5.2.1}$$

From this construction of the circuit, the size bound claimed in the lemma is obvious. Let us investigate its product-depth: In the final circuit computing $\mathsf{Hom}_{H,n}$, every path from the output gate to an input gate corresponds to a path in T from the root to a leaf. Analyzing (5.2.1), we see that every node t on this path contributes 1 to the product-depth iff t is active under the edge-representation rep. Indeed, a leaf p only contributes to the product-depth if two edges $e, e' \in E(H)$ are represented in its bag, i.e., $\mathsf{rep}(e) = \mathsf{rep}(e') = p$, as then there is a nontrivial product in the first product (over uv, shown in parentheses in (5.2.1)). A node p with one child only contributes if at least one edge is represented in its bag, as then the product between the parentheses and the remaining factor is nontrivial. A node p with at least two children always contributes to the product-depth.

To show that the circuit correctly computes $\mathrm{Hom}_{H,n}$, we need to show that the recursive expression for $\mathrm{Restr}_{p,h}$ in (5.2.1) is correct. Note that every edge is represented by rep in exactly one bag and thus appears precisely once in a monomial. Because X_p is a separator in H, any function $f : \gamma(p) \to [n]$ gives rise to $|N|$ functions $f_t : \gamma(t) \to [n]$ for $t \in N$ that all agree on their values for X_p (that is, on their values on $X_p \cap X_t$) and can otherwise be chosen independently. Conversely, any ensemble of such consistent functions can be merged to a function $h : \gamma(p) \to [n]$. The product over all $t \in N$ as in (5.2.1) thus yields $\mathrm{Restr}_{p,h}$. □

Finally, to prove the upper bound in Theorem 5.1.7, let $H^\dagger$ be the graph obtained from H by removing all degree-1 vertices. Given a tree-decomposition for $H^\dagger$ with underlying tree T of height Δ and width w witnessing that $\mathrm{ptw}_\Delta(H) = w$, we obtain a tree-decomposition with some tree T' for H and an edge-representation rep of H in T' of rep-height Δ as follows: For each vertex $v \in V(H)$ of degree 1, with neighbor $u \in V(H)$, choose some node $t \in T$ with $u \in X_t$ and add a node t' as a neighbor of t to T with bag $X_{t'} = \{v, u\}$. Choose an arbitrary representation rep of H in the resulting tree-decomposition with tree T' and observe that its rep-height is at most the height Δ of T, even though the height of T' may be $\Delta + 1$: The bags added for degree-1 vertices and their incident edges do not contribute towards the rep-height, as they are leaf nodes and represent single edges. The upper bound thus follows from Lemma 5.2.1.

Remark 5.2.2. The construction from Lemma 5.2.1 also yields an ABP of length $|V(T)|$ and size $O(|V(T)| \cdot n^w)$ when given a path-decomposition T of H with maximum bag size w. To see this, note that the product over $t \in N$ in (5.2.1) involves only a single factor when T is a path-decomposition, so (5.2.1) overall amounts to a skew-multiplication of a single monomial with the recursively computed polynomial.

5.3 Lower bounds in Theorem 5.1.7 and Theorem 5.1.8

We adapt the lower bound proofs of [KPR23] to prove the lower bounds in our theorems. Recall that proving the lower bound for $\mathsf{ColSub}_{H,n}$ is enough, since we can use a circuit computing $\mathsf{Hom}_{H,n}$ to obtain a circuit computing $\mathsf{ColSub}_{H,n}$ without changing the depth of the circuit using [KPR23, Lemma 8]. We summarize the results here for completeness.

Lemma 5.3.1. *Let k, Δ be positive integers and H be a fixed pattern graph on k vertices.*

- *If there is a monotone circuit of product-depth Δ and size s for $\mathsf{ColSub}_{H,n}$, then there is such a circuit of size $O(s)$ for $\mathsf{Hom}_{H,n}$.*
- *If there is a monotone circuit of product-depth Δ and size s for $\mathsf{Hom}_{H,n'}$, then there is such a circuit of size $O(s^{|E(H)|})$ for $\mathsf{ColSub}_{H,n}$, where $n' = kn$.*

The results also hold if circuits are replaced by ABPs, and product-depth is replaced by the length of the ABP, provided the length is at least the degree of the polynomials.

Proof. Given a monotone circuit of product-depth Δ that computes $\mathsf{ColSub}_{H,n}$, we replace each variable $x^{(uv)}_{f(u),f(v)}$ with $x_{f(u),f(v)}$ if $f(u) \neq f(v)$ and 0 otherwise. The circuit now computes $\mathsf{Hom}_{H,n}$.

For the other direction, let C be the monotone circuit of product-depth Δ computing the Hom_H polynomial over the vertex set $[k] \times [n]$. Note that a homomorphism ϕ from H to the complete graph on $[k] \times [n]$ maps a vertex $u \in [k]$, to (v, p) where $v \in [k]$ and $p \in [n]$. We introduce auxiliary variables y_{uv} for each edge $uv \in E(H)$. For $u, v \in [k]$ and $p, q \in [n]$, we replace the variable $x_{(u,p),(v,q)}$ with $x^{(uv)}_{p,q} y_{uv}$ if $uv \in E(H)$ and 0 otherwise.

Let C' be the new circuit obtained after the replacement, and consider the partial derivative $D := \frac{\partial^{|E(H)|}}{\partial y_{e_1} \cdots \partial y_{e_{|E(H)|}}} C'$, with respect to all the edge variables of H. Note that every monomial in D contains at least one variable corresponding to each edge of H. Further, set $y_{uv} = 0$ in D for all $uv \in E(H)$. This ensures that every monomial in $D|_{y_{uv}=0}$ contains exactly one variable corresponding to every edge of H, i.e., it counts only the color-preserving homomorphisms. The coefficient of each monomial is $|\mathrm{aut}(H)|$, the number of automorphisms of H, and dividing by this number gives us $\mathrm{ColSub}_{H,n}$.

We can compute D using partial derivatives' sum and product rules applied to every gate in a bottom-up fashion. For a gate g, we maintain both g and $\partial_{y_e} g$. The partial derivative of a sum gate, $\partial_{y_e} \sum_i g_i = \sum_i \partial_{y_e} g_i$ is straightforward and does not increase the depth. For a product gate, the derivative $\partial_{y_e} \prod_i g_i = \sum_i \left(\partial_{y_e} g_i \prod_{j \neq i} g_j \right)$ increases the depth by one, but this can be absorbed in the sum layer above. Note that the product-depth does not change in both cases. A partial derivative with respect to a single variable increases the circuit size by a factor of s. Hence, the final circuit for D is of size $O(s^{|E(H)|})$, and has product-depth Δ, the same as C.

We also note that both the constructions preserve monotonicity. Moreover, if the original circuit C was *skew* (i.e. an ABP), then so is the final circuit D. From Remark 2.2.12, we obtain the same results for ABPs as well. □

5.3.1 Tree decompositions from parse trees

Consider a pattern graph H on vertex set $V(H) := [k]$. An alternate and more intuitive way to think about the n-th colored subgraph isomorphism polynomial $\mathrm{ColSub}_{H,n}$ is to consider the blown-up graph G, where each vertex $u \in [k]$ of H is replaced by a 'cloud' of n vertices $C_u := \{(u, 1), \ldots, (u, n)\}$. Every edge $uv \in E(H)$ is replaced by a complete bipartite graph between C_u and C_v with an appropriate label for each of the n^2 edges; that is, an edge between (u, i) and (v, j) is labeled $x_{i,j}^{(uv)}$ where $u, v \in [k]$ and $i, j \in [n]$. The polynomial $\mathrm{ColSub}_{H,n}$ is now obtained by choosing a copy of H in G by picking a vertex from every cloud using a function $f : V(H) \to [n]$, and adding the monomial

$$m = \prod_{uv \in E(H)} x_{f(u),f(v)}^{(uv)}.$$

We say that the monomial $\mathtt{m}$ above is supported on a set $S \subseteq [k] \times [n]$ if every element of S looks like $(u, f(u))$ for $u \in [k]$. The polynomial $\mathtt{ColSub}_{H,n}$ is the sum over all such monomials $\mathtt{m}$

$$\mathtt{ColSub}_{H,n} = \sum_{f:V(H)\to[n]} \prod_{uv\in E(H)} x^{(uv)}_{f(u),f(v)}.$$

Claim 5.3.2. *Let Δ be a natural number and $\mathcal{T}$ be a monotone parse tree of product-depth Δ computing a monomial $\mathtt{m}$ of $\mathtt{ColSub}_{H,n}$. Let $H^\dagger$ be the pruned graph obtained by removing all degree-1 vertices from H. We can extract from $\mathcal{T}$, a tree decomposition of $H^\dagger$ with an underlying tree $T^\dagger$ of height Δ.*

Proof. Suppose that the monomial $\mathtt{m}$ is supported on vertices $(u, f(u))$ where $u \in [k]$ and $f : [k] \to [n]$ is a function. The parse tree $\mathcal{T}$ has height $\Delta + 1$. Note that since $\mathtt{ColSub}_{H,n}$ has $0/1$ coefficients, we can assume that a multiplication gate has only non-constant terms as its children. We build the tree decomposition bottom-up. We 'mark' certain vertices in the bags created during this procedure. All such marks are dropped at the end (see fig. 5.1).

