A Tale of Hardness, De-randomization and De-bordering in Complexity Theory

A Thesis

submitted to the

Chennai Mathematical Institute, India

for the degree of Doctor of Philosophy in

Computer Science

by

Pranjal Dutta

August, 2022

As the poems go into the thousands, you
realize that you've created very
little.
It comes down to the rain, the sunlight,
the traffic, the nights and the days of the
years, the faces.
Leaving this will be easier than living
it, typing one more line now as
a man plays a piano through the radio,
the best writers have said very
little
and the worst,
far too much.

— Charles Bukwoski, As The Poems Go.

DECLARATION

I declare that the thesis titled *A Tale of Hardness, De-randomization and De-bordering in Complexity Theory* has been authored by me. It presents the research carried out by me during the period from August 2018 to August 2022 under the guidance of Prof. Nitin Saxena (CSE, IIT Kanpur). 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.

August 2022 Pranjal Dutta

Chennai Mathematical Institute
Plot H1, SIPCOT IT Park, Siruseri,
Kelambakkam 603103
India

Certificate

This is is to certify that the Ph.D. thesis submitted by Pranjal Dutta to Chennai Mathematical Institute, titled *A Tale of Hardness, De-randomization and De-bordering in Complexity Theory* is a record of *bonafide* research work done during the period August, 2018 - August, 2022 under my guidance and supervision. The research work in this thesis has not been submitted elsewhere for a degree.

Prof. Nitin Saxena

Adjunct Faculty
Chennai Mathematical Institute
Plot H1, SIPCOT IT Park, Siruseri
Kelambakkam 603103
India

&

N Rama Rao Chair Professor
Indian Institute of Technology Kanpur
Department of CSE
Kanpur 208016
India

Synopsis

"It's still magic even if you know how it's done."

— Terry Pratchett, A Hat Full of Sky.

One of the main goals of theoretical computer science is to understand the complexity of various problems. This thesis proposes on looking at the 'simplest' algebraic models of computations, and their implications in fundamental complexity-theoretic questions. Further, we derandomize and 'de-border' (de-approximate) some problems that were previously known to have efficient randomized solutions, or approximations. The thesis is divided into two parts – 1) Results in Algebraic Complexity, and, 2) Results in Geometric Complexity Theory (GCT).

The most significant challenge in the field of algebraic complexity is to find an explicit polynomial that requires *super*polynomially many operations to compute it, equivalently, superpolynomial size circuits. The *permanent* is widely conjectured to be such a polynomial, though its illustrious sibling – the *determinant* – can indeed be computed by polynomial-sized circuits. Separating the complexity of the determinant from that of the permanent is the question of utmost importance in this field. In fact, this question, commonly known as the VP vs. VNP problem, was formalized by Valiant [Val79], as an algebraic analogue of the "P vs. NP" question. Surprisingly, a very closely related problem of prime importance is the *Polynomial Identity Testing* (PIT). PIT asks to check whether the polynomial computed by a given circuit is identically zero or not. Though a very straightforward randomized algorithm exists for PIT, a deterministic polynomial time solution has long been desired, but not yet achieved. Remarkably, [KI04; Agr05] showed that efficient deterministic algorithms for PIT would yield intriguing algebraic (and Boolean) circuit lower bounds, and vice versa.

We initiate our study by showing some curious connections between – (1) showing non-trivial lower bounds for representing a univariate polynomial as a sum-of-squares, (2) bounding the number of real roots of explicit univariate polynomials, and (3) the big-ticket consequences in algebraic complexity (separating VP from VNP, finding an efficient algorithm for PIT etc.). In particular, we show that it suffices to find an explicit univariate polynomial f, that can be written as a sum-of-squares (SOS) $f = \sum c_i f_i^2$, where $c_i \in \mathbb{F}$, for a field $\mathbb{F}$, such that the total sparsity of the polynomials f_i, called the "support-sum size" of f, is slightly 'largish'. *Most probably*, simple-looking 'real-rooted' polynomials, like $(x+1)^d$, or $\prod_{i=1}^{d}(x+i)$, already satisfy such a condition. The idea that polynomials with small size circuits should have small size SOS representations, plays a central role

in proving such consequences. We further investigate the sum-of-cube (SOC) and higher power representations similarly, and show *stronger* results.

Next, we study PIT for polynomials computed by some 'structured' algebraic circuits of depth-4. For some context, extremely simple models like constant-depth circuits (even depth-4) are known to be surprisingly powerful in algebraic complexity [AV08; Koi12]; proving strong lower bounds, or derandomization results will have direct consequences for general circuits! In this work, we *quasi*-derandomize two circuit classes – (i) $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, and, (ii) $\Sigma^{[k]}\Pi\Sigma\wedge$, where k and δ can be *arbitrary* constants. To put it in context, even for $k = 3$, and $\delta = 3$, nothing better than an exponential-time algorithm was known before our work. And, after our work, 'only' a subexponential-time blackbox PIT algorithm for *any* constant-depth circuits was obtained in the breakthrough result by Limaye, Srinivasan, Tavenas [LST21]. A key technical ingredient in our algorithms is how the *logarithmic derivative* operator, and its power-series, modify the top Π-gate to $\wedge$.

Next, we move onto the GCT results. GCT is a novel approach towards proving strong lower bounds in complexity theory (both algebraic & Boolean), via methods from algebraic geometry and representation theory. It was introduced by Mulmuley and Sohoni [MS01; Mul12; Mul17], and has gained significant momentum over the last few years. A polynomial P is said to be in the border of a circuit class $\mathscr{C}$ (denoted $\overline{\mathscr{C}}$), over a field $\mathbb{F}$, if there is a circuit $D \in \mathscr{C}$, which uses constants from the function field $\mathbb{F}(\epsilon)$, where ϵ is a formal variable, and $\lim_{\epsilon\to 0} D = P$. Over fields such as complex numbers, this notion is known to be *equivalent* to the fact that the polynomial can be approximated to arbitrary accuracy (say in ℓ_2 distance for the coefficient vectors), by a circuit in the class $\mathscr{C}$.

Unfortunately, very little is known about the properties of the border of very simple circuit classes. It is easy to see that depth-2 circuits are equal to their closure. But, what happens for border of depth-3 circuits? In a surprising result, Kumar [Kum20] showed that border fanin-2 depth-3 circuits ($\overline{\Sigma^{[2]}\Pi\Sigma}$) are 'universal', with an exponential blowup in the size. This motivates us to understand the power of polynomial size $\overline{\Sigma^{[2]}\Pi\Sigma}$ circuits. In this work, we show that $\overline{\Sigma^{[2]}\Pi\Sigma}$ are 'easy' – they can be computed by 'small' determinants. The same result holds, even when one replaces the fanin by an arbitrary constant k. Further, we can also *quasi*-derandomize the same class. We develop a new technique, called DiDIL – **di**vide, **de**rive, **i**nduct, with **l**imit. It 'almost' reduces $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits to read-once oblivious algebraic branching programs (ROABPs) in any-order, for which the closure and PIT are well-understood.

Finally, we extend our SOS results in the border paradigm. In particular, real-rooted polynomials should be hard to even *approximate* as a sum-of-squares, which suffices to prove the approximative (stronger) version of the VP vs. VNP, and derandomize the approximative version of PIT for general circuits.

Acknowledgements

The epidemic has been with us for more than two years. I owe a debt of gratitude to the doctors, nurses, and other frontline workers (including my father, a retired banker), who faced the worst of the crisis to keep the rest of us safe and healthy, and the economy running. Even a million thanks would be insufficient.

Every Ph.D. is perhaps an emotional journey, and mine was no exception. Moreover, I was stuck at home for 2 years during the pandemic and have been struggling with post-covid symptoms for about a year. So, please bear with me, because I feel obligated to name the people who kept me sane and helped me finish my Ph.D.

First and foremost, I want to thank my advisor, Nitin Saxena, for making me who I am today. You have always been extremely generous with your time, and we have had so many instructive discussions and endless streams of great (many times failed) research ideas. Your belief in hard work and incremental developments have been instrumental for my personal growth as a researcher. Your strong perseverance towards solving problems is the biggest lesson I have learned from you about research. There was a point, in 2020, when I was almost *about to quit*. Your constant encouragement made me get through that phase, and I cannot thank you enough. Outside of work, we also have had interesting long discussions on life, politics, daily lifestyle and what not! Thank you for treating me as a colleague from the first time we started working on. Thank you for sheltering me in a parental manner and uncountable suggestions regarding varied things, especially during the time of illness and pandemic. Finally, thank you for pushing me to apply for the academic jobs at the right time!

Thank you Thomas Thierauf, for being an important source of guidance for me. Working with you has always been a pleasure. The sum-of-squares (SOS) project started after your visit to IIT Kanpur in August 2018, and it has played a key role in my early days' growth during Ph.D. Among many other things which I learned from you is how to write better articles. Thank you for many varied and thoughtful suggestions throughout my Ph.D. And, last but no the least, thank you for encouraging me to write the additional results I had on sum-of-squares, and submit, which eventually turned out to be the *Best Paper Award* winner at CSR'21!

This thesis is based on joint works with Prateek Dwivedi, Thomas Thierauf, and my advisor. I am fortunate to have them as coauthors. Thanks to Prateek's numerous 'fundamental' questions, which have often perplexed me, and made me rethink about the particular problems in a much deeper and clearer way. Without them, this work would not have been possible.

Thanks to Amit Sinhababu (Amit da), Gorav Jindal, Anurag Pandey, Mahesh Rajasree, and Santanu Sarkar (Santanu da), for collaborating with me during my Ph.D. I have learned quite a lot from each one of you after long hours of discussions. Working with such wonderful individuals makes research much more fun. Thank you, Mahesh, for hours of conversation on the fascinating field of cryptography.

Thanks to Amit da and Sumanta da (Sumanta Ghosh), for being more than academic brothers to me. I have called you randomly and worriedly discussing both professional and personal issues. Every time you have been extremely patient and calm enough to listen to me, and offer excellent suggestions that have really worked.

My interest in complexity theory stemmed from the wonderful courses I had taken over the years, and my interaction with the faculty at CMI, IMSc and IIT Madras. I would specially like to thank Partha, KV, Samir, Jayalal, BV, and Meena for all the courses and discussions we have had. Special thanks to Jayalal, for introducing me to the beautiful world of algebraic complexity, during my bachelors.

Thank you KV and Partha, for always being a source of inspiration. I have written to you whenever I am in doubt (many times *without* even thinking about it), and each time you have responded very positively, which has often calmed me down. KV deserves special recognition for handling all the administrative hassles, and suggesting me doctors to visit, regarding long-covid issues. Let me also express my gratitude to CMI's administrative personnel (Rajeshwari, Ranjini, Sripathy, and Viji) for their efficiency and promptness. Finally, I'd want to express my gratitude to CMI for all the fantastic facilities, flexibility in allowing me to work at IIT Kanpur during my Ph.D., and for being the amazingly cool place that it is!

Besides my collaborators, I also thank Nitin (Saurabh), Nikhil, Jayalal, Raghu (BV Raghavendra Rao), Ramya, Rajit da, Utsab, Subhayan, Abhiroop, Pranav, Ashish, Samir, KV, Partha, Somenath sir, Zeyu, Vishwas, and Christian for several interesting discussions I have had over the years, on complexity theory and computer science in general.

During my Ph.D., I got the opportunity to attend some wonderful workshops, which had given me an exposition of wide range of topics in my research area. I want to thank the organizers of the workshops I participated in: *Algebraic Methods* 2019, at the Simons Institute for the Theory of Computing; *WACT* 2019 at ICTS Bangalore; *Caleidoscope : Complexity as a Kaleidoscope* 2019, at Institut Henri Poincaré, Paris; *CAALM* 2019 at CMI; and *GCT* 2022 virtually organized by CMI. Thanks to Rafael for hosting me at the University of Toronto.

Thanks to Google for the 4-year Ph.D. Fellowship, which allowed me to be financially self-sufficient in terms of research visits, and workshop attendance. Further, thanks to my Google Research Mentor Ravi Kumar. I have had quite a few interactions with you, and every time you have made me feel motivated. Finally, I would want to express my gratitude to Divy, for being highly approachable, when it came to any travel/expenditure-related difficulties or suggestions for using the Google grant.

Life at IIT Kanpur would not be so easy and joyful without the company I had here. Thanks to all of my lab-mates and friends for the wonderful time we had together. I sincerely thank

Amit da, Sumanta da, Rajendra, Mahesh, Prateek, Bhargav, Priyanka for being so helpful all the time.

The majority of the thesis writing was completed at CMI. Thanks to all of my CMI lab-mates and friends, specially Priya (Shanmugapriya), Kaberi, Sanchari, Aneesh, Utsab, Sayantani, Vishwa for making the thesis writing process more enjoyable! Special thanks to Kaberi and Sanchari, for all the pointless jokes, and hours of random discussions, that really helped to make the writing process less-stressful!

Thanks to all my non-academic friends (and juniors/seniors), specially Akash, Mouli, and Shreyasi, for being there, and having refreshing discussions. Many times, I have missed important occasions and gatherings! Thank you for putting up with me! Thank you for your unquestionable support throughout my Ph.D. journey. Thanks to all of my other unmentioned friends, for cherishable conversations. Please accept my apologies for not being able to put all of your names!

Special thanks to these four people: Abhijit Banerjee, Esther Duflo, Dan Ariely and Adam Grant, for keeping me sane and grounded throughout the pandemic. Banerjee and Duflo's "Poor Economics" and "Good Economics For Hard Times", Ariely's "Predictably Irrational" and "Irrationally yours", and Grant's "Think Again" have quite deeply impacted my thinking towards society, and life in general. Thank you, Dan, for personally responding to my email with a sweet voice-note, and permitting me to use some illustrations. Thanks to Bhaskar Chakraborty, Purnendu Potri, Budhdhadeb Halder, Shonkho Ghosh, Agha Shahid Ali, Kahlil Gibran for your existence, and the lovely pieces you have written. When I am feeling low, you have been one of my frequent go-to spots.

Finally, I would want to express my gratitude to Mammam, my sister, for putting up with me at home, despite all of my annoyances. Bui, my Labrador daughter, thank you for showering me with affection and making me feel stress-free on several occasions. Thank you, Baba and Ma, for your unwavering support and understanding, as well as your *countless* sacrifices. I cannot express my indebtedness to you. Thank you for trusting me, and being there throughout the highs-and-lows. And last, but not the last, Dada (my grandfather), I miss you.

Thanks to Ken Arroyo Ohori's Doctoral Thesis (tex), Tufte-Latex Class, and of course, Federico Marotta's Kaobook Class on which my class file is built; this thesis would not have looked the same without their amazingly stylish LaTeX files.

To Baba, Maa, and Bui,

whose unconditional love, and support made me survive Ph.D. during the pandemic.

Contents

List of Publications

Peer-reviewed works included (at least partially) in the thesis (in the chronological order, and authors in the alphabetical notation):

[DST21] A largish Sum-Of-Squares Implies Circuit Hardness and Derandomization,
Pranjal Dutta, Nitin Saxena, and Thomas Thierauf.
12^{th} Innovations in Theoretical Computer Science Conference (ITCS 2021).
Full version currently under review.
https://www.cse.iitk.ac.in/users/nitin/papers/sos_journal.pdf

[Dut21] Real τ-Conjecture for Sum-of-Squares: A Unified Approach to Lower Bound and Derandomization.
Pranjal Dutta
16^{th} International Computer Science Symposium in Russia (CSR 2021).
Won the Best Paper and Best Student Paper Award.
Invited in the ToCS Special Issue on CSR'21.
https://drive.google.com/file/d/1Ch9-l4hjkfsPka1Rfi-ZXMBg8mUn8Gge/view

[DDS21a] Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits,
Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena.
36^{th} Computational Complexity Conference (CCC 2021).
Full version currently under review.
https://www.cse.iitk.ac.in/users/nitin/papers/bdd-fanin-depth4-PIT.pdf

[DDS21b] Demystifying the border of depth-3 algebraic circuits,
Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena.
62^{nd} IEEE Symposium on Foundations of Computer Science (FOCS 2021).
Invited in the SICOMP Special Issue on FOCS'21.
https://www.cse.iitk.ac.in/users/nitin/papers/border-depth3-sicomp.pdf

The results of [DST21] and [Dut21] are completely included in the thesis. While, we have included only depth-4 blackbox PIT algorithm from [DDS21a], and, depth-3 border results from [DDS21b].

Peer-reviewed works published/accepted during Ph.D., but *not included* in the thesis (in the reverse chronological order):

[DS22] Separated borders: Exponential-gap fanin-hierarchy theorem for approximative depth-3 circuits,
Pranjal Dutta, and Nitin Saxena.
63^{rd} IEEE Symposium on Foundations of Computer Science (FOCS 2022)
https://www.cse.iitk.ac.in/users/nitin/papers/exp-hierarchy.pdf

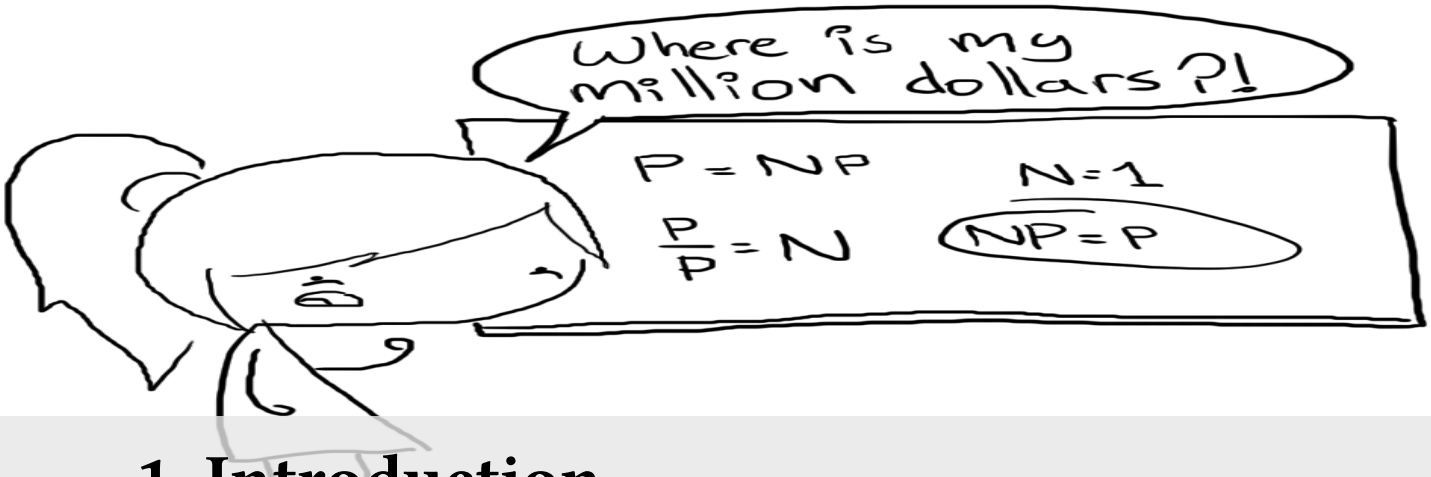
[DSS22] Discovering the roots: Uniform closure results for algebraic classes under factoring,
Pranjal Dutta, Nitin Saxena, and Amit Sinhababu.
Journal of the ACM (J.ACM), 2022.
Preliminary version in the 50^{th} Annual ACM Symposium on the Theory of Computing (STOC 2018).
https://www.cse.iitk.ac.in/users/nitin/papers/factor-closure-jacm.pdf

[DR22] Algebraic Algorithms for Variants of Subset Sum,
Pranjal Dutta, Mahesh Sreekumar Rajasree.
8^{th} International Conference on Algorithms and Discrete Applied Mathematics (CALDAM 2022).
One of the 3 winners of the Springer Best Student Presentation Award.
Full version currently under review.
https://arxiv.org/abs/2112.11020

[DRS22] On the hardness of monomial prediction and zero-sum distinguishers for Ascon,
Pranjal Dutta, Mahesh S. Rajasree, and Santanu Sarkar.
12th International Workshop on Coding and Cryptography (WCC 2022).
Invited in the DCC Special Issue on WCC'22.
https://www.wcc2022.uni-rostock.de/storages/uni-rostock/Tagungen/WCC2022/Papers/WCC_2022_paper_74.pdf

[Dut+21] Arithmetic circuit complexity of division and truncation,
Pranjal Dutta, Gorav Jindal, Anurag Pandey, and Amit Sinhababu.
36^{th} Computational Complexity Conference (CCC 2021).
https://eccc.weizmann.ac.il/report/2021/072/

List of Figures



1 Introduction

"If you're a scientist by trade, rethinking is fundamental to your profession...But being a scientist is not just a profession. It's a frame of mind – a mode of thinking that differs from preaching, prosecuting, and politicking".

— Adam Grant, Think Again.

Let us begin by asking some thought-provoking questions.

Q1. Is it possible to efficiently solve Sudoku for 'really large' squares?

Q2. How do the zeroes (roots) of a polynomial behave?

Q3. Do (algebraic) approximations help in computation?

Q4. Can we remove *randomness* with 'cosmetic' changes in resources (e.g., time, memory)?

Q5. Is there any connection between algorithms and lower bounds?

At a first glance, Q1-Q5 look like a hodgepodge of unrelated questions. Surprisingly, these questions are closely related and studied within the framework of *Computational Complexity Theory*. In today's technology-driven world, resources play a crucial role in determining the efficiency and usefulness of a particular algorithm. Complexity Theory tries to quantify the requirement of a certain resource to complete various computational tasks. The objective of my thesis is to fathom algebraic computations as a pervasive computational phenomenon, interrelate questions Q1-Q5, in the strongest

Throughout the thesis, we will keep referring Q1-Q5 and its relevance in the particular contexts.

possible way, and use the findings to answer fundamental questions in *Algebraic Complexity* and *Geometric Complexity Theory*. We will try to define these terms later.

Models of computation which deal with real (in fact, complex numbers) were introduced with mainly two aims. On one hand, it aims to build the Complexity Theory for the computations done within the numerical tradition. On the other hand, it uses the power of the methods and results of *continuous* mathematics (rather than *discrete* one). The basic point is, numbers should be considered as basic (indivisible) entities, along with *fixed* (unit) cost operations.

At the inception, it was really a novel idea that the number of operations, rather than the bit complexity, is *important enough* to measure the complexity of a numerical process. It is just as classical as the notion of computation! Interestingly, in 1948, Alan Turing [Tur48] already noted the following.

> *"It is convenient to have a measure of the amount of work involved in a computing process, even though it be a very crude one.* [⋯] *In the case of computing with matrices, most of the work consists of multiplications and writing down numbers, and we shall therefore only attempt to count the number of multiplications and recordings. For this purpose. a reciprocation will count as a multiplication. This is purely formal. A division will then count as two multiplications; this seems a little too much. and there may be other anomalies, but on the whole substantial justice should be done."*

1.1 Algebraic Complexity

Perhaps, the most important machine model which has the above-discussed features is the *algebraic circuits*. They are studied in algebraic complexity, formerly 'arithmetic circuit complexity', to understand formal multivariate polynomials over a field $\mathbb{F}$. The field may not necessarily be $\mathbb{F}_2$, like in the Boolean world. A paramount reason to focus on algebraic complexity is the so-called "yellow books argument", as mentioned by Aaronson [Aar16]: algebraic complexity brings us *closer* to continuous mathematics, where we have acquired

deep knowledge, for e.g., algebraic geometry, representation theory, which is harder to apply in the Boolean case.

1.1.1 Algebraic models of computation

Definition 1.1.1 (Algebraic circuit) *An algebraic circuit is a finite directed acyclic graph where each vertex (gate) is one of the following:*

(i) An input gate labeled by some variable x_i with in-degree zero.

(ii) A constant gate with in-degree zero, labeled by some constant $c \in \mathbb{F}$. Here $\mathbb{F}$ is the underlying field; for e.g., the field of rationals $\mathbb{Q}$, the field of complex numbers $\mathbb{C}$.

(iii) All the other internal gates are labelled by '$+$' and '$\times$'. They have the obvious operational semantics.

(iv) An output gate with out-degree zero; we assume there is exactly one output gate. It outputs the polynomial computed by the circuit.

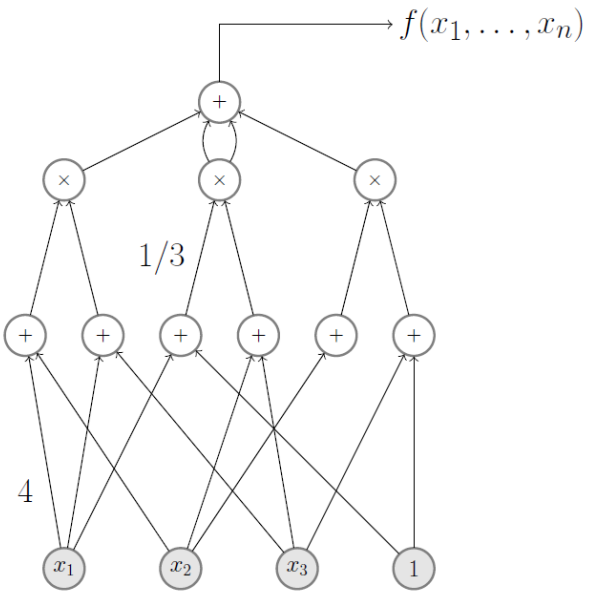


Figure 1.1: A depth-3 circuit of size 34

The *size* of the circuit denotes the number of edges and nodes

in the graph [1]. For any fixed polynomial f, size(f) denotes the size of the *smallest circuit* computing it; the notion of 'size' captures the number of additions and multiplications *required* to compute its value on any input. Some other very relevant and important complexity parameter of an algebraic circuit are –

1. the *depth* – it denotes the length of the longest path in the circuit,
2. formal *degree* – the maximum degree polynomial that can be computed by any node,
3. *fan-in* – maximum number of inputs to a node, and
4. *fan-out* – maximum number of outputs from a node.

1: At the inception of algebraic complexity, the size usually captured only the number of nodes in the circuit. Although it varies from case-to-case, nowadays, we usually use the number of nodes and edges. Since, in a graph, the number of edges can be at most *quadratic* in the number of nodes, it *hardly* matters, specailly while proving superpolynomial lower bounds.

Another very relevant model of computation is the algebraic formulas.

Definition 1.1.2 (Algebraic formula) *An algebraic circuit is called a formula if the underlying acyclic graph in Definition 1.1.1 is a tree.*

In a formula, the fan-out of the nodes is at most one, i.e. 'reuse' of intermediate computation is not allowed. An intermediate model between algebraic formulas and algebraic circuits, is the *algebraic branching programs* (ABPs).

Definition 1.1.3 (Algebraic Branching Program (ABP)) *An Algebraic Branching Program (ABP) in variables $x_1, x_2, \ldots, x_n$, over the field $\mathbb{F}$, is a directed acyclic graph with the following properties.*

(i) There is a distinguished vertex s of in-degree zero (the source).

(ii) There is a distinguished vertex t of out-degree zero (the sink).

(iii) Each edge e is labeled with a linear polynomial ℓ_e in the input variables $x_1, x_2, \ldots, x_n$.

They can be thought of as a linear projection of symbolic determinant polynomials, defined in subsection 1.1.3.

1.1.2 Algebraic complexity classes

Analogous to the idea of classical complexity classes such as P, NP [2], one can define the algebraic complexity classes. On a related note, we have defined the notion of algebraic complexity in a *non*-uniform manner. This means that we do not require that the description of an algebraic circuit computing the polynomial should be the output of some Turing Machine! And therefore, we have to define algebraic complexity classes using polynomial families. We refer the reader to [Bür13; Mah14] for a more comprehensive introduction to algebraic complexity classes.

2: The P versus NP problem lies in the heart of theoretical computer science. Informally, it asks whether every problem whose solution can be 'quickly' verified can also be *solved* quickly. It is wildely believed to be not the case. Sudoku is one such candidate; see Q1.

Definition 1.1.4 (p-family) *A family (or a sequence) $(f_n)_{n\in\mathbb{N}}$ of (multivariate) polynomials over the field $\mathbb{F}$ is said to be a p-family iff the number of variables as well as the degree[3] of f_n are p-bounded (polynomially bounded) functions of n.*

Now we define the notion of efficient polynomial families.

Definition 1.1.5 (Class VP) [a] *The (algebraic complexity) class* VP *is the set of all p-families $(f_n)_{n\in\mathbb{N}}$ such that* size(f_n) *is a p-bounded function of n.*

[a] Symbolically, one often refers to $\mathsf{VP}_{\mathbb{F}}$. However, when the underlying field is implicitly/explicitly clear, we drop the subscript $\mathbb{F}$.

3: There are several intuitive reasons for the "low degree" restriction, as discussed by Joshua Grochow on `cstheory.SE` [Gro13] and also in [Sap21]. Essentially, every Boolean function can be expressed as a multilinear polynomial, which is obviously a low-degree polynomial. Moreover, most interesting polynomials, such as $\det_n$, or perm_n are in fact of low degree. Finally, after the log-depth reduction result by [Val+83], it makes even more sense to restrict ourselves to the poly-degree regime.

Similarly, one can define VF and VBP for formulas and ABPs respectively. Finally, the class VNP, can be seen as a non-deterministic analog of the class VP; it is essentially an exponential sum of projection of VP polynomials.

Definition 1.1.6 *A p-family $(f_n)_{n\in\mathbb{N}}$ is said to be in the (algebraic complexity) class* VNP *if there exists a polynomial family $(g_n)_{n\in\mathbb{N}} \in \mathsf{VP}$, with $g_n \in \mathbb{F}[x_1, \ldots, x_{q(n)}]$ such that:*

$$f_n(x_1, \ldots, x_{p(n)}) = \sum_{e\in\{0,1\}^{r(n)}} g_n\left(x_1, \ldots, x_{p(n)}, e_1, \ldots, e_{r(n)}\right),$$

where $r(n) := q(n) - p(n)$.

We have the following easy containment:

$$\mathsf{VF} \subseteq \mathsf{VBP} \subseteq \mathsf{VP} \subseteq \mathsf{VNP}.$$

It is believed that each containment is *strict*. In fact, the following postulate is known as the *algebraic* analogue of the P ≠ NP conjecture.

Conjecture 1.1.1 (Valiant's Hypothesis [Val79]) Over any field $\mathbb{F}$, $\mathsf{VP}_{\mathbb{F}} \neq \mathsf{VNP}_{\mathbb{F}}$.

Remark 1.1.1
1. P ≠ NP 'negatively' answers Q1 that Sudoku *cannot* be solved efficiently.
2. It is in fact believed that VNP is 'exponentially far' from VP.

For a more concretized formulation of the above conjecture, see the next subsection.

1.1.3 Two fundamental problems

In general, algebraic complexity theorists are interested in the following two fundamental problems:

1. **Superpolynomial Lower Bound** – We want to find an *explicit* n-variate poly(n)-degree *hard* polynomial, i.e., it requires $n^{\omega(1)}$ size circuits. By explicit, we mean that the coefficients are *easily* computable[4]. One such candidate is the *permanent* polynomial. The permanent polynomial family $(\mathrm{perm}_n)_n$ is defined as follows:
   For an $n \times n$ matrix X_n, whose (i,j)-th entry is a variable $x_{i,j}$, we define:
   $$\mathrm{perm}_n(X_n) = \sum_{\sigma \in S_n} \prod_{j \in [n]} x_{j,\sigma(j)} .$$
   In the above, S_n is the symmetric group of degree n; i.e., it contains all the bijective functions from $[n] = \{1, \ldots, n\}$ to itself. Once we have defined perm_n, one can 'reformulate' VP ≠ VNP conjecture as follows.

4: It is known that a *random* polynomial, whose coefficients are picked *uniformly at random*, is hard, with high probability.

Conjecture 1.1.2 (Valiant's Hypothesis [Val79]) For any field $\mathbb{F}$ of characteristic 0, or ≥ 3, perm_n requires $n^{\omega(1)}$ size circuits.

Remark 1.1.2 Over $\mathbb{F}_2$, $\mathrm{perm}_n = \det_n$. Howbeit, one can define another explicit polynomial family, using the hamiltonian cycles in a graph, and conjecture its hardness over $\mathbb{F}_2$.

Colloquially, $(\mathrm{perm}_n)_n$ is the 'illustrious sister' of the *determinant* $\det_n$ family, which is defined as:

$$\det_n(X_n) \;=\; \sum_{\sigma\in S_n}\prod_{j\in[n]}(-1)^{\mathrm{sgn}(\sigma)}x_{j,\sigma(j)}\,.$$

Sgn, the *signum* of a permutation, is a function which takes values ± 1, according to the *parity* of the permutation [5]. Once we have defined $\det_n$, we can formulate the VBP $\neq$ VNP question, as the famous 'Permanent versus Determinant' problem.

5: One interesting way to compute sgn, is via: $\mathrm{sgn}(\sigma) = \prod_{i<j}\frac{\sigma(j)-\sigma(i)}{j-i}$.

Conjecture 1.1.3 (Permanent vs. Determinant) Let $n(m)$ be a function of m such that there exist affine linear maps $A_m : \mathbb{C}^{m^2} \longrightarrow \mathbb{C}^{n(m)^2}$, satisfying

$$\mathrm{perm}_m \;=\; \det_{n(m)} \circ A_m\,.$$

Then, $n(m) = m^{\omega(1)}$.

2. **Polynomial Identity Testing (PIT)** – We want to find an efficient deterministic algorithm to decide whether the given circuit computes just the zero polynomial or not. We already have a polynomial-time randomized algorithm for PIT due to the *Polynomial Identity Lemma* [Ore22; DL78; Zip79; Sch80]; see Lemma 1.3.2 for details. The challenge is to remove the randomness.

Why care about VP $\neq$ VNP**?** Although, we have abstracted out some interesting-looking lower bound questions, it is not apparent why they are so relevant in complexity theory, and how they relate to the 'flagship problem' of P $\neq$ NP, in Boolean complexity, or the question of derandomization. We will try to answer this in the next few points.

1. Bürgisser [Bür00] showed that P/poly $\neq$ NP/poly, the *non-uniform* version of P $\neq$ NP, implies that VP $\neq$ VNP over finite fields, and in fact it could be extended to characteristic 0, assuming GRH (Generalized Riemann Hypothesis).

2. Kabanets and Impagliazzo [KI04] showed that proving VP ≠ VNP actually gives a *subexponential*-time PIT algorithm for general circuits whilst the current upper bound is EXP (Exponential-time)!
3. It is also known that VP ≠ VNP implies that $\mathsf{P}_{\mathbb{C}} \neq \mathsf{NP}_{\mathbb{C}}$, i.e. P ≠ NP, over the complex numbers, in the Blum-Shub-Smale (BSS) model, assuming the *Factor Conjecture* (see [Bür13]). A BSS machine is a Random Access Machine (RAM) with registers that can store *arbitrary* real numbers, and compute rational functions over reals, in a single time step. Thus, the BSS machines are more powerful than Turing machines.

Pictorially, the following holds:

$$\mathsf{P/poly} \neq \mathsf{NP/poly} \xrightarrow{\text{GRH}} \mathsf{VP} \neq \mathsf{VNP} \xrightarrow{\text{Factor Conj.}} \mathsf{P}_{\mathbb{C}} \neq \mathsf{NP}_{\mathbb{C}}.$$

$$\downarrow$$

$$\mathsf{PIT} \in \mathsf{SUBEXP}$$

Due to the above implications, proving algebraic lower bounds may be 'easier' than proving boolean lower bounds. But the bigger question is perhaps the following:

> *Does proving* VP ≠ VNP *take us 'closer' to proving* P ≠ NP*?*

We will try to answer it in subsection 1.2.1.

Why care about PIT? PIT appears in several seemingly unrelated problems. For example, Shamir's famous result of IP = PSPACE [Sha92] used the idea of comparing two multivariate polynomials for *equality*. A much older application of PIT is due to the following theorem proved by Tutte : *A graph has no perfect matching iff the determinant of the Tutte matrix is zero.* In fact, the problem of deterministic *primality testing*, to check whether the given input is a prime number or not, was solved in an elementary way with a PIT formulation [AKS04]. Formally, n is prime iff $(x+1)^n - (x^n+1) = 0$, over the ring $\mathbb{Z}/n\mathbb{Z}$.

1.2 Geometric Complexity Theory

Geometric Complexity Theory (GCT) is quite an ambitious program, which is built on Valiant's algebraic complexity theory framework, and strengthens the $\mathsf{VP} \neq \mathsf{VNP}$ conjecture in terms of the *Border Complexity*. In a bigger aim, it also tries to prove $\mathsf{P} \neq \mathsf{NP}$, and some related conjectures using algebraic geometry and representation theory. GCT has been pursued since the late 1990s. It was started by Ketan Mulmuley and Milind Sohoni, with important contributions from others. The notion of border complexity was also independently defined by Bürgisser [Bür04]. We will define these notions later. Before moving on, here is an interesting quote by Scott Aaronson who likes to describe GCT as *"the string theory of computer science"* [Aar16, p. 84]:

> *I like to describe GCT as "the string theory of computer science". Like string theory, GCT has the aura of an intricate theoretical superstructure from the far future, impatiently being worked on today. Both have attracted interest partly because of "miraculous coincidences" (for string theory, these include anomaly cancellations and the prediction of gravitons; for GCT, exceptional properties of the permanent and determinant, and surprising algorithms to compute the multiplicities of irreps) ... And like with string theory, there are few parts of modern mathematics not known or believed to be relevant to GCT.*

1.2.1 A gentle introduction to GCT for non-geometers

We start by asking the following simple meta-question.

Algebraic approximations help?
Can 'approximations' help in algebraic computational models?

To understand this, we will define an important measure, called the *Waring rank*, $\mathrm{WR}(\cdot)$.

Definition 1.2.1 (Waring rank) *The Waring rank of a homogeneous degree-d polynomial h is the smallest r such that can be written as a sum of d-th power of linear forms ℓ_i, i.e., $h = \sum_{i=1}^{r} \ell_i^d$.*

It is a folklore that, for any homogeneous polynomial h, $\mathrm{WR}(h)$ is finite. The finiteness follows from the following argument –

1. $\mathrm{WR}(\cdot)$ is *sub*-additive: $\mathrm{WR}(f+g) \leq \mathrm{WR}(f) + \mathrm{WR}(h)$, for any $f, g \in \mathbb{F}[x]$.
2. $\mathrm{WR}(x^e)$ is *finite*, by *interpolating* the following polynomial, $(x_1 + t_2 x_2 + \cdots + t_n x_n)^{\sum_{i=1}^{n} e_i}$, where t_i are new variables.[6]

6: This upper bound can be significantly improved; see Lemma 2.2.3.

❑ Characterizing bivariate polynomials of Waring rank 1

If one tries to characterize the bivariate degree-2 polynomials $h(x,y) = ax^2 + bxy + cy^2$, such that $\mathrm{WR}(h) = 1$, then it is not hard to show that the following set

$$X_1 \;:=\; \{h \mid \mathrm{WR}(h) = 1\} \;=\; \{(a,b,c) \mid b^2 - 4ac = 0\},$$

exactly characterizes $\mathrm{WR}(h) = 1$. Interestingly, it helps to prove lower bounds. For, e.g., $\mathrm{WR}(xy) > 1$; this is because trivially $(0,1,0) \notin X_1$. Such a polynomial $f = b^2 - 4ac$, is called a *'polynomial obstruction'* or a *'separating polynomial'*. Most importantly X_1 is a *closed* set: If there are three sequences (a_n, b_n, c_n) such that $a_n \to a, b_n \to b, c_n \to c$, i.e., limits exist, such that $(a_n, b_n, c_n) \in X_1$, then $(a,b,c) \in X_1$.

Now, we move on to the most important example, which will help us to understand why the notion of *border complexity* is natural. Consider the polynomial $h := x^2 y$. Note that, $\mathrm{WR}(h) \leq 3$, because

$$x^2 y \;=\; \frac{1}{6}\cdot(x+y)^3 - \frac{1}{6}\cdot(x-y)^3 - \frac{1}{3}\cdot y^3 \,.$$

In fact, it is not hard to prove that $\mathrm{WR}(x^2 y) = 3$. Now, let

$$h_\epsilon \;:=\; \frac{1}{3\epsilon}\left((x+\epsilon y)^3 \;-\; x^3\right),$$

for some nonzero parameter ϵ. It is not hard to show that

$$\lim_{\epsilon \to 0} h_\epsilon \;=\; \lim_{\epsilon \to 0} \left(x^2 y + x\epsilon y^2 + \frac{\epsilon^2}{3} y^3 \right) \;=\; x^2 y.$$

However, by definition, $\mathrm{WR}(h_\epsilon) \leq 2$, for any fixed nonzero ϵ! This introduces a *subtlety*:

> If a *continuous* function f vanishes on all h with $\mathrm{WR}(h) \leq 2$, then f should also vanish on the limit point $x^2 y$, which *cannot* be true from the above example. Thus, to prove $\mathrm{WR}(x^2 y) > 2$, we need to find a *discontinuous* function f which vanishes on $\mathrm{WR}(h) \leq 2$ but *does not* vanish on $x^2 y$.