1. For an input gate $x^{(uv)}_{f(u),f(v)}$, we add the bag $\{u, v\}$ as a leaf in the tree decomposition. We *mark* all the vertices of degree 1. The rest are *unmarked*.

2. Let g be a multiplication gate. Suppose $X_1, \ldots, X_m$ are the bags corresponding to the children of g (that we have already constructed) and let $U_i \subseteq X_i$ be the *unmarked* elements of X_i. We then add the bag $X_g := \bigcup_{i\in[m]} U_i$ as the root of $X_1, \ldots, X_m$. If there are vertices $(u, f(u))$ such that the monomial computed at g includes all the edges incident on $(u, f(u))$ in the copy of H that f picked, we *mark* all such vertices u in the bag X_g.

3. Finally, after applying the procedure in the previous step to all the gates, we drop the bags (and edges) corresponding to input gates.

We claim that the tree decomposition we just constructed with underlying tree $T^\dagger$ and bags $\{X_u\}_{u\in V(T^\dagger)}$ is a tree decomposition of $H^\dagger$. Note that *all* the edges of H

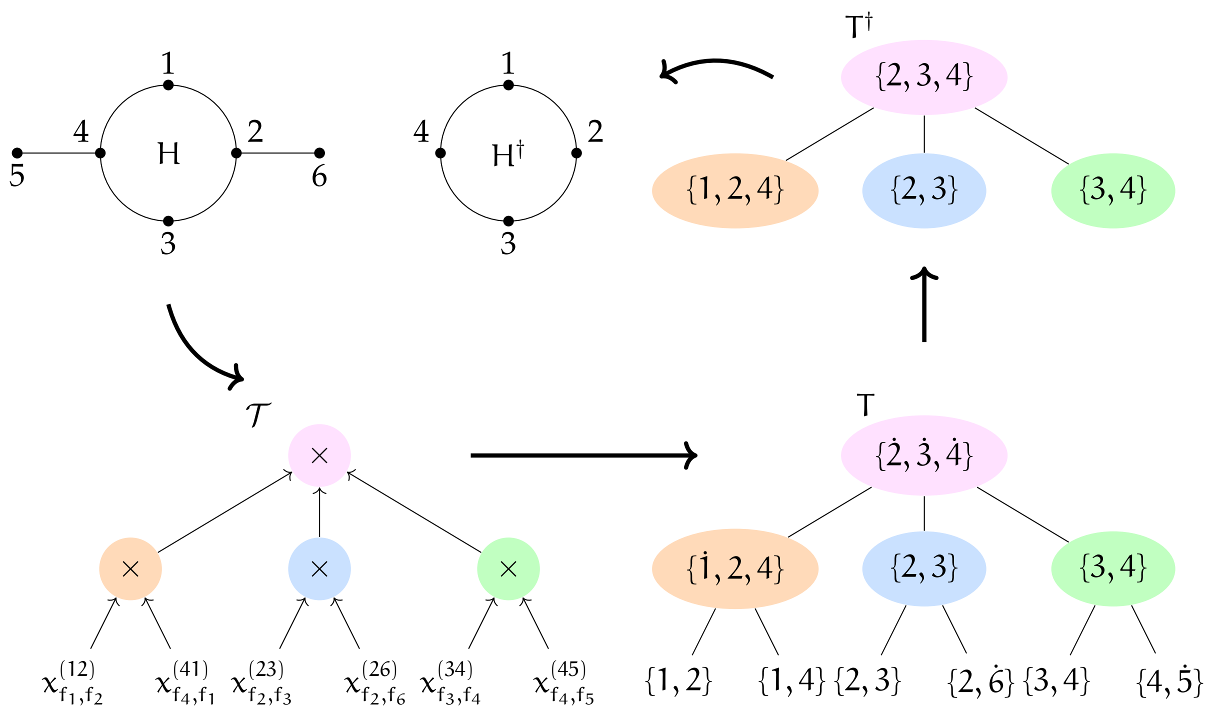


Figure 5.1: Extracting a tree decomposition

Extracting a tree decomposition of height 2 for $H^\dagger$ from a parse-tree of product-depth 2 for a monomial of $\mathsf{ColSub}_{H,n}$. We have for all $i \in [6]$, $f_i := f(i) \in [n]$.

were covered at the leaf bags (that we finally dropped), as they must be present in the monomial. Since only the degree-1 vertices in a leaf bag were *marked*, the parent bags of the leaves (which we include in our tree decomposition) will exactly have the vertices of $H^\dagger$, and thus cover all its edges.

We mark (forget) a vertex only after multiplying all its incident edges. Hence, the sub-graph induced by a vertex u (in $H^\dagger$) is *connected* in $T^\dagger$ and is, in fact, a subtree. As every multiplication gate of the parse tree has exactly one associated bag, the procedure does indeed result in a tree decomposition of $H^\dagger$ of height Δ. □

5.3.2 Lower bounds for $\mathsf{ColSub}_{H,n}$

Theorem 5.3.3. *Let Δ be a natural number and H be a pattern graph. Any monotone circuit of* product-depth Δ *computing the polynomial* $\mathsf{ColSub}_{H,n}$ *has size* $\Omega(n^{\mathrm{ptw}_\Delta(H)+1})$.

Proof. Let C be the monotone circuit computing $\mathsf{ColSub}_{H,n}$, and let the pruned

Δ-treewidth of H, $\mathrm{ptw}_\Delta(H) = t$. Consider a monomial m of $\mathsf{ColSub}_{H,n}$ supported on vertices $(u, f(u))$ for $u \in [k]$ and $f : [k] \to [n]$. Let $\mathcal{T}$ be a parse tree of C associated with m. Now, Claim 5.3.2 gives a tree decomposition of $H^\dagger$ with tree $T^\dagger$ and bags $\{X_u\}_{u \in V(T^\dagger)}$. Consequently, there is a bag X of size at least $t+1$ in the tree decomposition. Without loss of generality, we assume that $|X| = t+1$. If it is greater, we will only obtain a better lower bound. We also assume that the vertices in the bag are $1, \ldots, t+1$ (relabeling the vertices of H does not change the complexity of $\mathsf{ColSub}_{H,n}$). Let the corresponding gate in $\mathcal{T}$ associated with X be g.

We show that only a 'few' monomials can contain g in their parse tree. More precisely, we claim that any monomial m' (other than m) that contains g in its parse tree is supported on vertices $\{(u, f(u))\}_{u \in [t+1]}$. Suppose not. Let m' have a parse tree $\mathcal{T}'$ with gate g in it but vertex $(u, f'(u))$ for some $u \in [t+1]$, with $f(u) \neq f'(u)$. Recall that we obtained the tree decomposition using the parse tree $\mathcal{T}$ of m. For a gate g in a parse tree, we denote by $\mathcal{T}_g$ the subtree rooted at g. Note that if two parse trees contain a multiplication gate g, all the children of g are the same in both the parse trees. We now analyze two cases:

1. The vertex u is marked at the bag associated with g: There are at least two children g_1, g_2 of g in $\mathcal{T}$ that compute monomials with $(u, f(u))$ in them. This holds because there are no degree-1 vertices in the bags. If g_1 in $\mathcal{T}'$ contains the vertex $(u, f'(u))$, we replace $\mathcal{T}'_{g_2}$ with $\mathcal{T}_{g_2}$. Similarly, in the other case, when g_2 contains $(u, f'(u))$. If both g_1, g_2 do not contain $(u, f'(u))$ in $\mathcal{T}'$, we arbitrarily replace $\mathcal{T}'_{g_1}$ (say) with $\mathcal{T}_{g_1}$.