Ideally, we would like to avoid such scenario so that we could use continuous mathematics.

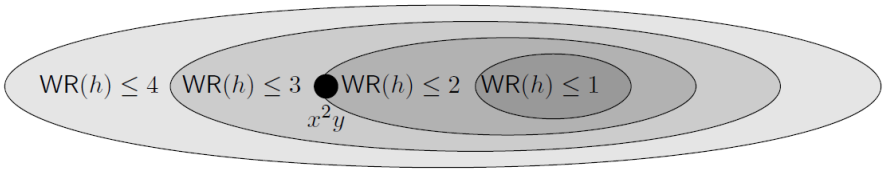


Figure 1.2: $x^2 y$ lies exactly in the boundary of the set $\{h \mid \mathrm{WR}(h) \leq 2\}$

This is one of the intuitions why the notion of *Border Waring rank* is natural to define. The border Waring rank of h, denoted, $\overline{\mathrm{WR}}(h)$ is defined as the smallest r such that h can be *approximated* arbitrarily closely by polynomials of Waring rank $\leq r$. In particular, from the above discussion, $\overline{\mathrm{WR}}(x^2 y) = 2$, but $\mathrm{WR}(x^2 y) = 3$. Most importantly, now the subtlety is *gone*: $X_r := \{h \mid \overline{\mathrm{WR}}(h) \leq r\}$, is now a *closed* set (in fact, it is a projective variety). Thus, to show a lower bound that $\overline{\mathrm{WR}}(p) > r$, for some p, it suffices to show that $p \notin X_r$, i.e. find a *continuous* function f that vanishes on X_r but not on p.

Now, one can analogously define the *border complexity* $\overline{\Gamma}$ with respect to any sensible measure Γ. For example, Γ can be size, determinantal complexity and so on. Using the notion of $\overline{\mathrm{size}}$, one can define $\overline{\mathsf{VP}}$, the *approximative closure* of VP.

In [MS01], Mulmuley and Sohoni strengthened Valiant's conjecture to: VNP $\not\subseteq \overline{\mathsf{VP}}$, or, *equivalently*, $\overline{\mathrm{size}}(\mathrm{perm}_n) = n^{\omega(1)}$; similarly VNP $\not\subseteq \overline{\mathsf{VBP}}$ can be phrased and conjectured; we will define these more geometrically later in this section. For the time being, let us define the working (algebraic) definition of approximation, in the context of GCT.

❑ Algebraic notion of approximation

The simplest notion of the approximative closure comes from the following definition [Bür04]:

> A polynomial $f(x) \in \mathbb{F}[x_1, \ldots, x_n]$ is *approximated* by $g(x, \epsilon) \in \mathbb{F}(\epsilon)[x]$ if there exists a $Q(x, \epsilon) \in \mathbb{F}[\epsilon][x]$ such that $g = f + \epsilon Q$.

We can also think analytically (in $\mathbb{F} = \mathbb{R}$ Euclidean topology) that $\lim_{\epsilon \to 0} g = f$. If g belongs to a circuit class $\mathscr{C}$ (over $\mathbb{F}(\epsilon)$, i.e. any *arbitrary* ϵ-power is allowed as 'cost-free' constants), then we say that $f \in \overline{\mathscr{C}}$, the approximative closure of $\mathscr{C}$.

The size of the circuit computing the polynomial g, over the field $\mathbb{F}(\epsilon)$, defines the *approximative* (or *border*) complexity of f, denoted $\overline{\mathrm{size}}(f)$. Interestingly, the topological definition of $\overline{\mathrm{size}}$ via limit exactly coincides with this algebraic definition [Mum95], over an algebraically closed field, like $\mathbb{C}$. Clearly, $\overline{\mathrm{size}}(f) \leq \mathrm{size}(f)$. Because of possible $1/\epsilon$ terms, one can not directly set $\epsilon = 0$. Since $g(x, \epsilon) = f(x) + \epsilon \cdot Q(x, \epsilon)$, we can extract the coefficient of ϵ^0 from g, using standard interpolation trick, by setting random ϵ-values from $\mathbb{F}$, if we know the degree upper bound M of ϵ (& $1/\epsilon$). However, the best known bound on M is $2^{\overline{\mathrm{size}(f)}^2}$, which is *exponentially* large [Bür04; Bür20, Theorem 5.7]. Therefore, the following relation is the best one known:

$$\overline{\mathrm{size}}(f) \leq \mathrm{size}(f) \leq 2^{\mathrm{poly}(\overline{\mathrm{size}}(f))}\,.$$

Understanding these relations is the main goal of Q4. We also mention the strengthened Valiant's Conjecture (of VP $\neq$ VNP) in the border context.

Conjecture 1.2.1 (Valiant's Conjecture [MS01]) VNP $\not\subseteq \overline{\mathsf{VP}}$, i.e., $\overline{\mathrm{size}}(\mathrm{perm}_n) = n^{\omega(1)}$.

❑ Valiant's Conjecture via orbit closure

To understand this in geometric language, let us denote by $\mathrm{Sym}^n V^*$, the space of homogeneous polynomial functions of degree n on a finite dimensional complex vector space V. The group $G := \mathrm{GL}(V)$ acts on $\mathrm{Sym}^n V^*$, in the *canonical* way:

$$(g \cdot f)(v) := f(g^{-1}v) \text{ for } g \in G, f \in \mathrm{Sym}^n V^*, \text{and } v \in V.$$

We denote by $G \cdot f := \{gf \mid g \in G\}$, the *orbit* of f. Let $V := \mathbb{C}^{n\times n}$, and think of the $n \times n$ symbolic determinant, $\det_n$, as an element of $\mathrm{Sym}^n(\mathbb{C}^{n\times n})^*$.

Now, consider its *orbit closure*:

$$\Omega_n := \overline{\mathrm{GL}_{n^2} \cdot \det_n} \subseteq \mathrm{Sym}^n(\mathbb{C}^{n\times n})^* ,$$

with respect to the Euclidean topology (*equivalent* to the closure with respect to the Zariski topology, see [Mum95, Appendix 2.C])[7].

For $n > m$, we consider the *padded permanent* defined as $\ell^{n-m} \cdot \mathrm{perm}_m \in \mathrm{Sym}^n(\mathbb{C}^{m\times m})^*$, where ℓ denotes the linear form providing the $(1,1)$-entry of a matrix in $\mathbb{C}^{m\times m}$. Strengthened Valiant's conjecture for VBP $\neq$ VNP, in the context of GCT program, is stated as follows:

Conjecture 1.2.2 (Mulmuley and Sohoni [MS01]) For all $c \in \mathbb{N}_{\geq 1}$, we have $\ell^{m^c-m} \cdot \mathrm{perm}_m \notin \Omega_{m^c}$, for infinitely many m.

7: Informally, Zariski closure means taking the closure of the set of polynomials (considered as points) of the class $\mathscr{C}$: Let $\mathscr{I}$ be the smallest (annihilating) ideal whose zeros cover {coefficient-vector of $g \mid g \in \mathscr{C}$}; then put in $\overline{\mathscr{C}}$ each polynomial f with coefficient-vector being a zero of $\mathscr{I}$.

1.2.2 Why care about GCT?

So far, it seems like all we have done is restated Valiant's Conjecture in a more abstract language, and perhaps slightly generalized it. But why should this formalization give us certain advantages, may not be instantly clear. Before going into particulars, we remark that essentially all of our lower bound techniques for algebraic circuits also work in the 'boundary'! Lower bound questions can be roughly translated into asymptotic geometry terms as follows:

Given a sequence of some 'nice' vector spaces V_n, and sequences of points and groups, does the inclusion (by inclusion, we mean the points under the group action in V_n) fail *for every $n \geq n_0$, for some n_0?*

So far, so good. However, the linchpin of the GCT program is that both the permanent and determinant are 'highly' symmetric functions (see below), and it is plausible that we can leverage that fact to learn more about their orbit closures than we could if they were arbitrary functions.

For starters, note that, $\mathsf{perm}(X)$ is symmetric under permuting X's rows or columns, transposing X, and multiplying the rows or columns by scalars that multiply to 1. Formally,

$$\mathsf{perm}(X) \;=\; \mathsf{perm}(X^T) \;=\; \mathsf{perm}(AXB) \;=\; \mathsf{perm}(PXQ)\,,$$

for all *permutation* matrices P and Q, and all *diagonal* matrices A and B such that $\mathsf{perm}(A) = 1/\mathsf{perm}(B)$. The determinant has an *even larger* symmetry group; we have

$$\det(X) \;=\; \det(X^T) \;=\; \det(AXB)\,,$$

where A and B are matrices (*not necessarily* diagonal) such that $\det(A) = 1/\det(B)$.

But there is a further point (and this is really what makes GCT powerful). It turns out that both are *uniquely* characterized (up to a constant factor) by their symmetries, among all homogeneous polynomials of the same degree. Formally,

Theorem 1.2.1 *Let f be any degree-d homogeneous polynomial in the entries of $X \in \mathbb{C}^{d\times d}$.*

1. *If f satisfies the following: $f(X) = f(PXQ) = f(AXB)$, for all permutation matrices P, Q, and diagonal A, B, with $\mathsf{perm}(A) = 1/\mathsf{perm}(B)$. Then,*

   $$f(X) \;=\; \alpha \cdot \mathsf{perm}(X)\,,$$

   for some $\alpha \in \mathbb{C}$.

2. *If f satisfies the following: $f(X) == f(AXB)$, with matrices A, B, with $\det(A) = 1/\det(B)$. Then,*

   $$f(X) \;=\; \alpha \cdot \det(X)\,,$$

   for some $\alpha \in \mathbb{C}$.

For a proof, we refer to [Gro12, Section 3.4].

Moreover, lower bounds are equivalent to orbit closure containment [Gro12, Section 3.3.2]. Therefore, the 'simplest'

way of proving lower bounds would be to find *occurrence obstructions*, i.e., finding an irreducible representation with multiplicity for permanent larger than that of determinant. This is the place where one hopes to use rich mathematics like algebraic geometry and representation theory.

❑ Escaping the Razborov–Rudich barrier

We point out that the symmetry-characterization of the permanent and determinant avoids the *Razborov–Rudich barrier*. Since very few functions are symmetry-characterized [8], the symmetry-characterization violates the 'largeness' criterion! For a proof, see [Gro12, Proposition 3.4.9].

Truth be told, $\mathsf{VNP} \not\subseteq \overline{\mathsf{VBP}}$ takes us 'closer' to $\#\mathsf{P} \neq \mathsf{NC}$. This is really because perm is similar to what #P captures in the Boolean complexity, while det is what NC captures. However, a similar-ish formulation (which also can be 'somewhat' symmetry-characterized) does imply $\mathsf{P} \neq \mathsf{NP}$!

To show the above, one can define an NP-function called E, via a product of some $0-1$ determinants, as well as a P-complete function, called H, via the *universal circuit* representations. Finally, one can show them to be characterized by symmetries in only a slightly *weaker* sense than the permanent and determinant are. However, if one can find explicit representation-theoretic obstructions, which in this case would be representations associated with the orbit of E, but not with the orbit of H, then such obstructions will suffice to show that $\mathsf{P} \neq \mathsf{NP}$. For details, we refer to [Gro12; Aar16].

8: A homogeneous polynomial f of degree d on n variables is *symmetry-characterized*, if it is the only such homogeneous polynomial that is fixed by the symmetries of f, up to scalar multiplication. In other words, if g is another n-variate, degree-d polynomial such that $A \cdot g = g$, for all $A \in \mathrm{Stab}_{\mathrm{GL}(\mathbb{C}^n)}(f)$, then $g = \alpha f$ for some scalar α. Here Stab denotes the *stablizer* group.

❑ Other GCT implications

Outside lower bound implications, GCT has deep connections with –

(i) computational invariant theory [FS13b; Mul12; Gar+16; Bür+18; IQS18],

(ii) algebraic natural proofs [Gro+17; Blä+21; Cha+20; Kum+22],

(iii) lower bounds [BI13; Gro15; LO15],

(iv) derandomization [Mul17; Muk16; DDS21b],

(v) optimization [All+18; Bür+19], and many more.

1.3 Contribution of the Thesis

This thesis studies various algebraic and geometric aspects of complexity theory and their interconnections. In particular, in this thesis, we study uni-/multi-variate polynomials computed (or, *approximated*) by restricted depth-3 and depth-4 algebraic circuits. We now informally state the motivation and the results of this thesis, before discussing them in more detail in subsequent sections.

1.3.1 Sum-of-squares

Sum-of-squares (SOS) optimization is an active area of research, which lies at the interface of algorithmic algebra and convex optimization. Over the last decade, it has made significant impact on both discrete and continuous optimization, as well as several other disciplines, notably control theory. A particularly exciting aspect of this research area is that it leverages classical results from real algebraic geometry, some dating back to prominent mathematicians like Hilbert. Yet, it offers a modern, algorithmic viewpoint on these concepts, which is amenable to computation and deeply rooted in *semidefinite* programming. The SOS can be motivated through the following polynomial optimization problem:

$$\begin{aligned} &\text{minimize} \quad p(x), \\ &\text{subject to} \quad x \in K := \{x \in \mathbb{R}^n \mid g_i(x) \geq 0, h_i(x) = 0\}, \end{aligned} \tag{1.1}$$

where p, g_i, and h_i are multivariate polynomials. The set defined by a finite number of polynomial inequalities (such as the set K above) is called *semialgebraic*. The special case of problem (Equation 1.1), where the polynomials p, g_i, h_i all have degree 1, is of course the *linear programming*, which we can solve in polynomial-time [Kha79; Kar84].

A more interesting perspective is that if we could optimize over the set of polynomials that take *nonnegative* values over given semialgebraic sets, then we could solve problem (Equation 1.1) globally. To see this, note that the optimal value of problem (Equation 1.1) is equal to the optimal value of the following problem:

$$\begin{aligned} &\text{maximize} && \gamma \\ &\text{subject to} && p(x) - \gamma \geq 0,\ \forall\, x \in K. \end{aligned} \tag{1.2}$$

It is a folklore result that testing membership to the set of polynomials that take nonnegative values over a semialgebraic set K is NP-hard, even when $K = \mathbb{R}^n$. [9]

9: One can show a *stronger* result that given a polynomial p of degree 4, it is NP-hard to decide if it is positive definite, i.e., if $p(x) > 0$, for all $x \in \mathbb{R}^n$.

If a polynomial is nonnegative, can we write it in a way that its nonnegativity becomes obvious? One way to achieve this goal is to try to write the polynomial as a sum of squares of polynomials. In analysis, a polynomial p is a sum-of-squares (SOS), if it can be written as $p(x) = \sum_i q_i^2(x)$, for some polynomials q_i. Existence of the polynomials q_i, of an SOS decomposition are *algebraic certificates* for nonnegativity. At this point, the obvious question is whether every nonnegative polynomial can be expressed as an SOS! As it turns out, at least for univariate polynomials they are *equivalent*! So, in the below, we will only consider univariate polynomials, unless specified otherwise.

Following up on the previous thread, as computer scientists, we could (& definitely should) ask how *large* the certificates are. One possible notion of largeness could be the large *sparsity* (the number of monomials) of the polynomials, denoted as $\mathrm{sp}(\cdot)$. So, here is an interesting polynomial optimization question.

$$\begin{aligned} &\text{minimize} && \sum_i \mathrm{sp}(q_i) \\ &\text{subject to} && p = \sum_i q_i^2\,. \end{aligned} \tag{1.3}$$

However, note that SOS is *not a complete* model, i.e., there are polynomials which *cannot* be written as sum-of-squares, e.g., $p(x) = -(x+1)^4$. To avoid this, one could work with *weighted SOS*, i.e., $p = \sum_i c_i q_i^2$, where $c_i \in \mathbb{R}$ [10]. In this case, it is not hard to show that for *any* polynomial $p \in \mathbb{R}[x]$, with $\deg(p) = d$, the above minimization problem has value roughly in the interval $[\sqrt{d}, d]$. Here are 3 meta questions that could be immediately asked.

10: The *nonnegativity* is gone now! But the question can still be framed. In this thesis, we will be working with the weighted version only, and still call it SOS.

Meta Questions on SOS

Q1. Are there *explicit*[11] polynomials that has a *large* minimum value (in the above sense)?

Q2. Is there any link between this optimization problem and strong complexity-theoretic lower bounds?

Q3. Can some properties of the polynomial p (e.g., roots, coefficients) be related to the minimum value attained?

11: Polynomials whose coefficients can be easily computed.

Surprisingly, it turns out that these questions are very much interrelated, and further they have serious implications in complexity theory. In fact, understanding the optimization problem for easy-looking polynomials like $p = (x+1)^d$, are strikingly hard with far-fetching consequences. Below, we informally state our meta-theorems on SOS.

We circumspect that some statements are deliberately left vague, and some may even be formally incorrect. Nonetheless, the statements are "morally" correct. Also, it is our opinion that the reader has little to gain and appreciate from simply reading the formal statement without getting the essence.

Theorem 1.3.1 (Informal results on SOS [DST21; Dut21]) *If the minimum value for the polynomial $(x+1)^d$, in (Equation 1.3), attains $\omega(d^{1/2})$, then* VP $\neq$ VNP. *Further, one can also derive a non-trivial derandomization of PIT.*

Moreover, any polynomial with real roots (e.g., $(x+1)^d$) should have large SOS representations (of size $\omega(d^{1/2})$), and assuming this, the consequences should be similar to those discussed above.

Essentially, we study Q2, Q4-Q5, and their interrelations in the SOS-model. For detailed results and discussions, we refer to Chapter 3.

1.3.2 Polynomial Identity Testing (PIT)

In the previous section, we focused on the real roots and its connection to the hardness questions. One can flip the coin and focus on the opposite direction, e.g. finding the non-roots and 'easiness', i.e., upper bounds/algorithms. This is exactly where Polynomial Identity Testing (PIT) comes in.

We already have an efficient (polynomial-time) randomized algorithm for PIT due to *Polynomial Identity Lemma* [Ore22; DL78; Zip79; Sch80], as stated below. The challenge is to remove the randomness, which leads to a bigger open question in the area of Complexity Theory, namely, the BPP vs. P question, which we will discuss after stating the lemma.

Lemma 1.3.2 (Polynomial Identity Lemma) *Let $f \in \mathbb{F}[x]$ be a n-variate nonzero polynomial of degree d, and $S \subseteq \mathbb{F}$. Suppose $a := (a_1, a_2, \ldots, a_n)$ such that each a_i are selected independently and uniformly at random from S. Then we have*

$$\Pr\left[f(a) \neq 0\right] \geq 1 - \frac{d}{|S|}.$$

❑ BPP vs. P: A personal take

We view computers as instruments that are able to achieve formidable results due to their high speed and extreme precision. Computers are indeed very precise; if programmed correctly, they almost never err. This is why precision appears to be at odds with randomness. We often associate randomness with disorganized behavior, which is apparently not good for solving hard problems! So, how can one make any good use of it? And, more importantly, if we could make good use of randomness, is it *necessary* to employ randomness for efficiency? To answer this formally, we define two classes, P and BPP.

The class P is the set of problems which can be solved in deterministic polynomial-time, and the class BPP is the set of problems which can be solved in randomized polynomial-time. Interestingly, by only a few repetitions, we can make the error probability so small that an error will never occur in practice. This infinitesimally small gap enables us to realize that BPP is 'tantalizingly close' to P. And in fact, the following is widely believed to be true.

Conjecture 1.3.1 BPP = P.

Conjecture 1.3.1 *does not* imply that randomness is completely useless. The conjecture merely says that we can eliminate randomness by incurring (polynomially) more time. It is likely that some problems need, say, quadratic-time if computed deterministically, but only linear-time when computed

probabilistically. In large scale computations done in practice this can make a big difference!

❑ Blackbox PIT

Coming back to the PIT[12] problem, we study it in the *blackbox* setting [*], where only evaluations at points are allowed. Designing a deterministic blackbox PIT algorithm for a circuit class is *equivalent* to finding a set of points such that for every nonzero circuit in that class, the set contains a point where it evaluates a nonzero value (or equivalently, a *non-root*). Such sets are called *hitting sets* (see Definition 2.7.1).

12: Unlike NP, we do not know any complete problem for BPP. Hence, we try to find "good" problems which have efficient randomized solutions and try to derandomize them. This gives us the hope that the newly developed tools for derandomizing that good problem may help to derandomize BPP class completely. By Lemma 1.3.2, PIT is already in BPP. This already makes derandomizing PIT an interesting 'theoretical struggle'!

For n-variate, degree-d polynomials, a slightly clever derandomization of the Lemma 1.3.2, gives the *optimal* $\binom{n+d}{d}$ size trivial hitting set [BP20]. Unfortunately, $\binom{n+d}{d}$ can be *exponential* in the circuit size s. It is also known that an $O(s)$ size hitting set exists for an s size circuit; see [HS80a] and [Mit13, Theorem 2.7.3]. Therefore, the goal in PIT is to design an *explicit* poly(*snd*) size hitting set.

In a surprising result, Agrawal and Vinay [AV08] showed that a complete derandomization of blackbox identity testing for just depth-4 algebraic circuits (ΣΠΣΠ) already implies a *near-complete* derandomization for the general PIT problem. ΣΠΣΠ circuits compute polynomials of the form $\Sigma_i \Pi_j g_{ij}$, where g_{ij} are sparse polynomials.

More recently, the bootstrapping phenomenon [AGS19; And20; KST19] shows that PIT for very restricted classes of depth-4 circuits, even depth-3, would have very interesting consequences for PIT of general circuits. These results make the identity testing regime for depth-4 circuits, a very meaningful pursuit. Very recently, a breakthrough result [LST21], by Limaye, Srinivasan and Tavenas, gave the *first* deterministic subexponential-time PIT algorithm for all constant-depth circuits! This automatically raises the following meta question.

Meta Question on PIT for constant-depth circuits

Can we design *better* than subexponential-time PIT for *all* constant-depth circuits?

[*] There is a *whitebox* setting as well, where we are allowed to see the internal structure of the circuit!

We partially answer this for bounded depth-4 circuits.

Theorem 1.3.3 (Informal result on depth-4 PIT [DDS21a]) *There is a deterministic* quasi*polynomial-time blackbox PIT for bounded top (& bottom)-fanin depth-4 circuits.*

Interestingly, efficient PITs for the depth-4 models, with the first Π-gate, replaced by ∧, are already known [For15; Gur+17]. We exploited and almost reduced our models to the corresponding ∧-models, and Eureka! For details, we refer to Chapter 4.

1.3.3 De-bordering and derandomizing restricted classes

From the definition, it is clear that the border of a circuit class can be *potentially* much richer than the circuit class itself. Quantitatively understanding *how much richer*, is an important and wide open question in algebraic complexity and GCT. On the other hand, one of the lessons of GCT is that algebraic complexity is algebraic geometry, sometimes in (thinly veiled) disguise! In the same spirit, here is a meta-question, that is seemingly natural.

De-bordering

Given a polynomial $f \in \overline{\mathscr{C}}$, for some class $\mathscr{C}$, find another 'nice' class $\mathscr{D}$, such that $f \in \mathscr{D}$.

If $\mathscr{C} = \overline{\mathscr{C}}$, then $\mathscr{C}$ is said to be closed under approximation. A few interesting examples are:

1. ΣΠ, the sparse polynomials, with complexity measure being sparsity.
2. ΠΣ, the product of linear polynomials (Lemma 6.2.1).
3. Monotone ABPs [Blä+].
4. ROABP (read-once ABP) respectively ARO (*any-order ROABP*), with measure being the width. ARO is an ABP with a natural restriction on the use of variables per layer; for definition and a formal proof, see Definition 2.6.3 and Lemma 6.2.3.

Interestingly, depth-2 circuits are *closed* under taking limit, i.e., $\overline{\Pi\Sigma} = \Pi\Sigma$, and $\overline{\Sigma\Pi} = \Sigma\Pi$; for a proof, see Lemma 6.2.1. Now this leads to the following meta-question.

Meta-question on de-bordering

Can we de-border *all* constant-depth circuits? Or, are they strictly more powerful?

In this thesis, we study border depth-3 circuits, and de-border them by introducing a general de-bordering technique; we call it DiDIL [DDS21b].

Theorem 1.3.4 (Results on border depth-3 circuits [DDS21b]) *Constant fanin border depth-3 circuits, $\overline{\Sigma^{[k]}\Pi\Sigma}$, are 'easy'. Moreover, PITs for these classes can be efficient derandomized.*

We also study τ-conjectures in the border setting, and show stronger complexity theoretic consequences. For details, we refer to Chapter 6-Chapter 8.

1.4 Organization of the Thesis

In Chapter 2, we shall fix some notations which will be maintained for the rest of the thesis (unless specified otherwise). It will also provide some necessary technical background and mathematical tools for our results. The work on hardness and roots of polynomials represented as a sum-of-squares (SOS), and its variants, is discussed in Chapter 3. Chapter 4 discusses *quasi*deranomization of two restricted depth-4 models, that had been left open for quite some time. We finish the first part of the thesis on algebraic complexity, by some concluding remarks and future directions in Chapter 5.

Next, we move on to the second part of the thesis where we discuss our works on GCT. Chapter 6 discusses efficient de-bordering of constant-fanin border depth-3 circuits, while Chapter 7 discusses effiient derandomization of the same. Next, Chapter 8 extends the results of Chapter 3 in the border paradigm. Finally, we provide some concluding remarks and some directions for future work on GCT in Chapter 9. Some familiarities with basic algebra (rings, fields, inequalities)

would be very helpful to the reader in understanding most of the thesis.

Success is failure turned inside out ...
And you can never tell how close you are.
It may be near when it seems so far.
So stick to the fight when you're hardest hit.
It's when things seem worst that you must not quit.

— Edgar A. Guest, Don't Quit.

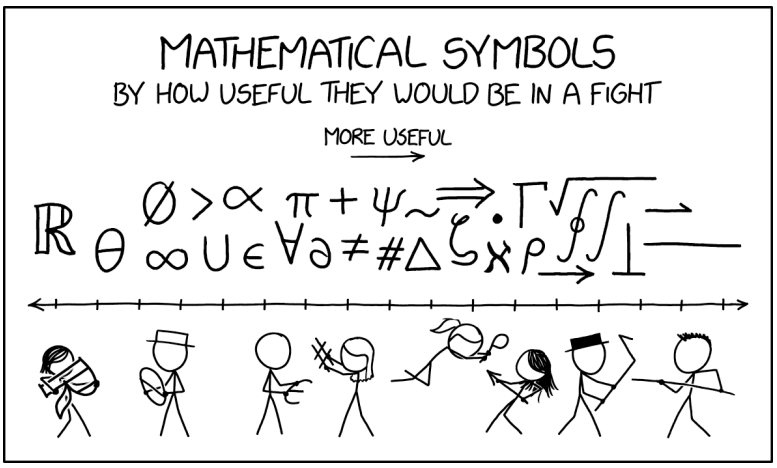
Preliminaries 2

"It starts with intellectual humility—knowing what we don't know.....If knowledge is power, knowing what we don't know is wisdom."

— Adam Grant, Think Again.

In this chapter, we will formally define and state some concepts and results that will be useful for the rest of the thesis. We will also give proof sketches of many of the results. Finally, we will fix some notation that we use throughout the thesis.

2.1 Basic Notation



xkcd.com

- We use lower-case boldface characters like $x, y, z, a, \ldots$, to denote vectors (ordered sets) of variables and constants, and use indexed lower-case letters to refer to individual elements, e.g., $x = \{x_1, x_2, \ldots, x_n\}$. The number of variables n may vary from context to context.

- We work with fields $\mathbb{F} = \mathbb{Q}, \mathbb{Q}_p$, or their fixed extensions. All of our results hold for fields with large enough characteristic.
- We denote $[n] = \{1, \ldots, n\}$. For nonnegative integers a, b, we denote the set $\{a, \ldots, b\}$ by $[a, b]$.
- For $i \in \mathbb{N}$ and $b \geq 2$, we denote by $\text{base}_b(i)$ the unique k-tuple $(i_1, \ldots, i_k)$ such that $i = \sum_{j=1}^{k} i_j \cdot b^{j-1}$.
- For an exponent vector $e = (e_1, \ldots, e_m)$, we use x^e to denote the monomial $x_1^{e_1} \cdots x_m^{e_m}$.
- We use $\mathbb{F}[x]$, as the ring over the field $\mathbb{F}$, which consists of polynomials on the variables x. And, $\mathbb{F}(x)$, as the fraction field, which consists of rational polynomials $g(x)/h(x)$, where $g, h \in \mathbb{F}[x]$, and h is nonzero. Finally, $\mathbb{F}[[x]]$ denotes the ring of formal power series on x. For definitions, see below.

Polynomials. For $p \in \mathbb{F}[x]$, where $x = (x_1, \ldots, x_m)$, for some $m \geq 1$, the *support of* p, denoted by $\text{supp}(p)$, is the set of nonzero monomials in p. The *sparsity* or *support size of* p is $\text{sp}(p) := |\text{supp}(p)|$. If p is m-variate of degree d, its sparsity is bounded by

$$\text{sp}(p) \leq \binom{m+d}{d}. \tag{2.1}$$

By $\text{coef}(p)$ we denote the *coefficient vector* of p (in some fixed order).

For a polynomial $p(x, y) \in \mathbb{F}[x, y]$, the *x-degree of* p, denoted by $\deg_x(p)$, is the maximum degree of x in p. That is, for $p(x, y) = \sum_e p_e(x) y^e$, we define $\deg_x(p) = \max_e \deg(p_e(x))$.

Notation for algebraic models. Here are some of the notations and symbols used uniformly for denoting algebraic models in the thesis. Unless there is an explicitly written bound on the top fanin, it is assumed to be unbounded (bounded by the size of the circuit).

- $\Sigma\Pi$ computes sparse polynomials of the form $\sum_i c_i x^{e_i}$, where each $e_i \in \mathbb{Z}_{\geq 0}^n$, and $c_i \in \mathbb{F}$.
- $\Sigma\wedge$ computes polynomials of the form $\sum_{i \in [n]} g_i(x_i)$.

- $\Sigma\wedge\Sigma$ computes polynomials of the form $\sum_i c_i \cdot \ell_i^{e_i}$, where ℓ_i are linear polynomials of the form $a_{i0} + \sum_{j\in[n]} a_{ij}x_j$. Each $c_i, a_{ij} \in \mathbb{F}$.
- $\Sigma\wedge\Sigma\wedge$ computes polynomials of the form $\sum_i c_i \cdot g_i^{e_i}$, where $g_i = \sum_{j\in[n]} g_{ij}(x_j)$, is a sum of univariates.
- $\Pi\Sigma\Pi$ computes product of sparse polynomials, i.e. $\prod_i g_i$, where each g_i is a sparse polynomial.
- $\Sigma\wedge\Sigma\Pi^{[\delta]}$ computes polynomials of the form $\sum_i f_i(x)^{e_i}$ where $\deg f_i \leq \delta$.
- $\Sigma\Pi\Sigma$ computes polynomials of the form $\sum_i \prod_j \ell_{ij}$, where ℓ_{ij} are linear polynomials. $\Sigma^{[k]}\Pi\Sigma$ is of the same form as above, with the top fanin being bounded by k.
- $\Sigma^{[k]}\Pi\Sigma\wedge$ computes polynomials of the form $\sum_{i\in[k]} \prod_j (g_{ij1}(x_1) + \cdots + g_{ijn}(x_n))$, where $g_{ij\ell} \in \mathbb{F}[x_\ell]$.
- $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ computes polynomials form $\sum_{i\in[k]} \prod_j g_{ij}(x)$, where $\deg(g_{ij}) \leq \delta$.

One important remark is that the definitions of the sizes of these circuits are rather nonuniform in the literature. Since, all these models have small formulas, we also include the exponents and the sparsity of the base polynomials in the circuit size, for brevity and simplicity of calculations. When needed for the ease of reading, we will explicitly define the models and sizes again, in the respective chapters. Finally, we will also be using ABP and ARO, extensively, throughout the thesis. For their definitions and properties, we refer to section 2.6.

Formal power series. The analytic tool that we will use quite heavily in the thesis is the *formal power series*; it often appears in algebra and complexity theory.

In mathematics, a formal power series is a strict generalization of a polynomial, where the number of terms is allowed to be infinite; this implies giving up the possibility of replacing the variables in the polynomial with arbitrary numbers. One may think of a formal power series as a power series in which we ignore questions of *convergence* by assuming the variables as really 'variables' and *not* assuming numerical values. For example, consider the series

$$f(x) = 1 - 3x + 5x^2 - 7x^3 + 9x^4 - \dots .$$

If we studied this as a power series, its properties would include, for example, that its *radius of convergence* is 1. However, as a formal power series, we may ignore this completely; the sequence of coefficients $[1, -3, 5, -7, 9, \ldots]$ is the only thing relevant to us. Basic arithmetic operations, like, addition and multiplication on formal power series are carried out by simply pretending that the series are polynomials (or vectors) of infinite length. Once we have defined *multiplication* for formal power series, we can define multiplicative inverses as follows. The multiplicative inverse of a formal power series A is a formal power series C such that $A \cdot C = 1$, provided that such a formal power series exists. It turns out that if A has a multiplicative inverse, it is unique, and we denote it by A^{-1}. For example, the following identity plays a very crucial role.

$$\frac{1}{1-x} = \sum_{i \geq 0} x^i \,, \qquad \textit{(The inverse identity)} \,.$$

Logarithmic derivative. Over a ring R and a variable y, the logarithmic derivative $\mathrm{dlog}_y : \mathsf{R}[y] \to \mathsf{R}(y)$ is defined as

$$\mathrm{dlog}_y(f) := \frac{\partial_y f}{f} \,.$$

Here ∂_y denotes the partial derivative with respect to the variable y. One important property of dlog is that it is additive over a product, as

$$\begin{aligned} \mathrm{dlog}_y(f \cdot g) = \frac{\partial_y(f \cdot g)}{f \cdot g} &= \frac{(f \cdot \partial_y g + g \cdot \partial_y f)}{f \cdot g} \\ &= \mathrm{dlog}_y(f) + \mathrm{dlog}_y(g) \,. \end{aligned}$$

We refer to this effect as *linearization* of product.

Kronecker map and its inverse. The *Kronecker substitution* is a bijective map between univariate and multivariate polynomials. We define two variants: The first one is the standard one, the second one is a multilinear version of it. In our application, we consider the sparsity of the polynomials. There it seems as the standard Kronecker substitution does not yield the bounds we need. Let $p(x)$ be a univariate polynomial of degree d.

1) `Standard Kronecker substitution.` Let k and n be such that $n = \lceil (d+1)^{1/k} \rceil - 1$. Introduce k variables $x = (x_1, \ldots, x_k)$. Define the *Kronecker map* $\phi_{k,n}$ by

$$\phi_{n,k} : x^i \mapsto x^{\text{base}_{n+1}(i)}, \tag{2.2}$$

for all $i \in [d]$. By linear extension, define polynomial $P_{n,k} = \phi_{n,k}(p)$. Note that $\phi_{k,d}$ maps each x^i to a *distinct* k-variate monomial of individual degree $\leq n$, for $0 \leq i \leq d$.

Next, we consider the inverse map. Let $P(x_1, \ldots, x_k)$ be a polynomial, where the variables have individual degree bounded by n. Define $\psi_{n,k}$ by

$$\psi_{n,k} : x_i \mapsto x^{(n+1)^{i-1}}, \tag{2.3}$$

for $0 \leq i \leq k$, and $\psi_{n,k}(P)$ by linear extension. Note that the degree of $\psi_{n,k}(P)$ is bounded by $\sum_{i=1}^{k} n(n+1)^{i-1} = (n+1)^k - 1$ [Kro82]. Also, we have $\psi_{n,k} \circ \phi_{n,k}(p) = p$.

2) `Multilinear Kronecker substitution.` Here, we choose k and n such that $(k-1)^n \leq d+1 \leq k^n$. Introduce kn variables $y_{j,\ell}$, where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$. For every $i = 0, 1, \ldots d$, write i in base-k representation, $\text{base}_k(i) = (i_1, \ldots, i_n)$. Define the injective map $\phi^{\text{lin}}_{n,k}$ by

$$\phi^{\text{lin}}_{n,k} : x^i \mapsto \prod_{j=1}^{n} y_{j,i_j}. \tag{2.4}$$

By linear extension, define polynomial $P_{n,k} = \phi^{\text{lin}}_{n,k}(p)$. Note that $P_{n,k}$ is a kn-variate multilinear polynomial of degree n.

Mapping $\phi^{\text{lin}}_{n,k}$ can be inverted by $\psi^{\text{lin}}_{n,k}$,

$$\psi^{\text{lin}}_{n,k} : y_{j,\ell} \mapsto x^{\ell \cdot k^{j-1}}. \tag{2.5}$$

Again by linear extension, we have $\psi^{\text{lin}}_{n,k} \circ \phi^{\text{lin}}_{n,k}(p) = p$.

It is also important to note that the sparsity of the polynomials stays the same,

for the standard and the multilinear Kronecker map and their inverses.

2.2 Basic Mathematical Tools

❑ Binomial inequalities

For binomial coefficients, we use the following well known bounds. For $1 \leq k \leq n$,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \tag{2.6}$$

There are multiple proofs known for the above inequality. Here is my favorite one.

Proof. Both side inequalities are extremely elegant.

The right inequality.

$$\begin{aligned}\binom{n}{k} = \frac{n!}{(n-k)!k!} &= \frac{n\cdot(n-1)\cdots(n-k+1)}{k!} \\ &\leq \frac{n^k}{k!} \\ &= \frac{n^k}{k^k}\cdot\frac{k^k}{k!} \\ &\leq \frac{n^k}{k^k}\cdot\left(\sum_{j\geq 0}\frac{k^j}{j!}\right) = \left(\frac{en}{k}\right)^k.\end{aligned}$$

In the above, we used the fact that $e^k = \sum_{j\geq 0}\frac{k^j}{j!}$.

The left inequality.

$$\begin{aligned}\binom{n}{k} = \frac{n!}{(n-k)!k!} &= \frac{n\cdot(n-1)\cdots(n-k+1)}{k!} \\ &= \prod_{j=0}^{k-1}\frac{n-j}{k-j} \\ &\geq \prod_{j=0}^{k-1}\frac{n}{k} = \left(\frac{n}{k}\right)^k.\end{aligned}$$

In the above, we used the fact that $\frac{n-j}{k-j} \geq \frac{n}{k}$, for any $n \geq k$. This finishes the proof. □

❑ A maximization problem

For the time-complexity bound in Chapter 4, we need to optimize the following function:

Lemma 2.2.1 *Let $k \in \mathbb{N}_{\geq 4}$, and $h(x) := x(k-x)7^x$. Then,*

$$\max_{i \in [k-1]} h(i) \;=\; h(k-1)\,.$$

Proof sketch. Differentiate to get

$$\begin{aligned} h'(x) &= (k-x)7^x - x7^x + x(k-x)(\log 7)7^x \\ &= 7^x \cdot \Big[x^2(-\log 7) + x(k\log 7 - 2) + k\Big]\,. \end{aligned}$$

Note that, it vanishes at

$$x \;=\; \left(\frac{k}{2} - \frac{1}{\log 7}\right) + \sqrt{\left(\frac{k}{2} - \frac{1}{\log 7}\right)^2 - \frac{k}{\log 7}}\,.$$

It is not hard to show that $x \approx k-1$. Thus, h is maximized at the integer $x = k-1$ (the maximization is only over positive integers). □

❑ Valuation and its properties

Valuation is a map $\mathrm{val}_y : \mathsf{R}[y] \to \mathbb{Z}_{\geq 0}$, over a ring R, such that $\mathrm{val}_y(\cdot)$ is defined to be the maximum power of y dividing the element. It can be easily extended to fraction field $\mathsf{R}(y)$, by defining $\mathrm{val}_y(p/q) := \mathrm{val}_y(p) - \mathrm{val}_y(q)$; where it can be negative.

Here is an important lemma to show that positive valuation with respect to y, lets us express a function as a power-series of y.

Lemma 2.2.2 (Valuation) *Let $f \in \mathbb{F}(x,y)$ such that $\mathrm{val}_y(f) \geq 0$. Then, $f \in \mathbb{F}(x)[[y]] \bigcap \mathbb{F}(x,y)$* [1].