2. The vertex u is not marked at the bag associated with g: The vertex $(u, f(u))$ appears in $\mathcal{T}_g$ *as well as* outside $\mathcal{T}_g$. In $\mathcal{T}'$, if $(u, f'(u))$ appears in $\mathcal{T}'_g$, we replace $\mathcal{T}_g$ with $\mathcal{T}'_g$ in $\mathcal{T}$. Otherwise, we replace $\mathcal{T}'_g$ with $\mathcal{T}_g$ in $\mathcal{T}'$.

In all cases, we obtain a valid parse tree $\mathcal{T}''$ of C that produces a monomial supported on $(u, f(u))$ *and* $(u, f'(u))$. This leads to a contradiction, since the monomial produced by $\mathcal{T}''$ is spurious and cannot be cancelled because the circuit is monotone. Every monomial (parse tree) m has a gate g whose corresponding bag has at least $t+1$ vertices. And any other monomial m' (parse tree) that contains this gate g must share at least $t+1$ vertices in its support with m. Thus, the maximum number of monomials containing this gate g equals the number of colored isomorphisms that fix $t+1$ vertices, which is n^{k-t-1}. Recall that there are n^k monomials in $\mathsf{ColSub}_{H,n}$, and so we need at least n^{t+1} gates in the circuit. $\square$

The lower bound proof for algebraic branching programs is very similar.

Theorem 5.3.4. *Let Δ be a positive integer and H be a pattern graph such that $\Delta \geq |E(H)|$. Any monotone ABP of* length Δ *computing the polynomial* $\mathsf{ColSub}_{H,n}$ *has size* $\Omega(n^{\mathrm{ppw}_\Delta(H)+1})$.

Proof. As mentioned earlier in Remark 2.2.12, the size-s monotone ABP of length Δ computing $\mathsf{ColSub}_{H,n}$ has an equivalent monotone skew-circuit C of size $O(s)$ and product-depth Δ. Consider a monomial m of $\mathsf{ColSub}_{H,n}$ supported on vertices $(u, f(u))$ for $u \in [k]$ and $f : [k] \to [n]$. Let $\mathcal{T}$ be a parse tree of C associated with m.

We observe that the procedure described in the proof of Claim 5.3.2 extracts a length-Δ *path decomposition* of the pruned graph $H^\dagger$ instead: as the circuit is skew, all the multiplication gates in $\mathcal{T}$ have at most one non-leaf child. Since we finally dropped the bags corresponding to input gates in our procedure, the tree decomposition we obtain is in fact a path decomposition!

Taking the pruned Δ-pathwidth of H to be t, the same proof implies that the number of monomials containing a particular gate g is n^{k-t-1}, thus implying a size lower bound of n^{t+1}. $\square$

5.4 Monotone depth hierarchy

Combining our previous results allows us to prove a depth-hierarchy theorem for bounded-depth monotone algebraic circuits.[2]

[2] A similar hierarchy can also be shown for monotone ABPs.

Theorem 5.4.1. *For all integers* n *and* Δ*, there exists a pattern graph* H_Δ *such that* $\mathrm{ColSub}_{H_\Delta,n}$ *can be computed by a monotone product-depth* $(\Delta+1)$ *circuit of size* $O(n|H_\Delta|)$ *but any product-depth* Δ *monotone circuit computing the polynomial needs size* $n^{\Omega(|H|^{1/\Delta})}$.

Proof. For an integer d, let $H_\Delta := T_{\Delta+2}$ be the full d-ary tree of height $\Delta + 2$. Note that $d = \Theta(|H_\Delta|^{1/\Delta})$. The pruned $(\Delta + 1)$-treewidth of H_Δ is equal to the $(\Delta + 1)$-treewidth of the full d-ary tree of height $(\Delta + 1)$. That is, $ptw_{\Delta+1}(H_\Delta) = tw_{\Delta+1}(T_{\Delta+1}) = 1$. So by Lemma 5.2.1, there exist a monotone circuit of product-depth $\Delta + 1$ and size $O(n|H_\Delta|)$, which computes $\mathrm{ColSub}_{H_\Delta,n}$.

On the other hand, $ptw_\Delta(H_\Delta) = tw_\Delta(T_{\Delta+1}) \geq d-1$ by Theorem 5.1.5. Hence, by Theorem 5.3.3, any monotone circuit of product-depth Δ computing $\mathrm{ColSub}_{H_\Delta,n}$ has size at least $\Omega(n^{ptw_\Delta(H_\Delta)+1}) = n^{\Omega(|H_\Delta|^{1/\Delta})}$. □

If we consider the case when the pattern graph is of size $\Theta(n)$ in Theorem 5.4.1, we obtain our main depth hierarchy result.

Theorem 5.4.2. *For any natural numbers* n *and* Δ*, there exists a pattern graph* H_Δ *of size* $\Theta(n)$ *such that* $\mathrm{ColSub}_{H_\Delta,n}$ *can be computed by a monotone circuit of size* $\mathrm{poly}(n)$ *and product-depth* $(\Delta + 1)$*, but any monotone circuit of product-depth* Δ *computing the polynomial needs size* $n^{\Omega(n^{1/\Delta})}$.

As an aside, we note that an analogous depth hierarchy *cannot* be obtained for the polynomials $\mathrm{Hom}_{H,n}$ using our methods, as the blow up in the size given by Lemma 5.3.1 is exponential, when $|H| = \Theta(n)$ is not a constant.

Chapter 6

Polynomial Factorization

In this chapter, we suggest natural conjectures and conditional results that make progress towards resolving the Factor Conjecture of Bürgisser. From the discussion in Section 1.4, it is enough to study the presentable border class $\overline{\mathsf{VP}}_{\varepsilon}$. We begin with a brief proof sketch of the $\overline{\mathsf{VP}}_{\varepsilon} \subseteq \mathsf{VNP}$ result over finite fields. For full details, refer [BDS24b] or [Dwi24].

Theorem 6.0.1 ([BDS24a, Theorem 1]). *Let $\mathbb{F}$ be a finite field, and $f \in \mathbb{F}[\mathbf{x}]$ be an n–variate polynomial of degree d such that the size of the smallest circuit (over $\mathbb{F}$) approximating f is of size s. Then, f can be written as*

$$f(\mathbf{x}) = \sum_{\mathbf{b} \in \{0,1\}^m} g(\mathbf{x}, \mathbf{b}),$$

where m, and $\mathsf{size}_{\mathbb{F}}(g(\mathbf{x}, \boldsymbol{y}))$ are bounded by $\mathsf{poly}(s, n, d)$.

Proof Sketch. The main idea is to use Valiant's criterion (Proposition 2.2.19) to show that the polynomial $f = \sum_{\mathbf{e}} c_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}$ has coefficients $c_{\mathbf{e}}$ that are "easy to describe". Notice that the polynomial F approximating f has a small circuit but exponential degree, where the high degree is only due to ε.

In the approximating expression for f (Equation (1.4.2)), we interpolate F on *all* the variables (including ε) using appropriate powers of unity. Consequently, each coefficient $c_{\mathbf{e}}$ of f can be written as a hypercube sum over a small-size high-degree circuit. When $\mathbb{F}$ is a finite field, we can *simulate* this algebraic circuit using a small boolean circuit. As a result, we obtain that the coefficient function of f is in $\#\mathsf{P}/\mathsf{poly}$, whence we can apply Valiant's criterion. $\square$

It is not clear how to use Valiant's criterion over fields of characteristic zero since in this case, there are polynomial families in $\overline{\mathsf{VP}}_\varepsilon$ with very large coefficients. Hence, we cannot hope to show that the coefficient function is in $\#\mathsf{P}/\mathsf{poly}$. We suggest a different approach.

6.1 The reverse Kronecker conjecture

Conjecture 6.1.1 (Reverse Kronecker)**.** Let $\mathbb{F}$ be any field and suppose $F \in \mathbb{F}[\varepsilon]$ is a *univariate* polynomial computed by a circuit of size s. Then, there is a polynomial $\tilde{F} \in \mathbb{F}[\mathbf{z}, \mathbf{w}]$ whose number of variables $|\mathbf{z}| + |\mathbf{w}|$, its *individual* degree r, and its size $\tilde{s}$ are all bounded by $\mathsf{poly}(s)$ satisfying

$$F(\varepsilon) = \tilde{F}(\mathbf{z}, \mathbf{a})\Big|_{\mathbf{z}=\varphi(\mathbf{z})},$$

under the Kronecker map $\varphi : z_i \mapsto \varepsilon^{(r+1)^{i-1}}$ applied to the $\mathbf{z}$ variables, and some assignment $\mathbf{a} \in \mathbb{F}^{|\mathbf{w}|}$ to the $\mathbf{w}$ variables.

Remark 6.1.2. If the univariate circuit is of low-degree, then the conjecture holds. For e.g., if the circuit is of constant depth and size s, then the degree, and hence number of monomials is itself small. Each univariate monomial (and hence the whole polynomial) can be obtained via Kronecker substitution to a small circuit.

The Reverse Kronecker conjecture helps in *debordering* presentable classes over any field. We begin by demonstrating this when we have a small circuit (over the base field) approximating our polynomial.

Theorem 6.1.3. *Let $\mathbb{F}$ be any field, and let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial such that there an* *approximating polynomial* $F \in F[\varepsilon][\mathbf{x}]$ *with* $\mathsf{size}_{\mathbb{F}}(F) = s$*, and for some integer* M*,*

$$F(\mathbf{x}, \varepsilon) = \varepsilon^{M} f(\mathbf{x}) + \varepsilon^{M+1} Q(\mathbf{x}, \varepsilon).$$

Suppose now that Conjecture 6.1.1 is true. Then, f *can be written as*

$$f(\mathbf{x}) = \sum_{\mathbf{b} \in \{0,1\}^{\ell}} g(\mathbf{x}, \mathbf{b}),$$

where ℓ *and* $\mathsf{size}(g)$ *are both bounded by* $\mathsf{poly}(n, s)$*.*

Proof. We are given that $f(\mathbf{x})$ can be approximated by a polynomial $F \in \mathbb{F}[\varepsilon][\mathbf{x}]$ with $\mathsf{size}_{\mathbb{F}}(F) = s$:

$$F(\mathbf{x}, \varepsilon) = \varepsilon^{M} f(\mathbf{x}) + \varepsilon^{M+1} Q(\mathbf{x}, \varepsilon) \tag{6.1.1}$$

Assuming Conjecture 6.1.1 to be true, we can replace all the univariate circuits of ε in F with $\mathsf{poly}(s)$-sized circuits that agree with the univariate circuit under the Kronecker map. Replacing every $\mathbb{F}[\varepsilon]$-circuit in this way, we get that there is a $\mathsf{poly}(n, s)$-variate polynomial $\tilde{F} \in \mathbb{F}[\mathbf{x}, \mathbf{z}, \mathbf{w}]$ of individual degree $r = \mathsf{poly}(s)$ and size $\tilde{s} = \mathsf{poly}(s)$ that agrees with F under the Kronecker map $\varphi : z_i \mapsto \varepsilon^{(r+1)^{i-1}}$, and an assignment $\mathbf{a} \in \mathbb{F}^{|\mathbf{w}|}$ to the $\mathbf{w}$ variables:

$$F(\mathbf{x}, \varepsilon) = \tilde{F}(\mathbf{x}, \mathbf{z}, \mathbf{a})\Big|_{\mathbf{z}=\varphi(\mathbf{z})}. \tag{6.1.2}$$

Combining Equation (6.1.1) and Equation (6.1.2), we also have

$$\tilde{F}(\mathbf{x}, \mathbf{z}, \mathbf{a})\Big|_{\mathbf{z}=\varphi(\mathbf{z})} = \varepsilon^{M} f(\mathbf{x}) + \varepsilon^{M+1} Q(\mathbf{x}, \varepsilon). \tag{6.1.3}$$

We claim that even without the map φ, $f(\mathbf{x})$ occurs as the trailing coefficient in $\tilde{F}(\mathbf{x}, \mathbf{z}, \mathbf{a})$. More precisely, we claim that the following equation holds.

$$\tilde{F}(\mathbf{x}, \mathbf{z}, \mathbf{a}) = \mathbf{z}^{\mathbf{M}} f(\mathbf{x}) + \mathbf{z}^{\succ \mathbf{M}} \tilde{Q}(\mathbf{x}, \mathbf{z}), \tag{6.1.4}$$

for some polynomial $\tilde{Q}$, and $\mathbf{M} = (M_1, M_2, \ldots, M_{|\mathbf{z}|})$ comes from the 'digits' in the base-$(r+1)$ expansion: $M = M_1 + M_2(r+1) + \ldots + M_{|\mathbf{z}|}(r+1)^{|\mathbf{z}|-1}$.

To see that Equation (6.1.4) holds, note that the map φ assigns different powers of ε to different $\mathbf{z}$ monomials of individual degree at most r [Kro82]. Hence, $\mathbf{z}^{\mathbf{M}}$ is the only monomial that gets mapped to ε^M. Since we know that Equation (6.1.3) holds under φ, we must have the term $\mathbf{z}^{\mathbf{M}} f(\mathbf{x})$ appearing somewhere in $\tilde{F}$. Moreover, this has to be the trailing term since any other term $\mathbf{z}^{\mathbf{L}}$ for $\mathbf{L} \prec \mathbf{M}$ would be mapped to a lower power of ε than M and hence not vanish in Equation (6.1.3).

Crucially, Equation (6.1.4) is much better than Equation (6.1.1) for the purpose of debordering since the low-degree small-size circuit $\tilde{F}$ is amenable to interpolation (Proposition 2.2.20). We can extract f as a hypercube sum

$$f(\mathbf{x}) = \sum_{\mathbf{b} \in \{0,1\}^{\ell}} g(\mathbf{x}, \mathbf{b}),$$

where ℓ is bounded by $\mathsf{poly}(|\mathbf{z}| + |\mathbf{w}|) = \mathsf{poly}(n, s)$ and $\mathsf{size}(g)$ is bounded by $\mathsf{poly}(\mathsf{size}(\tilde{F})) = \mathsf{poly}(n, s)$.

$\square$

Let (f_n) be a polynomial family in $\overline{\mathsf{VP}}_{\varepsilon}$. Then, the size s in Theorem 6.1.3 is $\mathsf{poly}(n)$ and thus we get that for any $n \in \mathbb{N}$, we can write f_n as a hypercube sum $f_n(\mathbf{x}) = \sum_{\mathbf{b} \in \{0,1\}^{\ell}} g_n(\mathbf{x}, \mathbf{b})$, where ℓ and $\mathsf{size}(g)$ are both bounded by $\mathsf{poly}(n)$. Hence, we get

Corollary 6.1.4 (of Theorem 6.1.3)**.** *Let $\mathbb{F}$ be any field, and suppose Conjecture 6.1.1 holds. Then, we have $\overline{\mathsf{VP}}_{\varepsilon} \subseteq \mathsf{VNP}$.*

Remark 6.1.5. Note that in the proof of Theorem 6.1.3, we did not necessarily need $\tilde{F}$ to be a small circuit. Even if it was a hypercube sum of small circuits, the proof would still go through. Hence, we can relax Conjecture 6.1.1 to allow the univariate circuit to be equivalent under the Kronecker map, to a hypercube sum of small circuits instead.

We also note that Conjecture 6.1.1 cannot be used to show $\overline{\mathsf{VP}}_\varepsilon \subseteq \mathsf{VP}$ since our proof works by extracting coefficients, and VP is *not* closed under taking coefficients, unless VP = VNP [Bür00a, Section 2.3].

For our problem of polynomial factorization, we also get the following explicitness of roots over all fields, assuming the reverse Kronecker conjecture.