1: $\mathbb{F}(x,y)$ is the fraction field, which consists of elements of the form $\frac{p}{q}$, where $p, q \in \mathbb{F}[x,y]$.

Proof sketch. Let $f = g/h$ such that $g, h \in \mathbb{F}[x,y]$. Now, $\mathrm{val}_y(f) \geq 0$, implies $\mathrm{val}_y(g) \geq \mathrm{val}_y(h)$. Let $\mathrm{val}_y(g) = d_1$ and $\mathrm{val}_y(h) = d_2$, where $d_1 \geq d_2 \geq 0$. Further, write $g = y^{d_1} \cdot \tilde{g}$ and

$h = y^{d_2} \cdot \tilde{h}$. Write, $\tilde{h} = h_0 + h_1\, y + h_2\, y^2 + \cdots + h_d\, y^d$, for some d; with $h_i \in \mathbb{F}[x]$. Note that $h_0 \neq 0$. Thus

$$\begin{aligned} f &= y^{d_1-d_2} \cdot \tilde{g}/(h_0 + h_1 y + \cdots + h_d y^d) \\ &= y^{d_1-d_2} \cdot (\tilde{g}/h_0) \cdot \left((h_1/h_0) + (h_2/h_0)y + \cdots + (h_d/h_0)y^d\right)^{-1} \\ &\in \mathbb{F}(x)[[y]] \end{aligned}$$

□

❑ Waring identity

We will often need an efficient way to write a product of a few powers as a sum of powers, using simple interpolation. For an algebraic proof, see [CCG12, Proposition 4.3].

Lemma 2.2.3 (Waring Identity for a monomial) *Let* $M = x_1^{b_1} \cdots x_k^{b_k}$, *where* $1 \leq b_1 \leq \ldots \leq b_k$, *and* roots of unity $\mathcal{Z}(i) := \{z \in \mathbb{C} : z^{b_i+1} = 1\}$. *Then,*

$$M = \sum_{\varepsilon(i) \in \mathcal{Z}(i) : i=2,\cdots,k} \gamma_{\varepsilon(2),\ldots,\varepsilon(k)} \cdot (x_1 + \varepsilon(2)x_2 + \ldots + \varepsilon(k)x_k)^d \,,$$

where $d := deg(M) = b_1 + \ldots + b_k$, *and* $\gamma_{\varepsilon(2),\ldots,\varepsilon(k)}$ *are scalars and ther are* $\mathrm{WR}(M) := \prod_{i=2}^{k} (b_i + 1)$ *many such scalars.*

Remark 2.2.1 We actually need not work with $\mathbb{F} = \mathbb{C}$. We can go to a small extension (at most d^k), for a monomial of degree d, to make sure that $\varepsilon(i)$ exists.

❑ Roots of univariate polynomials

Real-rooted polynomials will play a crucial role in Chapter 3. When are all the roots of a univariate polynomial *real* and *distinct*? Kurtz [Kur92] came up with the following *tight* and *sufficient* condition.

Theorem 2.2.4 ([Kur92]) *Let* f *be a real polynomial of degree* $n \geq 2$ *with* positive *coefficients. If*

$$a_i^2 \;>\; 4a_{i-1}a_{i+1}\,, \quad \forall\, i \in [n-1]\,,$$

then all the roots of f *are real and distinct.*

Remark 2.2.2 1. Kurtz [Kur92] further showed the following: Given $\epsilon > 0$, and an integer $n > 2$, there is a polynomial with positive coefficients of degree n which has some non-real roots and whose coefficients satisfy:

$$a_i^2 \ > \ (4-\epsilon)a_{i-1}a_{i+1} \ .$$

2. The requirement for *positive* coefficients is necessary. For e.g., $x^3 - 5x^2 + 6x + 1$, has two *non-real* roots, namely

$$\frac{1}{12}\Bigg(20 + 14(1-\iota\sqrt{3})\sqrt[3]{\frac{2}{47-3\sqrt{93}}} + 2^{2/3}(1+\iota\sqrt{3})\sqrt[3]{47-3\sqrt{93}}\Bigg),$$

and its conjugate.

Before going into details, we state a classical lemma due to Descartes, which will be used throughout the paper.

Lemma 2.2.5 (Descartes' rule of signs) *Let $p(x) \in \mathbb{R}[x]$ be a polynomial with t many monomials. Then, number of distinct positive roots in $p(x)$ can be at most $t-1$.*

Remark 2.2.3 An s-sparse polynomial $f \in \mathbb{C}[x]$ can have at most $2(s-1)$-many real roots. A real root a of f must be a real root of both the real part $\Re(f)$ and the imaginary part $\Im(f)$. By above, there can be at most $s-1$ many positive roots. The same bound holds for negative roots by $x \mapsto -x$.

2.3 Explicit Functions

It is known that most of the polynomials of degree d are *hard*, i.e. they require $\Omega(d)$ size circuits; for a self-contained proof, see [CKW11, Theorem 4.2] [2]. In fact, for p_i being the i-th prime, the polynomial $\sum_{i=0}^{d} \sqrt{p_i}\, x^i$ and $\sum_{i=0}^{d} 2^{2^i} x^i$, both require circuits of size $\Omega(d/\log d)$, see [BCS13, Corollary 9.4] & [Str74]. Such polynomials can be converted into an *exponentially hard* multilinear polynomial $f_n(x)$, using the inverse Kronecker map, section 2.1. Unfortunately, this *strong* lower bound is insufficient to separate VP and VNP because the

2: The size-bound in the previous such proofs usually counted only the number of nodes in the circuit, achieving square-root in the bound; we use the number of nodes and edges here.

polynomial family is *non-explicit*– so f_n may not be in VNP. For details, see [HS80b; Bür13].

Thus, the explicitness of the family plays a major role in its usefulness in algebraic complexity.

Definition 2.3.1 (Explicit functions) *Let $(f_d)_d$ be a polynomial family, where $f_d(x)$ is of degree d. The family is* explicit, *if its coefficient-function is computable in time* $\mathrm{poly}\log(d)$ *and each coefficient can be at most* $\mathrm{poly}(d)$*-bits long. The coefficient-function gets input (j,i,d) and outputs the j-th bit of the coefficient of x^i in f_d.*

In alternative versions, one can define explicitness via the coefficient-function, to be computable in $\#\mathsf{P}/\mathrm{poly}$, or in the *counting hierarchy* CH, which would be good enough for our purpose (see Theorem 2.4.3). To understand them well, let us go through the basic definitions of these complexity classes.

❑ Complexity classes

The counting hierarchy (CH) was first introduced in [Wag86]. It can be defined by a counting operator $\mathbf{C}$ that can be applied to complexity classes. We denote by $\{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*, (x,y) \mapsto \langle x,y\rangle$, a pairing function (e.g. by duplicating each bit of x and y and inserting 01 in between).

Definition 2.3.2 *If K is a complexity class, then we define an operator $\mathbf{C}$ acting on K. The action, denoted by $\mathbf{C}\cdot K$, produces a set of languages A, such that there exists a language $B \in K$ and a polynomial $p(\cdot)$, obeying :*

$$x \in A \iff \#\left\{y \in \{0,1\}^{p(|x|)} : \langle x,y\rangle \in B\right\} > \frac{1}{2}\cdot 2^{p(|x|)}.$$

The i-th level $C_i\mathsf{P}$ of the counting hierarchy is defined recursively as $C_0\mathsf{P} := \mathsf{P}$ and $C_i\mathsf{P} = \mathbf{C}\cdot C_{i-1}\mathsf{P}$. Finally, we define the counting hierarchy*: $\mathsf{CH} := \bigcup_{i\geq 0} C_i\mathsf{P}$. Often, $\mathsf{PP} := C_1\mathsf{P}$ is used in the literature. Observe that $C_2\mathsf{P} = \mathsf{PP}^{\mathsf{PP}}$.*

Let us recall the definition of other complexity classes. For a survey of complexity classes, see [Joh90].

- FP denotes the class of all string *functions* which can be computed by a polynomial time Turing machine.

- A *polynomial advice* is a function $\alpha : \mathbb{N} \to \{0,1\}^*$ such that $n \mapsto \alpha(n)$ is poly-bounded. The (non-uniform) class C/poly for a complexity class C consists of all string functions of the form $\psi(x) =: \phi(\langle x, \alpha(|x|)\rangle)$, where $\phi \in C$ and α is some polynomial advice function.
- The problem is in #P, the class of problems that can be defined as counting the number of accepting paths of a polynomial-time non-deterministic Turing machine.

❑ Some explicit polynomial families

Throughout the paper, we will be using some interesting examples of polynomial families. We also remark that different proofs would require different explicitness (& its relaxations)!

An *explicit* candidate for the hard family is the *Pochhammer-Wilkinson* [3] polynomial [*],

$$f_d(x) \;:=\; \prod_{i=1}^{d}(x-i)\,.$$

3: One can show that the coefficients of Pochhammer-Wilkinson polynomials are computable in CH, see [Bür09].

Other well known explicit families are: $(x+1)^d$ and the *Chebyshev* polynomial $T_d(x)$, that writes $\cos d\theta$ as a function of $\cos\theta$, i.e., $T_d(\cos\theta) = \cos d\theta$ [MH02]. [4] These three are quite relevant to this work.

4: Both $(x+1)^d$ and $T_d(x)$ can be computed by $O(\log d)$-size circuits. $(x+1)^d$ can be computed by repeated squaring, while the *Chebyshev polynomial* T_d can be computed using the following two nice identities: (i) $T_{2d}(x) = 2T_d^2(x)-1$, (ii) $T_{2d+1}(x) = 2T_{d-1}(x)\cdot T_d(x) - x$.

We will call a family CH-explicit if the coefficients are computable in CH. The proofs of Theorem 3.3.6 and Theorem 3.3.9 require CH explicitness of certain families.

Comment 2.3.1

One can show that the coefficients of $f_d(x) = (x+1)^d$, are computable in CH. Consider the identity $(x+1)^d = \sum_{k=0}^{d}\binom{d}{k}x^k$. For $x = 2^d$, we get

$$v(d) \;=\; (2^d+1)^d \;=\; \sum_{k=0}^{d}\binom{d}{k}2^{dk}\,.$$

[*] It can be showed that if $\text{size}(f_d) \le \text{poly}(\log d)$, then integer factoring is in P/poly. This connection is often dubbed as "factorials vs. factoring". For more details, we refer to [Sha79; Lip94].

Note that, $\binom{d}{k} < 2^d$. Thus, the bits of $\binom{d}{k}$, in the binary representation of $\nu(d)$, do not overlap for different k's, Hence, the bits of $\binom{d}{k}$ can be read off the bit-vector of $\nu(d)$. It is therefore sufficient to show that $\nu(d)$ is computable in CH.

Note that each bit of $2^d + 1$ can be computed in polynomial time. It is well-known that one can do exponential sum and products in CH, see [Bür09; KP11]. Therefore, we get that $\nu(d)$ is computable in CH.

2.4 VP vs VNP and the CH Collapse

In this section, we talk about the relation between Boolean and algebraic class collapses. Further, we talk about a sufficient condition for an explicit family, to be in VNP.

❑ Valiant's hypothesis and GRH

Valiant conjectured that VP ≠ VNP. Bürgisser [Bür00, Cor.1.2] showed that if Valiant's hypothesis is false and GRH holds, then the polynomial hierarchy collapses (to P/poly). In fact, something *stronger* holds: P/poly = NP/poly. From this, it is not hard to deduce the following.

Theorem 2.4.1 *If GRH is true and* VP = VNP, *then* CH ⊆ P/poly.

The proof basically follows by observing the fact that PH ⊆ P/poly $\implies$ PP ⊆ P/poly; inductively this implies that CH collapses as well. Over finite fields, GRH is not needed; GRH is required only for $\mathbb{Q}$.

A sufficient property. Valiant [Val79] showed a *sufficient* condition for a polynomial family $(f_n(x))_n$ to be in VNP. We use a slightly modified version of the criterion and formulate it only for multi-linear polynomials.

Theorem 2.4.2 (VNP criterion, [Val79], see also[Bür13]) *Let $f_n(x) = \sum_{e \in \{0,1\}^n} c_n(e)\, x^e$ be a polynomial family such that the coefficients $c_n(e)$ have length $\leq n$ in binary. Then*

$$c_n(e) \in \#\mathrm{P/poly} \implies f_n \in \mathrm{VNP}.$$

In Chapter 3 and Chapter 8, we will consider univariate polynomials, and define associated multivariate polynomials via Kronecker maps. We want all of these polynomials to be in VNP. For this, we will require further relaxation of Theorem 2.4.2 so that the coefficients $c_n(e)$ can actually be 2^n bits long. Koiran and Perifel [KP11, Lemma 3.2] used a similar idea. We also use the fact that VNP is closed under substitution. That is, for a family of polynomials $(f(x, y)) \in \mathsf{VNP}$, it also holds that $(f(x, y_0)) \in \mathsf{VNP}$, for any value $y_0 \in \mathbb{F}^n$ assigned to the variables in y.

Theorem 2.4.3 (Relaxed Valiant's criterion) *Let*

$$f_n(x) = \sum_{e \in \{0,1\}^n} c_n(e)\, x^e ,$$

be a polynomial family such that the coefficients $c_n(e)$ have length $\leq 2^n$ in binary. Let $c_{n,j}(e)$ be the j-th bit of $c_n(e)$. Then

$$c_{n,j}(e) \in \#\mathsf{P}/\mathsf{poly} \implies f_n \in \mathsf{VNP}.$$

Proof. For $j \in \{0, 1, \ldots, 2^n - 1\}$, let $\mathrm{bin}(j) = (j_1, \ldots, j_n)$ denote the n-bit base-2 representation of j such that $j = \sum_{i=1}^{n} j_i\, 2^{i-1}$. Introduce new variables $y = (y_1, \ldots, y_n)$ and define

$$\tilde{c}_n(e, y) = \sum_{j=0}^{2^n-1} c_{n,j}(e)\, y^{\mathrm{bin}(j)} .$$

Let $y_0 = (2^{2^0}, \ldots, 2^{2^{n-1}})$. Then, we have $\tilde{c}_n(e, y_0) = c_n(e)$. Finally, consider the $2n$-variate auxiliary polynomial $h_n(x, y)$.

$$h_n(x, y) = \sum_{e \in \{0,1\}^n} \tilde{c}_n(e, y)\, x^e = \sum_{e \in \{0,1\}^n} \sum_{j=0}^{2^n-1} c_{n,j}(e)\, y^{\mathrm{bin}(j)}\, x^e .$$

Then we have $h_n(x, y_0) = f_n(x)$. Since, $c_{n,j}(e)$ can be computed in $\#\mathsf{P}/\mathsf{poly}$, we have $(h_n(x, y))_n \in \mathsf{VNP}$. As VNP is closed under substitution, it follows that $(f_n(x))_n \in \mathsf{VNP}$. □

2.5 Matrix Rigidity

Before trying to prove a lower bound in the general settings, we would like to remark that one of the major open problems

in algebraic complexity is to prove any *super-linear* lower bound for *linear circuits*, defined below. These are simple circuits where we are only allowed to use addition and multiplication by a scalar. By definition, they can only compute linear (affine) functions. In fact, any algebraic circuit, computing a set of linear functions, can be converted into a linear circuit with only a constant blow-up in size, see [BCS13, Theorem 13.1]. Clearly, every set of n linear functions on n variables can be represented by a matrix in $\mathbb{F}^{n\times n}$, which can be computed by a linear circuit of size $O(n^2)$.

Given the ubiquitous role linear transformations play in computing, understanding the inherent complexity of explicit linear transformations is important. Using dimension argument/counting, it can be shown that a random matrix requires $\Omega(n^2)$-size circuit. However, showing the same for an explicit $A_n \in \mathbb{F}^{n\times n}$, still remains open. The standard notion of explicitness is that there is a deterministic algorithm which outputs the matrix A_n in poly(n)-time. Weak super-linear lower bounds are known for constant-depth linear circuits, using superconcentrators and their minimal size, see [Val75; Pip77; AP94; RT00]. It is also known that this technique alone is *insufficient* for proving lower bounds for logarithmic depth.

The quest for showing *super*linear lower bound for logarithmic depth lead to the notion of *matrix rigidity*, a pseudorandom property of matrices, introduced by Valiant [Val77], and independently by Grigoriev [Gri76].

Definition 2.5.1 (Matrix rigidity) *A matrix A over $\mathbb{F}$ is* (r, s)-rigid, *if one needs to change $> s$ entries in A to obtain a matrix of rank $\leq r$. That is, one* cannot *decompose A into $A = R + S$, where* $\mathrm{rank}(R) \leq r$ *and* $\mathrm{sp}(S) \leq s$, *where* $\mathrm{sp}(S)$ *is the* sparsity of S, *i.e., the number of nonzero entries in S.*

Valiant [Val77] showed that an explicit construction of a $(\epsilon \cdot n, n^{1+\delta})$-rigid matrix, for some $\epsilon, \delta > 0$, will imply a *super-linear* lower bound for linear circuits of depth $O(\log n)$; for a simple proof, refer to [SY10, Theorem 3.22]. Pudlak [Pud94] observed that similar rigidity parameters implies even *stronger* lower bounds for constant depth circuits. Here, we remark that a random matrix is $(r, (n-r)^2)$-rigid, but the best explicit constructions have rigidity $(r, n^2/r \cdot \log(n/r))$ [Fri93; SSS97],

which is *insufficient* for proving lower bounds. For recent works, we refer to [AC19; DGW19; Ram20].

2.6 Properties of Restricted Circuit Classes

In this section, we will be talking about properties of ROABP, $\Sigma \wedge \Sigma$ and other restricted classes and their properties, which will be used throughout the thesis.

❑ Cone-size measure and its relevance

We need to define cone-size, which will be crucially used in Chapter 7.

Definition 2.6.1 (Cone-size of monomials) *For a monomial x^a, the cone of x^a is the set of all sub-monomials of x^a. The cardinality of this set is called* cone-size *of x^a. It equals $\prod_{i\in[n]}(a_i+1)$, where $a=(a_1,\ldots,a_n)$. We will denote $cs(m)$, as the cone-size of the monomial m.*

Here is an important lemma, originally from [For14, Corollary 4.14], which shows that small partial derivative space implies existence of small cone-size monomial. For a detailed proof, we refer [Gho19, Lemma 2.3.15]

Theorem 2.6.1 (Cone-size concentration) *Let $\mathbb{F}$ be a field of characteristic 0 or greater than d. Let $\mathscr{P}$ be a set of n-variate d-degree polynomials over $\mathbb{F}$ such that for all $P \in \mathscr{P}$, the dimension of the partial derivative space of P is at most k. Then every nonzero $P \in \mathscr{P}$ has a cone-size-k monomial with nonzero coefficient.*

The next lemma shows that there are only few low-cone monomials in a non-zero n-variate polynomial.

Lemma 2.6.2 (Counting low-cones, [FGS18, Lemma 5]) *The number of n-variate monomials with cone-size at most k is $O(rk^2)$, where $r := (3n/\log k)^{\log k}$.*

The following lemma is the same as [FGS18, Lemma 4]. It is proved by multivariate interpolation.

Lemma 2.6.3 (Coefficient extraction) *Given a circuit C, over the underlying field $\mathbb{F}(\epsilon)$, we can 'extract' the coefficient of a monomial m in C; in* $\text{poly}(\text{size}(C), cs(m), d)$ *time, where $cs(m)$ denotes the cone-size of m.*

2.6.1 Properties of ABP

Some important properties of ABP, that we will use/assume throughout the thesis.

1. ABP is *closed* under both addition and multiplication: the size blow up is just *additive*. This is straightforward from the definition.
2. We will assume that the ABP is *layered*, i.e. the vertices are partitioned into layers and the edges only join successive layers. I.e. an edge from the i-th layer can only go to the $(i+1)$-th layer.

Two important measures of an ABP are:

1. *Length* – the longest path from the source to the sink.
2. *Width* – the maximum number of vertices in a layer.

We also need to eliminate division in ABPs. Here is an important lemma stated below.

Lemma 2.6.4 (Strassen's division elimination) *Let $g(x, y)$ and $h(x, y)$ be computed by ABPs of size s and degree $< d$. Further, assume $h(x, 0) \neq 0$. Then, $g/h \mod y^d$ can be written as $\sum_{i=0}^{d-1} C_i \cdot y^i$, where each C_i is of the form ABP/ABP of size $O(sd^2)$.*

Moreover, in case g/h is a polynomial, then it has an ABP of size $O(sd^2)$.

Proof. ABPs are closed under multiplication, which makes interpolation, wrt y, possible. Interpolating the coefficient C_i, of y^i, gives a sum of d ABP/ABP's; which can be rewritten as a single ABP/ABP of size $O(sd^2)$.

Next, assume that g/h is a polynomial. For a random $(a, a_0) \in \mathbb{F}^{n+1}$, write $h(x + a, y + a_0) =: h(a, a_0) - \tilde{h}(x, y)$ and define $g' := g(x + a, y + a_0)$. Clearly $0 \neq h(a, a_0) \in \mathbb{F}$ and $\tilde{h} \in \langle x, y \rangle$.

Of course, $\tilde{h}$ has a small ABP. Using the inverse identity in $\mathbb{F}[[x, y]]$, we have

$$\frac{g(x+a, y+a_0)}{h(x+a, y+a_0)} = \frac{\frac{g'}{h(a,a_0)}}{(1-\frac{\tilde{h}}{h(a,a_0)}} \equiv \frac{g'}{h(a,a_0)} \cdot \left(\sum_{0 \le i < d} \left(\frac{\tilde{h}}{h(a,a_0)} \right)^i \right) \mod \langle x, y \rangle^d .$$

Note that, the degree blowsup in the above summands to $O(d^2)$ and the ABP-size is $O(sd)$. ABPs are closed under addition/ multiplication; thus, we get an ABP of size $O(sd^2)$ for the polynomial $g(x+a, y+a_0)/h(x+a, y+a_0)$. This implies the ABP-size for g/h as well. □

2.6.2 Properties of any-order ROABP.

In this part, we will define ROABP and ARO (any-order ROABP), whose properties will be exploited throughout the thesis.

Definition 2.6.2 (ROABP) *An ABP is a read-once oblivious ABP (ROABP) if every variable x_i is present in only one of the layers of the ABP.*

Two examples. The following is an example of an ROABP computing the polynomial $\prod_{i\in[n]}(1+x_i) = \sum_{S\subseteq[n]} \prod_{i\in S} x_i$.

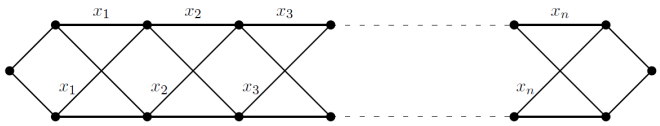


Figure 2.1: ROABP computing $\prod_{i\in[n]}(1+x_i)$

And the following is an example of an ROABP computing the *symmetric polynomial* of degree k over n variables,

$$e_{n,k} = \sum_{S\subseteq[n],|S|=k} \prod_{i\in S} x_i .$$

The following example might help the readers to think of layers as more 'robust' than what we think, in the above sense.

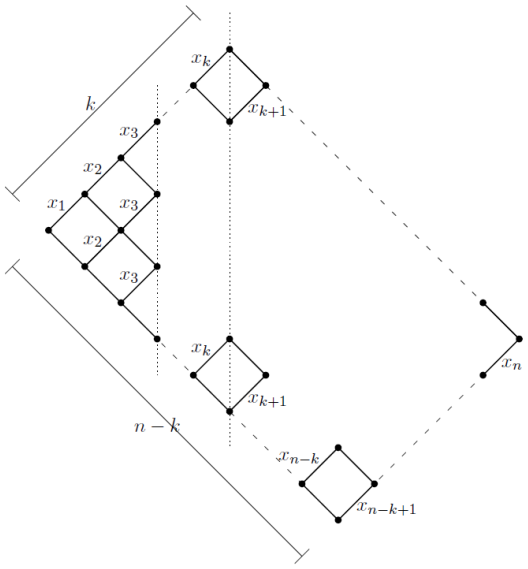


Figure 2.2: ROABP computing the symmetric polynomial $e_{n,k}$

Each path from s to t takes k-many northward steps and $n-k$-many southward steps. On every northward step, we associate a unique variable. Though the ROABP can compute some interesting polynomials, the 'read-once' restriction severely limits the power of the arithmetic branching program. E.g., Kayal, Nair and Saha [KNS20] showed a polynomial that cannot be computed by any ROABP of subexponential size.

Permutation associated with an ROABP. A *permutation* π of the variables $(x_i)_{i=1}^n$ can be associated with an ROABP. If the variables $(x_1, \dots, x_n)$ occur in the ROABP in the sequence $(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)})$, then the permutation π is associated with it. It is an important property of the ROABP. An ROABP of a small width for a polynomial may exist in one permutation π. But, it may not exist in some other permutation. E.g., The polynomial $f(x, y) = \prod_{i=1}^n (1 + x_i y_i)$ has a width 2 ROABP when the permutation is $(x_1, y_1, x_2, y_2, \dots, x_n, y_n)$. That is because $(1 + x_i y_i)$ has a width-2 ROABP, for each i.

But, it can be shown, using Lemma 2.6.5 (which also holds for individual ROABP of a particular permutation), that when the permutation is $(y_1, y_2, \dots, y_n, x_1, x_2, \dots, x_n)$, any ROABP which

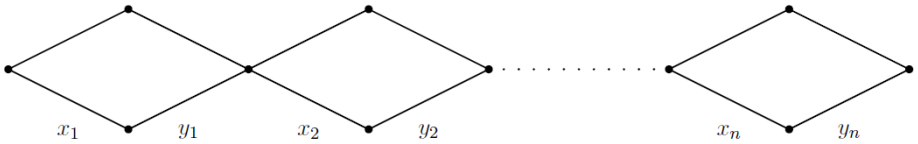


Figure 2.3: ROABP computing $\prod_{i=1}^{n}(1 + x_i y_i)$

computes $f(x, y)$ has width 2^n. We refer to [KNS20], for more examples.

Now, we defineARO.

Definition 2.6.3 (Any-order ROABP (ARO)) *A polynomial $f \in \mathbb{F}[x]$ is computable by* ARO *of size s if for all possible permutation of variables there exists a ROABP of size at most s in that variable order.*

We will start with defining the *partial coefficient space* of a polynomial f to 'characterise' the width of ARO. We can work over any field $\mathbb{F}$.

Let $A(x)$ be a polynomial over $\mathbb{F}$ in n variables with individual degree d. Denote the set $M := \{0, \ldots, d\}^n$. Note that, one can write $A(x)$ as

$$A(x) = \sum_{a \in M} \text{coef}_A(x^a) \cdot x^a .$$

Consider a partition of the variables x into two parts y and z, with $|y| = k$. Then, $A(x)$ can be viewed as a polynomial in variables y, where the coefficients are polynomials in $\mathbb{F}[z]$. For monomial y^a, let us denote the coefficient of y^a in $A(x)$ by $A_{(y,a)} \in \mathbb{F}[z]$. The coefficient $A_{(y,a)}$ can also be expressed as a partial derivative $\partial A / \partial y^a$, evaluated at $y = \mathbf{0}$ (and multiplied by an appropriate constant), see [FS13a, Section 6]. Moreover, we can also write $A(x)$ as

$$A(x) = \sum_{a \in \{0,\ldots,d\}^k} A_{(y,a)} \cdot y^a .$$

One can also capture the space by the coefficient matrix (also

known as the partial derivative matrix) where the rows are indexed by monomials p_i from y, columns are indexed by monomials q_j from $z = x \setminus y$ and (i, j)-th entry of the matrix is $\mathrm{coef}_{p_i \cdot q_j}(f)$.

The following lemma formalises the connection between ARO width and dimension of the coefficient space (or the rank of the coefficient matrix).

Lemma 2.6.5 ([Nis91]) *Let $A(x)$ be a polynomial of individual degree d, computed by an* ARO *of width w. Let $k \leq n$ and y be any prefix of length k of x. Then*

$$\dim_{\mathbb{F}}\{A_{(y,a)} \mid a \in \{0, \ldots, d\}^k\} \leq w\,.$$

We remark that the original statement was for a fixed variable order. Since, ARO affords any-order, the above holds for any-order as well. The following lemma is the converse of the above lemma and shows us that the dimension of the coefficient space is rightly captured by the width.

Lemma 2.6.6 (Converse lemma [Nis91]) *Let $A(x)$ be a polynomial of individual degree d with $x = (x_1, \ldots, x_n)$, such that for some w, for any $1 \leq k \leq n$, and y, any-order-prefix of length k, we have*

$$\dim_{\mathbb{F}}\{A_{(y,a)} \mid a \in \{0, \ldots, d\}^k\} \leq w\,.$$

Then, there exists an ARO *of width w for $A(x)$.*

2.6.3 Properties of $\Sigma\wedge\Sigma$ and $\Sigma\wedge\Sigma\wedge$ circuits

The key ingredient for seeing a sum of power of 'nice' forms is to convert it to ARO, via the *duality trick*.

Lemma 2.6.7 (Duality trick [Sax08]) *The polynomial $f = (x_1 + \ldots + x_n)^d$ can be written as*

$$f = \sum_{i \in [t]} f_{i1}(x_1) \cdots f_{in}(x_n),$$

where $t = O(nd)$, and f_{ij} is a univariate polynomial of degree at most d.

We remark that the above proof works for fields of characteristic $= 0$, or $> d$.

Using a simple interpolation, the coefficient of y^e can be extracted from $f(x, y) \in \Sigma\wedge\Sigma$ again as a small $\Sigma\wedge\Sigma$ representation.

Lemma 2.6.8 ($\Sigma\wedge\Sigma$ coefficient extraction) *Let $f(x, y) \in \mathbb{F}[x][y]$ be computed by a $\Sigma\wedge\Sigma$ circuit of size s and degree d. Then, $\mathrm{coef}_{y^e}(f) \in \mathbb{F}[x]$ is a $\Sigma\wedge\Sigma$ circuit of size $O(sd)$, over $\mathbb{F}[x]$.*

Proof sketch. Let $f =: \sum_i \alpha_i \cdot \ell_i^{e_i}$, with $e_i \leq s$ and $\deg_y(f) \leq d$. Thus, write $f =: \sum_{i=0}^{d} f_i \cdot y^i$, where $f_i \in \mathbb{F}[x]$. Interpolate using $(d+1)$-many distinct points $y \mapsto \alpha \in \mathbb{F}$, and conclude that f_i has a $\Sigma\wedge\Sigma$ circuit of size $O(sd)$. □

Like coefficient extraction, differentiation of $\Sigma\wedge\Sigma$ circuit is easy too.

Lemma 2.6.9 ($\Sigma\wedge\Sigma$ differentiation) *Let $f(x, y) \in \mathbb{F}[x][y]$ be computed by a $\Sigma\wedge\Sigma$ circuit of size s and degree d. Then, $\partial_y(f)$ is a $\Sigma\wedge\Sigma$ circuit of size $O(sd^2)$, over $\mathbb{F}[x][y]$.*

Proof sketch. Lemma 2.6.8 shows that each f_e has $O(sd)$ size circuit where $f =: \sum_e f_e\, y^e$. Doing this for each $e \in [0, d]$ gives a blowup of $O(sd^2)$ and the representation: $\partial_y(f) = \sum_e f_e \cdot e \cdot y^{e-1}$. □

Remark 2.6.1 Same property holds for $\Sigma\wedge\Sigma\wedge$ circuits.

$\Sigma\wedge\Sigma\wedge$ can be shown to be closed under multiplication i.e., product of two polynomials, each computable by a $\Sigma\wedge\Sigma\wedge$ circuit, is computable by a single $\Sigma\wedge\Sigma\wedge$ circuit. Using Lemma 2.2.3, we prove the closure result.

Lemma 2.6.10 *Let $f_i(x, y) \in \mathbb{F}[y][x]$, of syntactic degree $\leq d_i$, be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s_i, for $i \in [k]$ (wrt x). Then, $f_1 \cdots f_k$ has $\Sigma\wedge\Sigma\wedge$ circuit of size $O((d_2+1)\cdots(d_k+1) \cdot s_1 \cdots s_k)$.*

Proof. Let $f_i = \sum_j f_{ij}^{e_{ij}}$; by assumption $e_{ij} \leq d_i$ (by assumption). Then using Lemma 2.2.3, $f_{1j_1}^{e_{1j_1}} \cdots f_{kj_k}^{e_{kj_k}}$ has size at most

$(d_2+1)\cdots(d_k+1)\cdot\left(\sum_{i\in[k]}\text{size}(f_{ij_i})\right)$, for indices $j_1,\ldots,j_k$. Summing up for all $s_1\cdots s_k$ many products (atmost) gives the upper bound. □

Remark 2.6.2 The same proof works for $\Sigma\wedge\Sigma$ circuits as well.

Lemma 2.6.11 *If $f=\sum_{j\in[s]}f_j^{e_j}$ is computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s, then f can also be computed by an ROABP (of any order) of size at most $O(sn^2D^2)=O(s^5)$.*

Proof. Let $g^e=(g_1(x_1)+\cdots+g_n(x_n))^e$, where $\deg(g_i)\cdot e\le D$. Using Lemma 2.6.7 we get $g^e=\sum_{i=1}^{O(ne)}h_{i1}(x_1)\cdots h_{in}(x_n)$, where each h_{ij} is of degree at most D.

We do this for each power (i.e. each summand of f) individually, to get the final sum-of-product-of-univariates; of top-fanin $O(sne)$ and individual degree at most D. This is an ARO of size $O(sne)\cdot n\cdot D\le O(sn^2D^2)$.

Finally, if $f=\sum_{j\in[s]}f_j^{e_j}$, then applying this to each individual $f_j^{e_j}$, and using the fact that $\sum_j\text{size}(f_j)\le s$, the conclusion follows. □

2.7 Hitting Sets

Definition 2.7.1 (Hitting set) *A set $\mathscr{H}\subseteq\mathbb{F}^n$ is called a Hitting Set for a class polynomial $\mathscr{C}\subseteq\mathbb{F}[x]$, if for all $g\in\mathscr{C}$*

$$g\neq 0 \iff \exists a\in\mathscr{H}\ :\ g(a)\neq 0.$$

In literature, PIT has a close association with hitting set, as the two notions are provably equivalent (refer to [For14, Lemma 3.2.9 and 3.2.10]). Note that the set $\mathscr{H}$ works for every polynomial of the class. Instead of a PIT algorithm, we will usually design such a set.

The simplest PIT algorithm for any circuit in general is due to Polynomial Identity Lemma [Ore22; DL78; Zip79; Sch80]. When the number of variables is small, say $O(1)$, then this algorithm is very efficient.

Lemma 2.7.1 (Trivial PIT) *For a class of n-variate, individual degree $< d$ polynomial $f \in \mathbb{F}[x]$ there exists a deterministic PIT algorithm which runs in time $O(d^n)$.*

Remark 2.7.1 It can be improved to the *optimal* size of $\binom{n+d}{d}$ [BDI21]. However, in this thesis, this (asymptotically) does not give *better* results, since most of the time n, or d would be *constant* or very small compared to one another, in our particular applications. Thus, we will use Lemma 2.7.1 throughout.

In fact, a technical tool to solve blackbox PITs is to construct an efficient hitting-set generator, which is a polynomial map that "preserves the nonzeroness".

Definition 2.7.2 (Hitting-set generator (HSG)) *A polynomial map $G : \mathbb{F}^k \to \mathbb{F}^n$ given by $G(z) = (g_1(z), g_2(z), \dots, g_n(z))$ is a* hitting-set generator (HSG) *for a class $\mathscr{C} \subseteq \mathbb{F}[x]$ of polynomials, if for every nonzero $f \in \mathscr{C}$, we have that $f \circ G = f(g_1, g_2, \dots, g_n)$ is nonzero.*

We say that G is 't time' HSG, *if* coef(g_i) *can be computed in time t and the maximum degree of g_i is $\leq t$.*

Given a HSG, one can construct a *hitting-set*, a set H such that a non-zero circuit is non-zero at some points in H. Crucial here is the size of H which depends on the parameters of the HSG. A t-time HSG G gives a $(td)^{O(k)}$ time blackbox PIT algorithm, for circuits that compute polynomials of degree $\leq d$, over popular fields like rationals $\mathbb{Q}$ or their extensions, local fields $\mathbb{Q}_p$ or their extensions, or finite fields $\mathbb{F}_q$. When k is constant, we get a poly-time blackbox PIT.

Very recently, Guo et al. [Guo+19] showed how to use the hardness of a *constant* variate explicit polynomial family to derandomize PIT. They need the algebraic circuit hardness to be more than d^3; which requires $k \geq 4$ for the family to exist.

Theorem 2.7.2 *[Guo+19] Let $P \in \mathbb{F}[x]$ be a k-variate polynomial of degree d such that* coef(P) *can be computed in* poly(d)*-time. If* size$(P) > s^{10k+2}\, d^3$*, then there is a* poly(s)*-time* HSG *for $\mathscr{C}(s, s, s)$.*

The following lemma is useful to construct hitting set for product of two circuit classes when the hitting set of individual circuit is known.

Lemma 2.7.3 *Let $\mathscr{H}_1, \mathscr{H}_2 \subseteq \mathbb{F}^n$ of size s_1 and s_2 respectively be the hitting set of the class of n-variate degree d polynomials computable by $\mathscr{C}_1$ and $\mathscr{C}_2$ respectively. Then, for the class of polynomials computable by $\mathscr{C}_1 \cdot \mathscr{C}_2$ there is an explicit hitting set $\mathscr{H}$ of size $s_1 \cdot s_2 \cdot O(d)$.*

Proof. Let $f = f_1 \cdot f_2 \in \mathscr{C}_1 \cdot \mathscr{C}_2$ such that $f_1 \in \mathscr{C}_1$ and $f_2 \in \mathscr{C}_2$. For each $a_i \in \mathscr{H}_1$, $b_j \in \mathscr{H}_2$ define a 'formal-sum' evaluation point (over $\mathbb{F}[t]$) $c := (c_\ell)_{1 \le \ell \le n}$ such that $c_\ell := a_{i\ell} + t \cdot b_{j\ell}$; where t is a formal variable. Collect these points, going over i, j, in a set H. It can be seen, by shifting and scaling, that non-zeroness is preserved: there exists $c \in H$ such that $0 \neq f(c) \in \mathbb{F}[t]$ and $\deg f(c) = O(d)$. Using trivial hitting set from Lemma 2.7.1 we obtain the final hitting set $\mathscr{H}$ of size $O(s_1 \cdot s_2 \cdot d)$. □

Remark 2.7.2 The above argument easily extends to circuit classes $(\mathscr{C}_1/\mathscr{C}_1) \cdot (\mathscr{C}_2/\mathscr{C}_2)$, which compute rationals of the form $(g_1/g_2) \cdot (h_1/h_2)$, where $g_i \in \mathscr{C}_1$ and $h_i \in \mathscr{C}_2$ $(g_2 h_2 \neq 0)$.

2.7.1 PIT for ΠΣΠ circuits

Sparse PIT is testing the identity of polynomials with bounded number of monomials. There have been a lot of work on sparse-PIT, interested readers can refer [BT88; KS01] and references therein. For the proof of poly-time hitting set of Sparse PIT see [Sax09, Thm. 2.1].

Theorem 2.7.4 (Sparse-PIT map [KS01]) *Let $p(x) \in \mathbb{F}[x]$, be an n-variate polynomial with individual degree at most d, and sparsity at most m. Then, there is a deterministic algorithm to test its identity which runs in time* poly(mnd).

Indeed, if identity of sparse polynomial can be tested efficiently, product of sparse polynomial can be tested efficiently. We formalize and prove this in the following, which would also trivially prove Theorem 2.7.4.

Theorem 2.7.5 ([Sap21, Lemma 2.4]) *For a class of n-variate polynomial $f \in \mathbb{F}[x]$ computable by $\Pi\Sigma\Pi$ of size s, there is an explicit hitting Set of size* $\mathrm{poly}(s)$.

Proof. We will show the following.