Corollary 6.1.6 (of Theorem 6.1.3)**.** *Let $\mathbb{F}$ be any field, and suppose Conjecture 6.1.1 is true over $\mathbb{F}$. Let $f(\mathbf{x}, y)$ be an $(n+1)$-variate polynomial computed by a circuit of size s. Suppose $\phi(\mathbf{x})$ is a power series root of f with respect to y (i.e., $f(\mathbf{x}, \phi(\mathbf{x})) = 0$). Then the truncation of ϕ up to degree $d = \mathrm{poly}(s)$ can be written as*

$$[\phi]_{\leq d} = \sum_{\mathbf{b} \in \{0,1\}^m} g(\mathbf{x}, \mathbf{b}),$$

where m, and $\mathrm{size}_{\mathbb{F}}(g(\mathbf{x}, \boldsymbol{y}))$ are bounded by $\mathrm{poly}(s, n, d)$.

Proof. We know that $[\phi]_{\leq d}$ can be approximated by a presentable circuit of size $\mathrm{poly}(s)$. Applying Theorem 6.1.3 then gives the required result. □

As a consequence of the above corollary, assuming Conjecture 6.1.1 also implies that over any field, low-degree factors of polynomial families with small circuits (of high degree) are in VNP.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity*. Cambridge University Press, Cambridge, 2009. ISBN: 978-0-521-42426-4.

[AF24] Isolde Adler and Eva Fluck. *Monotonicity of the Cops and Robber Game for Bounded Depth Treewidth*. In: *49th International Symposium on Mathematical Foundations of Computer Science*. Vol. 306. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2024, Art. No. 6, 18. ISBN: 978-3-95977-335-5.

[Agr06] Manindra Agrawal. *Determinant versus Permanent*. In: *International Congress of Mathematicians. Vol. III*. Eur. Math. Soc., Zürich, 2006, pp. 985–997.

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. *Bootstrapping variables in algebraic circuits*. In: *Proceedings of the National Academy of Sciences of the United States of America* 116.17 (2019), pp. 8107–8118. ISSN: 0027-8424,1091-6490.

[All+98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. *Non-Commutative Arithmetic Circuits: Depth Reduction and Size Lower Bounds*. In: *Theoretical Computer Science* 209.1-2 (1998), pp. 47–86.

[Ami+23] Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. *Low-Depth Arithmetic Circuit Lower Bounds: Bypassing Set-Multilinearization*. In: *50th International Colloquium on Automata, Languages, and Programming*. Vol. 261. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023, Art. No. 12, 20. ISBN: 978-3-95977-278-5.

[And+18] Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. *Identity testing and lower bounds for read-k oblivious algebraic branching programs*. In: *ACM Transactions on Computation Theory* 10.1 (2018), Art. 3, 30. ISSN: 1942-3454,1942-3462.

[AR16] V. Arvind and S. Raja. *Some lower bound results for set-multilinear arithmetic computations*. In: *Chicago Journal of Theoretical Computer Science* (2016), Art. 6, 26. ISSN: 1073-0486.

[AV08] Manindra Agrawal and V. Vinay. *Arithmetic circuits: A chasm at depth four*. In: *49th annual IEEE symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2008, pp. 67–75.

[BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory*. Vol. 315. Grundlehren der mathematischen wissenschaften [fundamental principles of mathematical sciences]. Springer-Verlag, Berlin, 1997. ISBN: 3-540-60582-7.

[BDS24a] C. S. Bhargav, Sagnik Dutta, and Nitin Saxena. *Improved Lower Bound, and Proof Barrier, for Constant Depth Algebraic Circuits*. In: *ACM Transactions on Computation Theory* 16.4 (2024), Art. 23, 22. ISSN: 1942-3454,1942-3462. Conf. version - MFCS 2022.

[BDS24b] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *Learning the Coefficients: A Presentable Version of Border Complexity and Applications to Circuit Factoring*. In: *STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. ACM, New York, 2024, pp. 130–140.

[BDS25a] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *A Primer on the Closure of Algebraic Complexity Classes under Factoring*. 2025. arXiv: `2506.19604 [cs]`. The preprint is under review in the special issue of the workshop RTCA'23 Paris.

[BDS25b] C. S. Bhargav, Prateek Dwivedi, and Nitin Saxena. *Lower Bounds for the Sum of Small-Size Algebraic Branching Programs*. In: *Theoretical Computer Science* 1041 (2025), Paper No. 115214, 11. ISSN: 0304-3975. Conf. version - TAMC 2024.

[BES87] C. A. Barefoot, Roger Entringer, and Henda Swart. *Vulnerability in Graphs—a Comparative Survey*. In: *Journal of Combinatorial Mathematics and Combinatorial Computing* 1 (1987), pp. 13–22. ISSN: 0835-3026,2817-576X.

[BH98] Hans L. Bodlaender and Torben Hagerup. *Tree Decompositions of Small Diameter*. In: *Mathematical Foundations of Computer Science, 1998 (Brno)*. Vol. 1450. Lecture Notes in Comput. Sci. Springer, Berlin, 1998, pp. 702–712. ISBN: 978-3-540-64827-7.

[Bha+25] C. S. Bhargav, Shiteng Chen, Radu Curticapean, and Prateek Dwivedi. *Monotone Bounded-Depth Complexity of Homomorphism Polynomials*. 2025. arXiv: `2505.22894 [cs]`. To appear in 50th International Symposium on Mathematical Foundations of Computer Science (MFCS) 2025.

[BI25] Markus Bläser and Christian Ikenmeyer. *Introduction to Geometric Complexity Theory*. In: *Theory of Computing. An Open Access Journal*. Graduate Surveys 10 (2025), p. 166.

[BIZ18] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. *On Algebraic Branching Programs of Small Width*. In: *Journal of The ACM* 65.5 (2018), Art. 32, 29. ISSN: 0004-5411.

[BS83] Walter Baur and Volker Strassen. *The complexity of partial derivatives*. In: *Theoretical Computer Science* 22.3 (1983), pp. 317–330. ISSN: 0304-3975.

[Bür+11] Peter Bürgisser, J. M. Landsberg, Laurent Manivel, and Jerzy Weyman. *An overview of mathematical issues arising in the geometric complexity theory approach to $\mathsf{VP} \neq \mathsf{VNP}$*. In: *SIAM Journal on Computing* 40.4 (2011), pp. 1179–1209. ISSN: 0097-5397,1095-7111.

[Bür00a] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*. Vol. 7. Algorithms and computation in mathematics. Springer-Verlag, Berlin, 2000. ISBN: 3-540-66752-0.

[Bür00b] Peter Bürgisser. *Cook's versus Valiant's Hypothesis*. In: *Theoretical Computer Science* 235 (2000), pp. 71–88. ISSN: 0304-3975,1879-2294.

[Bür04] Peter Bürgisser. *The Complexity of Factors of Multivariate Polynomials*. In: *Foundations of Computational Mathematics* 4.4 (2004), pp. 369–396. ISSN: 1615-3375.

[Bür20] Peter Bürgisser. *Correction to: The Complexity of Factors of Multivariate Polynomials*. In: *Foundations of Computational Mathematics* 20.6 (2020), pp. 1667–1668. ISSN: 1615-3375,1615-3383.

[Bür24] Peter Bürgisser. *Completeness Classes in Algebraic Complexity Theory*. 2024. arXiv: `2406.06217 [cs]`.

[Bür99] Peter Bürgisser. *On the Structure of Valiant's Complexity Classes*. In: *Discrete Mathematics & Theoretical Computer Science. DMTCS. An Electronic Journal* 3.3 (1999), pp. 73–94. ISSN: 1365-8050.

[CDM17] Radu Curticapean, Holger Dell, and Dániel Marx. *Homomorphisms Are a Good Basis for Counting Small Subgraphs*. In: *STOC'17—Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, New York, 2017, pp. 210–223. ISBN: 978-1-4503-4528-6.

[Cha+22] Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. *Quadratic lower bounds for algebraic branching programs and formulas*. In: *Computational Complexity* 31.2 (2022), Paper No. 8, 54. ISSN: 1016-3328.

[Cha+24] Prerona Chatterjee, Deepanshu Kush, Shubhangi Saraf, and Amir Shpilka. *Lower bounds for set-multilinear branching programs*. In: *39th Computational Complexity Conference*. Vol. 300. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2024, Art. No. 20, 20. ISBN: 978-3-95977-331-7.

[Chi+18] Suryajith Chillara, Christian Engels, Nutan Limaye, and Srikanth Srinivasan. *A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits*. In: *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*. IEEE Computer Soc., Los Alamitos, CA, 2018, pp. 934–945. ISBN: 978-1-5386-4230-6.