Lemma 2.7.6 *Let $f(x_1, \dots, x_n) = \prod_{j=1}^{r} f_i(x_1, \dots, x_n)$, where each f_i is an s-sparse polynomial with individual degree $< d$. Then, there exists $1 \leq m \leq \mathrm{poly}(rsn \log d)$, such that*

$$f(y, y^{d \bmod m}, \dots, y^{d^{n-1} \bmod m}) \neq 0 \,(\mathrm{mod}\ \ y^m - 1) \ .$$

If the above lemma is true, one can simply use the following hitting set for Theorem 2.7.5, since by the definition of size $r, d, n \leq s$:

$$H := \left\{(\alpha, \alpha^{d \bmod m}, \dots, \alpha^{d^{n-1} \bmod m}) \mid 1 \leq \alpha, m \leq \mathrm{poly}(s)\right\} .$$

Proof of Lemma 2.7.6. Let us first fix an $i \in [r]$, and a prime m. Let $g_i := f_i(y, y^d, \dots, y^{d^{n-1}}) \in \mathbb{F}[y]$. Note that, each monomial in f_i gets uniquely mapped to a monomial in g_i. Therefore, trivially, $g_i(y) \neq 0$.

Now, suppose y^a is a monomial in g_i (we will write $y^a \in \mathrm{supp}(g_i)$. If $g_i(y) = 0 \pmod{y^m - 1}$, then there must be another monomial y^b in g_i, such that $y^b \equiv y^a \,(\mathrm{mod}\ \ y^m - 1)$. This is possible only when $m \mid (b - a)$. Let us call a prime m 'bad' if $g_i(y) = 0 \pmod{y^m - 1}$, for some $i \in [r]$. To avoid such a bad prime, it suffices the following to be satisfied:

$$m \nmid \prod_{i=1}^{r} \prod_{y^a, y^b \in \mathrm{supp}(g_i), b \neq a} (b - a) \ =: \ R \ .$$

This integer R can be at most $(d^n)^{s^2 \cdot r}$; this is because $|b - a| < d^n$, and each g_i is s-sparse implying there can be at most s^2 many pairs of different (a, b), and finally there are r many polynomials g_i. Since, R has at most $\log R$ many prime factors, and there are $\log R + 1$ many primes in the range $[1, (\log R)^2]$, we will find a good prime m within $(\log R)^2 = \mathrm{poly}(srn \log d)$. This finishes the proof. □

□

There have been quite a few results on blackbox PIT for ROABPs as well [FS13a; FSS14; GKS17]. The current best known algorithm works in quasipolynomial time.

Theorem 2.7.7 (Theorem 4.9 [GKS17]) *For an n-variate polynomial of individual-degree d, which is computable by width-w ROABPs in any order, a $(ndw)^{O(\log\log w)}$ time hitting set can be constructed.*

Known PIT for Depth-4 Circuits

Recall that a polynomial $f(x) \in \mathbb{F}[x]$ is computable by a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit if $f(x) = \sum_{i\in[s]} f_i(x)^{e_i}$ where $\deg f_i \leq \delta$. The first nontrivial PIT algorithm for this model was designed in [For15].

Theorem 2.7.8 (Proposition 4.18 [For15]) *For the class of n-variate, degree-$(\leq d)$ polynomials $f(x)$, computed by $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits of size s, there is a $\mathrm{poly}(n, d, \delta\log s)$-explicit hitting set of size $(nd)^{O(\delta\log s)}$*

Similarly, $\Sigma\wedge\Sigma\wedge$ circuits compute polynomials of the form $f(x) = \sum_{i\in[s]} f_i^{e_i}$ where f_i is a sum of univariate polynomials. Using duality trick [Sax08] and PIT results from [RS05; GKS17], one can design efficient PIT algorithm for $\Sigma\wedge\Sigma\wedge$ circuits.

Lemma 2.7.9 (PIT for $\Sigma\wedge\Sigma\wedge$ circuits) *Let $P \in \Sigma\wedge\Sigma\wedge$ of size s. Then, there exists an $s^{O(\log\log s)}$) time blackbox PIT for the same.*

Proof sketch. If $f = \sum_{j\in[s]} f_j^{e_j}$ is computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s, then clearly, f can also be computed by an ROABP (of any order) of size at most $O(s^5)$ Lemma 2.6.11. So, the blackbox PIT follows from Theorem Theorem 2.7.7. □

2.8 Jacobian and Algebraic Dependence

We will give the proof of Theorem 4.2.1 in Chapter 4. Before the details, we will state a few important definitions and

lemmas from [Agr+16] to be referenced later.

Definition 2.8.1 *The Jacobian of a set of polynomials* $\mathbf{f} = \{f_1, \ldots, f_m\}$ *in* $\mathbb{F}[x]$ *is defined to be the matrix*

$$\mathcal{J}_x(\mathbf{f}) \ := \ \left(\partial_{x_j}(f_i)\right)_{m \times n} .$$

Notation. Let $S \subseteq x = \{x_1, \ldots, x_n\}$ and $|S| = m$. Then, polynomial $J_S(\mathbf{f})$ denotes the minor (i.e. determinant of the submatrix) of $\mathcal{J}_x(\mathbf{f})$, formed by the columns corresponding to the variables in S.

Definition 2.8.2 (Transcendence Degree) *Polynomials* T_i, *for* $i \in [m]$, *are called* algebraically dependent *if there exists a nonzero* annihilator A *s.t.* $A(T_1, \ldots, T_m) = 0$. Transcendence degree *is the size of the largest subset* $S \subseteq \{T_1, \ldots, T_m\}$ *that is algebraically independent. Then* S *is called a* transcendence basis.

The next definition we need is that of a faithful homomorphism.

Definition 2.8.3 (Faithful homomorphism) *A homomorphism* $\Phi : \mathbb{F}[x] \to \mathbb{F}[y]$ *is faithful for* $\mathbf{T}$ *if* $\mathrm{trdeg}_{\mathbb{F}}(\mathbf{T}) = \mathrm{trdeg}_{\mathbb{F}}(\Phi(\mathbf{T}))$.

The reason for interest in faithful maps is due its usefullness in preserve the identity as shown in the following theorem.

Theorem 2.8.1 (Theorem 2.4 [Agr+16]) *Let* $C \in \mathbb{F}[y_1, \ldots, y_m]$. *Then,* $C(\mathbf{T}) = 0 \iff C(\Phi(\mathbf{T})) = 0$.

It would be useful to point out why Theorem 2.8.1 is *surprising*! Let us say $\{f_1, \ldots, f_r\}$ is one maximal algebraically independent, and Φ is a map that ensures that $\{\Phi(f_1), \ldots, \Phi(f_r)\}$ continue to remain algebraically independent. However, the circuit C could involve only $f_{s+1}, \ldots, f_m$, and it is *far from obvious* why just this suffices to preserve relations between them. But the fact that $\{f_1, \ldots, f_s\}$ is a maximal algebraically independent set forces all relations amongst the polynomials f_i to be preserved exactly by Φ.

Proof. $\Longrightarrow$ is 'obvious'. To show the other side, since Φ is faithful to $\mathbf{f}$, there is a transcendence basis (say, $f_1, \ldots, f_s$) of $\mathbf{f}$ such that $\Phi(f_1), \ldots, \Phi(f_s)$ is a transcendence basis of $\Phi(\mathbf{f})$. The function field $K = \mathbb{F}(\mathbf{f})$ 'essentially' consists of elements that are polynomials in $f_{s+1}, \ldots, f_m$, with coefficients from $\mathbb{F}(f_1, \ldots, f_s)$. To see this, since $\{f_1, \ldots, f_s\}$ are algebraically independent, the field $\mathbb{F}(f_1, \ldots, f_s)$ is isomorphic to $\mathbb{F}(y_1, \ldots, y_s)$ for some new variables $y_1, \ldots, y_s$. Further, since every other f_i is algebraically dependent on $\{f_1, \ldots, f_s\}$, it is also algebraic

over $\mathbb{F}(f_1, \dots, f_s)$. Hence,

$$\begin{aligned}\mathbb{F}(f_1, \dots, f_m) &\equiv (\mathbb{F}(f_1, \dots, f_s))(f_{s+1}, \dots, f_m) \\ &\equiv \mathbb{F}(f_1, \dots, f_s)[f_{s+1}, \dots, f_m] \ .\end{aligned}$$

Now, suppose $C(\mathbf{f})$ is a nonzero element of K, then there is a unique inverse $Q \in K$ such that $Q \cdot C(\mathbf{f}) = 1$. Since Q is a polynomial in $f_{s+1}, \dots, f_m$ with coefficients from $\mathbb{F}(f_1, \dots, f_s)$, by clearing off the denominators of these coefficients in Q, we get an equation that $\tilde{Q} \cdot C = P(f_1, \dots, f_s)$, where Q is a nonzero polynomial in $\mathbf{f}$ and P is a *nonzero* polynomial in $f_1, \dots, f_s$. Applying Φ to both sides of the equation, we conclude that

$$C(\Phi(\mathbf{f})) \ = \ \Phi(C(\mathbf{f})) \ \neq \ 0 \, ,$$

otherwise,

$$P(\Phi(f_1), \dots, \Phi(f_s)) \ = \ \Phi(P(f_1, \dots, f_s)) \ = \ 0 \, ,$$

which is a contradiction as $\Phi(f_1), \dots, \Phi(f_s)$ are algebraically independent and P is a nonzero nontrivial polynomial! This finishes the proof.

□

Here is an important criterion about the Jacobian matrix which basically shows that it *preserves* algebraic independence. For a proof, see [BMS13].

Theorem 2.8.2 (Jacobian criterion) *Let* $\mathbf{f} \subset \mathbb{F}[x]$ *be a finite set of polynomials of degree at most d, and* $\text{trdeg}_{\mathbb{F}}(\mathbf{f}) \leq r$. *If* $char(\mathbb{F}) = 0$, *or* $char(\mathbb{F}) > d^r$, *then* $\text{trdeg}_{\mathbb{F}}(\mathbf{f}) = \text{rank}_{\mathbb{F}(x)}\mathcal{J}_x(\mathbf{f})$.

Jacobian criterion together with faithful maps give a recipe to design a map which drastically reduces number of variables, if trdeg is small; for a proof see [Agr+16].

Lemma 2.8.3 (Lemma 2.7 [Agr+16]) *Let* $\mathbf{T} \in \mathbb{F}[x]$ *be be a finite set of polynomials of degree at most d and* $\text{trdeg}_{\mathbb{F}}(\mathbf{T}) \leq r$, *and char(F)=0 or* $> d^r$. *Let* $\Psi' : \mathbb{F}[x] \longrightarrow \mathbb{F}[z_1]$ *such that* $\text{rank}_{\mathbb{F}(x)}\mathcal{J}_x(\mathbf{T}) = \text{rank}_{\mathbb{F}(z_1)}\Psi'(\mathcal{J}_x(\mathbf{T}))$.

Then, the map $\Phi : \mathbb{F}[x] \longrightarrow \mathbb{F}[z_1, t, y]$, *such that* $x_i \mapsto (\sum_j y_j t^{ij}) + \Psi'(x_i)$, *is a faithful homomorphism for* $\mathbf{T}$.

❑ An important Jacobian identity

We have the following identity [Agr+16, Equation 3.1], where $T_i = \prod_j g_{ij}$, and $L(T_i)$ the *multiset* of polynomials that constitute T_i:

Lemma 2.8.4 *For $\mathbf{T}_k = (T_1, \dots, T_k)$, we have*

$$J_{x_k}(\mathbf{T}_k) = \sum_{g_1 \in L(T_1), \dots, g_k \in L(T_k)} \left(\frac{T_1 \dots T_k}{g_1 \cdots g_k}\right) \cdot J_{x_k}(g_1, \dots, g_k) \,.$$

Proof. The proof essentially exploits linearity of determinants. Here is a simple fact.

Fact

For any set of vectors $\mathbf{v_{ij}} \in \mathbb{F}^n$, for $i \in [n]$, and $j \in [k]$,

$$\det\left[\sum_{j=1}^{k} \mathbf{v_{1j}}, \dots, \sum_{j=1}^{k} \mathbf{v_{nj}}\right] = \sum_{1 \le j_1 \le \dots \le j_n \le k} \det\left[\mathbf{v_{1j_1}}, \dots, \mathbf{v_{nj_n}}\right] \,.$$

Now, if $T_i = \prod_{j \in [d]} g_{ij}$, then

$$\mathrm{dlog}_x(T_i) = \sum_{j \in [d]} \mathrm{dlog}_x\left(g_{ij}\right) \implies \partial_x(T_i) = T_i \cdot \left(\sum_{j \in [d]} \frac{\partial_x g_{ij}}{g_{ij}}\right) \,.$$

Using this with the linearity fact as above, $J_{x_k}(\mathbf{T}_k)$ takes the following form:

$$\begin{aligned} J_{x_k}(\mathbf{T}_k) &= \sum_{g_1 \in L(T_1), \dots, g_k \in L(T_k)} \frac{T_1 \cdots T_k}{g_1 \cdots g_k} \cdot \det \begin{bmatrix} \partial_{x_1} g_1 & \cdots & \partial_{x_k} g_1 \\ \vdots & \cdots & \vdots \\ \partial_{x_1} g_k & \cdots & \partial_{x_k} g_k \end{bmatrix} \\ &= \sum_{g_1 \in L(T_1), \dots, g_k \in L(T_k)} \frac{T_1 \cdots T_k}{g_1 \cdots g_k} \cdot J_{x_k}(g_1, \dots, g_k) \,. \end{aligned}$$

□

2.9 Limitations of Bounded Fan-in Depth-3 circuits

The following theorem is a folkore [Kum20], which shows that bounded depth-3 circuits are *not universal*. We present a

proof here.

Theorem 2.9.1 *The inner product polynomial*

$$\mathsf{IP}_n = \sum_{i=1}^{n} x_i \cdot y_i ,$$

cannot be computed by any $\Sigma^{[n-1]}\Pi\Sigma$ *circuit, no matter how large the product fan-in is.*

Proof. Assume, for the sake of contradiction, IP_n can be computed by a $\Sigma\Pi\Sigma$ circuit of top fan-in $n-1$:

$$\mathsf{IP}_n = \sum_{i=1}^{n-1} \prod_{j=1}^{d_i} \ell_{ij} . \tag{2.7}$$

The proof idea is as follows: we reduce Equation 2.7 modulo $n-1$ many linear polynomials suitably picked from *each* summand of the RHS. We get a contradiction, as the RHS becomes zero modulo those linear polynomials, but the LHS remains nonzero. Formally, we implement the proof below.

Assume, wlog, $\prod_{j=1}^{d_1} \ell_{1j}$ contains the variable x_1 in one of its factors. For some j, wlog we have $\ell_{1j} = x_1 - r_1(x_2, \ldots, x_n, y)$. If $\ell_{1j} = a_1 x_1 - r_1(x_2, \ldots, x_n, y)$, we can take out a_1, and work with $(x_1 - r_1/a_1)$. Taking mod by $(x_1 - r_1/a_1)$ essentially means substituting $x_1 = r_1/a_1$, in Equation 2.7.

Now,we go modulo $x_1 - r_1$ in both sides of Equation 2.7. This changes the IP_n polynomial in the LHS, which becomes $r_1 y_1 + x_2 y_2 + \cdots + x_n y_n$. In the RHS, the first summand vanishes! So, we get

$$r_1 y_1 + x_2 y_2 + \cdots + x_n y_n = \sum_{i=2}^{n-1} \prod_{j=1}^{d_i} \ell_{ij}(r_1, x_2, \ldots, x_n, y) . \tag{2.8}$$

Now, note that $r_1 y_1 + x_2 y_2 + \cdots + x_n y_n$ is *not free* of x_2, as $r_1 y_1$ cannot cancel the term $x_2 y_2$. Thus, x_2 must be present also in the RHS of Equation 2.8. Wlog, assume that x_2 is present in $\ell_{2j}(r_1, x_2, \ldots, x_n, y)$, for some j. Assume it is of the form $x_2 - r_2(x_3, \ldots, x_n, y)$.

Now, we reduce Equation 2.8 modulo $x_2 - r_2$. This changes the LHS to

$$r_1(r_2, \ldots, x_n, y) y_1 + r_2 y_2 + x_3 y_3 + x_n y_n .$$

At least one term in the RHS gets vanished. The resulting polynomial in the LHS is not free of x_3. Thus, x_3 must be present in the RHS too. We assume there is a linear term in RHS of the form $(x_3 - r_3)$. Next, we go modulo $(x_3 - r_3)$.

We can continue this, $n-1$ times, and in the end RHS would completely vanish. In the LHS, we would have $\tilde{r}_1 y_1 + \tilde{r}_2 y_2 + \cdots + \tilde{r}_{n-1} y_{n-1} + x_n y_n$, for some linear polynomials $\tilde{r}_1, \ldots, \tilde{r}_{n-1}$. This polynomial would have a nonzero term $x_n y_n$, as $x_n y_n$ *cannot* be cancelled by $\tilde{r}_1 y_1 + r2y2 + \cdots + \tilde{r}_{n-1} y_{n-1}$. This leads to a contradiction.

□

Results in Algebraic Complexity

A τ-Conjecture for sum-of-squares and its consequences

3

"An unexamined life is not worth living".

— Socrates, Trial of Socrates.

Abstract. For a univariate polynomial f, a *sum-of-squares representation (SOS)* has the form $f = \sum_{i\in[s]} c_i f_i^2$, where c_i are field elements and the f_i are polynomials. The size of the representation is the number of monomials that appear across the polynomials f_i. We denote its minimum as the *support-sum* $S(f)$ of f.

For a polynomial f of degree d of full support, a trivial lower bound for the support-sum is $S(f) \geq \sqrt{d}$. We show that the existence of an explicit polynomial f with support-sum *slightly larger* than the lower bound implies that VP $\neq$ VNP. We also consider the *sum-of-cubes representation (SOC)* of polynomials. In a similar way, we show that an explicit hard polynomial implies that blackbox PIT is in P.

On the other hand, the famous Shub-Smale τ-conjecture [SS95] is a conjecture in algebraic complexity, asserting that a univariate polynomial which is computable by a small algebraic circuit, has a small number of integer roots. This has strong implications in complexity theory. In this work, we conjecture that the number of real roots of a univariate polynomial can be at most a constant multiple of the support-sum. We connect this conjecture with two central open questions in algebraic complexity – matrix rigidity and VP vs. VNP.

Furthermore, a (stronger) conjecture for sum-of-cubes (SOC) implies that blackbox PIT is in P. This is the *first* time a τ-conjecture has been shown to give a polynomial-time PIT. We also establish some special cases of this conjecture, and prove tight lower bounds for restricted depth-2 models.

This chapter is based on two closely related works –1) the article titled *A Largish Sum-of-Squares Implies Circuit Hardness and Derandomization*, which is a joint work with Nitin Saxena and Thomas Thierauf, that appeared in ITCS, 2021 [DST21], and 2) the article titled *Real τ-Conjecture for sum-of-squares: A unified approach to lower bound and derandomization*, which is a single author paper that appeared in CSR, 2021 [Dut21].
Some statements/results may differ from those written in the original articles. However, because the themes of both the works are similar, we compiled and wrote it in a single chapter.

3.1 Set-up and Our Results

The sum-of-squares representation (SOS) is one of the most fundamental in number theory and algebra. Lagrange's four-squares theorem has inspired generations of mathematicians [Ram17]. Hilbert's *17th problem* asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions [Pfi76]. In engineering, SOS has found many applications in approximation, optimization and control theory, see [Rez78; Las07; Lau09; BM22].

The famous Shub-Smale τ-conjecture [SS95] is a conjecture in algebraic complexity, asserting that a univariate polynomial which is computable by a small algebraic circuit has 'a few' integer roots. It was established in [SS95] that the τ-conjecture implies $\mathsf{P}_{\mathbb{C}} \neq \mathsf{NP}_{\mathbb{C}}$, for the Blum–Shub–Smale (BSS) model of computation over the complex numbers [BSS89; Blu+00]. Bürgisser [Bür09] obtained a similar result for VP vs. VNP.

Can real roots help? One possible disadvantage of the τ-conjecture is the reference of *integer roots*. As a natural approach to the τ-conjecture, one can try to bound the number of *real roots* instead of integer roots. However, a mere replacement of "integer roots" by "real roots" fails miserably as the number of real roots of a univariate polynomial can be *exponential* in its circuit size; for e.g., the Chebyshev polynomials [Sma98][1].

Interestingly, Koiran [Koi11] came up with the following τ-conjecture for the *restricted* (depth-4) circuits.

If $f \in \mathbb{R}[x]$ is a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}$, where each $f_{ij} \in \mathbb{R}[x]$ is t-sparse [2], then the number of distinct *real roots of f can at most be* $\mathsf{poly}(kmt)$.

Note that, the conjecture is true for $m = 1$, by Descartes' rule of signs (Lemma 2.2.5). Using the celebrated depth-4 reduction [AV08; Koi12], it was established that real τ-conjecture with $m = \omega(1)$, yields a strong separation in the *constant-free* setting, i.e. $\mathsf{VP}_0 \neq \mathsf{VNP}_0$ [3]. Later, it was shown to imply $\mathsf{VP} \neq \mathsf{VNP}$, see [Tav14; GKT15].

1: We already discussed in section 2.3 that the d-th chebyshev polynomial T_d can be computed by an $O(\log d)$ size circuit. But, it has d-many *real* roots in the interval $[-1, 1]$.

2: $f := \sum_{i=0}^{n} a_i x^i$ is t-sparse if at most t of the coefficients $a_0, \ldots, a_n$ are non-zero.

3: In the constant-free setting, you are not allowed to use arbitrary constant from the field. Instead, you have to compute it as a circuit starting from ± 1 (which is cost-free), and the size of the constants contribute to the original size as well. For e.g., $(2^{2^{2^n}} x_1 \cdots x_n)_n$ requires $\Omega(2^n)$ size VP_0 circuits while it has trivial $O(n)$ size VP circuits.

An interesting anecdote

Consider a computational model where multiplication is *free*, and we only count the number of additions to compute a certain polynomial. It is also called the *Additive Complexity* of a polynomial.

Suppose, $f \in \mathbb{R}[x]$ has additive complexity at most k, i.e., it can be computed by k many number of additions. Can we *bound* the number of *real* roots of f, in terms of k?

Theorem [Borodin-Cook'76 [BC76]]. Let $f \in \mathbb{R}[x]$ is computable by k additions. Then, there is an explicit (astronomical) function ϕ such that f has at most $\phi(k)$ many real roots.

In [Ris85], it was further established that one can take $\phi(k) = 2^{(4k)^2}$. These results lead to concretize the modern Shub-Smale τ-conjecture and later Koiran's real τ-conjecture.

3.1.1 Sum-of-squares model (SOS)

We give some background on sum-of-square representation, give some examples, and define our hardness condition. We first define the model and a complexity measure.

Definition 3.1.1 (SOS and support-sum size $S_R(f)$) *Let R be a ring. An n-variate polynomial $f(x) \in R[x]$ is represented as a* (weighted) sum-of-squares (SOS), *if*

$$f = \sum_{i=1}^{s} c_i {f_i}^2 , \tag{3.1}$$

for some top-fanin *s, where $f_i(x) \in R[x]$ and $c_i \in R$.*

The size *of the representation of f in Equation 3.1 is the* support-sum, *the sum of the support size (or sparsity) of the polynomials f_i. The* support-sum size *of f, is defined as the minimum support-sum of f, denoted by $S_R(f)$, or simply $S(f)$, when the ring R is clear from the context.*

Remark 3.1.1 In real analysis, the SOS representation of a polynomial is defined without the coefficients c_i, that is, only for non-negative polynomials f. In these terms, what

we define in Equation 3.1 is a *weighted* SOS. However, we will skip the term "weighted" in the following.

If we consider the expression in Equation 3.1 as a $\Sigma \wedge^{[2]} \Sigma\Pi$-formula, then the support-sum is the number of Π-operations directly above the input level.

For any N-variate polynomial f, let $\mathrm{sp}(f)$ denote the sparsity of f. For any field $R = \mathbb{F}$ of characteristic $\neq 2$, we have

$$\mathrm{sp}(f)^{1/2} \;\leq\; S_{\mathbb{F}}(f) \;\leq\; 2\,\mathrm{sp}(f) + 2\,. \tag{3.2}$$

The lower bound can be shown by counting monomials. The upper bound is because

$$f \;=\; (f+1)^2/4 - (f-1)^2/4\,. \tag{3.3}$$

In particular, when f is univariate and has full sparsity, $\mathrm{sp}(f) = d+1$, we get

$$\sqrt{d+1} \;\leq\; S(f) \;\leq\; 2d + 2\,. \tag{3.4}$$

By Equation 3.3, the SOS-model is *complete* for any field of characteristic $\neq 2$. It can be argued by a geometric-dimension argument that for most N-variate (constant $N \geq 1$) polynomials f of degree d, we have $S_{\mathbb{F}}(f(x)) = \Theta(d^N)$, as for random f, we know that $\mathrm{sp}(f) = \Theta(d^N)$. Note that this matches the upper bound given in Equation 3.4 for univariate f.

We give two examples.

Example 3.1.1 Let $f(x) = \sum_{k=0}^{d-1} x^k$. Note that

$$\sum_{k=0}^{d-1} x^k \;=\; \left(\sum_{k=0}^{\sqrt{d}-1} x^k\right)\left(\sum_{k=0}^{\sqrt{d}-1} x^{k\sqrt{d}}\right).$$

Hence, we have a representation of f as $f = gh$, where $\mathrm{sp}(g)$,$\mathrm{sp}(h) \leq \sqrt{d}$. Such a product can be written as a SOS,

$$gh = \frac{(g+h)^2}{4} - \frac{(g-h)^2}{4} \tag{3.5}$$

Because $\mathrm{sp}(g \pm h) \leq 2\sqrt{d}$ we get that $S(f) \leq 4\sqrt{d}$.

Observe that $S(f)$ essentially hits the lower bound in Equation 3.4, except for a constant factor.

Example 3.1.2 Let $f(x) = (x+1)^d$. This has a trivial SOS-representation with one summand: $(x+1)^d = \left((x+1)^{d/2}\right)^2$, for even d. So we get $S(f) \leq d/2 + 1$.

Note that this bound meets the upper bound in Equation 3.4, except for a constant factor. We **conjecture** that it is *optimal*, i.e. that $S(f) = \Omega(d)$. This is somewhat in contrast to the fact that f has small circuits. By repeated squaring, the circuit size of f_d is $O(\log d)$. For more on SOS-complexity versus circuit complexity, see subsection 3.1.3.

We call a polynomial family SOS-hard, if its support-sum is just *slightly* larger than the trivial lower bound from Equation 3.4. For our results, it actually suffices to consider *univariate* polynomials.

Definition 3.1.2 (SOS-hardness) *We call a polynomial family* $(f_d(x))_d$, SOS-hard with hardness ε, *if* $S(f_d) = \Omega(d^{0.5+\varepsilon})$.

SOS is provably the *simplest* model to show lower bound than higher powers. For detailed argument, we refer to section 3.5.

Main results (assuming SOS-hardness). Our main results with respect to SOS-representation show that the existence of explicit SOS-hard families of polynomials imply circuit lower bounds. The precise bounds depend on the size of ε:

1. For $\varepsilon = \omega(1/\sqrt{\log d})$, we show that the permanent cannot be computed by small ABPs, i.e., VBP $\neq$ VNP (Corollary 3.3.3).
2. For $\varepsilon = \omega(\sqrt{\log\log d / \log d})$ we show that the permanent cannot be computed by small circuits, i.e., VP $\neq$ VNP (Theorem 3.3.2).
3. For $\varepsilon > 0$ constant, we show that the permanent requires exponential size circuits, i.e., we have an exponential separation of VP and VNP (Theorem 3.3.6).

The technical foundation for these results are SOS decompositions for circuits (Lemma 3.3.1 and 3.3.5) that are based on the known depth-reductions techniques. We show how to express a polynomial $p(x)$ of degree d, given by a circuit of size s, as a sum of squares

- of $\texttt{quasi-poly}(d, s)$ many polynomials, each of degree at most $d/2$, in case of Lemma 3.3.1, and

- of poly(s) many polynomials, each of degree 'close' to $d/2$, in case of Lemma 3.3.5.

Hence, by our results, the major challenge in arithmetic complexity, to separate VP from VNP, can be solved by exhibiting an explicit univariate polynomial family $f_d(x) \in \mathbb{C}[x]$ of degree d with SOS-hardness parameter ε, just slightly above the general lower bound, even for vanishing small $\varepsilon = \varepsilon(d)$.

This would also have consequences for PIT, because Kabanets and Impagliazzo [KI04, Theorem 7.7] showed that VP $\neq$ VNP implies blackbox PIT $\in$ SUBEXP. In the case of constant ε, where we have an exponential separation of VP and VNP, we get blackbox PIT $\in$ QP (*quasi*polynomial-time).

❑ A τ-conjecture for SOS

The following question was originally raised by Arkadev Chattopadhyay, as the simplest case of Koiran's τ-conjecture.

Conjecture 3.1.1 (Simplest τ-conjecture) If the polynomials $f(x)$ and $g(x)$ have sparsity at most s, then the number of real roots of the polynomial $h := fg + 1$, can be at most $O(s)$.

Interestingly, fg can have at most $O(s)$ many real roots, which follows from the fact that the real roots of fg can be either of f or g and the Descartes' rule of signs gives bound on the number of real roots (Lemma 2.2.5).

In general, we are interested in the number of real roots of f, in terms of $S(f)$. Since the sparsity of f can be at most $S(f)^2$, f can have at most $S(f)^2$-many real roots by Descartes' rule of signs (Lemma 2.2.5). Further, it can be shown that a *random* polynomial f can have at most $O(S(f))$-many real roots, similar to [BB20, Theorem 1.1 with $k = 2$]. Motivated thus, we conjecture the following.

Conjecture 3.1.2 (SOS-τ-Conjecture) [4] Consider any nonzero polynomial $f(x) \in \mathbb{R}[x]$. Then, there exists a positive constant $c > 0$ such that the number of distinct real roots of f is at most $c \cdot S_{\mathbb{R}}(f)$.

4: An explicit d-degree polynomial f_d, with d-many real roots implies $S(f_d) \geq \Omega(d)$ (optimal $\epsilon = 1$). Thus, SOS-τ-conjecture is a much more *stronger* postulation!

Remarks. 1. One can show that Conjecture 3.1.2 implies $S_{\mathbb{C}}((x+1)^d) \geq \Omega(d)$; see Lemma 3.9.1. The proof is identical to [Hru13; Hru20], where strong distribution property of complex roots with multiplicities were shown to be implied, from the real τ-conjecture. We present it for

2. In Equation 3.1, we could restrict the degrees of f_i to be $O(d \log d)$. This might help us proving the conjecture; for details, see subsection 3.3.4 (& remark).

3. The $fg + 1$ case happens to be a special case of this new conjecture for 3 squares. For the two squares,i.e. any $f \in \mathbb{R}[x]$ of the form $c_1 f_1^2 + c_2 f_2^2$, can have at most $O(|f_1|_0 + |f_2|_0)$-many real roots; for details see Theorem 3.8.1 in subsection 3.8.1.

Main results (assuming Conjecture 3.1.2). Our main results with respect to SOS-τ-conjecture show that the conjecture implies *strong* circuit lower bounds.

1. We show that the conjecture implies "very" explicit family of real rigid matrices.
2. We show that the conjecture implies that the permanent requires exponential size circuits, i.e., we have an exponential separation between VP and VNP.

3.1.2 Sum-of-cubes model (SOC)

It is not clear whether a strong lower bound in the SOS-model can give a polynomial-time blackbox PIT. However, a different complexity measure on the sum-of-cube representation of polynomials indeed leads to a complete derandomization of blackbox PIT. So, we start by defining the model and give some background on it.

Definition 3.1.3 (SOC and support-union size $U_R(f, s)$) *Let R be a ring. An n-variate polynomial $f(x) \in R[x]$ is represented as a* sum-of-cubes (SOC)*, if*

$$f = \sum_{i=1}^{s} c_i {f_i}^3 , \tag{3.6}$$

for some top-fanin s, where $f_i(x) \in R[x]$ and $c_i \in R$.

The size *of the representation of f in Equation 3.6 is the size of the* support-union*, namely the number of distinct monomials in the representation, $\left|\bigcup_{i=1}^{s} \text{supp}(f_i)\right|$, where* support $\text{supp}(f_i)$ *denotes the set of monomials with a nonzero coefficient in $f_i(x)$. The* support-union size *of f with respect to s, denoted $U_R(f, s)$, is defined as the minimum support-union size when f is written as in Equation 3.6.*

If we consider the expression in Equation 3.6 as a $\Sigma \wedge^{[3]} \Sigma\Pi$ circuit, then the support-union size is the number of *distinct* Π-operations directly above the input level.

The two measures– support-union and support-sum –are largely *incomparable*, since $U(\cdot)$ has the extra argument s. Still one can show: $S_{\mathbb{F}}(f) \geq \min_s (U_{\mathbb{F}}(f, 4s) - 1)$ (Lemma 3.6.8).

For any polynomial f of sparsity $\mathrm{sp}(f)$, we have

$$\mathrm{sp}(f)^{1/3} \;\leq\; U_{\mathbb{F}}(f,s) \;\leq\; \mathrm{sp}(f) + 1, \tag{3.7}$$

where the upper bound is for $s \geq 3$ and for fields $R = \mathbb{F}$ of characteristic $\neq 2, 3$. The lower bound can be shown by counting monomials. The upper bound is because

$$f \;=\; (f+2)^3/24 + (f-2)^3/24 - f^3/12\,. \tag{3.8}$$

Hence, the SOC-model is *complete* for any field of characteristic $\neq 2, 3$.

In particular, when f is univariate and has full sparsity, $\mathrm{sp}(f) = d + 1$, we get

$$(d+1)^{1/3} \;\leq\; U_{\mathbb{F}}(f,s) \;\leq\; d+1\,. \tag{3.9}$$

More bounds and examples for the trade-off between s and the measure $U(f,s)$ can be found in section 3.5. Here, we summarize:

Example 3.1.3 1. For small $s = \Theta(d^{1/2})$, we have

$$U(f,s) \;= O(d^{1/2})\,, \qquad (\textit{Corollary } 3.6.5)\,.$$

2. For large $s = \Omega(d^{2/3})$, we have

$$U(f,s) = \Theta(d^{1/3})\,, \quad (\textit{Theorem } 3.6.7)\,.$$

However, it is unclear whether it is possible to have a very small fanin s, like $s = o(\sqrt{d})$, and at the same time a support-union of $o(d)$. This motivated us to define the hardness of univariate polynomials in the SOC-model as follows.

Definition 3.1.4 (SOC-hardness) *We call a polynomial family* $(f_d(x))_d$, SOC-hard, *if there is a constant* $0 < \varepsilon < 1/2$ *such that* $U_{\mathbb{F}}(f_d, d^{\varepsilon}) = \Omega(d)$.

Main results (assuming SOC-hardness). Our main result with respect to SOC-representation shows that the existence of an explicit SOC-hard family of polynomials leads to a *complete* derandomization of blackbox PIT (Theorem 3.4.2). From this, also the separation of VP and VNP follows.

The technical basis for our result is again a decomposition lemma (Lemma 3.4.1), an extension of Lemma 3.3.5. It shows how to express a polynomial $p(x)$ of degree d, given by a circuit of size s, as a sum of cubes of poly(s) many polynomials, each of degree close to $d/3$.

A τ-conjecture for SOC. With the aim of coming up with a tenable approach to connect PIT with the number of real roots, we conjecture the following.

Conjecture 3.1.3 (SOC-τ-conjecture) Consider any non-zero polynomial $f \in \mathbb{R}[x]$. Then, there exist positive constants $\varepsilon < 1/2$, and c such that the number of distinct real roots of f is at most $c \cdot U_{\mathbb{R}}(f, d^{\varepsilon})$.

Remark. We show that $f = c_1 f_1^3 + c_2 f_2^3$, has at most $O(\text{supp}(f_1) \cup \text{supp}(f_2))$-many real roots (see Theorem 3.8.2).

3.1.3 Hard polynomial candidates

Some interesting *explicit* candidates for the SOS/SOC-hard families are:

$$f_d(x) \in \left\{ \prod_{i=1}^{d} (x-i), (x+1)^d, T_d(x), \sum_{i=0}^{d} 2^{i^2} \cdot x^i, \sum_{i=0}^{d} 2^{2i(d-i)} \cdot x^i \right\}.$$

$T_d(x)$ is the *Chebyshev* polynomial that writes $\cos d\theta$ as a function of $\cos\theta$, i.e. $T_d(\cos\theta) := \cos d\theta$.

SOS-complexity vs. circuit complexity

- The polynomials: $\prod_{i=1}^{d}(x-i), \sum_{i=0}^{d} 2^{i^2} \cdot x^i, \sum_{i=0}^{d} 2^{2i(d-i)} \cdot x^i$, are all *conjectured* to be *hard* in the general circuit model; i.e., they require $\Omega(d)$ size circuits. However, both $(x+1)^d$ and $T_d(x)$ are *easy* polynomials, since both can be computed by $O(\log d)$ size circuits! But,

in this work, we conjecture these easy polynomials (in the circuit model) to be SOS-hard.

- On the other hand, a *random* $d^{1/2}$-sparse polynomial is trivially SOS-easy (*not* SOS-hard), but it requires $\omega(\log d)$ size circuit!

- Finally, it is not hard to show that $f_d(x) := \sum_{i=0}^{d} x^i$, is easy both in SOS and circuit model! The key idea is to use the identity:

  $$f_{n^2-1}(x) = (1 + x + \cdots x^{n-1}) \cdot (1 + x^n + \cdots + x^{n(n-1)}).$$

  When $n^2-1 \le d < (n+1)^2-1$, use the fact that $f_d(x) = f_{n^2-1}(x) + x^{n^2} \cdot f_{d-n^2}(x)$, and compute f_d recursively.

Requirement of GRH. To separate VP from VNP, we will eventually try to create a multivariate hard polynomial family from the SOS-hard polynomial family, via some explicit transformation. However, by Theorem 2.4.3, it is sufficient to show that the coefficients are #P/poly-explicit, in order to show that the family is in VNP. As a result, without assuming GRH, our current proof techniques *fail* for CH-explicit polynomial families, such as $(x+1)^d$, $\prod_{i=1}^{d}(x-i)$. Essentially, if both GRH and VP = VNP are assumed, then Theorem 2.4.1 shows that the transformed polynomial family is explicit enough to be in VNP = VP; however the SOS-hardness parameter itself shows that it cannot be computed by polynomial-sized circuits (i.e., $\notin$ VP). Thus, by simple logical deduction, it follows that assuming GRH and SOS-hardness, VP $\neq$ VNP!

However, GRH is *not required* for many explicit polynomial families, such as, $\sum_{i=0}^{d} 2^{i^2} x^i$.[5]

5: One could wonder why not just work with very explicit polynomials, like $\sum_{i=0}^{d} 2^{i^2} x^i$, without even mentioning GRH! Howbeit, because of the ingrained patterns and well-behaved (explicit) roots, polynomials such as $\prod_i (x-i)$, or $(x+1)^d$, could be *easier* to deal with.

3.2 Comparison with Prior Works

SOS to non-commutative hardness. Hrubeš, Wigderson, and Yehudayoff [HWY11] considered the sum-of-squares representation in the *non-commutative* setting. They showed that lower bounds for the SOS-representation of a specific multivariate polynomial imply exponential lower bounds on the circuit size of the permanent. Besides the non-commutative

algebra, their setting differs in the precise SOS-model and the complexity measure. So, hardly any comparison is possible.

Depth-4 circuits with unbounded powering. Much of the previous works are concerned with multivariate depth-4 circuits that are a sum of unbounded-powers, i.e., $\Sigma \wedge^{[\omega(1)]} \Sigma\Pi$ circuits, because this is the form one gets after applying the depth-reduction results [AV08; Koi12; Gup+16; AGS19]. The sufficiency of proving lower bound on restricted models of *univariate* polynomials was shown by Koiran [Koi11]. He considered univariate explicit polynomials f_d of degree d over an algebraically closed field $\mathbb{F}$ that are written as

$$f_d(x) = \sum_{i=1}^{s} c_i \, Q_i^{e_i}(x)\,,$$

where $c_i \in \mathbb{F}$, and Q_i are polynomials with sparsity $\mathrm{sp}(Q_i) \leq t$ with unbounded exponents $e_i \geq 1$. He showed that when every such presentation of f_d requires $s = (d/t)^{\Omega(1)}$ summands, then VP $\neq$ VNP.

Some initial lower bounds have been established for this model.

- When $\deg(Q_i) \leq t$, [Kay+15] showed existence of an explicit family such that $s \geq \Omega(\sqrt{d/t})$.
- For $\deg(Q_i) \leq 1$, the bound $s \geq \Omega(d)$ has been established for certain polynomials using the concept of *Birkhoff Interpolation* [GK17; KPG18].