[CIP16] Krishnendu Chatterjee, Rasmus Ibsen-Jensen, and Andreas Pavlogiannis. *Optimal Reachability and a Space-Time Tradeoff for Distance Queries in Constant-Treewidth Graphs*. In: *24th Annual European Symposium on Algorithms*. Vol. 57. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, Art. No. 28, 17. ISBN: 978-3-95977-015-6.

[CKR22] Bruno Pasqualotto Cavalar, Mrinal Kumar, and Benjamin Rossman. *Monotone Circuit Lower Bounds from Robust Sunflowers*. In: *Algorithmica. An International Journal in Computer Science* 84.12 (2022), pp. 3655–3685. ISSN: 0178-4617,1432-0541.

[CKW10] Xi Chen, Neeraj Kayal, and Avi Wigderson. *Partial derivatives in arithmetic complexity and beyond*. In: *Foundations and Trends in Theoretical Computer Science* 6.1-2 (2010), front matter, 1–138. ISSN: 1551-305X,1551-3068.

[CLS19] Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. *Small-depth multilinear formula lower bounds for iterated matrix multiplication with applications*. In: *SIAM Journal on Computing* 48.1 (2019), pp. 70–92. ISSN: 0097-5397,1095-7111.

[CLV21] Prasad Chaugule, Nutan Limaye, and Aditya Varre. *Variants of Homomorphism Polynomials Complete for Algebraic Complexity Classes*. In: *ACM Transactions on Computation Theory* 13.4 (2021), Art. 21, 26. ISSN: 1942-3454,1942-3462.

[CM14] Radu Curticapean and Dániel Marx. *Complexity of Counting Subgraphs: Only the Boundedness of the Vertex-Cover Number Counts*. In: *55th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2014*. IEEE Computer Soc., Los Alamitos, CA, 2014, pp. 130–139. ISBN: 978-1-4799-6517-5.

[CM16] Hubie Chen and Stefan Mengel. *Counting Answers to Existential Positive Queries: A Complexity Classification*. In: *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*. Ed. by Tova Milo and Wang-Chiew Tan. ACM, 2016, pp. 315–326.

[con25] Wikipedia contributors. *Bernoulli's inequality*. In: *Wikipedia* (May 2025).

[Coo71] Stephen A. Cook. *The complexity of theorem-proving procedures*. In: *Proceedings of the 3rd annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1971, pp. 151–158.

[DG24] Pranjal Dutta and Sumanta Ghosh. *Complexity Theory Column 121: Advances in Polynomial Identity Testing*. In: *ACM SIGACT News* 55.2 (2024), pp. 53–88. ISSN: 0163-5700.

[DJ04] Víctor Dalmau and Peter Jonsson. *The Complexity of Counting Homomorphisms Seen from the Other Side*. In: *Theoretical Computer Science* 329.1-3 (2004), pp. 315–323. ISSN: 0304-3975,1879-2294.

[DSS22] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. *Discovering the Roots: Uniform Closure Results for Algebraic Classes under Factoring*. In: *Journal of The ACM* 69.3 (2022), Art. 18, 39. ISSN: 0004-5411.

[Dur+16] Arnaud Durand, Meena Mahajan, Guillaume Malod, Nicolas de Rugy-Altherre, and Nitin Saurabh. *Homomorphism polynomials complete for* VP. In: *Chicago Journal of Theoretical Computer Science* (2016), Art. 3, 25. ISSN: 1073-0486.

[Dut23] Sagnik Dutta. *Lower Bounds for Constant Depth Algebraic Circuits*. Chennai Mathematical Institute, 2023. Master's Thesis.

[Dvi+12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. *Separating multilinear branching programs and formulas*. In: *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*. ACM, New York, 2012, pp. 615–623. ISBN: 978-1-4503-1245-5.

[Dwi24] Prateek Dwivedi. *Treading the Borders for Explicitness, Circuit Factoring, and Identity Testing*. PhD thesis. Indian Institute of Technology Kanpur, 2024.

[For14] Michael Andrew Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis. Massachusetts Institute of Technology, 2014.

[For24] Michael A. Forbes. *Low-Depth Algebraic Circuit Lower Bounds over Any Field*. In: *39th Computational Complexity Conference*. Vol. 300. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2024, Art. No. 31, 16. ISBN: 978-3-95977-331-7.

[Fou+15] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. *Lower bounds for depth-4 formulas computing iterated matrix multiplication*. In: *SIAM Journal on Computing* 44.5 (2015), pp. 1173–1201. ISSN: 0097-5397,1095-7111.

[FS13] Michael A. Forbes and Amir Shpilka. *Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs*. In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science—FOCS 2013*. IEEE Computer Soc., Los Alamitos, CA, 2013, pp. 243–252. ISBN: 978-0-7695-5135-7.

[Gat06] Joachim von zur Gathen. *Who Was Who in Polynomial Factorization: 1*. In: *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation*. ISSAC '06. Association for Computing Machinery, 2006, p. 2. ISBN: 1-59593-276-3.

[GG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Third. Cambridge University Press, Cambridge, 2013. ISBN: 978-1-107-03903-2.

[Gim+25] Tatsuya Gima, Tesshu Hanaka, Yasuaki Kobayashi, Ryota Murai, Hirotaka Ono, and Yota Otachi. *Structural Parameterizations of Vertex Integrity*. In: *Theoretical Computer Science* 1024 (2025), Paper No. 114954, 16. ISSN: 0304-3975,1879-2294.

[GP01] Joachim von zur Gathen and Daniel Panario. *Factoring Polynomials over Finite Fields: A Survey*. In: *J. Symbolic Comput.* 31.1 (2001), pp. 3–17. ISSN: 0747-7171.

[GR21] Purnata Ghosal and B. V. Raghavendra Rao. *Limitations of sums of bounded read formulas and ABPs*. In: *Computer science—theory and applications*. Vol. 12730. Lecture Notes in Comput. Sci. Springer, Cham, 2021, pp. 147–169. ISBN: 978-3-030-79415-6.

[Gro12] Joshua A. Grochow. *Symmetry and Equivalence Relations in Classical and Geometric Complexity Theory*. PhD thesis. University of Chicago, 2012.

[Gro13] Joshua Grochow. *Degree restriction for polynomials in* VP. Theoretical Computer Science Stack Exchange. URL: https://cstheory.stackexchange.com/q/19268 (version: 2013-10-03). 2013.

[GS12] Sergey B. Gashkov and Igor' S. Sergeev. *A Method for Deriving Lower Bounds for the Complexity of Monotone Arithmetic Circuits Computing Real Polynomials*. In: *Sbornik: Mathematics* 203.10 (Oct. 2012), p. 1411. ISSN: 1064-5616.

[GST20] Nikhil Gupta, Chandan Saha, and Bhargav Thankey. *A super-quadratic lower bound for depth four arithmetic circuits*. In: *35th Computational Complexity Conference*. Vol. 169. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020, Art. No. 23, 31. ISBN: 978-3-95977-156-6.

[Guo+22] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. *Derandomization from algebraic hardness*. In: *SIAM Journal on Computing* 51.2 (2022), pp. 315–335. ISSN: 0097-5397,1095-7111.

[Gup+14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. *Approaching the chasm at depth four*. In: *Journal of the ACM* 61.6 (2014), Art. 33, 16. ISSN: 0004-5411,1557-735X.

[Gup+16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. *Arithmetic circuits: a chasm at depth 3*. In: *SIAM Journal on Computing* 45.3 (2016), pp. 1064–1079. ISSN: 0097-5397,1095-7111.

[HY11] Pavel Hrubeš and Amir Yehudayoff. *Arithmetic Complexity in Ring Extensions*. In: *Theory of Computing. An Open Access Journal* 7 (2011), pp. 119–129.

[HY16] Pavel Hrubeš and Amir Yehudayoff. *On Isoperimetric Profiles and Computational Complexity*. In: *43rd International Colloquium on Automata, Languages, and Programming*. Vol. 55. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, Art. No. 89, 12. ISBN: 978-3-95977-013-2.

[IL17] Christian Ikenmeyer and J. M. Landsberg. *On the Complexity of the Permanent in Various Computational Models*. In: *Journal of Pure and Applied Algebra* 221.12 (2017), pp. 2911–2927. ISSN: 0022-4049,1873-1376.

[IP01] Russell Impagliazzo and Ramamohan Paturi. *On the Complexity of K-SAT*. In: *Journal of Computer and System Sciences* 62 (2001), pp. 367–375. ISSN: 0022-0000,1090-2724.

[JS82] Mark Jerrum and Marc Snir. *Some Exact Complexity Results for Straight-Line Computations over Semirings*. In: *Journal of the Association for Computing Machinery* 29.3 (1982), pp. 874–897. ISSN: 0004-5411,1557-735X.

[Kal82] Erich Kaltofen. *Factorization of Polynomials*. In: *Computer Algebra: Symbolic and Algebraic Computation*. Vienna: Springer, 1982, pp. 95–113. ISBN: 978-3-7091-3406-1.

[Kal85a] K. A. Kalorkoti. *A lower bound for the formula size of rational functions*. In: *SIAM Journal on Computing* 14.3 (1985), pp. 678–687. ISSN: 0097-5397.

[Kal85b] Erich Kaltofen. *Polynomial-Time Reductions from Multivariate to Bi- and Univariate Integral Polynomial Factorization*. In: *SIAM Journal On Computing* 14.2 (1985), pp. 469–489. ISSN: 0097-5397.

[Kal86] Erich L. Kaltofen. *Uniform Closure Properties of P-Computable Functions*. In: *STOC1986*. ACM, 1986, pp. 330–337.

[Kal87] Erich Kaltofen. *Single-Factor Hensel Lifting and Its Application to the Straight-Line Complexity of Certain Polynomials*. In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. Association for Computing Machinery, 1987. ISBN: 0-89791-221-7.

[Kal89] Erich Kaltofen. *Factorization of Polynomials Given by Straight-Line Programs*. In: *Adv. Comput. Res.* 5 (1989), pp. 375–412.

[Kal90] Erich Kaltofen. *Polynomial Factorization 1982–1986*. In: *Computers in mathematics*. CRC Press, 1990, pp. 285–309.

[Kal92] Erich Kaltofen. *Polynomial Factorization 1987–1991*. In: *LATIN '92 (São Paulo, 1992)*. Vol. 583. Lecture Notes in Comput. Sci. Springer, Berlin, 1992, pp. 294–313.

[Kar+24] C. S. Karthik, Dániel Marx, Marcin Pilipczuk, and Uéverton Souza. *Conditional Lower Bounds for Sparse Parameterized 2-CSP: A Streamlined Proof*. In: *2024 Symposium on Simplicity in Algorithms (SOSA)*. SIAM, Philadelphia, PA, 2024, pp. 383–395. ISBN: 978-1-61197-793-6.

[Kay+17] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. *An exponential lower bound for homogeneous depth four arithmetic formulas*. In: *SIAM Journal on Computing* 46.1 (2017), pp. 307–335. ISSN: 0097-5397,1095-7111.

[Kay12] Neeraj Kayal. *An exponential lower bound for the sum of powers of bounded degree polynomials*. 2012. Electronic Colloquium on Computational Complexity: `TR12-081`.

[Koi12] Pascal Koiran. *Arithmetic circuits: the chasm at depth four gets wider*. In: *Theoretical Computer Science* 448 (2012), pp. 56–65. ISSN: 0304-3975,1879-2294.

[KPR23] Balagopal Komarath, Anurag Pandey, and C. S. Rahul. *Monotone Arithmetic Complexity of Graph Homomorphism Polynomials*. In: *Algorithmica. An International Journal in Computer Science* 85.9 (2023), pp. 2554–2579. ISSN: 0178-4617.

[Kro82] L. Kronecker. *Grundzüge Einer Arithmetischen Theorie Der Algebraische Grössen*. In: *Journal Fur Die Reine Und Angewandte Mathematik* 92 (1882), pp. 1–122. ISSN: 0075-4102.

[KS06] Adam R. Klivans and Amir Shpilka. *Learning restricted models of arithmetic circuits*. In: *Theory of Computing. An Open Access Journal* 2 (2006), pp. 185–206.

[KS15] Mrinal Kumar and Shubhangi Saraf. *The limits of depth reduction for arithmetic formulas: it's all about the top fan-in*. In: *SIAM Journal on Computing* 44.6 (2015), pp. 1601–1625. ISSN: 0097-5397,1095-7111.

[KS17] Neeraj Kayal and Chandan Saha. *Multi-k-ic depth three circuit lower bound*. In: *Theory of Computing Systems* 61.4 (2017), pp. 1237–1251. ISSN: 1432-4350,1433-0490.

[KS19] Mrinal Kumar and Ramprasad Saptharishi. *Hardness-randomness tradeoffs for algebraic computation*. In: *Bulletin of the European Association for Theoretical Computer Science. EATCS* 129 (2019), pp. 56–87. ISSN: 0252-9742.

[KS22] Deepanshu Kush and Shubhangi Saraf. *Improved low-depth set-multilinear circuit lower bounds*. In: *37th Computational Complexity Conference*. Vol. 234. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022, Art. No. 38, 16. ISBN: 978-3-95977-241-9.

[KS23] Deepanshu Kush and Shubhangi Saraf. *Near-optimal set-multilinear formula lower bounds*. In: *38th Computational Complexity Conference*. Vol. 264. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2023, Art. No. 15, 33. ISBN: 978-3-95977-282-2.

[KSS14] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. *A superpolynomial lower bound for regular arithmetic formulas*. In: *STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing*. Ed. by David B. Shmoys. ACM, 2014, pp. 146–153.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. *An almost cubic lower bound for depth three arithmetic circuits*. In: *43rd International Colloquium on Automata, Languages, and Programming*. Vol. 55. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016, Art. No. 33, 15. ISBN: 978-3-95977-013-2.

[KST18] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. *On the size of homogeneous and of depth-four formulas with low individual degree*. In: *Theory of Computing. An Open Access Journal* 14 (2018), Paper No. 16, 46. ISSN: 1557-2862.

[KST23] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. *Near-optimal bootstrapping of hitting sets for algebraic models*. In: *Theory of Computing. An Open Access Journal* 19 (2023), Paper No. 12, 30. ISSN: 1557-2862.

[Lan17] J. M. Landsberg. *Geometry and Complexity Theory*. Vol. 169. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 2017. ISBN: 978-1-107-19923-1.

[LMS11] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. *Lower Bounds Based on the Exponential Time Hypothesis*. In: *Bulletin of the European Association for Theoretical Computer Science. EATCS* 105 (2011), pp. 41–71. ISSN: 0252-9742.

[LS78] Richard J. Lipton and Larry J. Stockmeyer. *Evaluation of Polynomials with Super-Preconditioning*. In: *Journal of Computer and System Sciences* 16.2 (1978), pp. 124–139. ISSN: 0022-0000.

[LST21] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. *Superpolynomial lower bounds against low-depth algebraic circuits*. In: *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science—FOCS 2021*. IEEE Computer Soc., Los Alamitos, CA, 2021, pp. 804–814. ISBN: 978-1-6654-2055-6.

[LST22] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. *On the partial derivative method applied to lopsided set-multilinear polynomials*. In: *37th Computational Complexity Conference*. Vol. 234. LIPIcs. Leibniz Int. Proc. Inform. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022, Art. No. 32, 23. ISBN: 978-3-95977-241-9.

[Mah14] Meena Mahajan. *Algebraic complexity classes*. In: *Perspectives in computational complexity*. Vol. 26. Progr. Comput. Sci. Appl. Logic. Birkhäuser/Springer, Cham, 2014, pp. 51–75. ISBN: 978-3-319-05445-2.

[Mal03] Guillaume Malod. *Polynômes et coefficients*. PhD thesis. Université Claude Bernard - Lyon, 2003.

[Mal07] Guillaume Malod. *The Complexity of Polynomials and Their Coefficient Functions*. In: *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*. 2007, pp. 193–204.

[Mar10] Dániel Marx. *Can You Beat Treewidth?* In: *Theory of Computing. An Open Access Journal* 6 (2010), pp. 85–112. ISSN: 1557-2862.

[Mit70] Dragoslav S. Mitrinović. *Analytic inequalities*. Vol. Band 165. Die Grundlehren der mathematischen Wissenschaften. Springer-Verlag, New York-Berlin, 1970. ISBN: 978-3-642-99970-3.