Clearly, allowing arbitrary exponents gives much more flexibility than fixed exponents as in SOS and SOC. In that sense, it should be easier to obtain lower bounds in the SOS- or SOC-model. Also the complexity measure is different, as Koiran considers the number of summands, whereas we consider the support-sum.

Existence of $(r, 2)$-elusive function vs. SOS-hardness. Raz [Raz10] formalized a notion of elusive maps and established a connection between the existence of explicit elusive maps and VP vs. VNP. A polynomial map $L : \mathbb{F}^n \to \mathbb{F}^m$ is $(r, 2)$-*elusive*, if for every polynomial of degree 2 that maps $M : \mathbb{F}^r \to \mathbb{F}^m$, we have $\mathrm{Image}(L) \not\subseteq \mathrm{Image}(M)$. Formally, he showed that any explicit polynomial map which is $(r, 2)$-elusive, with $m = n^{\omega(1)}$ and $r = n^{0.9}$, implies VP $\neq$ VNP.

Observe that one can reinterpret the coefficients of the f_i^2's in Equation 3.1 as expressing $\mathrm{coef}(f)$ via quadratic forms, like M. However, the elusiveness notion is *too* general in the following sense: the parameters r and m have a super-polynomial large gap, and still M has to elude all L. On the other hand, SOS-hardness, say for $N = 1$, goes a step further and optimizes the gap to be vanishingly close to *square*. Further, SOS gives a rather specialized degree-2 polynomial mapping.

From hardness to derandomization. With respect to the derandomization of the blackbox PIT, there are a few (strong) conditional results. For example, it has been shown that multivariate hard polynomials lead to blackbox PIT $\in$ QP (*quasipolynomial-time*) [KI04; AGS19]. Closer to our result is the work of [Guo+19]. They showed that the circuit hardness of a constant-variate polynomial family yields blackbox PIT $\in$ P (Theorem 2.7.2). Still, our hardness assumption is merely in the SOC-model and for univariate polynomials. For now, SOC seems to be the simplest model where hardness implies a complete derandomization.

Earlier τ-conjectures vs. our SOS- or, SOC-τ-conjecture. Technically, our SOS-τ-conjecture is incomparable to the earlier τ-conjectures, since all the previous works [Koi11; GKT15; Koi+15] used the standard depth-reduction results [AV08; Koi12; Gup+16; AGS19]; hence, they were concerned with the sum-of *unbounded*-powers $\Sigma\Pi^{[m]}\Sigma\Pi$, with $m = \omega(1)$, while we work with $m = 2$. This is the *first* time we are showing connections to matrix-rigidity and PIT; these were perhaps always desired of, nonetheless *never achieved*.

Moreover, the measure $S(f)$ in the τ-conjecture is different from the usual circuit size. If we consider the expression in Equation 3.1 as a $\Sigma \wedge^{[2]} \Sigma\Pi$-formula, then the support-sum is the number of Π-operations directly above the input level. However, the usual measure is the size of the depth-4 circuit $\Sigma^{[k]}\Pi^{[m]}\Sigma^{[t]}\Pi$. Even if we substitute $m = 2$, there is *no* direct dependence of t, the individual sparsity of the intermediate polynomials f_i, on $S(f)$, which implies that the sparsity of some f_i could be *large*. However, the upper bound requirement in [Koi11] is $\mathrm{poly}(kmt)$ while the SOS-τ-conjecture *demands* a linear (*stronger*) dependence on $S(f)$.

Further, the polynomial family and the proof used in [Koi11; GKT15] are *different* from those in Theorem 3.3.9, as it relies on depth-4 reduction and the usual Kronecker map while our proof relies on multilinearization ([DST21]) and a folklore decomposition (Lemma 3.3.4), see [SY10; Raz10; Sap21].

3.3 Sum-of-Squares

In this section, let $\mathbb{F}$ be a field of characteristic $\neq 2, 3$.

3.3.1 From SOS-hardness to VP $\neq$ VNP

The connection between the SOS-model and general circuits is mainly established by the next lemma. It shows that a multivariate polynomial $p(x)$ of degree d, computed by a circuit of size s, has a SOS-representation with $(sd)^{O(\log d)}$ summands, where each summand polynomial has the degree precisely $d/2$.

This is achieved by transforming the given circuit for $p(x)$ in several steps into a *homogeneous* ABP. The point here is that degrees of the polynomials computed at the intermediate nodes of the ABP increase gradually, as the labels are linear forms. In particular, there exists a layer of vertices that computes polynomials of degree exactly $d/2$. By cutting the ABP at that layer, we get a representation of p as a sum of products of two polynomials of degree $d/2$ each. This yields the desired SOS-representation.

Below, we present an SOS decomposition lemma, which is similar to Lemma 3.3.5 below. It uses the frontiers based depth-reduction technique of [Val+83]. However, this approach yields intermediate polynomials of degree only close to $d/2$, whereas we want degree exactly $d/2$ here.

Lemma 3.3.1 (SOS Decomposition) *Let $p \in \mathbb{F}[x]$ be an n-variate polynomial of degree d, with $\text{size}(p) = s$. Then there exist $p_i \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$ such that*

$$p(x) = \sum_{i=1}^{s'} c_i p_i(x)^2 , \tag{3.10}$$

for $s' = (sd)^{O(\log d)}$ *and* $\deg(p_i) \leq \lceil d/2 \rceil$, *for all* $i \in [s']$.

Proof. Let C be a circuit of size s that computes p. Let us first assume that p is a homogeneous polynomial. We transform C by the following steps.

1. We apply *depth-reduction* to C [Val+83], and get a homogeneous circuit C' of depth $\log d$ and size $\text{poly}(s)$ that computes p.
2. Then we convert C' into a formula F by unfolding the gates with fan-out larger than one. By induction on the depth of the circuit, one can show that F has size $s^{O(\log d)}$.
3. Next, we convert F to an ABP B. It is well known that for any formula of size t, there exists an equivalent ABP of size at most $t + 1$, for details see [Sau12, Lemma 2.14]. Thus, the ABP B that computes p has size at most $s^{O(\log d)}$.
4. Finally, we *homogenize* B to a *layered* ABP B' as explained at the end of the preliminary section. Its size is $|B'| = \text{poly}(s^{O(\log d)}) = s^{O(\log d)}$.

To obtain the representation (Equation 3.10) of p, we cut ABP B' in half. That is, we split B' along the nodes in the $\lceil d/2 \rceil$-th layer. The i-th node v_i in that layer (in some order) defines two ABPs, one between the starting node of B' and v_i as end node, and a second one between v_i as starting node and the end node of B'. Let $p_{i,1}$ and $p_{i,2}$ be the two polynomials computed by these ABPs, respectively. By the definition of how ABPs compute polynomials, we have

$$p = \sum_{i=1}^{|B'|} p_{i,1}\, p_{i,2}\,,$$

where the degree of $p_{i,1}, p_{i,2}$ is at most $\lceil d/2 \rceil$. Now each product can be written as a SOS by Equation 3.5, as $p_{i,1}\, p_{i,2} = \frac{1}{4}\left((p_{i,1} + p_{i,2})^2 - (p_{i,1} - p_{i,2})^2\right)$ to obtain (Equation 3.10). Hence, we get a SOS-representation of p with top fan-in $s' = 2|B'|$.

For a non-homogeneous polynomial p, it is known that the homogeneous parts can be computed by homogeneous circuits of size $O(sd^2)$. Thus, for non-homogeneous polynomials, we can replace the s from above by $O(sd^2)$. Then the top-fanin of the SOS-representation is $(sd^2)^{O(\log d)} = (sd)^{O(\log d)}$. □

Now we come to our main result. We show how to lift the hardness of univariate polynomial f_d of degree d in the SOS-model to a multivariate polynomial that has circuits of super-polynomial size, it will be in VNP and not in VP.

Our technique is to convert f_d into a multivariate polynomial $P_{n,k}$ via the multilinear Kronecker substitution defined in the preliminary section. Polynomial $P_{n,k}$ will have kn variables and degree n, for carefully chosen parameters k and n that depend on d and the SOS-hardness parameter ε for f_d. Since n is a function in k, it would actually suffice to index the family over k. We will eventually show that $\text{size}(P_{n,k}) = (kn)^{\omega(1)}$.

The proof goes via contradiction. If the size is smaller than claimed, then, by Lemma 3.3.1, we can write $P_{n,k}$ as the sum of $d^{o(\varepsilon)}$-many Q_i^2's, where the polynomials Q_i have kn variables and degree at most $n/2$. Thus, the support-sum of $P_{n,k}$, and hence of f_d as well, is bounded by $d^{o(\varepsilon)}\binom{kn+n/2}{n/2}$. We show that, for carefully chosen parameters, the latter expression is bounded by $o(d^{1/2+\varepsilon})$. Hence, we get a contradiction to the SOS-hardness of f_d.

Theorem 3.3.2 *If there exists an SOS-hard explicit family (f_d) with hardness $\varepsilon = \varepsilon(d) = \omega\left(\sqrt{\frac{\log\log d}{\log d}}\right)$, then* VP $\neq$ VNP.

Proof. Let $f_d(x)$ be an explicit SOS-hard polynomial with hardness ε as in the theorem statement. We define parameters k and n as follows. Choose k large enough such that

$$(k-1)^{\varepsilon} \geq 6\,. \tag{3.11}$$

That is, define $k = \lceil 6^{\frac{1}{\varepsilon}} + 1 \rceil$. Then choose n such that

$$(k-1)^n \leq d+1 \leq k^n\,. \tag{3.12}$$

Note that $n = \Theta(\varepsilon \cdot \log d) = O(\log d)$.

Now we apply the multilinear Kronecker map $\phi_{n,k}^{\text{lin}}$ from Equation 2.4 to f_d and define polynomial

$$P_{n,k}(y) = \phi_{n,k}^{\text{lin}}(f_d(x))\,.$$

Recall that $P_{n,k}$ is multilinear of degree n and has kn variables $y_{j,\ell}$, where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$. We show that

$P_{n,k} \in \mathsf{VNP}$ and $\notin \mathsf{VP}$, thereby separating the classes.

Part 1: $P_{n,k} \in \mathsf{VNP}$ Let

$$P_{n,k} = \sum_{e \in \{0,1\}^{kn}} c_n(e)\, y^e .$$

By the inverse multilinear Kronecker map $\psi^{\mathrm{lin}}_{n,k}$ from Equation 2.5, we get an exponent e such that $x^e = \psi^{\mathrm{lin}}_{n,k}(y^e)$. Note that coefficient $c_n(e)$ in $P_{n,k}$ is the coefficient of x^e in f_d. We can compute e in time $\mathrm{poly}(n,k)$ and each bit of $c_n(e)$ in time $\mathrm{poly}(\log d) = \mathrm{poly}(n \log k)$, by the explicitness of f_d. Hence, $P_{n,k} \in \mathsf{VNP}$ by Theorem 2.4.3.

Part 2: $P_{n,k} \notin \mathsf{VP}$ Define

$$\mu = \frac{1}{\sqrt{\log d \log\log d}} .$$

We will show that $\mathrm{size}(P_{n,k}) \geq d^{\mu}$.

Assume to the contrary that $\mathrm{size}(P_{n,k}) \leq d^{\mu}$. By Lemma 3.3.1, there exist polynomials Q_i such that $P_{n,k} = \sum_{i=1}^{s} c_i\, Q_i^2$, where $s = (d^{\mu} n)^{O(\log n)}$ and $\deg(Q_i) \leq \lceil n/2 \rceil$.

We apply the inverse multilinear Kronecker map $\psi^{\mathrm{lin}}_{n,k}$ to the Q_i's: Define $g_i(x) = \psi^{\mathrm{lin}}_{n,k}(Q_i(y))$. Note that the Q_i's might no longer be multilinear. Anyway we can apply the ψ^{lin}-transformation. Then we get

$$f_d = \sum_{i=1}^{s} c_i\, g_i^2 .$$

Recall that sparsity of g_i can be at most that of Q_i. For the sparsity of Q_i, we use the general bound Equation 2.1. That is, $\mathrm{sp}(Q_i) \leq \binom{kn+\lceil n/2 \rceil}{\lceil n/2 \rceil}$, for all $i \in [s]$. Thus,

$$S(f_d) \leq s \binom{kn + \lceil n/2 \rceil}{\lceil n/2 \rceil}. \tag{3.13}$$

In the following two claims, we give upper bounds for s and the binomial coefficient in Equation 3.13. Let

$$\delta = \sqrt{\frac{\log\log d}{\log d}} .$$

Note that $\delta = \mu \log\log d = o(\varepsilon)$.

Claim 3.3.1 $s = d^{O(\delta)} = d^{o(\varepsilon)}$.

Proof. Recall that $s = (d^\mu n)^{O(\log n)}$. We show that $(d^\mu n)^{O(\log n)} = d^{O(\delta)}$. Taking logarithms, we have to show that

$$\log n\,(\mu \log d + \log n) = O(\delta)\log d\,. \tag{3.14}$$

Recall that $n = O(\log d)$. Hence, we have $\log n = O(\log\log d)$. Now it suffices to show that

$$\mu \log\log d + \frac{(\log\log d)^2}{\log d} = O(\delta)\,. \tag{3.15}$$

But this holds because by the definitions of μ and δ, for large enough d, we have

$$\frac{(\log\log d)^2}{\log d} < \mu \log\log d = \delta\,.$$

This proves the claim. □

Claim 3.3.2 $\binom{kn+\lceil n/2\rceil}{\lceil n/2\rceil} \le d^{\frac{1+\varepsilon}{2}}$.

Proof. We use Equation 2.6 to bound the binomial coefficient. We omit the ceiling brackets for better readability.

$$\binom{kn + n/2}{n/2} \le \left(\frac{e(kn+\frac{n}{2})}{\frac{n}{2}}\right)^{\frac{n}{2}} = (2ek+e)^{\frac{n}{2}} \le (6(k-1))^{\frac{n}{2}}\,. \tag{3.16}$$

The last inequality is because $2e < 6$. Hence, we get that $2ek + e \le 6(k-1)$, for large enough k.

By Equation 3.11, we get that $6(k-1) \le (k-1)^{1+\varepsilon}$. Hence, we can continue Equation 3.16 by

$$(6(k-1))^{\frac{n}{2}} \le (k-1)^{\frac{n}{2}(1+\varepsilon)} \le d^{\frac{1+\varepsilon}{2}}\,.$$

The last inequality follows by our choice of n such that $(k-1)^n \le d$. This proves the claim. □

We plug in the bounds from the two claims in Equation 3.13 and get

$$S(f_d) = d^{o(\varepsilon)}\, d^{\frac{1+\varepsilon}{2}} = o(d^{1/2+\varepsilon})\,.$$

This is a contradiction to the SOS-hardness of f_d. So $\mathrm{size}(P_{n,k}) \geq d^{\mu}$.

It remains to show that d^{μ} is super-polynomial in parameters k and n.

Claim 3.3.3 $d^{\mu} = (kn)^{\omega(1)}\,.$

Proof. Taking logarithms, we have to show that

$$\mu \log d = \omega(\log k + \log n)\,. \tag{3.17}$$

For the left hand side of Equation 3.17, we have

$$\mu \log d = \sqrt{\frac{\log d}{\log\log d}} = \omega(1/\varepsilon)\,.$$

For the right-hand side of Equation 3.17, we have

$$\log k = \log\lceil 6^{1/\varepsilon} + 1\rceil = O(1/\varepsilon)\,, \tag{3.18}$$

$$\log n \leq \log\log d = o(1/\varepsilon)\,. \tag{3.19}$$

This proves Equation 3.17, and the claim follows. □

We conclude that $P_{n,k}$ requires super-polynomial size circuits, and therefore, $P_{n,k} \notin \mathrm{VP}$. This proves the theorem. □

Remark 3.3.1

1. We used the multilinear Kronecker substitution ϕ^{lin} because the standard one ϕ from Equation 2.2 would *not* give our result. For d, k, n as above, polynomial $\phi_{n,k}(f_d)$ would have only k variables but higher degree, kn, compared to $P_{n,k}$ from above. Then the binomial coefficient in Equation 3.13 would become $\binom{k+kn/2}{k} > (n+1)^k > d$. Hence, Claim 3.3.2 would no longer hold.
2. Recall from the proof that $\deg(Q_i) \leq n/2$. Hence, for $g_i(x) = \psi_{n,k}(Q_i(y))$, by the definition of ψ, we have

$$\deg(g_i) \leq \frac{n}{2}\,(k-1)\,k^{n-1} < n\,k^n = O(nd) = O(d\log d)\,.$$

Thus, in the SOS-hardness assumption for f_d we could additionally restrict the degree of the polynomials in the SOS-representation to $O(d \log d)$, and still Theorem 3.3.2 would hold.

3. Similarly, by Claim 3.3.1, we could additionally restrict the top fan-in s in the SOS-representation to $s = d^\delta$ and still Theorem 3.3.2 would hold. Note that this is very small compared to d since $d^\delta = d^{o(\varepsilon)}$.

Separating VBP and VNP. Recall that VBP $\subseteq$ VP. If we are interested in the weaker separation of VBP and VNP, then actually a smaller hardness parameter ε suffices in the assumption. The reason comes from Lemma 3.3.1: When we start with a polynomial p given by an ABP of size s, we can skip transformation steps 1, 2, 3, and only do the homogenization step 4. Then the resulting ABP has size only poly(s), i.e., we do not have the $\log d$-term in the exponent. So we can modify the proof of Theorem 3.3.2 and set $\varepsilon = \omega(1/\sqrt{\log d})$ and $\mu = \delta = 1/\sqrt{\log d}$, and still all the calculations go through, in particular Claim 3.3.1.

Corollary 3.3.3 (Determinant vs Permanent) *If there exists an SOS-hard explicit family (f_d) with hardness parameter $\varepsilon = \omega(1/\sqrt{\log d})$, then* VBP $\neq$ VNP.

3.3.2 An exponential separation of VP and VNP

The argument for an exponential separation of VP and VNP follows the proof of Theorem 3.3.2. However, we use a different decomposition lemma and a different parameter setting. The decomposition lemma is based on the circuit depth-reduction technique of [Val+83]. See for example [Sap21] for a very well written survey on *frontier decomposition*, the technique to prove the following lemma.

Lemma 3.3.4 (Sum of product-of-2) *Let $p \in \mathbb{F}[x]$ be an n-variate homogeneous polynomial of degree d, computed by a homogeneous circuit of size s. Then there exist polynomials*

$p_{i,j} \in \mathbb{F}[x]$ *such that*

$$p = \sum_{i=1}^{s} p_{i,1}\, p_{i,2}\,, \tag{3.20}$$

and for all $i \in [s]$ *and* $j = 1, 2$,

1. $\frac{d}{3} \leq deg(p_{i,j}) \leq \frac{2d}{3}$,
2. $deg(p_{i,1}) + deg(p_{i,2}) = d$, *and*
3. $p_{i,j}$ *has a homogeneous circuit of size* $O(s)$.

Remark 3.3.2 For a non-homogeneous polynomial $p(x)$, we can apply Lemma 3.3.4 for each homogeneous part of $p(x)$. It is well known that each homogeneous part can be computed by a homogeneous circuit of size $O(sd^2)$. Thus, for non-homogeneous polynomials, s can be replaced by $O(sd^2)$ and we get a similar conclusion.

The following lemma iterates the decomposition in Lemma 3.3.4 to bring the degree of the intermediate polynomials close to $d/2$, while keeping the circuit size polynomial s. Note the contrast to Lemma 3.3.4 where we got intermediate polynomials of degree precisely $d/2$ but paid with super-polynomial circuit size.

Lemma 3.3.5 *Let* $\frac{1}{2} < \gamma < 1$ *be a constant. Then there exists a constant* c, *such that for any* n*-variate homogeneous polynomial* $p \in \mathbb{F}[x]$ *of degree* d *that can be computed by a homogeneous circuit of size* s, *we have a representation*

$$p = \sum_{i=1}^{s^c} q_i^2\,, \tag{3.21}$$

where $q_i \in \mathbb{F}[x]$, *for all* $i \in [s^c]$, *such that*

1. $deg(q_i) < \gamma d$,
2. q_i *has a homogeneous circuit of size* $O(s)$.

Proof. By Lemma 3.3.4, we can write $p(x) = \sum_{i=1}^{s} \tilde{p}_{i,1}\, \tilde{p}_{i,2}$, where $\deg(\tilde{p}_{i,j}) \leq 2d/3$, $\deg(\tilde{p}_{i,1}) + \deg(\tilde{p}_{i,2}) = d$, and $\tilde{p}_{i,j}$ has circuits of size $O(s)$.

Let $\delta = \gamma - 1/2$. Choose constant m such that $(2/3)^m < \delta$. That is, let $m = \lceil \log_{3/2}(1/\delta) \rceil$. Now we apply Lemma 3.3.4 recur-

sively m-times to each $\tilde{p}_{i,j}$. It follows that we can write $p(x)$ as

$$p(x) = \sum_{i=1}^{s^{2^m-1}} \hat{p}_{i,1}\, \hat{p}_{i,2} \cdots \hat{p}_{i,2^m}\,, \tag{3.22}$$

where $\deg(\hat{p}_{i,j}) \leq (2/3)^m\, d < \delta d$. For all $i \in [s^{2^m-1}]$, we have $\sum_{j=1}^{2^m} \deg(\hat{p}_{i,j}) = d$ and $\mathrm{size}(\hat{p}_{i,j}) = O(s)$, for all $j \in [2^m]$.

For each product $\hat{p}_{i,1} \cdots \hat{p}_{i,2^m}$, pick the smallest $j_0 \in [2^m]$ such that

$$\frac{d}{2} \leq \sum_{j=1}^{j_0} \deg(\hat{p}_{i,j}) < \gamma d\,.$$

Define $p_{i,1} = \hat{p}_{i,1} \cdots \hat{p}_{i,j_0}$ and $p_{i,2} = \hat{p}_{i,j_0+1} \cdots \hat{p}_{i,2^m}$. Then we have

$$p = \sum_{i=1}^{s^{2^m-1}} p_{i,1}\, p_{i,2}\,. \tag{3.23}$$

By definition, $d/2 \leq \deg(p_{i,1}) < \gamma d$, and therefore, $\deg(p_{i,2}) = d - \deg(p_{i,1}) \leq d/2 < \gamma d$. Because each $\hat{p}_{i,j}$ has a homogeneous circuit of size $O(s)$, so does $p_{i,j}$. Finally, we use equality Equation 3.5 as $p_{i,1}\, p_{i,2} = \frac{1}{4}\left((p_{i,1} + p_{i,2})^2 - (p_{i,1} - p_{i,2})^2\right)$ to obtain Equation 3.21 with $c = 2^m$. □

Remark 3.3.3 Similar as remarked for Lemma 3.3.4, for a non-homogeneous polynomial $p(x)$, the size s can be replaced by $O(sd^2)$ and we get a similar conclusion.

Lemma 3.3.5 provides the tool for an exponential separation of VP and VNP. The argument follows the proof of Theorem 3.3.2. Instead of Lemma 3.3.1, we use Lemma 3.3.5. Also we use a different parameter setting.

Theorem 3.3.6 (Constant ε) *If there exists an SOS-hard explicit family with constant hardness parameter $\varepsilon > 0$, then* VNP *is exponentially harder than* VP.

Proof. Let $f_d(x)$ be an explicit SOS-hard polynomial with constant hardness parameter $\varepsilon < \frac{1}{2}$. First, we define parameters k and n. Let

$$\gamma = \frac{1}{2} + \frac{\varepsilon}{4}\,. \tag{3.24}$$

Choose constant k large enough such that

$$(k-1)^{\frac{\varepsilon}{12}} \geq 6^{\gamma}\,, \tag{3.25}$$

and n again such that $(k-1)^n \leq d+1 \leq k^n$. Note that $n = O(k \log d) = O(\log d)$. Then define $P_{n,k}(y) = \phi_{n,k}(f_d(x))$. Again, we have $P_{n,k} \in \mathsf{VNP}$.

We show that $P_{n,k}$ requires exponential size circuits. We apply Lemma 3.3.5 to $P_{n,k}$ with parameter γ. Let c be the constant such that s^c bounds the top fan-in in Equation 3.21. That is, when $\text{size}(P_{n,k}) = s$, we get a representation

$$P_{n,k} = \sum_{i=1}^{s^c} c_i Q_i^2 , \tag{3.26}$$

where $\deg(Q_i) \leq \gamma n$. Define constant

$$\mu = \frac{\varepsilon}{3c} . \tag{3.27}$$

Claim 3.3.4 $\text{size}(P_{n,k}) > d^\mu$.

Proof. Assume that $\text{size}(P_{n,k}) \leq d^\mu$. Via the inverse Kronecker map applied to the Q_i's, we get a bound similar to Equation 3.13:

$$S(f_d) \leq d^{c\mu} \binom{kn + \lceil \gamma n \rceil}{\lceil \gamma n \rceil} . \tag{3.28}$$

We bound the binomial coefficient:

$$\binom{kn+\gamma n}{\gamma n} \leq \left(\frac{e(kn+\gamma n)}{\gamma n}\right)^{\gamma n} \leq (2ek+e)^{\gamma n} \leq (6(k-1))^{\gamma n} . \tag{3.29}$$

The last inequality is again for large enough k. By Equation 3.25 and the definition of γ, we get that

$$(6(k-1))^{\gamma n} \leq (k-1)^{n(\frac{\varepsilon}{12}+\gamma)} \leq d^{\frac{1}{2}+\frac{\varepsilon}{3}} .$$

Plugging the bound into Equation 3.28, we get by definition of μ

$$S(f_d) \leq d^{c\mu} d^{\frac{1}{2}+\frac{\varepsilon}{3}} = d^{\frac{1}{2}+\frac{2}{3}\varepsilon} = o(d^{\frac{1}{2}+\varepsilon}) . \tag{3.30}$$

This is a contradiction to the SOS-hardness of f_d. This proves the claim. □

Finally observe that by the definition of n, and since μ and k are constants, we have $d^\mu = \Omega(k^{\mu n}) = 2^{\Omega(kn)}$. Hence, $P_{n,k}$

requires exponential size circuits. This shows an exponential separation between VP and VNP. □

3.3.3 From SOS-τ-conjecture to Matrix Rigidity

Assuming Conjecture 3.1.2, one can show existence of explicit rigid matrices.

Theorem 3.3.7 *If Conjecture 3.1.2 holds, then there exist $\epsilon > 0$ and a "very"-explicit family of real matrices $(A_n)_n$ such that A_n is $(\epsilon \cdot n, n^{1+\delta})$-rigid, for any $\delta < 1$.*

Proof. The essential idea is to show that a non-rigid matrix can be factored into two 'sparse' matrices (Part 1). And then, one construct matrices from the 'explicit' polynomial family defined by $f_d := \prod_{i\in[d]}(x - i)$, which *cannot* be factored into sparse matrices (Part 2). These two parts together imply that the matrix family must be rigid.

Part 1: Non-rigid matrices as 'small' depth-2 circuits. We argue via *linear circuits* which we have defined in section 2.5. Linear circuits can compute linear functions; for details, refer to [KV21, Section 1.2]). As a graph, the nodes of a linear circuit are either input nodes or addition nodes, and the edges are labeled by scalars. If an edge from u to v is labeled by $c \in \mathbb{F}$, then the output of u is multiplied by c and then given as input to v.

We eventually establish that any matrix $A \in \mathbb{R}^{n\times n}$ which is not (r, s)-rigid, for some r, s, can be computed by a depth-2 circuit of size $2rn + s + n$; see item 1-4 below. This will be crucial in the proof of Theorem 3.3.7. We give this bound over any general field $\mathbb{F}$.

1. Let $a = (a_1, \dots, a_n)$ be a vector. Consider a as a linear function $\mathbb{F}^n \to \mathbb{F}$. It can be computed by a linear circuit of depth 1 with n inputs and one addition-gate as output gate. The edge from the i-th input is labeled by a_i. The size of the circuit is n. However, we omit edges labeled by 0. Hence, the size of the circuit is actually $\mathrm{sp}(a) \le n$, the sparsity of a.

Similarly, we consider an $n \times n$ matrix A as a linear transformation $\mathbb{F}^n \to \mathbb{F}^n$. For each row vector of A we get a linear circuit as described above. Hence we represent A by circuit of depth 1 with n output gates and size $\mathrm{sp}(A) \leq n^2$.

2. The model gets already interesting for linear circuits of depth 2. Suppose $A = BC$, where B is a $n \times r$ matrix and C is a $r \times n$ matrix. Then we can take the depth-1 circuit for C at the bottom as in item 1 and combine it with the depth-1 circuit for B on top. The resulting depth-2 circuit is *layered*: all edges go either from the bottom to the middle layer, or from the middle to the top layer. The size of the circuit is $\mathrm{sp}(B) + \mathrm{sp}(C) \leq 2rn$.
   In particular, there is a representation $A = BC$ with $r = \mathrm{rank}(A)$. Hence the rank of A is involved in the circuit size bound for A. Also note that r is bounded by the size of the circuit because be omit all zero-edges.
   Note that any layered linear circuit of depth 2 in turn gives a factorization of A as a product of 2 matrices, $A = BC$, where the top edges define B and the bottom edges C.

3. Let $A = BC + D$, where B, C are as above and D is a $n \times n$ matrix. Then we can represent A by a depth-2 circuit for BC as in item 2 plus edges from the inputs directly to the output nodes to represent D as in item 1. The resulting circuit has depth 2 and size $\mathrm{sp}(B) + \mathrm{sp}(C) + \mathrm{sp}(D) \leq 2rn + n^2$, but it would not be layered. We can transform it into a layered circuit by writing A as $A = BC + ID$, where I is the $n \times n$ identity matrix. Then we get a depth-2 circuit for ID similar to BC and can combine the two circuits into one. The size increases by $\leq n$ edges for I.

4. Now consider matrix A that is not (r, s)-rigid, for some r, s. Hence, we can write A as $A = R+S$, where $\mathrm{rank}(R) = r$ and $\mathrm{sp}(S) = s$. Then R can be written as as $R = BC$, where B is a $n \times r$ matrix and C is a $r \times n$ matrix. From item 3, we have that $A = BC + S$ has a layered linear circuit of depth 2 of size $\leq 2rn + s + n$.

Part 2: Construction of rigid matrices. Consider the polynomial family $f_d := \prod_{i \in [d]}(x - i)$. Let $d =: n^2 - 1$, for some $n \in \mathbb{N}$. Conjecture 3.1.2 implies that $S_{\mathbb{R}}(f_d) > \delta' \cdot d$, for

some $\delta' > 0$. Note that $\delta' \leq 2$, as $S_{\mathbb{R}}(f_d) \leq 2d + 2$, from the upper bound (see Equation 3.2). This δ' will play a crucial role in the proof.

Define the bivariate polynomial $g_n \in \mathbb{R}[x_1, x_2]$ from f_d such that after the Kronecker substitution, $g_n(x, x^n) = f_d$. It is easy to construct g_n from a given d; just convert every x^e, for $e \in [0, d]$ to $x_1^{e_1} \cdot x_2^{e_2}$, where $e =: e_1 + e_2 \cdot n$, and $0 \leq e_i \leq n - 1$. Thus, the individual degree of each x_i in g_n is at most $n - 1$.

Let $g_n(x_1, x_2) = \sum_{1 \leq i,j \leq n} a_{i,j}\, x_1^{i-1} x_2^{j-1}$. By the definition of f_d, $a_{i,j} = \mathrm{coef}_{x^{(i-1)+(j-1)n}}(f_d)$. Define the $n \times n$ matrix $A_n = \left(a_{i,j}\right)_{1 \leq i,j \leq n}$ and vectors

$$[x_1]_n = \begin{pmatrix} 1 & x_1 & \cdots & x_1^{n-1} \end{pmatrix}, \quad [x_2]_n = \begin{pmatrix} 1 & x_2 & \cdots & x_2^{n-1} \end{pmatrix}.$$

Thus, $g_n(x_1, x_2) = [x_1]_n A_n [x_2]_n^T$. Further, $a_{i,j}$ is poly(n)-computable implies A_n is poly(n)-explicit. Next we show a lower bound on the linear circuit size of A_n.

Lemma 3.3.8 *Conjecture 3.1.2 $\implies$ any layered linear circuit of depth 2 that computes A_n, has size $> (\delta'/2) \cdot d$.*

Proof of Lemma 3.3.8. Conjecture 3.1.2 implies that $S_{\mathbb{R}}(f_d) > \delta' \cdot d$, for some $\delta' > 0$. We show that, size of the linear circuit computing A_n has size $> (\delta'/2) \cdot d$.

Assume that this is false. Then we can write $A_n = BC$, where $B \in \mathbb{R}^{n \times t}$, $C \in \mathbb{R}^{t \times n}$, such that $t \leq \mathrm{sp}(B) + \mathrm{sp}(C) \leq (\delta'/2) \cdot d$.

Denote

$$[x_1]_n B = \begin{pmatrix} \ell_1(x_1) & \ell_2(x_1) & \cdots & l_t(x_1) \end{pmatrix},$$

and

$$C[x_2]_n^T = \begin{pmatrix} \tilde{\ell}_1(x_2) & \tilde{\ell}_2(x_2) & \cdots & \tilde{\ell}_t(x_2) \end{pmatrix}^T.$$

Then

$$g_n(x_1, x_2) = [x_1]_n A_n [x_2]_n^T = [x_1]_n BC[x_2]_n^T = \sum_{i=1}^{t} \ell_i(x_1)\, \tilde{\ell}_i(x_2).$$

Since $\mathrm{sp}(B) + \mathrm{sp}(C) \leq (\delta/2) \cdot d$, we have $\sum_{i=1}^{t} (|\ell_i|_0 + |\tilde{\ell}_i|_0) \leq$

$(\delta'/2)\cdot d$. Substituting $x_1 = x$ and $x_2 = x^n$, we get

$$f_d(x) \;=\; g(x,x^n) \;=\; \sum_{i=1}^{t} \ell_i(x)\,\tilde{\ell}_i(x^n)$$
$$= \sum_{i=1}^{t} \left(\frac{\ell_i(x)+\tilde{\ell}_i(x^n)}{2}\right)^2 - \sum_{i=1}^{t} \left(\frac{\ell_i(x)-\tilde{\ell}_i(x^n)}{2}\right)^2 .$$

Thus, we have a representation of f_d as $\leq 2t \leq \delta'\cdot d$ sum of squares. Note that, this means

$$S_{\mathbb{R}}(f_d) \;\leq\; \sum_{i=1}^{t} 2\cdot\left(|\ell_i|_0 \;+\; |\tilde{\ell}_i|_0\right) \;\leq\; \delta'\cdot d, \tag{3.31}$$

contradicting the assumption on the hardness of f_d. This proves Lemma 3.3.8.

We now show that A_n is $((\delta'/8)\cdot n, n^{1+\delta})$-rigid, for any $\delta < 1$. For the sake of contradiction, assume that this is false. Then there is a $\delta < 1$, and a decomposition $A_n = R + S$, where $\mathrm{rank}(R) = r = (\delta'/8)\cdot n$, and $\mathrm{sp}(S) = s = n^{1+\delta}$. By item 4 above, A_n has a layered linear circuit C_n of depth 2 of size

$$\mathrm{size}(C_n) \;\leq\; 2rn + s + n \;\leq\; \frac{\delta'\cdot n^2}{4} + 2n^{1+\delta}. \tag{3.32}$$

Recall that δ' is a constant and $\delta < 1$. Hence, for large enough n, we have $2n^{1+\delta} \leq \delta'\cdot(\frac{n^2-2}{4})$. Note: $\delta = 1$ is not achievable as $\delta' \leq 2$. Now, we can continue the inequalities in Equation 3.32 by

$$\mathrm{size}(C_n) \;\leq\; \delta'\cdot(\frac{n^2-1}{2}) \;=\; (\delta'/2)\cdot d. \tag{3.33}$$

For the last equation, recall that $d = n^2-1$. The bound in Equation 3.33 contradicts Lemma 3.3.8. Therefore we conclude that A_n is $(\epsilon\cdot n, n^{1+\delta})$-rigid for any $\delta < 1$, where $\epsilon := \delta'/8$ (remember δ' was fixed at the beginning). □

Remark 3.3.4 1. The matrix A_n is not only poly(n)-explicit, it is 'very' explicit in the good common sense: one could consider them as simple as binomial coefficients, recorded one row at a time. This is quite interesting given the recent dramatic developments that have killed virtually all known candidates.

2. Our proof requires the upper bound of $O(S(f))$ in Conjecture 3.1.2; any weaker upper bound *does not* yield the same rigidity parameter. Also, the same proof holds over $\mathbb{C}$, using Lemma 3.9.2.

3. One can also work with $f_d = (x+1)^d$, but one needs to use Lemma 3.9.1.

3.3.4 From SOS-τ-conjecture to VP $\neq$ VNP

Theorem 3.3.9 *Conjecture 3.1.2 implies that* $\mathsf{VNP}_{\mathbb{C}}$ *is exponentially harder than* $\mathsf{VP}_{\mathbb{C}}$.

Proof. We will construct an explicit (multivariate) polynomial family from the univariate $f_d := \sum_{i=0}^{d} 2^{2i(d-i)} \cdot x^i$, and show that it requires *exponential size* circuit (assuming Conjecture 3.1.2). Moreover, we show that the family is in VNP, and the conclusion would directly follow. The hardness and VNP-inclusion proof are 'specialized' versions of the proof of Theorem 3.3.6 in subsection 3.3.2. However for the sake of completeness, we present the detailed version.

Kurtz condition. We show that the coefficients $a_i := 2^{2i(d-i)}$ satisfies the Kurtz condition (Theorem 2.2.4). For that, it suffices to check that

$$4\,i\,(d-i) \;>\; 2 \;+\; 2\,(i-1)\,(d-i+1) \;+\; 2\,(i+1)\,(d-i-1)\,,$$

which is true since LHS - RHS=2. Therefore, roots of f_d are all distinct and real.

Construction. We will construct $(P_{n,k})_n$ from f_d, where $P_{n,k}$ is a multilinear degree-n and kn-variate polynomial, where k is a fixed constant (to be fixed in Lemma 3.3.11), and $n = O(\log d)$; thus $kn = O(\log d)$.

The basic relation between d, n and k is that $k^n \;\geq\; d+1 \;>\; (k-1)^n$. Introduce kn many new variables $y_{j,\ell}$, where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$. Let $\phi_{n,k}$ be the map,

$$\phi_{n,k} \;:\; x^i \;\mapsto\; \prod_{j=1}^{n} y_{j,i_j} \;, \text{where } \; i \;=:\; \sum_{j=1}^{n} i_j k^{j-1}, \;\; 0 \;\leq\; i_j \;\leq\; k-1\,.$$

For $i \in [0,d]$, $\phi_{n,k}$ maps x^i uniquely to a multilinear monomial of degree n. By linear extension, define $\phi_{n,k}(f_d) =: P_{n,k}$. By construction, $P_{n,k}$ is n-degree, kn-variate multilinear polynomial. Let $\psi_{n,k}$ be the homomorphism that maps any degree-n multilinear monomial, defined on variables $y_{j,\ell}$, such that $y_{j,\ell} \mapsto x^{\ell \cdot k^{j-1}}$. Trivially, $\psi_{n,k} \circ \phi_{n,k}(f) = f$, for any degree $\leq d$ polynomial $f \in \mathbb{C}[x]$.

Lemma 3.3.10 $(P_{n,k})_n \in \mathsf{VNP}$.

Proof. By construction, $P_{n,k}$ is a kn-variate, individual degree-n multilinear polynomial. Hence,

$$P_{n,k} = \sum_{e \in \{0,1\}^{kn}} \gamma(e) \cdot y^e \,.$$

Here, y denotes the kn variables $y_{j,\ell}$ where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$, and e denotes the exponent-vector. As each x^e in $\mathrm{supp}(f_d)$ maps to a monomial y^e *uniquely*; given e, one can easily compute $e := \sum_{j=1}^{n} e_j \cdot k^{j-1}$, and thus $\gamma(e) = \mathrm{coef}_{x^e}(f_d) = 2^{2e(d-e)}$. Note that, $\gamma(e) < 2^{d^2}$, for all e. We also remark that each bit of $\gamma(e)$ is computable in $\mathrm{poly}(\log d) = \mathrm{poly}(kn)$-time.