[MP08] Guillaume Malod and Natacha Portier. *Characterizing Valiant's Algebraic Complexity Classes*. In: *Journal of Complexity* 24.1 (2008), pp. 16–38. ISSN: 0885-064X.

[MS01] Ketan D. Mulmuley and Milind Sohoni. *Geometric Complexity Theory. I. An Approach to the P vs. NP and Related Problems*. In: *Siam Journal On Computing* 31.2 (2001), pp. 496–526. ISSN: 0097-5397.

[MS08] Ketan D. Mulmuley and Milind Sohoni. *Geometric Complexity Theory. II. Towards Explicit Obstructions for Embeddings among Class Varieties*. In: *Siam Journal On Computing* 38.3 (2008), pp. 1175–1206. ISSN: 0097-5397.

[MS18] Meena Mahajan and Nitin Saurabh. *Some Complete and Intermediate Polynomials in Algebraic Complexity Theory*. In: *Theory of Computing Systems* 62.3 (2018), pp. 622–652. ISSN: 1432-4350,1433-0490.

[Mul11] Ketan D. Mulmuley. *On P vs. NP and Geometric Complexity Theory*. In: *Journal of the ACM* 58.2 (2011), Art. 5, 26. ISSN: 0004-5411.

[Mul12] Ketan D. Mulmuley. *The GCT Program toward the P vs. NP Problem*. In: *Communications of the ACM* 55.6 (2012), pp. 98–107. ISSN: 0001-0782.

[Nis91] Noam Nisan. *Lower bounds for non-commutative computation (extended abstract)*. In: *Proceedings of the 23rd annual ACM Symposium on Theory of Computing*. ACM, 1991, pp. 410–418.

[NW96] Noam Nisan and Avi Wigderson. *Lower bounds on arithmetic circuits via partial derivatives*. In: *Computational Complexity* 6.3 (1996), pp. 217–234. ISSN: 1016-3328.

[Raz06] Ran Raz. *Separation of multilinear circuit and formula size*. In: *Theory of Computing. An Open Access Journal* 2 (2006), pp. 121–135. ISSN: 1557-2862.

[Raz09] Ran Raz. *Multi-linear formulas for permanent and determinant are of super-polynomial size*. In: *Journal of the ACM* 56.2 (2009), Art. 8, 17. ISSN: 0004-5411,1557-735X.

[Raz10] Ran Raz. *Elusive functions and lower bounds for arithmetic circuits*. In: *Theory of Computing. An Open Access Journal* 6 (2010), pp. 135–177. ISSN: 1557-2862.

[Raz13] Ran Raz. *Tensor-rank and lower bounds for arithmetic formulas*. In: *Journal of The ACM* 60.6 (2013), Art. 40, 15. ISSN: 0004-5411.

[Reg02] Kenneth W. Regan. *Understanding the Mulmuley-Sohoni Approach to P vs. NP*. In: *Bulletin of the European Association for Theoretical Computer Science* 78 (2002), pp. 86–99. ISSN: 0252-9742.

[RR20] C. Ramya and B. V. Raghavendra Rao. *Lower bounds for special cases of syntactic multilinear ABPs*. In: *Theoretical Computer Science* 809 (2020), pp. 1–20. ISSN: 0304-3975,1879-2294.

[RY08] Ran Raz and Amir Yehudayoff. *Balancing syntactically multilinear arithmetic circuits*. In: *Computational Complexity* 17.4 (2008), pp. 515–535. ISSN: 1016-3328,1420-8954.

[RY09] Ran Raz and Amir Yehudayoff. *Lower bounds and separations for constant depth multilinear circuits*. In: *Computational Complexity* 18.2 (2009), pp. 171–207. ISSN: 1016-3328,1420-8954.

[RY11] Ran Raz and Amir Yehudayoff. *Multilinear Formulas, Maximal-Partition Discrepancy and Mixed-Sources Extractors*. In: *Journal of Computer and System Sciences* 77.1 (2011), pp. 167–190. ISSN: 0022-0000,1090-2724.

[Sap21] Ramprasad Saptharishi. *A survey of lower bounds in arithmetic circuit complexity*. Github Survey, 2021.

[Sax09] Nitin Saxena. *Progress on polynomial identity testing*. In: *Bulletin of the European Association for Theoretical Computer Science. EATCS* 99 (2009), pp. 49–79. ISSN: 0252-9742.

[Sax14] Nitin Saxena. *Progress on polynomial identity testing-II*. In: *Perspectives in computational complexity*. Vol. 26. Progr. Comput. Sci. Appl. Logic. Birkhäuser/Springer, Cham, 2014, pp. 131–146. ISBN: 978-3-319-05445-2.

[Sch76] C.-P. Schnorr. *A Lower Bound on the Number of Additions in Monotone Computations*. In: *Theoretical Computer Science* 2.3 (1976), pp. 305–315. ISSN: 0304-3975,1879-2294.

[Sch91] Wolfgang M. Schmidt. *Diophantine Approximations and Diophantine Equations*. Vol. 1467. Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1991. ISBN: 978-3-540-54058-8.

[Sho09] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. 2nd ed. Cambridge University Press, Cambridge, 2009. ISBN: 978-0-521-51644-0.

[Sni80] Marc Snir. *On the Size Complexity of Monotone Formulas*. In: *Automata, Languages and Programming (Proc. Seventh Internat. Colloq., Noordwijkerhout, 1980)*. Vol. 85. Lecture Notes in Comput. Sci. Springer, Berlin-New York, 1980, pp. 621–631. ISBN: 978-3-540-10003-4.

[Sri20] Srikanth Srinivasan. *Strongly Exponential Separation between Monotone VP and Monotone VNP*. In: *ACM Transactions on Computation Theory* 12.4 (2020), Art. 23, 12. ISSN: 1942-3454,1942-3462.

[SS96] Victor Shoup and Roman Smolensky. *Lower bounds for polynomial evaluation and interpolation problems*. In: *Computational Complexity* 6.4 (1996), pp. 301–311. ISSN: 1016-3328,1420-8954.

[Str73] Volker Strassen. *Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten*. In: *Numerische Mathematik* 20 (1973), pp. 238–251. ISSN: 0029-599X.

[SW01] Amir Shpilka and Avi Wigderson. *Depth-3 arithmetic circuits over fields of characteristic zero*. In: *Computational Complexity* 10.1 (2001), pp. 1–27. ISSN: 1016-3328,1420-8954.

[SY09] Amir Shpilka and Amir Yehudayoff. *Arithmetic circuits: a survey of recent results and open questions*. In: *Foundations and Trends in Theoretical Computer Science* 5.3-4 (2009), 207–388 (2010). ISSN: 1551-305X.

[Tav15] Sébastien Tavenas. *Improved bounds for reduction to depth 4 and depth 3*. In: *Information and Computation* 240 (2015), pp. 2–11. ISSN: 0890-5401,1090-2651.

[TLS22] Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. *Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication*. In: *STOC '22—Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, New York, 2022, pp. 416–425. ISBN: 978-1-4503-9264-8.

[Val+83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. *Fast parallel computation of polynomials using few processors*. In: *SIAM Journal on Computing* 12.4 (1983), pp. 641–644. ISSN: 0097-5397.

[Val79a] L. G. Valiant. *Completeness classes in algebra*. In: *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing (STOC)*. ACM, New York, 1979, pp. 249–261. ISBN: 978-0-89791-003-3.

[Val79b] L. G. Valiant. *The Complexity of Computing the Permanent*. In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201. ISSN: 0304-3975,1879-2294.

[Val80] L. G. Valiant. *Negation Can Be Exponentially Powerful*. In: *Theoretical Computer Science* 12.3 (1980), pp. 303–314. ISSN: 0304-3975,1879-2294.

[Val82] L. G. Valiant. *Reducibility by Algebraic Projections*. In: *Logic and Algorithmic (Zurich, 1980)*. Vol. 30. Monogr. Enseign. Math. Univ. Genève, Geneva, 1982, pp. 365–380.

[Ven92] H. Venkateswaran. *Circuit Definitions of Nondeterministic Complexity Classes*. In: *SIAM Journal on Computing* 21.4 (1992), pp. 655–670. ISSN: 0097-5397.

[Yeh19] Amir Yehudayoff. *Separating Monotone VP and VNP*. In: *STOC'19—Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. ACM, New York, 2019, pp. 425–429. ISBN: 978-1-4503-6705-9.