Write each $\gamma(e)$ in binary, i.e. $\gamma(e) =: \sum_{j=0}^{d^2-1} \gamma_j(e) \cdot 2^j$, where $\gamma_j(e) \in \{0,1\}$ is computable in P. As $d^2 - 1 < k^{2n}$, introduce new variables $z = (z_1, \ldots, z_m)$, where $m := 2n \log k = O(n)$ [so that, $d^2 - 1 \leq 2^m - 1$]; and consider the auxiliary polynomial $\tilde{\gamma}(e,z) := \sum_{j \in \{0,1\}^m} \gamma_j(e) \cdot z^{\mathrm{bin}(j)}$. Here, we identify $j \in [0, 2^m - 1]$ as a unique $j \in \{0,1\}^m$, via $\mathrm{bin}(j)$,i.e. $\gamma_j = \gamma_j$. Let $z_0 := (2^{2^0}, \ldots, 2^{2^{m-1}})$. Note that, $\tilde{\gamma}(e, z_0) = \gamma(e)$. Finally, consider the $(m+kn)$-variate (where $m + kn = O(n)$) auxiliary polynomial $h_{n,k}(y,z)$ as:

$$h_{n,k}(y,z) := \sum_{e \in \{0,1\}^{kn}} \tilde{\gamma}(e,z) \cdot y^e = \sum_{e \in \{0,1\}^{kn}} \sum_{j \in \{0,1\}^m} \gamma_j(e) \cdot z^j \cdot y^e \,.$$

Then, we have $h_{n,k}(y, z_0) = P_{n,k}(y)$. Since each bit $\gamma_j(e)$ is computable in P, thus by Valiant's criterion (Theorem 2.4.3), we have $(h_{n,k}(y,z))_n \in \mathsf{VNP}$. As VNP is *closed* under substitution, it follows that $(P_{n,k}(y))_n \in \mathsf{VNP}$. □

Next we show that $P_{n,k}$ is exponentially hard assuming Conjecture 3.1.2.

Lemma 3.3.11 *Conjecture 3.1.2 implies $P_{n,k}$ requires exponential size circuit.*

Proof. We show that over $\mathbb{C}$, size of the minimal circuit computing $P_{n,k}$, namely $\text{size}(P_{n,k}) > d^{1/7} = 2^{\Omega(kn)}$. If not, then apply Lemma 3.3.4 to conclude that

$$P_{n,k} = \sum_{i=1}^{s} c_i \cdot Q_i^2 \implies f_d = \sum_{i=1}^{s} c_i \cdot \psi_{n,k}(Q_i)^2 ,$$

where, $\deg(Q_i) \leq 2n/3$, and $s = O(d^{1/7} \cdot n^2)$. Above equation implies: $S_{\mathbb{C}}(f_d) \leq s \cdot \binom{kn+2n/3}{2n/3}$. We want to show that $S_{\mathbb{C}}(f_d) \leq o(d)$, this will contradict Conjecture 3.1.2. This is because the coefficients of f_d satisfies the Kurtz condition implying f_d has all distinct real roots, then Conjecture 3.1.2 implies that $S_{\mathbb{R}}(f_d) \geq \Omega(d) \implies S_{\mathbb{C}}(f_d) \geq \Omega(d)$, from Lemma 3.9.2.

By assumption, $s \leq O(d^{1/7} \cdot \log^2 d)$. It suffices to show that $\binom{kn+2n/3}{2n/3} \leq d^{5/7}$, so that $S(f_d) \leq O(d^{6/7} \cdot \log^2 d) = o(d)$, the desired contradiction. Use Equation 2.6 to show the upper bound on the binomial:

$$\begin{aligned}\binom{kn + 2n/3}{2n/3} \leq (e + 3ek/2)^{2n/3} &\leq (5(k-1))^{2n/3} \\ &\leq (k-1)^{5n/7} \leq d^{5/7} .\end{aligned}$$

The second inequality holds for $e + 3ek/2, \leq 5(k-1)$; so $k \geq 9$ suffices. For the third inequality to be true, $(k-1)^{5/7} \geq (5(k-1))^{2/3}$ suffices; this holds true for $(k-1)^{1/21} \geq 5^{2/3} \iff k \geq 5^{14} + 1$. We also used $d \geq (k-1)^n$ (by assumption). □

Both the above Lemma 3.3.10-3.3.11 imply the desired conclusion. □

Remark 3.3.5 1. As $\deg(Q_i) \leq 2n/3$, we have

$$\begin{aligned}\deg(\psi_{n,k}(Q_i)) &\leq 2n/3 \cdot (k-1) \cdot k^{n-1} \\ &< n.k^n = O(nd) \\ &= O(d \log d) .\end{aligned}$$

Thus, it is enough to consider the restricted-degree SOS representation, and prove the conjecture.

2. One could directly obtain that Conjecture 3.1.2 implies $S_{\mathbb{R}}(f_d) \geq \Omega(d)$, where $f_d := \prod_{i=1}^{d}(x-i)$. However, to separate VP and VNP, using the proof techniques of [DST21] (with $\varepsilon = 1$) *require* GRH (Generalized Riemann Hypothesis).
3. To show an unconditional lower bound, we work with $f_d := \sum_{i=0}^{d} 2^{2i(d-i)} \cdot x^i$ (a similar family was considered in [GKT15, Equation 8]). However, the hardness proof is completely different from [GKT15], due to disparate settings and parameters.

3.4 Sum-of-Cubes

In this section, let $\mathbb{F}$ be a field of characteristic $\neq 2, 3$. The following lemma is the crucial ingredient to connect general circuits to a SOC-representation. It is similar to Lemma 3.3.5. There, we represented a polynomial p as a sum of squares of polynomials with degree close to $1/2$. Now, we write p as a sum of cubes of polynomials with degree close to $1/3$.

Lemma 3.4.1 (SOC decomposition) *There exists a constant c, such that for any n-variate homogeneous polynomial $p \in \mathbb{F}[x]$ of degree d that can be computed by a homogeneous circuit of size s, we have a representation*

$$p = \sum_{i=1}^{s^c} q_i^3, \tag{3.34}$$

where $q_i \in \mathbb{F}[x]$, for all $i \in [s^c]$, such that

1. *$\deg(q_i) < \frac{4}{11} d$,*
2. *q_i has a homogeneous circuit of size $O(s)$.*

Proof. We start exactly as in the proof of Lemma 3.3.5, with parameters $\gamma = 4/11$ and $\delta = \gamma - 1/3 = 1/33$. Then we choose m such that $(2/3)^m < \delta$. Hence, we can set $m = 9$ and we can write p as in Equation 3.22:

$$p = \sum_{i=1}^{s^{2^m-1}} \hat{p}_{i,1}\, \hat{p}_{i,2} \cdots \hat{p}_{i,2^m}, \tag{3.35}$$

where $\deg(\hat{p}_{i,j}) \leq (2/3)^m d < \delta d$. For all $i \in [s^{2^m-1}]$, we have $\sum_{j=1}^{2^m} \deg(\hat{p}_{i,j}) = d$ and $\text{size}(\hat{p}_{i,j}) = O(s)$, for all $j \in [2^m]$.

In Lemma 3.3.5, we split each product $\hat{p}_{i,1} \cdots \hat{p}_{i,2^m}$ into two parts of degree close to $d/2$. Now, we similarly split it into three parts of degree close to $1/3$. So we first pick the smallest $j_0 \in [2^m]$ such that

$$\frac{d}{3} \leq \sum_{j=1}^{j_0} \deg(\hat{p}_{i,j}) < \gamma d,$$

and define $p_{i,1} = \hat{p}_{i,1} \cdots \hat{p}_{i,j_0}$. Then we pick the smallest j_1, where $j_0 < j_1 \leq 2^m$, such that

$$\frac{d}{3} \leq \sum_{j=j_0+1}^{j_1} \deg(\hat{p}_{i,j}) < \gamma d,$$

and define $p_{i,2} = \hat{p}_{i,j_0+1} \cdots \hat{p}_{i,j_1}$ and $p_{i,3} = \hat{p}_{i,j_1+1} \cdots \hat{p}_{i,2^m}$. Then we have

$$p = \sum_{i=1}^{s^{2^m-1}} p_{i,1}\, p_{i,2}\, p_{i,3}\,, \tag{3.36}$$

where $d/3 \leq \deg(p_{i,j}) < \gamma d$, for all $i \in [s^c]$ and $j = 1, 2, 3$.

Finally, we write the products in Equation 3.36 as sums of cubes by the following identity:

$$24abc = (a+b+c)^3 - (a-b+c)^3 - (a+b-c)^3 + (a-b-c)^3\,. \tag{3.37}$$

□

Remark 3.4.1 In case of non-homogeneous polynomials, we can consider the homogeneous parts separately. The size s has then again to by replaced by $O(sd^2)$.

We now come to the main result of this section, that the existence of a SOC-hard family implies the derandomization of blackbox PIT. The proof outline is roughly similar to the proof of Theorem 3.3.2, but with some crucial modifications. Given a SOC-hard polynomial $f_d(x)$, we apply the standard Kronecker map to construct a polynomial $P_{n,k}$ that is k-variate, for some constant k, and the variables have individual degree n. We show that $\text{size}(P_{n,k}) = n^{\Omega(1)}$.

The proof of the size lower bound goes again by contradiction, and this is where Lemma 3.4.1 comes into the play. Via the

SOC decomposition of $P_{n,k}$ and the inverse Kronecker map, we get a SOC-representation of f_d that would be smaller than the assumed SOC-hardness of f_d.

Thus $P_{n,k}$ fulfills the assumptions made in Theorem 2.7.2, by [Guo+19], and we can conclude that blackbox PIT $\in$ P .

6: The coefficients are computable in poly(d)-time. This is a much more lenient condition, because we may use $f_d = \prod(x - i)$, or, $f_d = (x + 1)^d$, as candidate SOC-hard families without requiring GRH! This occurs due to the fundamental difference between the VP $\neq$ VNP and PIT $\in$ P proofs, where the former translates the SOS-hard polynomial into an $O(\log d)$-variate polynomial and the latter into a *constant* variate polynomial.

Theorem 3.4.2 *If there is a* poly(d)*-explicit*[6] *SOC-hard family* $(f_d)_d$*, then blackbox-PIT* $\in$ P.

Proof. Let $f_d(x)$ be an explicit SOC-hard polynomial such that $U(f_d, d^\varepsilon) \geq \delta d$, for constants $0 < \varepsilon < 1/2$ and $\delta > 0$. Let furthermore c be the constant from Lemma 3.4.1.

We define parameters k and n as follows. Let $\alpha = 1 - \frac{1}{110}$. Choose k large enough such that

$$k > \frac{9c}{\varepsilon} \quad \text{and} \quad \alpha^k < \delta, \tag{3.38}$$

and define $n = \lceil (d+1)^{1/k} \rceil - 1$. Now we apply the Kronecker map $\phi_{n,k}$ from Equation 2.2 to f_d and define polynomial

$$P_{n,k}(y) = \phi_{n,k}(f_d(x)) .$$

Recall that $P_{n,k}$ has k variables of individual degree n, and therefore total degree kn. Since f_d is explicit, we have $P_{n,k} \in$ VNP.

Define μ as

$$\mu = \frac{1}{2}\left(\frac{\varepsilon}{c} - \frac{1}{k}\right). \tag{3.39}$$

Note that $\mu > 0$ by our choice of k in Equation 3.38.

Claim 3.4.1 (Hardness of $P_{n,k}$) size$(P_{n,k}) > d^\mu$, for large enough n.

Proof. Assume to the contrary that size$(P_{n,k}) \leq d^\mu$. By Lemma 3.4.1, there exist polynomials Q_i such that $P_{n,k} = \sum_{i=1}^{s_0} c_i Q_i^3$, where $s_0 \leq (d^\mu kn)^c$ and $\deg(Q_i) \leq \frac{4}{11} kn$.

We apply the inverse Kronecker map $\psi_{n,k}$ to the polynomials Q_i: Define $g_i(x) = \psi_{n,k}(Q_i(y))$. Then we get

$$f_d = \sum_{i=1}^{s_0} c_i g_i^3 .$$

Recall that g_i and Q_i have the same sparsity. Therefore

$$s_1 = \left|\bigcup_i \operatorname{supp}(g_i)\right| \leq \left|\bigcup_i \operatorname{supp}(Q_i)\right| \leq \binom{k + \frac{4}{11}kn}{k}.$$

Thus, $U(f_d, s_0) \leq s_1$.

We want to show that $s_0 < d^{\varepsilon'}$ and $s_1 < \delta d$, for large enough n. Then, we have $U(f_d, d^{\varepsilon}) < \delta d$, for large enough d, which contradicts the SOC-hardness of f_d.

Bound on s_0 Recall that $d = (n+1)^k - 1 > n^k$, for large n. Therefore we get for large enough n, and thus d,

$$s_0 \leq (d^{\mu} kn)^c < (d^{\mu} k d^{\frac{1}{k}})^c = (k d^{\mu + \frac{1}{k}})^c < d^{\varepsilon}. \tag{3.40}$$

In the last inequality we used that $\mu + 1/k < \varepsilon/c$, by the definition of μ.

Bound on s_1 By Equation 2.6, we have

$$s_1 = \binom{k + \frac{4}{11}kn}{k} \leq \left(e\left(1 + \frac{4}{11}n\right)\right)^k < (\alpha n)^k < \alpha^k d < \delta d. \tag{3.41}$$

As $4e \approx 10.873$, we used that $e\left(1 + \frac{4}{11}n\right) < \alpha n$ and $d > n^k$, for large n. The last inequality is by our choice of k. This proves Claim 3.4.1. □

It remains to show that from the hardness of $P_{n,k}$, the assumption in Theorem 2.7.2 can be fulfilled, that $\operatorname{size}(P_{n,k}) > s^{10k+2}\deg(P_{n,k})^3$, for some growing function $s = s(n)$. Recall that $\deg(P_{n,k}) \leq kn$. We define, $s(n) = n^{\frac{1}{10k+3}}$. Then we have

$$s^{10k+2}(kn)^3 = n^{\frac{10k+2}{10k+3}}(kn)^3 = k^3 n^{4 - \frac{1}{10k+3}} < n^4, \tag{3.42}$$

for large enough n. By the first condition in our choice of k in Equation 3.38, we have

$$\mu = \frac{1}{2}\left(\frac{\varepsilon}{c} - \frac{1}{k}\right) \geq \frac{1}{2}\left(\frac{9}{k} - \frac{1}{k}\right) = \frac{4}{k},$$

and therefore $k\mu \geq 4$. Recall also that $n^k < d$, for large n. Hence, we can continue Equation 3.42 as

$$n^4 \leq n^{k\mu} < d^{\mu} < \operatorname{size}(P_{n,k}). \tag{3.43}$$

Equation 3.42) and Equation 3.43 give the desired hardness of $P_{n,k}$. Thus, Theorem 2.7.2 gives a poly(s)-time HSG for $\mathscr{C}(s,s,s)$. Hence, blackbox PIT $\in$ P. □

Remark 3.4.2 The degree of the Q_i's in the above proof is bounded by $\frac{4}{11}kn$. Hence, the degree of the g_i's obtained via the inverses Kronecker substitution is bounded by

$$(n+1)^{k-1}\,\frac{4}{11}\,kn \;<\; \frac{4}{11}\,k(n+1)^k \;\leq\; \frac{4}{11}\,k(d+1) \;=\; O(d)\,,$$

where the last equality is because k is a constant. Thus, it suffices to study the representation of f_d as sum-of-cubes g_i^3, where $\deg(g_i) = O(d)$, and still Theorem 3.4.2 would hold.

3.5 Sum-of-Constant-Powers (SOCP)

A brief history of struggle: We started with $r = 25$, and showed that a (really) large lower bound on $S_{\mathbb{F}}(f,25)$ separates VP $\neq$ VNP [DST20]. It was then improved to $r = 4$ with *slightly relaxed* requirement on the bound [DS20]. Finally, we settled for $r = 2$, with the requirement on the lower bound tantalizingly close to the *trivial* one!

In this section, let $\mathbb{F}$ be a field of characteristic 0 or large. Similar to SOS or SOC model, one can study higher constant powers; the model is to represent a given univariate polynomial as a *sum-of-constant-powers* (SOCP), namely, writing polynomial $f(x)$ as

$$f(x) \;=\; \sum_{i=1}^{s} c_i f_i^r$$

for some top-fanin s, constant r, polynomials f_i and coefficients c_i. This representation corresponds to a depth-four circuit that has the form $\Sigma^{[s]}\wedge^{[r]}\Sigma\Pi$. One can similarly define the support-sum size $S_{\mathbb{F}}(f,r)$, and support-union size $U_{\mathbb{F}}(f,r,s)$[7].

7: Recall support-union measure has an extra s parameter which denotes the top-fanin.

By trivial monomial counting, $S_{\mathbb{F}}(f,r), U_{\mathbb{F}}(f,r,s) \geq \mathrm{sp}(f)^{1/r}$. To upper bound $S_{\mathbb{F}}(f,r)$, we prove the following.

Lemma 3.5.1 *Let $\mathbb{F}$ be a field of characteristic 0 or large. Let $h(x) \in \mathbb{F}[x]$ and $0 \leq m \leq r$. There exist $c_{m,i} \in \mathbb{F}$ and* distinct *$\lambda_i \in \mathbb{F}$, for $0 \leq i \leq r$, such that*

$$h(x)^m \;=\; \sum_{i=0}^{r} c_{m,i}\,(h(x)+\lambda_i)^r\,. \tag{3.44}$$

Proof. Consider the polynomial $(h(x)+t)^r$, where t is a new indeterminate different from x. We have

$$(h(x)+t)^r \;=\; \sum_{i=0}^{r} \binom{r}{i} h(x)^i\, t^{r-i}\,.$$

Choose $r+1$ many distinct λ_i's and put $t=\lambda_i$, for $i=0,1,\ldots,r$. We get $r+1$ many linear equations which can be represented in matrix form $Av=b$, for matrix $A=\left(\binom{r}{j}\lambda_i^{r-j}\right)_{0\le i,j\le r}$, and vectors $v=\left(h^i\right)_{0\le i\le r}$ and $b=\left((h+\lambda_i)^r\right)_{0\le i\le r}$.

Note that except for the binomial factors, A is a Vandermonde matrix. When computing the determinant, one can pull out the binomial factor $\binom{r}{j}$ from the j-th column, for $j=0,1,\ldots,r$. Then a Vandermonde matrix remains, and hence

$$\det(A)=\prod_{j=0}^{r}\binom{r}{j}\prod_{0\le i<j\le r}(\lambda_j-\lambda_i)\neq 0\,.$$

Therefore, A is invertible and we have $v=A^{-1}b$. Let c_m be the $(m+1)$-th row of A^{-1}. Then we have $h(x)^m = c_m b$ which is exactly Equation 3.44. □

SOS is the *easiest* model

The SOCP-hardness (wrt. both support-sum and support-union size), can be defined in the same way as the SOS-, or SOC-hardness, using the same parameters, and shown to have identical complexity theoretic consequences.

Due to Equation 3.44, if a polynomial f has a small SOS- or, SOC-representation, the support-sum, or the support-union in the sum of r-th powers changes insignificantly. When h^2, for a univariate polynomial h, is written as a sum of r-th powers changes the top-fanin (and hence the support-sum) by a factor of $(r+1)$, which is a constant. While the adjustment is additive for the support-union! Therefore, proving hardness in the higher power model is more *difficult* than showing results in the SOS! In this sense, our SOS model is really *optimal* to work with.

3.5.1 Strong lower bound over $\mathbb{Z}$

The SOS-hardness of $(x+1)^d$ over $\mathbb{F} = \mathbb{Q}$ is open, however we show a *strong* lower bound over localized integer rings (e.g. $\mathbb{Z}$) for the same, giving substantial evidence for the hardness being provably true. [8] For the algebraic number theory terms, see [Lan13].

8: In many cases, proving lower bound over $\mathbb{Z}$, could already be very challenging. It is conjectured that perm_n, over $\mathbb{Z}$, *cannot* be computed by a small determinant. Moreover, this is already implied by assuming a stronger (*nonuniform*) version of the $\mathsf{P} \neq \mathsf{NP}$ conjecture, namely $\mathsf{NP} \not\subseteq \mathsf{P/poly}$, or even the *weaker* $\#\mathsf{P} \not\subseteq \mathsf{NC}$. For a detailed discussion, we refer to [Mul12].

For any number field K, let $\mathcal{O}_K$ be the *ring of integers* in K, e.g. $\mathbb{Z}$ in $\mathbb{Q}$. Let $\mathbb{P}$ be a prime ideal of $\mathcal{O}_K$, e.g. $\langle p \rangle$ of $\mathbb{Z}$. Define the *localization* $(\mathcal{O}_K)_{\mathbb{P}} := \{r/s \mid r, s \in \mathcal{O}_K, s \notin \mathbb{P}\}$ which is a domain larger than $\mathcal{O}_K$, e.g. $\mathbb{Z}_{\langle p \rangle}$; it has all fractions except the ones like $1/p$. We show the hardness (wrt. support-union measure) over $R := (\mathcal{O}_K)_{\mathbb{P}}$, whenever $\mathbb{P} \mid \langle r \rangle_{\mathcal{O}_K}$ (equivalently $\mathbb{P} \supseteq \langle r \rangle_{\mathcal{O}_K}$).

Theorem 3.5.2 (Unconditional lower bound) *Fix a prime-power r, any $s \geq 1$, and $f_d(x) := (x+1)^d$. Fix a number field K and its prime ideal $\mathbb{P}$ such that $\mathbb{P} \mid \langle r \rangle_{\mathcal{O}_K}$. Then, $U_{(\mathcal{O}_K)_{\mathbb{P}}}(f_d, r, s) > d^9$, for infinitely many d.*

9: The same proof also works for support-sum measure.

Remark 3.5.1

1. he lower bound of $d+1$ is the *strongest* one could possibly achieve. This suggests that constants like $1/r \in \mathbb{Q} = \mathbb{F}$ may help a bit in writing as sum-of-r-th-powers.
2. The flow of all our proofs is such that proving the hardness for infinitely many d (over $\mathbb{Q}$) actually suffices to separate $\mathsf{VP} \neq \mathsf{VNP}$. Thus, 'infinitely many' is *not a weakness* at all.

The main technical lemma is a celebrated theorem due to Lucas [Luc78].

Theorem 3.5.3 (Lucas's Theorem, [Luc78]) *For $m, n \in \mathbb{N}$ and a prime p, let*

$$m = m_k p^k + m_{k-1} p^{k-1} + \ldots + m_1 p + m_0$$
$$n = n_k p^k + n_{k-1} p^{k-1} + \ldots + n_1 p + n_0$$

be the base-p representation of m and n. Then

$$\binom{m}{n} \equiv \prod_{i=0}^{k} \binom{m_i}{n_i} \pmod{p}.$$

Lemma 3.5.4 *Let $d = r^\ell - 1$, for some prime r and $\ell \in \mathbb{N}$. Then*

$$(x+1)^d \equiv \sum_{k=0}^{d} (-1)^k x^k \pmod{r}. \tag{3.45}$$

Therefore, we have for the support size $|(x+1)^d \bmod r|_1 = d+1$.

Proof. The base-r representation of d is $d = \sum_{i=0}^{\ell-1}(r-1)\,r^i$. Let $0 \le k \le d$ and write k in base-r representation, $k = \sum_{i=0}^{\ell-1} k_i r^i$.

By Lucas's Theorem, we have

$$\binom{d}{k} \equiv \prod_{i=0}^{\ell-1} \binom{r-1}{k_i} \pmod{r}. \tag{3.46}$$

Now observe that $\binom{r-1}{k_i} \equiv (-1)^{k_i} \pmod{r}$. This is because

$$(r-1)(r-2)\cdots(r-k_i) \;\equiv\; (-1)^{k_i} k_i! \pmod{r},$$

and hence

$$\binom{r-1}{k_i} \;\equiv\; \frac{(-1)^{k_i} k_i!}{k_i!} \equiv (-1)^{k_i} \pmod{r}.$$

Plugging this into Equation 3.46, we get

$$\binom{d}{k} \;\equiv\; (-1)^{\sum_{i=0}^{\ell-1} k_i}.$$

Finally observe that $k = \sum_{i=0}^{\ell-1} k_i r^i \equiv \sum_{i=0}^{\ell-1} k_i \pmod{2}$, because r is odd. This proves Equation 3.45. □

Proof of Theorem 3.5.2. Let r be a power of a prime r_0 and $d = r^\ell - 1$, for some $\ell \in \mathbb{N}$.

By Lemma 3.5.4, we have $|(x+1)^d \bmod r_0|_1 = d+1$. Moreover, r_0 does not divide any of the coefficients $\binom{d}{k}$ because $\binom{d}{k} \equiv (-1)^k \pmod{r_0}$, for any $0 \le k \le d$.

Consider the given *prime* ideal $\mathbb{P}$ of $\mathcal{O}_K$ that contains $\langle r \rangle_{\mathcal{O}_K}$, and hence contains $\langle r_0 \rangle_{\mathcal{O}_K}$. Suppose $\binom{d}{j} \in \langle r_0 \rangle_{(\mathcal{O}_K)_\mathbb{P}}$, for some $0 \le j \le d$. Then, simply by ideal definition, there exists $m \in (\mathcal{O}_K)_\mathbb{P}$ such that $\binom{d}{j} = m r_0$. Since r_0 does not divide $\binom{d}{j}$ and $r_0 \in \mathbb{P}$, the quotient $\binom{d}{j}/r_0$ cannot lie in the localization $(\mathcal{O}_K)_\mathbb{P}$, which is a contradiction.

Thus, $\binom{d}{j} \notin \langle r_0 \rangle_{(\mathcal{O}_K)_{\mathbb{P}}}$, for all $0 \leq j \leq d$. Whence,

$$\begin{aligned} f_d(x) = \sum_{i\in[s]} c_i\, \ell_i^r \implies f_d(x) &\equiv \sum_{i\in[s]} c_i\, \ell_i(x^r) \bmod \langle r_0 \rangle_{(\mathcal{O}_K)_{\mathbb{P}}} \\ &\implies \Big|\bigcup_{i\in[s]} \operatorname{supp}(\ell_i(x^r))\Big| \geq d+1 \\ &\implies \Big|\bigcup_{i\in[s]} \operatorname{supp}(\ell_i)\Big| \geq d+1, \end{aligned}$$

which gives a lower bound on the support-union size as promised. □

Remark 3.5.2

1. The fact that $\mathbb{P}$ is a *prime ideal* is crucial in the above proof. This proof works for the polynomial $g := \sum_{i=0}^{d} 2^{i^2} x^i$ as well, as long as r is odd. This is simply because $2^{i^2} \neq 0 \bmod r_0$, for any odd prime r_0. The rest of the proof remains unchanged. For even r (say $r = 2$), one can work with the alternative $h := \sum_{i=0}^{d} 3^{i^2} x^i$.
2. This also proves that for any prime-power r, for any integer m coprime to r, and for all d of the form $r^\ell - 1$, we have $U_{\mathbb{Z}}(mf_d, r, \cdot) > d$. This behavior changes when m, r are *not* coprime.

3.6 Sum-of-Constant-Powers with *Small* Support

In this section, we talk about different constructions of 'small support-union' representations of any univariate polynomial in different sum-of-constant-power models.

3.6.1 Upper bounding $U_{\mathbb{F}}(f, r, s)$ with *large* s

Let us ask the following question. Can we write f as sum-of-constant-powers with the support-union as close as the lower bound $d^{1/r}$? In this subsection, we show that indeed it is possible when we can *relax* s to be as large as $\Omega(d)$; see Corollary 3.6.2!

Minkowski sum. Here we use the notion of sumsets. In additive combinatorics, the *sumset*, also called the *Minkowski sum* of two subsets A and B of an abelian group G, is defined to be the set of all sums of an element from A with an element from B,

$$A + B = \{a + b \mid a \in A,\ b \in B\}.$$

The *n-fold iterated sumset* of A is $nA = A + \cdots + A$, where there are n summands.

We want a *small support-union* representation of a d-degree polynomial f as a sum of r-th powers, where r is constant.

Let t be the unique non-negative integer such that $(t-1)^r < d+1 \leq t^r$. Define set B as

$$B = \{a t^\ell \mid 0 \leq a \leq t-1 \text{ and } 0 \leq \ell \leq r-1\}.$$

Hence $|B| = rt = O(d^{1/r})$. Let $k \in \{0, 1, \ldots, d\}$. The base-$t$ representation of k is a sum of at most r elements from B. Hence, $\{0, 1, \ldots, d\} \subseteq rB$. The largest element in B is $m = (t-1)t^{r-1} = O(d)$. Since r is a constant, the largest element in rB is $rm = O(d)$.

We show next that any polynomial can be written as a sum of r-th powers of polynomials with support in B.

Theorem 3.6.1 *For any $f \in \mathbb{F}[x]$ of degree d, there exist $\ell_i \in \mathbb{F}[x]$ with $\mathrm{supp}(\ell_i) \subseteq B$ and $c_i \in \mathbb{F}$, for $i = 0, 1, \ldots, mr$, such that $f = \sum_{i=0}^{mr} c_i \ell_i^r$.*

Proof. Let us set up the polynomials ℓ_i we seek as

$$\ell_i(x) = \sum_{j \in B} a_{i,j} x^j,$$

for unknown coefficients $a_{i,j} \in \mathbb{F}$, for $i = 0, 1, \ldots, rm$ and $j \in B$. We determine the $a_{i,j}$'s via the multivariate polynomial

$$L_i(z_i, x) = \sum_{j \in B} z_{i,j} x^j,$$

where we replaced the coefficients of ℓ_i by distinct indeterminates $z_{i,j}$.

Note that $\deg_x(L_i) \le m$. Taking the r-th power, we can write

$$L_i^r \;=\; \sum_{j=0}^{mr} Q_j(z_i)\, x^j\,,$$

for $0 \le i \le rm$, for polynomials Q_j of degree r with $|B| = rt$ many variables, $0 \le j \le rm$.

Let $S = \{\, j \mid Q_j \neq 0 \,\} \subseteq \{0, 1, \ldots, mr\}$. Note that from any monomial in Q_j we can recover j. This follows because $\mathrm{supp}(Q_{j_1}) \cap \mathrm{supp}(Q_{j_2}) = \emptyset$, for any $j_1 \neq j_2$ in S. Therefore, the polynomials $\{\, Q_j \mid j \in S \,\}$ are $\mathbb{F}$-linearly independent.

Note that by the definition of B, we have $\{0, 1, \ldots, d\} \subseteq S$.

We want to find $c = \begin{pmatrix} c_1 & c_2 & \cdots & c_{|S|} \end{pmatrix} \in \mathbb{F}^{|S|}$ and $a = (a_{i,j})_{i,j}$ such that

$$f(x) \;=\; \sum_{i=0}^{mr} c_i\, \ell_i^r(x) = \sum_{i=0}^{mr} c_i\, L_i^r(a, x)\,. \tag{3.47}$$

Let $f(x) = \sum_{i=0}^{d} f_i\, x^i$. We set up a linear system to determine the unknowns. Define the coefficient vector $\overline{f}$ of f over S and a $|S| \times |S|$-matrix A as

$$\overline{f} = \begin{pmatrix} f_0 & f_1 & \cdots & f_d & 0 & \cdots & 0 \end{pmatrix}, \tag{3.48}$$

$$A = \begin{pmatrix} Q_{j_1}(z_1) & Q_{j_2}(z_1) & \cdots & Q_{j_s}(z_1) \\ Q_{j_1}(z_2) & Q_{j_2}(z_2) & \cdots & Q_{j_s}(z_2) \\ \vdots & \vdots & \cdots & \vdots \\ Q_{j_1}(z_{|S|}) & Q_{j_2}(z_{|S|}) & \cdots & Q_{j_s}(z_{|S|}) \end{pmatrix}. \tag{3.49}$$

Then Equation 3.47 is equivalent to

$$c\, A(a) = \overline{f}.$$

As the z_i's are distinct variables, the first column of A consists of different variables at each coordinate. Moreover, the first row of A contains $\mathbb{F}$-linearly independent Q_j's. Thus, for a random $a = (a_{i,j})$, matrix $A(a)$ has full rank over $\mathbb{F}$. Fix such an a. This yields $c = \overline{f}\,(A(a))^{-1}$. For these values c and a, we get Equation 3.47 as desired. □

Remark 3.6.1 1. The above calculation does *not* give small support-sum representation of f, as the top-fanin is already $\Omega(d)$.

2. The above representation crucially requires a *field* $\mathbb{F}$. E.g. it does not exist for f_d over the ring $\mathbb{Z}$.

The number of *distinct* monomials across the ℓ_j's in the above proof is $|B| = O(d^{1/r})$, while the top-fanin is $\leq mr + 1 = \Theta(d)$. Of particular interest for us are the cases $r = 2, 3$.

Corollary 3.6.2 *Any polynomial $f \in \mathbb{F}[x]$ of degree d has a SOS- and a SOC-representation with top-fanin $O(d)$ and support-union $O(\sqrt{d})$, respectively $O(\sqrt[3]{d})$.*

In the following, we improve Theorem 3.6.1 for $r = 2, 3$. We show a SOS- and SOC-representation for any polynomial $f(x)$, wherein both the top-fanin *and* the support-union size are small, namely $O(\sqrt{d})$. We assume that characteristic of $\mathbb{F}$ is $\neq 2$ in case of SOS, and $\neq 3$, in case of SOC. The representations are based on discussions with Agrawal [Agr20].

3.6.2 Constructing small SOS

By Corollary 3.6.2, any polynomial f of degree d has a SOS-representation with top-fanin $O(d)$ and support-union $O(\sqrt{d})$ We show that also the top-fanin can be reduced to $O(\sqrt{d})$. The technical key for this is the following lemma. It shows how to decrease the top-fanin in a representation without increasing the support-union.

Lemma 3.6.3 *Let $f \in \mathbb{F}[x]$ be written as $f = \sum_{i=1}^{s} c_i f_{i,1} f_{i,2}$, with support-union $t = |\bigcup_{i,j} \mathrm{supp}(f_{i,j})|$. Then there exists a representation $f = \sum_{i=1}^{t} c'_i f'_{i,1} f'_{i,2}$ with support-union $\leq t$.*

Proof. For the given representation of f, we assume w.l.o.g. that $\deg(f_{i,1}) \geq \deg(f_{i,2})$ and that $f_{i,1}, f_{i,2}$ are monic, for $i = 1, 2, \ldots, s$. Let $S = \bigcup_{i,j} \mathrm{supp}(f_{i,j})$.

We construct the representation claimed in the lemma by ensuring the following properties:

1. For every $x^e \in S$, there is exactly one i such that $\deg(f'_{i,1}) = e$,
2. $\bigcup_{i,j} \mathrm{supp}(f'_{i,j}) \subseteq S$,

Since we also maintain that $\deg(f'_{i,1}) \geq \deg(f'_{i,2})$, it follows that the top-fanin is indeed bounded by $t = |S|$ as claimed.

We handle the monomials in S successively according to decreasing degree. Let $x^e \in S$ be the monomial with the largest e that occurs more than once as the degree of a $f_{i,1}$, say $\deg(f_{1,1}) = \deg(f_{2,1}) = e$.

Define $g_1 = f_{2,1} - f_{1,1}$. Then we have $f_{2,1} = f_{1,1} + g_1$ and $\deg(g_1) < e$. Moreover, the support of g_1 is contained in the support of $f_{1,1}$ and $f_{2,1}$ If $\deg(f_{2,2}) = e$, then we define similarly $g_2 = f_{2,2} - f_{1,1}$. Then $f_{2,2} = f_{1,1} + g_2$ and $\deg(g_2) < e$. Now we can write

$$\begin{aligned} & c_1 f_{1,1} f_{1,2} + c_2 f_{2,1} f_{2,2} \\ = \; & c_1 f_{1,1} f_{1,2} + c_2 (f_{1,1} + g_1)(f_{1,1} + g_2) \\ = \; & f_{1,1} \left(c_1 f_{1,2} + c_2 f_{1,1} + c_2 g_1 + c_2 g_2 \right) + c_2 g_1 g_2 \; . \end{aligned} \tag{3.50}$$

The second line is a new sum of two products, where only the first product has terms of degree e, whereas in the second product, g_1, g_2 have smaller degree. Also, the support-union set has not increased.

In case when $\deg(f_{2,2}) < e$, we can just work with $f_{2,2}$ directly instead of $f_{1,1} + g_2$, and the above equations gets even simpler. □

So when we start with the SOS-representation for polynomial f provided by Theorem 3.6.1 and apply Lemma 3.6.3, It follows that f can be re-written as $f(x) = \sum_{i=1}^{O(\sqrt{d})} c'_i \, f_{i,1} \, f_{i,2}$, where $|\bigcup_{i,j} \operatorname{supp}(f_{ij})| = O(\sqrt{d})$. This can be turned into a SOS-representation by $f_{i,1} \, f_{i,2} = (f_{i,1} + f_{i,2})^2/4 - (f_{i,1} - f_{i,2})^2/4$. Note that the last step does not change the support-union, and at most doubles the top-fanin. Hence, we get

Theorem 3.6.4 (Small SOS-Representation) *Any polynomial $f \in \mathbb{F}[x]$ of degree d has a SOS-representation such that the top-fanin and the support-union are bounded by $O(\sqrt{d})$.*

3.6.3 Constructing small SOC

We show two small SOC-representation with different parameters. First, we show a $\sqrt{d}$ SOC-representation that follows essentially from Theorem 3.6.4. In particular, one can

use Equation 3.44, for $m = 2$ and $r = 3$, to rewrite a SOS-representation as a SOC-representation.

Observe that the support on both sides of Equation 3.44 is the same, except maybe for an extra constant term on the right hand side. Hence, for any given polynomial f, we can take the SOS-representation from Theorem 3.6.4 and rewrite each square as a sum of four cubes by Lemma 3.5.1. Then we get

Corollary 3.6.5 ($\sqrt{d}$ SOC-representation) *Any polynomial $f \in \mathbb{F}[x]$ of degree d has a SOC-representation such that the top-fanin and the support-union are bounded by $O(\sqrt{d})$.*

Remark 3.6.2 Recall Definition 3.1.4 that f_d is SOC-hard if $U_{\mathbb{F}}(f_d, d^{\varepsilon}) = \Omega(d)$, for some $0 < \varepsilon < 1/2$. Corollary 3.6.5 shows, that SOC-hardness is not possible for $\varepsilon = 1/2$.

The second way to get a small SOC-representation technically follows the way we got Theorem 3.6.4. We first show a reduction similar to Lemma 3.6.3 for the sum of product-of-3.

Lemma 3.6.6 *Let $f \in \mathbb{F}[x]$. If $f = \sum_{i=1}^{s} c_i\, f_{i,1} f_{i,2} f_{i,3}$, with support-union t, then there exists a representation of the form $f = \sum_{i=1}^{t^2} c'_i\, f'_{i,1} f'_{i,2} f'_{i,3}$ with support-union $\leq t$.*

Proof. The argument is similar to the proof of Lemma 3.6.3. For the given representation of f, we assume that $\deg(f_{i,1}) \geq \deg(f_{i,2}) \geq \deg(f_{i,3})$ and that $f_{i,1}, f_{i,2}, f_{i,3}$ are monic, for $i = 1, 2, \ldots, s$. Let $S = \bigcup_{i,j} \operatorname{supp}(f_{i,j})$.

Let $x^e \in S$ be the monomial with the largest e that occurs more than once as the degree of a $f_{i,1}$. W.l.o.g. assume $\deg(f_{1,1}) = e$. Write all the other $f_{i,j}$'s where x^e occurs as

$$f_{i,j} = f_{1,1} + g_{i,j}, \tag{3.51}$$

for $j \in [s]$ and $k \in [3]$. Note that $\deg(g_{i,j}) < e$.

Now we plug in Equation 3.51 in the representation of f given by assumption and multiply out. This gives

$$f = \sum_{i \in [m]} c_i\, f_{i,1} f_{i,2} f_{i,3} = f_{1,1}\, P + R, \tag{3.52}$$

where P is a sum of product-of-2 and R is a sum of product-of-3, where each intermediate polynomial has degree $< e$. Note that the last expression still has the same support-union.

Apply Lemma 3.6.3 on P, to reduce its top-fanin to t. Observe that then $f_{1,1}P$ has a sum of product-of-3 expression with top fanin at most t. Iterating the procedure to R, we finally get a representation of f with top fanin bounded by t^2. □

By Corollary 3.6.2, any polynomial f of degree d has a SOC-representation with top-fanin $O(d)$ and support-union $O(\sqrt[3]{d})$. By Lemma 3.6.6, this can be re-written as a sum product-of-3 with top-fanin $O(d^{2/3})$. Finally, any product-of-3 can be written as a sum of four cubes, by Equation 3.37. Hence, we get

Theorem 3.6.7 ($d^{2/3}$ SOC-representation) *Any polynomial $f \in \mathbb{F}[x]$ of degree d has a SOC-representation with top-fanin $O(d^{2/3})$ and support-union $O(d^{1/3})$.*

Finally, we observe that Lemma 3.5.1 also provides a connection between the two complexity measures $S(f)$ from SOS and $U(f, s)$ from SOC.

Lemma 3.6.8 *For any $f \in \mathbb{F}[x]$, we have*

$$S(f) \geq \min_s \left(U(f, 4s) - 1\right) .$$

Proof. Suppose $f = \sum_{i=1}^{s} c_i f_i^2$. By Lemma 3.5.1, each f_i^2 can be written as $f_i^2 = \sum_{j=1}^{4} c_{ij}(f_i + \lambda_{ij})^3$, for distinct $\lambda_{ij} \in \mathbb{F}$. Thus, $U(f, 4s) \leq 1 + \sum_{i=1}^{s} \mathrm{sp}(f_i)$. Taking minimum over s gives the desired inequality. □

Corollary 3.6.9 *For $s = \Omega(d^{2/3})$, we have $U(f, s) = \Theta(d^{1/3})$.*

3.7 Lower Bound for Restricted Models

Kumar and Volk [KV21] showed a strong connection between matrix rigidity and depth-2 linear circuit lower bound. They argued (similarly done in [Pud94] in a different language) that

depth-2 $\Omega(n^2)$ lower bound for an explicit matrix is *necessary* and *sufficient* for proving *super*-linear lower bound for general $O(\log n)$-depth circuits (or matrix rigidity).

Symmetric depth-2 circuit. Over $\mathbb{R}$, it is a circuit of the form $B^T \cdot B$, for some $B \in \mathbb{R}^{m \times n}$. [Over $\mathbb{C}$, one should take the conjugate-transpose B^* instead of B^T.] Symmetric circuits are a natural computational model for computing a positive semi-definite (*PSD*) matrix.

Invertible depth-2 circuit. It is a circuit $B \cdot C$, where at least one of the matrices B, C is invertible. We stress that invertible circuits can compute non-invertible matrices. Invertible circuits generalize many of the common matrix decompositions, such as QR decomposition, eigen decomposition, singular value decomposition (SVD), and LUP decomposition.

[KV21, Theorem 1.3 & Theorem 1.5] prove asymptotically optimal lower bounds for both the models.

Theorem 3.7.1 [KV21] *There exists an explicit family of real $n \times n$ PSD matrices $(A_n)_{n \in \mathbb{N}}$ such that every symmetric circuit (respectively invertible circuits) computing A_n (over $\mathbb{R}$) has size $\Omega(n^2)$.*

We present a simple, *alternative* proof of Theorem 3.7.1 using lower bounds on the SOS representation (with restriction) of two different explicit families f_d over $\mathbb{R}$. For details, see 3.7.5, and 3.7.7, in Theorem 3.7.

3.7.1 Lower bound for symmetric circuits over $\mathbb{R}$: Proof of the first part of Theorem 3.7.1

We state a lemma from classical mathematics for the study of fewnomials and give a simple proof. This would be critical to prove explicit lower bounds.

Lemma 3.7.2 (Hajós Lemma) *Suppose $f(x) \in \mathbb{C}[x]$ be a univariate polynomial with $t \geq 1$ monomials. Let α be a non-zero root of $f(x)$. Then, the multiplicity of α in f can be at most $t - 1$.*

Proof. We will prove this by induction on t. When $t = 1$, $f(x) = a_m x^m$ for some m. It has no non-zero roots and we are trivially done. Assume that, it is true upto t. We want to prove the claim for $t+1$.

Suppose $|f|_0 = t+1$. There exists $m \geq 0$ such that $f(x) = x^m.g(x)$, with $|g|_0 = t+1$ and $g(0) \neq 0$. It suffices to prove the claim for g. Let, α be a non-zero root of $g(x)$. Suppose, $g(x) = (x-\alpha)^s \cdot h(x)$, for some $s \geq 1$ and $h(\alpha) \neq 0$. Observe that, multiplicity of α in g' is $s-1$. As $g(0) \neq 0$, $|g'|_0 = t$. Therefore by induction hypothesis, $s-1 \leq t-1 \implies s \leq t$. Hence, multiplicity of α in g (thus in f) can be at most t. This finishes the induction step. □

Corollary 3.7.3 *Suppose $f(x) = (x+\alpha)^t \cdot g(x)$, for some non-zero α and $g(\cdot)$, then we must have $|f|_0 \geq t+1$.*

We prove an important lower bound on SOS representation for a non-zero multiple of $(x+1)^d$; it will be important to prove the first part of Theorem 3.7.1.

Lemma 3.7.4 *Let $f(x)$ be a non-zero polynomial in $\mathbb{R}[x]$. Suppose, there exist non-zero $\ell_i \in \mathbb{R}[x]$, for $i \in [m]$ such that $(x+1)^d \cdot f(x) = \sum_{i=1}^m \ell_i^2$. Then, $\sum_{i\in[m]} |\ell_i|_0 \geq m \cdot (\lfloor d/2 \rfloor + 1)$.*

Proof. Denote $g(x) := \gcd(\ell_1, \ldots, \ell_m)$. We will prove that $(x+1)^t \mid g(x)$ where $t := \lfloor d/2 \rfloor$. Suppose not, assume that $(x+1)^k \| g(x)$ (i.e $(x+1)^{k+1} \nmid g(x)$) such that $k < t$ (and thus $d - 2k > 0$). Then, $g(x) = h(x) \cdot (x+1)^k$ where $h(x) \in \mathbb{R}[x]$ with $h(-1) \neq 0$. Define $\tilde{\ell}_i := \ell_i/(x+1)^k$. By assumption, $(x+1) \nmid \gcd(\tilde{\ell}_1, \ldots, \tilde{\ell}_m) =: h(x)$. Thus,

$$\begin{aligned}
\sum_{i=1}^k \ell_i(x)^2 = (x+1)^d \cdot f(x) &\implies \sum_{i=1}^m \tilde{\ell}_i(x)^2 = (x+1)^{d-2k} \cdot f(x) \\
&\implies \sum_{i=1}^m \tilde{\ell}_i(-1)^2 = 0 \\
&\implies \tilde{\ell}_i(-1) = 0\,, \quad \forall i \in [1,m] \\
&\implies (x+1) \mid \tilde{\ell}_i(x)\,, \quad \forall i \in [1,m] \\
&\implies (x+1) \mid \gcd(\tilde{\ell}_1, \ldots, \tilde{\ell}_m) = h(x)
\end{aligned}$$

which is a contradiction. Thus, $k \geq t$.

This implies, each ℓ_i is non-zero polynomial multiple of $(x+1)^t$. Since Corollary 3.7.3 implies that $|\ell_i|_0 \geq t+1$, for all $i \in [m]$; the lemma follows. □

Recall that a *symmetric* depth-2 circuit (over $\mathbb{R}$) is a circuit of the form $B^T \cdot B$ for some $B \in \mathbb{R}^{m \times n}$. We prove the *first* part of Theorem 3.7.1.

Theorem 3.7.5 (Reproving Theorem 1.3 of [KV21]) *There exists an explicit family of real $n \times n$ PSD matrices $\{A_n\}_{n \in \mathbb{N}}$ such that every symmetric circuit computing A_n (over $\mathbb{R}$) has size $\Omega(n^2)$.*

Proof. Denote $[x]_n := \begin{bmatrix} 1 & x & \dots & x^{n-1} \end{bmatrix}$. Denote $k := \lfloor n/2 \rfloor$. Define $g_i(x) := (x+1)^k \cdot x^{\lfloor (i-1)/2 \rfloor}$, for $i \in [n]$. Note that, $\deg(g_i) = k + \lfloor (i-1)/2 \rfloor \leq k + \lfloor (n-1)/2 \rfloor = n-1$. Define $n \times n$ matrix M_n such that

$$M_n \cdot [x]_n^T := \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_n(x) \end{bmatrix}.$$

It is easy to see that $g_1, g_3, g_5, \dots$ are linearly independent over $\mathbb{R}$. Therefore, $\operatorname{rank}(M_n) = \operatorname{rank}_{\mathbb{R}}(g_1(x), \dots, g_n(x)) = \lfloor (n-1)/2 \rfloor + 1 = \lfloor (n+1)/2 \rfloor$.

Define $A_n := M_n^T \cdot M_n$. By definition, A_n is PSD and $\operatorname{rank}(A_n) = \lfloor (n+1)/2 \rfloor$. This follows from the classical fact that for any matrix A over $\mathbb{R}$, $\operatorname{rank}(A^T A) = \operatorname{rank}(A)$. Also A_n is *explicit* (entries are P-computable from definition). Now, assume there is some $m \times n$ matrix B such that $A_n = B^T \cdot B$. Then, denote $B[x]_n := \begin{bmatrix} \ell_1 & \ell_2 & \dots & \ell_m \end{bmatrix}^T$, where $\ell_i \in \mathbb{R}[x]$ are univariate polynomials of degree at most $n-1$. Observe that number of *non-zero* entries in B is precisely $\sum_{i \in [m]} |\ell_i|_0$. Thus, it suffices to show that $\sum_{i \in [m]} |\ell_i|_0 \geq \Omega(n^2)$.

As $\operatorname{rank}(B) = \operatorname{rank}(B^T B) = \operatorname{rank}(A_n) = \lfloor (n+1)/2 \rfloor$, we must

have $m \geq \lfloor (n+1)/2 \rfloor$. Thus,

$$\begin{aligned} A_n = B^T \cdot B &\implies [x]_n M_n^T \cdot M_n [x]_n^T = [x]_n B^T \cdot B [x]_n^T \\ &\iff \sum_{i=1}^{n} g_i(x)^2 = \sum_{i=1}^{m} \ell_i^2 \\ &\iff (x+1)^{2k} \cdot f(x) = \sum_{i=1}^{m} \ell_i^2 \text{, where } f(x) := \sum_{i=1}^{n} x^{2\cdot\lfloor (i-1)/2 \rfloor} \\ &\implies \sum_{i=1}^{m} |\ell_i|_0 \geq (\lfloor (n+1)/2 \rfloor) \cdot (k+1) \geq \frac{n^2}{4} . \end{aligned}$$

The last line follows by Lemma 3.7.4. □

3.7.2 Lower bound for invertible circuits over $\mathbb{R}$: Proof of the second part of Theorem 3.7.1

This subsection is devoted to proving the *second* part of Theorem 3.7.1. This proof uses SOS lower bound for a bivariate polynomial. Investigating sum of product of two polynomials is similar to looking at the SOS; as, one can write $f \cdot g = ((f+g)/2)^2 - ((f-g)/2)^2$. The summand fan-in at most doubles. Thus, proving lower bound for sum of product of two polynomials is 'same' as proving SOS lower bound. The following lemma proves certain lower bound on sum of sparsity when a specific *bivariate* polynomial is written as sum of product of two polynomials (with certain restrictions).

Lemma 3.7.6 *Let $f_d := f_{d,t}(x,y) := \left(\prod_{i\in[d]}(x-i)(y-i)\right) \cdot p(x,y)$, for some polynomial $p \in \mathbb{R}[x,y]$ such that $deg_x(p) = deg_y(p) = t$. Suppose, $f_d = \sum_{i\in[d+t+1]} \ell_i(x) \cdot \tilde{\ell}_i(y)$, where $\ell_i, \tilde{\ell}_i$'s are polynomials of degree at most $d+t$; with the additional property that $\tilde{\ell}_1, \ldots, \tilde{\ell}_{d+t+1}$ are $\mathbb{R}$-linearly independent.*

Then, $\sum_{i=1}^{d+t+1} |\ell_i|_0 \geq m \cdot (d+1)$, where m is the number of non-zero ℓ_i.

Proof. Suppose, $g(x) := \gcd(\ell_1, \ldots, \ell_{d+t+1})$. We claim that $\prod_{i=1}^{d}(x-i) \mid g(x)$. Note that, it suffices to prove the claim; as, $\prod_{i=1}^{d}(x-i) \mid \ell_i(x)$ for each non-zero ℓ_i implies $|\ell_i|_0 \geq d+1$ by Lemma 2.2.5.

We prove the claim by contradiction. Suppose, there exists $j \in [d]$ such that $x - j \nmid g(x)$, so $g(j) \neq 0$. Fix this j. Hence, there exists i such that $\ell_i(j) \neq 0$.

In particular, $v := \begin{bmatrix} \ell_1(j) & \ell_2(j) & \dots & \ell_{d+t+1}(j) \end{bmatrix}^T \neq \mathbf{0}$. Define the $(d+t+1) \times (d+t+1)$ matrix A as

$$[y]_{d+t+1} \cdot A \ := \ \begin{bmatrix} \tilde{\ell}_1 & \tilde{\ell}_2 & \dots & \tilde{\ell}_{d+t+1} \end{bmatrix},$$

where

$$[y]_{d+t+1} \ := \ \begin{bmatrix} 1 & y & \dots & y^{d+t} \end{bmatrix}.$$

Observe: $\text{rank}_{\mathbb{R}}(\tilde{\ell}_1, \dots, \tilde{\ell}_{d+t+1}) = d+t+1 \iff A$ is invertible. But,

$$\begin{aligned} v \neq \mathbf{0} \text{ and } A \text{ is invertible} &\implies A \cdot v \ \neq \ \mathbf{0} \\ &\implies [y]_{d+t+1} \cdot Av \ \neq \ 0 \\ &\implies \sum_{i=1}^{d+t+1} \tilde{\ell}_i(y) \cdot \ell_i(j) \ \neq \ 0 \\ &\implies f_{d,t}(j, y) \ \neq \ 0 \end{aligned}$$

which is a contradiction! Therefore, $\prod_{i=1}^{d}(x-i) \mid g(x)$ and so we are done. □

Recall that an *invertible* depth-2 circuit computes a matrix A such that whenever $A = BC$, either B or C has to be invertible. We prove the *second* part of Theorem 3.7.1.

Theorem 3.7.7 (Reproving Theorem 1.5 of [KV21]) *There exists an explicit family of $n \times n$ PSD matrices $\{A_n\}_{n \in \mathbb{N}}$ such that every invertible circuit over $\mathbb{R}$ computing A_n has size $\Omega(n^2)$.*

Proof. Denote $k := \lfloor n/2 \rfloor$. Define $g_i(x) := \prod_{i=1}^{k}(x-i) \cdot x^{\lfloor (i-1)/2 \rfloor}$, for $i \in [n]$. Note that $\deg(g_i) = k + \lfloor (i-1)/2 \rfloor \leq k + \lfloor (n-1)/2 \rfloor = n-1$. Define the $n \times n$ matrix M_n as

$$M_n \cdot [x]_n^T \ := \ \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_n(x) \end{bmatrix}.$$

It is easy to see that $g_1, g_3, g_5, \dots$ are linearly independent over $\mathbb{R}$. Therefore, $\text{rank}(M_n) = \text{rank}_{\mathbb{R}}(g_1(x), \dots, g_n(x)) = \lfloor (n-1)/2 \rfloor + 1 = \lfloor (n+1)/2 \rfloor$.

Define $A_n := M_n^T \cdot M_n$. By definition, A_n is PSD and $\text{rank}(A_n) = \lfloor (n+1)/2 \rfloor$. This follows from the classical fact that for any matrix A, $\text{rank}(A^T A) = \text{rank}(A)$ over $\mathbb{R}$. Also A_n is *explicit* (entries are P-computable from definition).

Suppose, there exists $n \times n$ invertible matrix B and some $n \times n$ matrix C such that $A_n = B \cdot C$ (the other case where C is invertible is similar). Note that, from classical property of rank of matrices, $\text{rank}(C) \geq \text{rank}(A_n) = \lfloor (n+1)/2 \rfloor$. With the usual notation of $[x]_n$ and $[y]_n$ used before, denote

$$[y]_n \cdot B := \begin{bmatrix} \tilde{\ell_1}(y) & \tilde{\ell}_2(y) & \dots & \tilde{\ell}_n(y) \end{bmatrix},$$

and

$$C \cdot [x]_n^T := \begin{bmatrix} \ell_1(x) & \ell_2(x) & \dots & \ell_n(x) \end{bmatrix}^T.$$

Note that the degree of each $\ell_i, \tilde{\ell}_i$ can be at most $n-1$. Thus,

$$\begin{aligned} A_n = B \cdot C &\implies [y]_n M_n^T \cdot M_n [x]_n^T = [y]_n \cdot B \cdot C \cdot [x]_n^T \\ &\iff \sum_{i=1}^{n} g_i(x) \cdot g_i(y) = \sum_{i=1}^{n} \ell_i(x) \cdot \tilde{\ell}_i(y) \\ &\iff \left(\prod_{i=1}^{k} (x-i)(y-i) \right) \cdot p(x,y) = \sum_{i=1}^{n} \ell_i(x) \cdot \tilde{\ell}_i(y) \end{aligned}$$

where $p(x,y) := \sum_{i \in [n]} (xy)^{\lfloor (i-1)/2 \rfloor}$. The LHS is actually of the form $f_{k,\lfloor (n-1)/2 \rfloor}(x,y)$ as in Lemma 3.7.6. From the lower bound on rank of C, we know that there must be at least $\lfloor (n+1)/2 \rfloor$ many non-zero ℓ_i's. Therefore, by Lemma 3.7.6, it follows that

$$\sum_{i=1}^{n} |\ell_i|_0 \geq \lfloor (n+1)/2 \rfloor \cdot (k+1) \geq n^2/4 .$$

□

Remark 3.7.1 The defined matrix A_n in the above proof *also* works for the Theorem 3.7.5. For that, one needs to replace the polynomial $\prod_{i=1}^{d}(x-i) \cdot f(x)$, in Lemma 3.7.4, and prove similar lower bound on sum of sparsity. The proof details of theorem remains *almost* unchanged until at the very end, one has to use Descartes' rule (Lemma 2.2.5) instead of Corollary 3.7.3.

3.8 τ-conjectures for Top-fanin 2 Hold True

In this section, we show that both SOS-τ-conjecture and SOC-τ-conjecture hold true for top fanin-2.

3.8.1 SOS-τ-conjecture for sum of two squares

We show that when f is a sum of two squares, number of real roots is indeed linear in the support-sum.

Theorem 3.8.1 *If $f = \sum_{i=1}^{s} c_i \cdot f_i^2 \in \mathbb{R}[x]$, where $s \leq 2$, then f can have at most $O(\sum_{i=1}^{s} |f_i|_0)$-many real roots.*

Proof. There are two cases to consider:

Case I ($s = 1$): In this case, $f = c_1 \cdot f_1^2$. Thus, the real roots of f are precisely the roots of f_1. However, by Descartes' rule (Lemma 2.2.5), f_1 can have at most $2(|f_1|_0 - 1)$-many real roots.

Case II ($s = 2$): Without loss of generality, assume that c_1 and c_2 are of opposite signs ; otherwise, any real root of f must also be roots of f_1 and f_2, and trivially we are done by Lemma 2.2.5. When, the signs are opposite, note that, f has the following factoring over $\mathbb{R}[x]$:

$$f = c_1 \cdot (f_1 + \gamma \cdot f_2) \cdot (f_1 - \gamma \cdot f_2)\,, \text{ where } \gamma := \sqrt{-c_2/c_1} \in \mathbb{R}\,.$$

It directly follows that $|f_1 \pm \gamma \cdot f_2|_0 \leq |f_1|_0 + |\gamma \cdot f_2|_0 = |f_1|_0 + |f_2|_0$. However, the real roots of f must also be real roots of $f_1 \pm \gamma \cdot f_2$. Each $f_1 \pm \gamma \cdot f_2$ can have at most $2(|f_1|_0 + |f_2|_0) - 2$ many real roots, by Descartes' rule (Lemma 2.2.5). Therefore, the conclusion follows. □

Remark 3.8.1 We could strengthen the above theorem by replacing $O(|\bigcup_{i\in[2]} \text{supp}(f_i)|)$. Since, $|\text{supp}(f_1 \pm \gamma \cdot f_2)| \leq |\text{supp}(f_1) \bigcup \text{supp}(f_2)|$, using Lemma 2.2.5, the conclusion follows.

3.8.2 SOC-τ-conjecture for sum of two cubes

We show that when f is a sum of two squares, number of real roots is indeed linear in the support-union.

Theorem 3.8.2 *If $f = \sum_{i=1}^{s} c_i \cdot f_i^3 \in \mathbb{R}[x]$ where $s \leq 2$, then f can have at most $O(|\bigcup_{i=1}^{s} \text{supp}(f_i)|)$-many real roots.*

Proof. There are two cases to consider:

Case I ($s = 1$): In this case, $f = c_1 \cdot f_1^3$. Thus, the real roots of f are precisely the roots of f_1. However, by Descartes' rule (Lemma 2.2.5), f_1 can have at most $2(|f_1|_0 - 1)$-many real roots.

Case II ($s = 2$): Note that, f has the following factoring over $\mathbb{R}[x]$:

$$f = c_1 \cdot (f_1 + \gamma \cdot f_2) \cdot (f_1^2 - \gamma \cdot f_1 f_2 + \gamma^2 \cdot f_2^2) \text{ , where } \gamma := \sqrt[3]{c_2/c_1} \in \mathbb{R} .$$

However,

$$f_1^2 - \gamma \cdot f_1 f_2 + \gamma^2 \cdot f_2^2 = (f_1 - \frac{\gamma}{2} \cdot f_2)^2 + (\frac{3\gamma^2}{4}) \cdot f_2^2 ,$$

which has $O(|\bigcup_{i=1}^{2} \text{supp}(f_i)|)$-many real roots by Theorem 3.8.1 (and its remark). Also $f_1 + \gamma \cdot f_2$ has at most $O(|\bigcup_{i=1}^{2} \text{supp}(f_i)|)$-many real roots by Descartes' rule (Lemma 2.2.5). Moreover, any real root of f must also be real roots of either $f_1 + \gamma \cdot f_2$ or $f_1^2 - \gamma \cdot f_1 f_2 + \gamma^2 \cdot f_2^2$. Therefore, the conclusion follows. □

3.9 SOS-τ-conjecture to SOS Lower Bound on $(x+1)^d$

Lemma 3.9.1 *If Conjecture 3.1.2 is true, then $S_{\mathbb{C}}(f_d) \geq \Omega(d)$, where $f_d := (x + a)^d$, for any $0 \neq a \in \mathbb{R}$.*

Before proving the above, we establish an interesting lemma. For $f \in \mathbb{C}[x]$, we denote $\mathfrak{R}(f)$ as the real part of f, and $\mathfrak{I}(f)$ as the imaginary part, i.e. $f = \mathfrak{R}(f) + \iota \cdot \mathfrak{I}(f)$. Note, $|\mathfrak{R}(f)|_0, |\mathfrak{I}(f)|_0 \leq |f|_0$.

Lemma 3.9.2 *$S_{\mathbb{R}}(\mathfrak{R}(f)) \leq 2 \cdot S_{\mathbb{C}}(f)$, for any $f \in \mathbb{C}[x]$.*

Proof. Suppose, $f(x) = \sum_{i=1}^{s} f_i^2$, for $f_i \in \mathbb{C}[x]$ is a *minimal* representation in SOS-model over $\mathbb{C}$ (we ignore the constants c_i as in Equation 3.1 as $\sqrt{c_i}$ can be taken inside), i.e. $S_{\mathbb{C}}(f) = \sum_{i=1}^{s} |f_i|_0$. Note that

$$\mathfrak{R}(f) = \sum_{i=1}^{s} \mathfrak{R}(f_i^2) = \sum_{i=1}^{s} \mathfrak{R}(\mathfrak{R}(f_i) + \iota \cdot \mathfrak{I}(f_i))^2$$
$$= \sum_{i=1}^{s} (\mathfrak{R}(f_i)^2 - \mathfrak{I}(f_i)^2) .$$

The last expression implies that

$$S_{\mathbb{R}}(\mathfrak{R}(f)) \leq \sum_{i=1}^{s} |\mathfrak{R}(f_i)|_0 + \sum_{i=1}^{s} |\mathfrak{I}(f_i)|_0 \leq \sum_{i=1}^{s} 2|f_i|_0 = 2 \cdot S_{\mathbb{C}}(f) .$$

□

Proof of Lemma 3.9.1. It suffices to prove the bound for $f_d = (x+1)^d$, as $S_{\mathbb{C}}((x+a)^d) = S_{\mathbb{C}}((x+1)^d)$ [just by replacing $x \mapsto x/a$]. Consider the complex polynomial $g_d(x) := f_d(\iota x) + f_d(-\iota x)$. Its degree is either d, if d is even, or $d-1$, if it is odd. The roots are of the form

$$\iota \cdot \frac{1-\zeta}{1+\zeta},$$

where ζ is d-th root of -1 ($\zeta \neq 1$). There are again either d or $d-1$ such roots, depending on the parity of d. Further, they are all *distinct*. Since $|\zeta| = 1$, each root

$$\iota \cdot \frac{1-\zeta}{1+\zeta} = \iota \cdot \frac{(1-\zeta)(1+\bar{\zeta})}{(1+\zeta)(1+\bar{\zeta})} = \iota \cdot \frac{\bar{\zeta}-\zeta}{|1+\zeta|^2} = \frac{2\mathfrak{I}(\zeta)}{|1+\zeta|^2}$$

is real. Therefore, $g_d(x)$ must be a real polynomial with *distinct real* roots. Hence Conjecture 3.1.2 implies that $S_{\mathbb{R}}(g_d) = \Omega(d)$. Using Lemma 3.9.2, one can directly conclude that $S_{\mathbb{C}}(g_d) = \Omega(d)$. It is straightforward to see that $S_{\mathbb{C}}(f)$ remains unchanged under the map $x \mapsto c \cdot x$, for any $c \neq 0$. Therefore, in particular, $S_{\mathbb{C}}(f_d(\iota x)) = S_{\mathbb{C}}(f_d(-\iota x)) = S_{\mathbb{C}}(f_d)$. Finally, we must have

$$\Omega(d) \;=\; S_{\mathbb{C}}(g_d) \;\leq\; S_{\mathbb{C}}(f_d(\iota x)) \;+\; S_{\mathbb{C}}(f_d(-\iota x)) \;=\; 2\cdot S_{\mathbb{C}}(f_d)\,.$$

Hence, the conclusion follows.

3.10 Newton Polygon and Bivariate SOS Lower Bound

In [Koi+15], Koiran proposed a τ-conjecture for Newton polygons of bivariate polynomials. Like the real τ-conjecture, it deals with sums of products of sparse polynomials and implies that the permanent is hard. A common idea to all these τ-conjectures is that "simple" arithmetic circuits should compute only "simple" polynomials. In the original τ-conjecture, the simplicity of a polynomial is measured by the number of its integer roots; in the real τ-conjecture (or its counterpart in SOS-, or, SOC-model), it is measured by the number of its real roots; and in [Koi+15], by the number of edges of its Newton polygon; for basic definitions see below [10].

10: These conjectures are *independent* in the sense that we know of no implication between two of them.

Consider a bivariate polynomial $f \in \mathbb{F}[X,Y]$. To each monomial X^iY^j appearing in f with a nonzero coefficient, we associate a point with coordinate (i,j) in the Euclidean plane. Let $\mathrm{Mon}(f)$ denotes this finite set of points. If A is a set of points in the plane, we denote by $\mathrm{conv}(A)$ the *convex hull* of A. By definition, the *Newton polygon* of f, denoted by $\mathrm{Newt}(f)$, is the convex hull of $\mathrm{Mon}(f)$, i.e., $\mathrm{Newt}(f) = \mathrm{conv}(\mathrm{Mon}(f))$. Note that $\mathrm{Newt}(f)$ has at most t edges if f has t monomials.

In this section, we use similar techniques to [Koi+15] (of Newton polygon) to prove an interesting lower bound for a bivariate polynomial in the sum-of-constant-power model; see Theorem 3.10.4. Before proving that, we state some basics to build the foundation of the proof. The following result is well known in the literature.

Theorem 3.10.1 [Ost75] *$Newt(fg) = Newt(f)+Newt(g) := \{p+q \mid p \in Newt(f), q \in Newt(g)\}$.*

From the above theorem, one can deduce that $\mathrm{Newt}(f^2) = 2\cdot\mathrm{Newt}(f)$. However, if S is a convexly independent subset of

2·Newt(f), how large can S be? [A set is called *convexly independent* if its elements are exactly the vertices of its convex hull.]

This will be crucial in the next section. Here is an important theorem (which is optimal up to constant factors) regarding the size of S; compare the bound with the trivial mn.

Theorem 3.10.2 [Eis+08] *Let P and Q be two planar point sets with $|P| = m$ and $|Q| = n$. Let S be a convexly independent subset of the Minkowski sum $P + Q$. Then, we have $|S| \leq O(m^{2/3}n^{2/3} + m + n)$.*

Corollary 3.10.3 *Let P be a planar point set with $|P| = n$. Let S be a convexly independent subset of rP (r is a constant). Then, $|S| \leq O(n^{r^{\log(4/3)}})$.*

Proof. Let $T(r)$ be the maximum size of convexly independent subset of rP. Thus, we must have $T(r) \leq O(T(r/2)^{4/3})$ with $T(1) \leq n$. Thus, $T(r) \leq O(n^{(4/3)^{\log r}}) = O(n^{r^{\log(4/3)}})$. □

Using convexity theory, we establish the lower bound of $\Omega(d^{1/r^{\log(4/3)}})$ for the bivariate polynomial $\sum_{i=0}^{d} x^i y^{i^2}$. This polynomial was studied in [Koi+15].

Theorem 3.10.4 *For $f(x, y) := \sum_{i=0}^{d} x^i y^{i^2}$, we have $S_{\mathbb{R}}(f, r, s) \geq \Omega(d^{1/r^{\log(4/3)}})$, for any $s \geq 1$ and constant r.*

Proof sketch. Write $f(x, y) = \sum_{i\in[s]} \ell_i(x, y)^r$. Let S_i be the set of points in the plane corresponding to the monomials of ℓ_i^r which appear in f with a nonzero coefficient. Since Newt(f) is the convex hull of $\cup_i \text{conv}(S_i)$, it is enough to bound the number of vertices of conv(S_i).

Of course, the vertices of conv(S_i) is a convexly independent subset of Mon(ℓ_i^r) $\subseteq r\text{Mon}(\ell_i)$. Hence, by Corollary 3.10.3, we get that conv(S_i) has at most $O(|\ell_i|_1)^{r^{\log(4/3)}}$ many vertices. Thus, the convex hull of $\bigcup_i \text{conv}(S_i)$ has at most $O(\sum_i \|\ell_i\|^{r^{\log(4/3)}})$ vertices. On the other hand, as $y = x^2$ is a convex function, Newt(f) has $d + 1$ many vertices. Therefore,

$$\sum_i (|\ell_i|_1)^{r^{\log(4/3)}} \geq d + 1 \implies \sum_i |\ell_i|_1 \geq \Omega(d^{1/r^{\log(4/3)}}) .$$

By definition, we must have $S_{\mathbb{R}}(f, r, s) \geq \Omega(d^{1/r^{\log(4/3)}})$, for any $s \geq 1$. □

Remark 3.10.1 As $\log(4/3) \approx 0.415 < 1$, the above is a better lower bound on $S_{\mathbb{R}}(\cdot)$ than the trivial lower bound of $\mathrm{sp}(f)^{1/r} = (d+1)^{1/r}$.

3.11 Discussion

The findings in this chapter suggest that proving the "simple looking" lower bounds (in this context, the lower bounds on the size of the SOS representations) - is perhaps *more difficult* than one might assume. Flipping the coin, it simply conveys that proving an upper bound on the number of real roots, of a polynomial, represented as the 'simplest' model is more difficult, and has very fundamental consequences.

Moreover, an important ingredient of these results essentially establish that a small algebraic circuit has a small SOS or SOC representation. These results can be thought as a "fine-grained" version of the famous chasm results [AV08; Gup+16]. It would be nice to know some more potential application of these compact, small representations.

Finally, the idea behind "multivariate ↔ univariate" is old, due to Kronecker substitution. However, in this work, we used a different map. Unfortunately, both of these transformations are *uni*-directional, in the sense that the easiness in the multivariate setting translates to the univariate one; however, we *do not* know whether the opposite direction is also true! We conclude this chapter with the following meta-question.

Food for thought

Are there any interesting 'explicit' map which transforms univariate polynomials to multivariate ones uniquely, while preserving the hardness in both the directions?

Depth-4 Identity Testing | 4

"Everything we care about lies somewhere in the middle, where pattern and randomness interlace."

— James Gleick, The Information: A History, a Theory, a Flood.

Abstract. Polynomial Identity Testing (PIT) is a fundamental computational problem. The famous depth-4 reduction result by Agrawal and Vinay [AV08], and its subsequent improvements have made PIT for depth-4 circuits an enticing pursuit. A restricted depth-4 circuit computing a n-variate degree-d polynomial of the form $\Sigma_{i=1}^{k}\Pi_j g_{ij}$, where, $\deg(g_{ij}) \leq \delta$, is called $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuit. On further restricting g_{ij} to be sum of univariates we obtain $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits. The largely open, special-cases of $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ for constant k and δ, and $\Sigma^{[k]}\Pi\Sigma\wedge$ have been a source of many great ideas in the last two decades. For e.g., depth-3 ideas of Dvir and Shpilka [DS07], Kayal and Saxena [KS07], and Saxena and Seshadhri [SS11; SS12]; depth-4 ideas of Beecken, Mittmann and Saxena [BMS13], Saha, Saxena and Saptharishi [SSS13], Forbes [For15], and additionally, geometric Sylvester-Gallai ideas of Kayal and Saraf [KS09], and Peleg and Shpilka [PS20; PS21] have been quite diverse and rich in mathematics.

Very recently, a subexponential-time *blackbox* PIT algorithm for constant-depth circuits was obtained via lower bound breakthrough of Limaye, Srinivasan, Tavenas [LST21]. We solve one of the basic underlying open problem in this work.

We give the *first* quasipolynomial-time *blackbox* PIT for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits. A common technical ingredient for both of them is how the *low algebraic rank*, along with the *logarithmic derivative* operator, modify the top Π-gate to $\wedge$.

This chapter is based on the second half of the article title *Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits*, which is a joint work with Prateek Dwivedi and Nitin Saxena, that appeared in CCC 2021 [DDS21a].

4.1 Set-up: Bounded Depth-4 Circuits

In a series of surprising results, Agrawal and Vinay [AV08], followed by Koiran [Koi12] and Tavenas [Tav15] showed that any VP family $(f_n)_n$ has depth-4 algebraic circuits ($\Sigma\Pi\Sigma\Pi$) of size $n^{O(\sqrt{d_n}\log d_n)}$, where d_n is the degree of d_n. For instance, this result shows that if perm_n has $\mathrm{poly}(n)$ size circuits, then it also has depth-4 circuits of size $n^{O(\sqrt{n}\log n)}$. This is potentially useful for a lower bound proof: to show that the permanent does not have polynomial size circuits, we "only" have to show that it *requires* depth-4 circuits of size $n^{\omega(\sqrt{n}\log n)}$.

Further, efficient derandomization of PIT for just depth-4 $\Sigma\Pi\Sigma\Pi$ circuits implies an *exponential* lower bound for general circuits, and an efficient derandomization of PIT for general circuits of poly-degree [AV08; AS09]. We briefly sketch the proof.

Proposition 4.1.1 *If there is a PIT algorithm for depth-4 circuit running in deterministic polynomial-time, then there is a deterministic* sub*exponential PIT algorithm for any general circuit computing a low degree polynomial.*

Proof sketch. Given any circuit of size s, we can convert it to a depth-4 circuit of size $2^{O(\sqrt{s}\log s)} = 2^{o(s)}$. Further, this conversion can be done in time $2^{o(s)}$ as well. Therefore, a polynomial-time PIT algorithm for depth-4 would yield a $2^{o(s)}$ time algorithm for general circuits of size s (and poly-degree). □

Due to Proposition 4.1.1, in the last two decades, there has been an incredibly large number of results for $\Sigma\Pi\Sigma\Pi$-circuits with diverse restrictions; e.g. 'locally' bounded independence, bounded read/occur, bounded variables.

Our Bounded Depth-4 Models. In this chapter, we consider two models.

1. The bounded sum of product of sum of univariate polynomials $\Sigma^{[k]}\Pi\Sigma\wedge$ (formally first studied in [SSS13]). These circuits compute polynomials of the form

$$\sum_{i\in[k]}\prod_{j}\Big(g_{ij1}(x_1)+\cdots+g_{ijn}(x_n)\Big)\text{, where } g_{ij\ell}\in\mathbb{F}[x_\ell]\,.$$

2. The bounded sum of product of bounded degree polynomials $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$; these circuits compute polynomials which are of the form

$$\sum_{i\in[k]}\prod_{j} g_{ij}(x)\,,\ \text{where}\ \ \deg(g_{ij})\ \le\ \delta\,.$$

We remark that even $\delta = 2$, has been quite a challenging open problem [KS16, Open Problem 2].

Circuit size. The size of the circuit could be defined as follows: For circuits $g_i \in \Sigma\Pi^{[\delta]}$*, respectively $\in \Sigma\wedge$, the *size* of their product is simply the sum of the individual size (the degree included):

$$\text{size}(g_1\cdots g_s) = \sum_{i\in[s]}(\text{sp}(g_i)+\deg(g_i))\,.$$

Similarly, for $T_i \in \Pi\Sigma\Pi^{[\delta]}$, respectively $\Pi\Sigma\wedge$,size is defined as $\text{size}(\sum_i T_i) = \sum_i \text{size}(T_i)$.

4.2 Our Results and Main Techniques

Throughout the chapter, we will work with $\mathbb{F} = \mathbb{Q}$, though all the results hold for the field of *large* characteristic.

Theorem 4.2.1 (Blackbox depth-4 PIT) *Let k and δ be arbitrary fixed positive integers. Then,*

(a) There is a $s^{O(\delta^2 k\log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits of size s, over $\mathbb{F}$.

(b) There is a $s^{O(k\log\log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits of size s, over $\mathbb{F}$.

Remark 4.2.1

1. Our results are quasipolynomial-time even up to $k, \delta = \text{poly}(\log s)$.
2. Theorem 4.2.1 (b) is better than Theorem 4.2.1 (a), because $\Sigma\wedge\Sigma\wedge$ has a *faster* algorithm known than $\Sigma\wedge\Sigma\Pi^{[\delta]}$.
3. Even for $\Sigma^{[3]}\Pi\Sigma\wedge$ and $\Sigma^{[3]}\Pi\Sigma\Pi^{[3]}$ models, we leave the *poly*-time blackbox question open! When $k = 3, \delta = 2$,

* $\Sigma\Pi^{[\delta]}$ circuits compute polynomials of the form g, where $\deg(g) \le \delta$.

A well-known theorem in incidence geometry, called the *Sylvester-Gallai* (SG) theorem, states that : if there are *n distinct* points on the real plane such that, for every pair of distinct points, the line through them also contains a third point, then they all lie on the same line. Ankit Gupta [Gup14] proposed a new line of SG theorems for *non-linear* polynomials. And this can be devised to give PIT algorithms for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits.

To see this, they defined SG-type circuits which essentially means if $C = \sum_{i\in[k]} T_i$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$, then $\forall\, i \in [k], \bigcap_{j\neq i} V(T_j) \subseteq V(T_i)$. Here, $V(f)$ denotes the *variety*, the set of zeroes, of the polynomial f.

In [Gup14], it was conjectured that for SG-type circuits, $C = 0$ implies small (constant) *transcendence* degree, depending on k, and δ. Assuming the conjecture, one can show a polynomial-time PIT for SG-type $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits. Further, non-SG-type circuits already have a polynomial-time PIT [Gup14; Guo21]. We remark that in this work, we *donot* go via this algebraic-gemetric approach of solving bounded depth-4 PITs.

[PS20; PS21] showed a polynomial-time algorithm for the same.

Comment 4.2.1

A recurring theme in the blackbox PIT research on depth-3/depth-4 circuits has been that of rank. If we consider a circuit $C = \Sigma^{[k]}\Pi\Sigma = \sum_{i\in[k]} \prod_j \ell_{ij}$, where $\ell_{i,j}$ are linear forms, then rank(C) is defined to be the linear rank of the set of forms $\{\ell_{i,j}\}_{i,j}$, each viewed as a vector in $\mathbb{F}^n$.

In [BMS13] the authors generalized this notion of rank for depth-4 circuits as well, and more importantly, one that is useful in blackbox PIT. It was via transcendence degree. Further, it was established that the depth-4 circuits with top-fanin $k = 2$, after reducing it to the *simplest minimal* form, must have the rank=1. Finally, using it, one can give a polynomial-time blackbox PIT for $\Sigma^{[2]}\Pi\Sigma\Pi^{[\delta]}$, *even* when $\delta = O(\sqrt{\log s})$.

[Gup14] further generalized the approach of [BMS13], in a more geometric way, as discussed in the margin note. Moreover, the classification into SG-type and non-SG-type circuits, along with efficient PIT for non-SG circuits, in [Gup14], showed something "more" remarkable: it gives polynomial-time PIT unconditionally for "most" $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits. By this we mean that if one fixes the parameters k, δ, and samples the g_{ij}'s randomly over any large enough subset of $\mathbb{F}$, one can show that w.h.p., the circuit C *will not* be an SG-type circuit!

❑ Jacobian hits again

In Theorem 4.2.1 we exploit the prowess of the Jacobian polynomial, which was first introduced in the context of PIT in [BMS13], and later explored in [Agr+16] to unify known PIT algorithms and design new ones; for the basic definitions and properties, we refer to Chapter 2.

Suppose, we want to test $\sum_{i\in[k]} T_i \stackrel{?}{=} 0$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ (respectively $\Pi\Sigma\wedge$). We associate the Jacobian $J(T_1, \ldots, T_r)$ to captures the algebraic independence of, $T_1, \ldots, T_r$ assuming this to be a transcendence basis of the T_i's. We design a *variable reducing linear* map Φ which preserves the algebraic

Independence of $T_1, \ldots, T_r$ and show that for any C:

$$C(T_1, \ldots, T_k) \;=\; 0 \quad \Longleftrightarrow \quad C(\Phi(T_1), \ldots, \Phi(T_k)) \;=\; 0\,.$$

Such a map is called 'faithful' [Agr+16]; see Definition 2.8.3. The map Φ ultimately provides a hitting set for $T_1 + \ldots + T_k$, as we reduce to a PIT of a polynomial over 'few' (roughly equal to k) variables, yielding a *quasi*polynomial-time algorithm.

Finding such an 'explicit' map Φ is very analytical as we use *logarithmic derivative*, and its power series expansion, which greatly transform the respective models. Both the proofs are *one-shot* proofs. In both the cases, we essentially reduce the respective models to the well-understood *wedge* models, that have unbounded top-fanin[1]; yet PITs for these models are known.

1: These reductions are quite powerful, also in the sense that usually a Π gate to $\wedge$ gate would incur an *exponential* blowup in the fanin, due to the Waring identity (Lemma 2.2.3); clearly we cannot do that.

4.2.1 Prior works on related models

There have been numerous results on PIT for depth-3 circuits with bounded top fanin (known as $\Sigma^{[k]}\Pi\Sigma$-circuits). Dvir and Shpilka [DS07] gave the first quasipolynomial-time deterministic whitebox algorithm for $k = O(1)$, using rank based methods, which finally lead Karnin and Shpilka [KS11] to design an algorithm of the same complexity in the blackbox setting. Kayal and Saxena [KS07] gave the first polynomial-time algorithm of the same. Later, a series of works in [SS11; SS12; SS13; Agr+16] generalized the model and gave $n^{O(k)}$-time algorithm when the algebraic rank of the product polynomials are bounded.

There has also been some progress on PIT for restricted classes of depth-4 circuits. A quasipolynomial-time blackbox PIT algorithm for *multilinear* $\Sigma^{[k]}\Pi\Sigma\Pi$-circuits was designed in [Kar+13], which was further improved to a $n^{O(k^2)}$-time deterministic algorithm in [SV18]. A quasipolynomial blackbox PIT was given in [BMS13; KS16] when the algebraic rank of the irreducible factors in each multiplication gate as well as the bottom fanin are bounded. Further, interesting restrictions like sum of product of fewer variables, and more structural restrictions have been exploited, see [FS13a; ASS13; For15; Muk16; KS17]. Some progress has also been made for bounded top-fanin and bottom-fanin depth-4 circuits via incidence geometry [Gup14; Shp20; PS20]. In fact,

very recently, [PS21] gave a polynomial-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$-circuits.

Model	Time	Ref.
$\Sigma^{[k]}\Pi^{[d]}\Sigma$	$\mathrm{poly}(s, d^k)$	[SS12]
Multilinear $\Sigma^{[k]}\Pi\Sigma\Pi$	$\mathrm{poly}(s^{O(k^2)})$	[SV18; Agr+16]
$\Sigma\Pi\Sigma\Pi$ of bounded trdeg	$\mathrm{poly}(s^{\mathrm{trdeg}})$	[BMS13]
$\Sigma^{[k]}\Pi\Sigma\Pi^{[d]}$ of bounded *local* trdeg	$\mathrm{QP}(s)$	[KS17]
$\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$	$\mathrm{poly}(s, d)$	[PS21]
$\Sigma\Pi\Sigma\Pi$	$\mathsf{SUBEXP}(n)$	[LST21]
Whitebox $\Sigma^{[k]}\Pi\Sigma\wedge$	$s^{O(k7^k)}$	[DDS21a].
$\Sigma^{[k]}\Pi\Sigma\wedge$	$s^{O(k\log\log s)}$	This chapter.
$\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$	$s^{O(\delta^2 k\log s)}$	This chapter.

[LST21] breakthrough. In a breakthrough result by Limaye, Srinivasan and Tavenas [LST21], the *first* superpolynomial lower bound for constant depth circuits was obtained. Their lower bound result, together with the 'hardness vs randomness' tradeoff result of [CKS18] gives the *first* deterministic blackbox PIT algorithm for general depth-4 circuits which runs in $s^{O(n^\epsilon)}$ for all real $\epsilon > 0$. Their result is the first *sub*exponential-time PIT algorithm for depth-4 circuits. Moreover, compared to their algorithm, our quasipolynomial-time blackbox and polynomial-time whitebox algorithms are significantly faster.

PIT in the border. Recently, with my coauthors Dwivedi and Saxena, we generalized the DiDI-technique introduced in [DDS21a], to solve 'border PIT' of depth-4 circuits [DDS21b]. Specifically, we give a $s^{O(k\cdot 7^k\cdot\log\log s)}$ time and $s^{O(\delta^2\cdot k\cdot 7^k\cdot\log s)}$ time blackbox PIT algorithm for $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ and, $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$ respectively. By definition, border classes capture exact complexity classes, hence border PIT results seemingly subsumes the results we present in this paper. However, the time complexity of blackbox PIT algorithms has a way better dependence on k and δ! Moreover, the Jacobian criteria seems to not work in the border. We have not included these results in this thesis.

❑ **Limitations of known techniques**

People have studied depth-4 PIT only with extra restrictions, mostly due to the limited applicability of the existing techniques, as they were tailor-made for the specific models and do not generalize. E.g. the previous methods handle $\delta = 1$ (i.e. linear polynomials at the bottom) or $k = 2$ (via *factoring*, [SSS13]). While $k = 2$ to 3, or $\delta = 1$ to 2 (i.e. 'linear' to 'quadratic') already demands a qualitatively different approach.

Whitebox $\Sigma^{[k]}\Pi\Sigma\wedge$ model generalizes the famous bounded top fanin depth-3 circuits $\Sigma^{[k]}\Pi\Sigma$ of [KS07]; but their Chinese Remaindering (CR) method, loses applicability and thus fails to solve even a slightly more general model. The blackbox setting involved similar 'certifying path' ideas in [SS12] which could be thought of as general CR. It comes up with an ideal I such that $f \neq 0 \bmod I$ and finally preserves it under a constant-variate linear map. The preservation gets harder (for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$) due to the increased non-linearity of the ideal I generators. Intuitively, larger δ via ideal-based routes, brings us to the Gröbner basis method (which is doubly-exponential-time in n) [Vas04]. We know that ideals even with 3-generators (analogously $k = 4$) already capture the whole ideal-membership problem [Sap21].

The algebraic-geometric approach to tackle $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ has been explored in [BMS13; Gup14; Muk16; Guo21]. The families which satisfy a certain Sylvester–Gallai configuration (called SG-circuits) is the harder case which is conjectured to have constant transcendence degree [Gup14, Conj. 1]. Non-SG circuits is the case where the nonzeroness-certifying-path question reduces to radical-ideal non-membership questions [GS20]. This is really a variety question where one could use algebraic-geometry tools to design a poly-time blackbox PIT. In fact, very recently, Guo [Guo21] gave a s^{δ^k}-time PIT by constructing explicit variety evasive subspace families. Unfortunately, this is not the case in the ideal non-membership; this scenario makes it much harder to solve $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$. From this viewpoint, radical-ideal-membership explains well why the intuitive $\Sigma^{[k]}\Pi\Sigma$ methods do not extend to $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$.

Interestingly, Forbes [For15] found a quasipolynomial-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ using shifted-partial derivative techniques; but it naively fails when one replaces the $\wedge$-gate by Π (because the 'measure' becomes too large). The duality trick of [Sax08]

completely solves whitebox PIT for $\Sigma\wedge\Sigma\wedge$, by transforming it to a read-once oblivious ABP (ROABP); but it is inapplicable to our models with the top Π-gate (due to large Waring rank and ROABP-width). A priori, our models are incomparable to ROABP, and thus the famous PIT algorithms for ROABP [FS13a; FSS14; GKS17] are not expected to help either.

Similarly, a naive application of the *Jacobian* and *certifying path* technique from [Agr+16] fails for our models because it is difficult to come up with a *faithful* map for constant-variate reduction. Kumar and Saraf [KS16] crucially used that the computed polynomial has low individual degree (such that [DSY10] can be invoked), while in [KS17] they exploit the low algebraic rank of the polynomials computed below the top Π-gate. Neither of them hold in general for our models. Very recently, Peleg and Shpilka [PS21] gave a poly-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$, via incidence geometry (e.g. Edelstein-Kelly theorem involving 'quadratic' polynomials), by solving [Gup14, Conj. 1] for $k = 3, \delta = 2$. The method seems very strenuous to generalize even to 'cubic' polynomials ($\delta = 3 = k$).

PIT for other models. Blackbox PIT algorithms for many restricted models are known. Egs. ROABP related models [RS05; JQS10; Agr+15; GKS17; Gur+17; FSS14; And+18], log-variate circuits [FGS18; BS21], and non-commutative models [Gar+16; LMP19].

4.2.2 Some basic tools and notations

The analytic tools used in this chapter are inspired from power series, Wronskian (linear dependence) [KPT15, Theorem 7] [Kay+15], Jacobian (algebraic dependence) [BMS13; Agr+16; PSS18], and logarithmic derivative operator $\mathrm{dlog}_z(f)$.

We will be working with the division operator (e.g. $\mathsf{R}(z)$, over a certain ring R). However, the divisions do not come for free as they require invertibility with respect to z throughout (again landing us in $\mathsf{R}[[z]]$. For circuit classes, C and D, we define class

$$\mathscr{C}/\mathscr{D} := \{f/g \mid f \in \mathscr{C}, \mathscr{D} \ni g \neq 0\}.$$

Similarly $\mathscr{C}\cdot\mathscr{D}$ to denotes the class taking respective products. We will also use rank, in the fraction field $\mathbb{F}(x)$, which conveys

the obvious meaning of linear rank.

4.3 PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ Circuits

We solve the PIT for a more general model than $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ by solving the following problem.

Problem 4.3.1 Let $\{T_i \mid i \in [m]\}$ be $\Pi\Sigma\Pi^{[\delta]}$ circuits of (syntactic) degree at most d and size s. Let the transcendence degree of T_i's, $\mathrm{trdeg}_{\mathbb{F}}(T_1, \ldots, T_m) = k \ll s$. Further, $C(x_1, \ldots, x_m)$ be a circuit of (size+deg) $< s'$. Design a blackbox PIT algorithm for $C(T_1, \ldots, T_m)$.

Trivially, $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ is a very special case of the above setting.

Solution to Problem 4.3.1. Let $\mathbf{T} := \{T_1, \ldots, T_m\}$. Let $\mathbf{T}_k := \{T_1, \ldots, T_k\}$ be a transcendence basis. For $T_i = \prod_j g_{ij}$, we denote the set

$$L(T_i) := \{g_{ij} \mid j\} .$$

We sketch the road-map of the solution below.

1. Construct a map Φ which is *faithful* (Lemma 2.8.3). To do that, we –
   - find an explicit homomorphism $\Psi : \mathbb{F}[x] \to \mathbb{F}[x, z]$ such that $\Psi(\mathcal{J}_x(\mathbf{T}))$ is of a 'nice' form (product of small $\Pi\Sigma\Pi^{[\delta]}$ and $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits),
   - fix x suitably, in the image, to get a composed map $\Psi' : \mathbb{F}[x] \longrightarrow \mathbb{F}[z]$ such that
   $$\mathrm{rank}_{\mathbb{F}(x)}\mathcal{J}_x(\mathbf{T}) = \mathrm{rank}_{\mathbb{F}(z)}\Psi'(\mathcal{J}_x(\mathbf{T})) ,$$
   - extend this map to $\Phi : \mathbb{F}[x] \longrightarrow \mathbb{F}[z, y, t]$ such that
   $$\Phi : x_i \mapsto (\sum_{j=1}^{k} y_j t^{ij}) + \Psi'(x_i) .$$
2. We show that the map Φ can be *efficiently* constructed using a scaling and shifting map Ψ, which is eventually fixed by the hitting set H', of a polynomial size $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit, which defines Ψ'.

3. Overall, $\Phi(f)$ is a $k+2$-variate polynomial. It can be shown that $f \equiv 0 \iff \Phi(f) \equiv 0$. And, for $\Phi(f)$, a trivial hitting set exists.

Wlog, $\mathscr{J}_x(\mathbf{T})$ is full rank with respect to the variable set $x_k = (x_1, \dots, x_k)$. Thus, by assumption, $J_{x_k}(\mathbf{T}_k) \neq 0$ (for notation, see Chapter 2).

Recall, we want to construct a Ψ such that $\Psi(J_{x_k}(\mathbf{T}_k))$ has an 'easier' PIT. We have the following identity (see Lemma 2.8.4) where $T_i = \prod_j g_{ij}$:

$$J_{x_k}(\mathbf{T}_k) = \sum_{g_1 \in L(T_1), \dots, g_k \in L(T_k)} \left(\frac{T_1 \dots T_k}{g_1 \cdots g_k}\right) \cdot J_{x_k}(g_1, \dots, g_k)\,. \quad (4.1)$$

❑ The homomorphism Ψ

To ensure the invertibility of all $g \in \bigcup_i L(T_i)$, consider

$$h := \prod_{i \in [k]} \prod_{g \in L(T_i)} g = \prod_{i \in [\ell]} g,$$

where $g \in \bigcup_i L(T_i)$, and $\ell \leq k \cdot s$. Note that, $\deg(h) \leq d \cdot k \cdot s$, and h is computable by a $\Pi\Sigma\Pi$ circuit of size $O(s)$. Theorem 2.7.5 gives the relevant polynomial-time hitting set $\mathscr{H} \subseteq \mathbb{F}^n$, which contains an evaluation point $a = (a_1, \dots, a_n)$, such that, $h(a) \neq 0$ implying $g(a) \neq 0$, for all $g \in \bigcup_i L(T_i)$.

Since, we are in the blackbox setting, we *do not* have individual access of g, to verify for the correct a. Thus, we try out all $a \in \mathscr{H}$, to see whichever works. If the input polynomial f is non-zero, then one such a must exist. This search adds a *multiplicative* blowup of $\mathrm{poly}(s)$, since the size of $\mathscr{H}$ is $\mathrm{poly}(s)$.

Fix an $a = (a_1, \cdots, a_n) \in \mathscr{H}$ and define $\Psi : \mathbb{F}[x] \to \mathbb{F}[x, z]$ as

$$\Psi : x_i \mapsto z \cdot x_i + a_i\,.$$

Let us denote the ring $\mathsf{R}[x]$, where $\mathsf{R} := \mathbb{F}[z]/\langle z^D \rangle$, and $D := k \cdot (d-1) + 1$. Being 1-1, Ψ is clearly a non-zero preserving map. Moreover, we claim the following.

Claim 4.3.1 $J_{x_k}(\mathbf{T}_k) = 0 \iff \Psi(J_{x_k}(\mathbf{T}_k)) = 0$, in $\mathsf{R}[x]$.

Proof. As $\deg(T_i) \leq d$, each entry of the matrix can be of degree at most $d-1$; therefore $\deg(J_{x_k}(\mathbf{T}_k)) \leq k(d-1) = D-1$.

Thus, $\deg_z(\Psi(J_{x_k}(\mathbf{T}_k))) < D$. Hence, the conclusion. □

Equation 4.1 implies that

$$\Psi(J_{x_k}(\mathbf{T}_k)) \;=\; \Psi(T_1 \cdots T_k) \cdot \sum_{g_1 \in L(T_1), \ldots, g_k \in L(T_k)} \frac{\Psi(J_{x_k}(g_1, \ldots, g_k))}{\Psi(g_1 \cdots g_k)} \,. \tag{4.2}$$

As T_i has product fanin s, the top-fanin in the sum in Equation 4.2 can be at most s^k. Then define,

$$\tilde{F} \;:=\; \sum_{g_1 \in L(T_1), \ldots, g_k \in L(T_k)} \frac{\Psi(J_{x_k}(g_1, \ldots, g_k))}{\Psi(g_1 \cdots g_k)} \,, \text{ in } \mathsf{R}[x]. \tag{4.3}$$

Some immediate questions are:

1. Why is $\tilde{F}$ well-defined?
2. How does $\tilde{F}$ help in the whole scenario, since, it may seem that we are working with the *partial* information of $\Psi(J_{x_k}(\mathbf{T}_k))$, in Equation 4.2? See Claim 4.3.2.

We discuss the above issues below.

Well-definability of $\tilde{F}$. Note that,

$$\Psi(g_i) \equiv \Psi_1(g_i) \bmod z \neq 0 \implies \frac{1}{\Psi(g_1 \cdots g_k)} \in \mathbb{F}[[x, z]] \,.$$

Thus, RHS is an element in $\mathbb{F}[[x, z]]$, and taking mod z^D, it is in $\mathsf{R}[x]$. We remark that instead of minimally reducing mod z^D, we will work with an $F \in \mathbb{F}[z, x]$ such that $F = \tilde{F}$ in the ring $\mathsf{R}[x]$. Further, we ensure that the degree of z is polynomially bounded. Moreover, the following claim is trivial.

Claim 4.3.2 Over $\mathsf{R}[x]$, $\Psi(J_{x_k}(\mathbf{T}_k)) = 0 \iff F = 0$.

Proof sketch. This follows from the invertibility of $\Psi(T_1 \cdots T_k)$ in $R[x]$. □

❑ Size bound for circuit size of F

Next, Claim 4.3.3 bounds the size of F.

Claim 4.3.3 (Main size bound) $F \in \mathsf{R}[x]$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size $(s3^{\delta})^{O(k)}$.

The proof studies the two parts of Equation 4.3—

1. The numerator $\Psi(J_{x_k}(g_1,\dots,g_k))$ has $O(3^{\delta}2^k k!ks)$ size $\Sigma\wedge\Sigma\Pi^{[\delta-1]}$-circuit (see Lemma 4.3.1), and
2. $1/\Psi(g_1\cdots g_k)$, for $g_i \in L(T_i)$ has $(s3^{\delta})^{O(k)}$ size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit; both in $\mathsf{R}[x]$ (see Lemma 4.3.2).

We need the following two claims to prove the numerator size bound.

Claim 4.3.4 Let $g_i \in L(T_i)$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ of size at-most s, then the polynomial $J_{x_k}(g_1,\dots,g_k)$ is computable by $\Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$ of size $O(k!\,ks)$.

proof-sketch. Each entry of the matrix has a degree at most $\delta-1$. Trivial expansion gives $k!$ top-fanin where each product (of fanin k) has size $\sum_i \text{size}(g_i)$. As, $\text{size}(T_i) \le s$, trivially each $\text{size}(g_i) \le s$. Therefore, the total size is $k!\cdot\sum_i \text{size}(g_i) = O(k!\,ks)$. □

Claim 4.3.5 Let $g \in \Sigma\Pi^{[\delta]}$, then, $\Psi(g) \in \Sigma\Pi^{[\delta]}$ of size $3^{\delta}\cdot\text{size}(g)$ (for $n \gg \delta$).

Proof. Since, $\sum_i e_i \le \delta$, each monomial x^e of degree δ, can produce

$$\prod_i (e_i+1) \le \left(\frac{\sum_i e_i + n}{n}\right)^n \le (\frac{\delta}{n}+1)^n$$

many monomials, by AM-GM inequality. As $\delta/n \to 0$, we have $(1+\delta/n)^n \to e^{\delta}$. Since, $e < 3$, the upper bound follows. □

Lemma 4.3.1 (Numerator size) $\Psi(J_{x_k}(g_1,\dots,g_k))$ *can be computed by* $\Sigma\wedge\Sigma\Pi^{[\delta-1]}$ *circuits of size* $O(3^{\delta}\,2^k k\,k!s) =: s_2$.

Proof. In Claim 4.3.4, we showed that $J_{x_k}(g_1,\dots,g_k)$ can be computed by a $\Sigma^{[k!]}\Pi\Sigma\Pi^{[\delta-1]}$ circuit of size $O(k!ks)$. Moreover,

for a $g \in \Sigma\Pi^{[\delta-1]}$, we have $\Psi(g) \in \Sigma\Pi^{[\delta-1]}$ of size at most $3^\delta \cdot \text{size}(g)$, in $\mathsf{R}[x]$ due to Claim 4.3.5).

Combining these, one concludes that

$$\Psi(J_{x_k}(g_1, \ldots, g_k)) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$$

of size $O(3^\delta k!ks)$. We *convert* the Π-gate to $\wedge$ gate using Waring identity (Lemma 2.2.3) which blowsup the size by a multiple of 2^{k-1}. Thus, $\Psi(J_{x_k}(g_1, \ldots, g_k)) \in \Sigma \wedge \Sigma\Pi^{[\delta-1]}$, of size $O(3^\delta 2^k k k!s)$. □

In the following lemma, using power series expansion of expressions like $1/(1 - a \cdot z)$, we conclude that $1/\Psi(g)$ has a small $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit, which would further imply the same for $1/\Psi(g_1 \cdots g_k)$.

Lemma 4.3.2 (Denominator size) *Let $g_i \in L(T_i)$. Then, $1/\Psi(g_1 \cdots g_k)$ can be computed by a $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size $s_1 := (s3^\delta)^{O(k)}$, in $\mathsf{R}[x]$.*

Proof. Let $g \in L(T_i)$ for some i. Assume, $\Psi(g) = A - z \cdot B$, for some $A \in \mathbb{F}$ and $B \in \mathsf{R}[x]$ of degree δ, with $\text{size}(B) \leq 3^\delta \cdot s$, from Claim 4.3.5. Note that, in $\mathsf{R}[x]$,

$$\frac{1}{\Psi(g)} = \frac{1}{A(1 - \frac{B}{A} \cdot z)} = \frac{1}{A} \cdot \sum_{i=0}^{D-1} \left(\frac{B}{A}\right)^i \cdot z^i . \tag{4.4}$$

As, $\text{size}(B^i)$ has a trivial $\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathsf{R}[x]$) of size $\leq 3^\delta \cdot s + i$; summing over $i \in [D-1]$, the overall size is at most $D \cdot 3^\delta \cdot s + O(D^2)$. As $D < k \cdot d$, we conclude that $1/\Psi(g)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$ of size $\text{poly}(s \cdot k \cdot d3^\delta)$, in $\mathsf{R}[x]$. Multiplying k-many such products directly gives an upper bound of $(s \cdot 3^\delta)^{O(k)}$, using Lemma 2.6.10 (basically, waring identity). □

Proof of Claim 4.3.3. Combining Lemmas 4.3.1-4.3.2, observe that $\Psi(J_{x_k}(\cdot))/\Psi(\cdot)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size at most $(s_1 \cdot s_2)^2 = (s \cdot 3^\delta)^{O(k)}$, in $\mathsf{R}[x]$, using Lemma 2.6.10. Summing up at most s^k many terms (by definition of F), the size still remains $(s \cdot 3^\delta)^{O(k)}$. □

Finally, we construct the hitting set H', as follows.

The hitting set H'. By $J_{x_k}(\mathbf{T}_k) \neq 0$, and Claims 4.3.1-4.3.2, we have $F \neq 0$, in $\mathsf{R}[x]$. We want to find $H' \subseteq \mathbb{F}^n$, such that

$$\Psi(J_{x_k}(\mathbf{T}_k))|_{x=a} \neq 0\,,$$

for some $a \in H'$, which will ensure the rank-preservation. Since, Claim 4.3.3 shows that F has $s^{O(\delta k)}$ size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit in $\mathsf{R}[x]$, by Theorem 2.7.8, it also immediately gives the hitting set H' in time $s^{O(\delta^2 k \log s)}$.

We also require the syntactic degrees, not to blowup much. The next paragraph is dedicated to argue that this is indeed the case.

Degree bound. As, syntactic degree of T_i are bounded by d, and Ψ maintain $\deg_x = \deg_z$, we must have

$$\deg_z(\Psi(J_{x_k}(g_1,\ldots,g_k)) \;=\; \deg_x(J_{x_k}(g_1,\ldots,g_k)) \;\leq\; D-1\,.$$

Note that, Lemma 4.3.1 actually works in $\mathbb{F}[x,z]$ and thus there is no additional degree-blow up (in z). However, there is some degree blowup in Lemma 4.3.2, due to Equation 4.4.

Note that Equation 4.4 shows that in $\mathsf{R}[x]$,

$$\frac{1}{\Psi(g)} = \left(\frac{1}{A^D}\right)\cdot\left(\sum_{i=0}^{D-1} A^{D-1-i}z^i\cdot B^i\right) =: \frac{p(x,z)}{q},$$

where $q = A^D$. We think of $p \in \mathbb{F}[x,z]$ and $q \in \mathbb{F}$. Note, $\deg_z(\Psi(g)) \leq \delta$ implies $\deg_z(p) \leq \deg_z((Bz)^{D-1}) \leq \delta\cdot(D-1)$.

Finally, denote $1/\Psi(g_1\cdots g_k) =: P_{g_1,\ldots,g_k}/Q_{g_1,\ldots,g_k}$, in $\mathsf{R}[x]$. This is just multiplying k-many (p/q)'s; implying a degree blowup by a multiple of k. In particular – $\deg_z(P_{(\cdot)}) \leq \delta\cdot k\cdot(D-1)$ Thus, in Equation 4.3, summing up s^k-many terms gives an expression (over $\mathsf{R}[x]$):

$$F \;=\; \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \Psi(J_{x_k}(g_1,\ldots,g_k))\cdot\left(\frac{P_{g_1,\ldots,g_k}}{Q_{g_1,\ldots,g_k}}\right) \;=:\; \frac{P(x,z)}{Q}\,.$$

Verify that $Q \in \mathbb{F}$. The degree of z also remains bounded by

$$\max_{g_i\in L(T_i),i\in[k]} \deg_z(P_{g_1,\ldots,g_k}) + \delta k \leq \mathrm{poly}(s).$$

Using the degree bounds, we finally have $P \in \mathbb{F}[x, z]$ as a $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathbb{F}(z)$) of size

$$n^{O(\delta)}(s3^{\delta})^{O(k)} = 3^{O(\delta k)}s^{O(k+\delta)} =: s_3 .$$

❑ Final algorithm

We know that

$$C(T_1, \dots, T_m) = 0 \iff E := \Phi(C(T_1, \dots, T_m)) = 0.$$

We wanted to *construct* a set $H' \subseteq \mathbb{F}^n$, such that the action $P(H', z) \neq 0$. By previous discussion and using [For15] (Theorem 2.7.8), we conclude that it has $s^{O(\delta \log s_3)} = s^{O(\delta^2 k \log s)}$ size hitting set which is constructible in a similar time.

Since, H' can be constructed in $s^{O(\delta^2 k \log s)}$-time, it is trivial to find a hitting set for $E|_{H'}$ (which is just a $k+2$-variate polynomial with the aforementioned degree bounds). Hence, the construction of Φ follows, making $\Phi(f)$, a $(k+2)$-variate polynomial. Finally, by the obvious degree bounds of y, z, t from the definition of Φ, we get the blackbox PIT algorithm with time-complexity $s'^{O(k)} \cdot s^{O(\delta^2 k \log s)}$ time.

In particular, we get the blackbox PIT algorithm with time-complexity $s^{O(\delta^2 k \log s)}$; finishing Theorem 4.2.1 (a). □

Remark 4.3.1

1. As Jacobian Criterion (Theorem 2.8.2) holds when the characteristic is $> d^{\text{trdeg}}$, it is easy to conclude that our theorem holds for all fields of char $> d^k$.
2. The above proof gives an efficient reduction from blackbox PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits to $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits. In particular, a poly-time hitting set for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits would put PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ in P.
3. An alternative proof based on the DiDI-technique, introduced in the same paper [DDS21a] directly gives a blackbox algorithm, but the complexity is *exponentially* worse (in terms of k in the exponent) for its recursive blowups.

4.4 PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$ Circuits

The proof of Theorem 4.2.1(b) is similar to the one we discussed in section 4.3. Here we sketch the proof, stating some relevant changes. Similar to Theorem 4.2.1(a), we generalize this theorem and prove for a much bigger class of polynomials.

Problem 4.4.1 Let $\{T_i \mid i \in [m]\}$ be $\Pi\Sigma\wedge$ circuits of (syntactic) degree at most d and size s. Let the transcendence degree of T_i's, $\mathrm{trdeg}_{\mathbb{F}}(T_1, \ldots, T_m) =: k \ll s$. Further, $C(x_1, \ldots, x_m)$ be a circuit of size + degree $< s'$. Design a blackbox PIT algorithm for $C(T_1, \ldots, T_m)$.

It is trivial to see that $\Sigma^{[k]}\Pi\Sigma\wedge$ is a very special case of the above settings. We will use the same idea (& notation) as in Theorem 4.2.1(a), using the Jacobian technique. The main idea is to come up with Ψ map, and correspondingly the hitting set H'. If $g \in L(T_i)$, then $\mathrm{size}(g) \leq O(dn)$. The D (and hence $R[x]$) remains as before. Claims 4.3.1-4.3.2 hold similarly. We will construct the hitting set H' by showing that F has a small $\Sigma\wedge\Sigma\wedge$ circuit in $R[x]$.

Note that, Claim 4.3.4 remains the same for $\Sigma\wedge\Sigma\wedge$ (implying the same size blowup). However, Claim 4.3.5, the size blowup is $O(d\,\mathrm{size}(g))$, because each monomial x^e can only produce $d+1$ many monomials. Therefore, similar to Lemma 4.3.2, one can show that $\Psi(J_{x_k}(g_1, \ldots, g_k)) \in \Sigma\wedge\Sigma\wedge$, of size $O(2^k k!kds)$. Similarly, the size in Lemma 4.3.1 can be replaced by $s^{O(k)}$. Therefore, we get (similar to Claim 4.3.3):

Claim 4.4.1 $F \in R[x]$ has $\Sigma\wedge\Sigma\wedge$-circuit of size $s^{O(k)}$.

Next, the degree bound also remains the same. Following the same footsteps, it is not hard to see that while degree bound on z remains $\mathrm{poly}(ksd)$. Therefore, $P \in \mathbb{F}[x, z]$ has $\Sigma\wedge\Sigma\wedge$-circuit of size $s^{O(k)}$.

We want to *construct* a set $H' \subseteq \mathbb{F}^n$ such that the action $P(H', z) \neq 0$. By Lemma 2.7.9, we conclude that it has $s^{O(k \log\log s)}$ size hitting set which is constructible in a similar time. Hence, the construction of map Φ and the theorem follows (from z-degree bound).

Solution to Problem 4.4.1. We know that

$$C(T_1, \ldots, T_m) = 0 \iff E := \Phi(C(T_1, \ldots, T_m)) = 0.$$

Since, H' can be constructed in $s^{O(k \log\log s)}$ time, it is trivial to find the hitting set for $E|_{H'}$ (which is just a $k+2$-variate polynomial with the aforementioned degree bounds). The final hitting set for E can be constructed in $s'^{O(k)} \cdot s^{O(k \log\log s)}$ time. □

4.5 Discussion

At hindsight, the Jacobian criterion did the magic for bounded depth-4 circuits; however, the inverse power series identity played a crucial role to convert Π-gate to ∧. Unfortunately, it is not clear what happens when we work with the inverse of a general sparse polynomial (or its shift): $\frac{1}{\Sigma\Pi(x+a)}$. We conclude this chapter by asking the following related open question.

Rational identity testing for sum-of-inverse-of-sparse
Design an efficient PIT for rational functions of the form $\Sigma\,(1/\Sigma \wedge \Sigma)$ or $\Sigma\,(1/\Sigma\Pi)$ (for *un*bounded top-fanin).

Future Directions in Algebraic Complexity

5

" To make progress, we have to constantly go back to the facts, acknowledge our errors, and move."

— Abhijit Banerjee, Good Economics for Hard Times.

We investigate the significance of roots and non-roots in algebraic complexity (Q2, Q4 and Q5). Chapter 3 establishes that studying the number of real roots of univariate polynomials for sum-of-squares representation (respectively cubes) is fecund, and leads to strong lower bound results. While, Chapter 4 demonstrates that algebraic dependence with its inherent analytic nature can lead to efficient derandomization algorithms. Both the chapters leave scope for future directions. These are some obvious steps to take towards advancing the current state-of-the-art.

5.1 τ-conjecture and SOS Lower Bounds

We showed that an SOC-hard family will lead to complete derandomization of PIT. This immediately raises the following question.

Open Problem 3.1

Does the existence of an SOS-hard family solve PIT completely?

The current proof technique fails to reduce from cubes to squares. Basically, in the proof of Theorem 3.4.2, the main goal was to devise an amply hard polynomial with a *constant* number of variables only. Even if we are able to get an optimal SOS decomposition of $d/2$ with polynomial blowup in the top-fanin, the current transformations would eventually lead

to the support-union size to be $\binom{k+kn/2}{k}$, via naive monomial counting. It is not hard to show that for a constant $k \geq 3$,

$$\binom{k+kn/2}{k} \geq (n+1)^k > d\,.$$

To prove this, note that, $\binom{k+kn/2}{k} = (1+n/2)\cdot\binom{k+kn/2-1}{k-1} > 2(1+n/2)\cdot\binom{(k-1)\cdot(n/2+1)}{k-1}$. Here, we used the fact that $\binom{k+kn/2-1}{k-1} > 2\cdot\binom{(k-1)\cdot(n/2+1)}{k-1}$, for large enough n, and constant $k \geq 3$. Inductively, the rest follows.

So, there could be two approaches to the above problem –

1. either one needs to better understand the decomposition, and characterize all $kn/2$-degree monomials that appear, to show a better bound (than the current naive monomial-counting), or,
2. one has to come up with an entirely different lifting-map.

In either case, this appears to be intriguingly challenging. The other related question is to show some concrete lower bound on the support-sum for the sum-of-constantly-many-squares. In particular, the following question seems to be tractable.

Open Problem 3.2

Come up with an explicit polynomial f such that $S(f) = \Theta(d)$, when the SOS-representation is restricted to the top-fanin $s = 3$.

Spurred by the intriguing connection between SOS-τ-conjecture and two fundamental lower bound questions, namely matrix rigidity and VP vs. VNP, one could ask to prove the 'simplest' SOS-τ-conjecture.

Open Problem 3.3

If the polynomials $f(x)$ and $g(x)$ have sparsity at most s, then the number of real roots of the polynomial $h := fg+1$, can be at most $O(s)$

In fact, in any 'relevant' model, understanding deeper connections between roots, and representations, would be nice

to explore in the near future.

5.2 Some Interesting PIT Questions

Here are some natural questions in the spirit of the results from Chapter 4.

Open Problems 3.4

1. Can we design a polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$?
2. Design a polynomial-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits (i.e. unbounded top-fanin)?
3. Can we solve PIT for $\Sigma^{[k]}\Pi\Sigma\wedge^{[2]}$ efficiently, i.e. in polynomial-, or, quasipolynomial-time?

Moreover, our Jacobian technique does not infer anything about the *rank* of the circuit, as informally defined in Chapter 4. This remains an exciting open question to prove any interesting rank bound for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, which only (perhaps *polynomially*) depends on k and δ, as conjectured in [Gup14], and discussed in the introduction of this chapter.

Results in Geometric Complexity Theory

De-bordering approximative depth-3 circuits

6

"Although this may seem a paradox, exact science is dominated by the idea of approximation."

— Bertrand Russell, World Unity, Vol. IX, 3rd edition (1931).

Abstract. *Border* complexity of polynomials plays an integral role in GCT (Geometric complexity theory) approach to $\mathsf{P} \neq \mathsf{NP}$. It tries to formalize the notion of approximation (of polynomials) via limits (Bürgisser FOCS'01). This raises the open question $\overline{\mathsf{VP}} \stackrel{?}{=} \mathsf{VP}$; as the approximation involves *exponential precision* which may not be efficiently simulable. Recently (Kumar ToCT'20) proved the universal power of the border of top-fanin-2 depth-3 circuits ($\overline{\Sigma^{[2]}\Pi\Sigma}$). Here we answer some related open questions. We show that the border of bounded top-fanin depth-3 circuits ($\overline{\Sigma^{[k]}\Pi\Sigma}$ for constant k) is relatively easy– it can be computed by a polynomial size algebraic branching program (ABP). There were hardly any de-bordering results known for prominent models before our result.

Our de-bordering paradigm is a multi-step process; in short we call it DiDIL –divide, derive, induct, with limit. It 'almost' reduces $\overline{\Sigma^{[k]}\Pi\Sigma}$ to special cases of read-once oblivious algebraic branching programs (ROABPs) in any-order.

This chapter is based on the first half of the article titled *Demystifying the border of depth-3 algebraic circuits*, which is a joint work with Prateek Dwivedi and Nitin Saxena, that appeared in FOCS 2021 [DDS21b], and *invited* in the special SICOMP issue on FOCS'21.

6.1 Why Care About Upper Bounds?

One of the fundamental questions in GCT is whether $\overline{\mathsf{VP}} \stackrel{?}{=} \mathsf{VP}$ [Mul12]. Confirmation or refutation of this question has multiple consequences, both in the algebraic complexity and at the frontier of algebraic geometry. If $\mathsf{VP} = \overline{\mathsf{VP}}$, then any proof of $\mathsf{VP} \neq \mathsf{VNP}$ will in fact also show that $\mathsf{VNP} \not\subseteq \overline{\mathsf{VP}}$, as conjectured in [Mul12]; however a refutation would imply that any realistic approach to the VP vs. VNP conjecture would even

have to separate the permanent from the families in $\overline{\mathsf{VP}}\backslash\mathsf{VP}$ (and for this, one needs a far better understanding than the current state of the art).

The other significance of the upper bound result arises from the *flip* [Mul10; Mul12] whose basic idea in a nutshell is to understand the theory of upper bounds first, and then use this theory to prove lower bounds later. Taking this further to the realm of algorithms: showing de-bordering results, for even restricted classes (e.g., depth-3, small-width ABPs), could have potential identity testing implications. For details, see Chapter 7.

De-bordering results in GCT are in a very nascent stage; for example, the boundary of 3×3 determinants was only recently understood [HL16]. Note that here both the number of variables n and the degree d are constant. In this work, however, we target polynomial families with both n and d unbounded. So getting exact results about such border models is highly nontrivial considering the current state of the art.

Known de-bordering results. The exponential degree dependence of ϵ [Bür04; Bür20] suggests us to look for separation of restricted complexity classes or try to upper bound them by some other means. In [BIZ18], the authors showed that $\mathsf{VBP}_2 \subseteq \overline{\mathsf{VBP}_2} = \overline{\mathsf{VF}}$; here VBP_2 denotes the class of polynomials computed by width-2 ABP. Surprisingly, we also know that $\mathsf{VBP}_2 \subseteq \mathsf{VF} = \mathsf{VBP}_3$ [BC92; AW16]. Very recently, [Blä+] showed polynomial gap between ABPs and border-ABPs, in the trace model, for noncommutative and also for commutative monotone settings (along with $\mathsf{VQP} \neq \overline{\mathsf{VNP}}$).

Unfortunately, de-bordering results in GCT are in a very nascent stage. Some well known de-bordering results are –

1. the boundary of 3×3 determinants [HL16],
2. $\overline{\mathsf{VBP}_2} \subseteq \mathsf{VF}$ [BIZ18]; here VBP_2 denotes the class of polynomials computed by width-2 ABP,
3. $\overline{\mathsf{VBP}_{\text{noncom}}} = \mathsf{VBP}_{\text{noncom}}$, i.e. VBP, in the non-commutative setting is *closed* under taking limit [Nis91],
4. $\overline{\Sigma\wedge\Sigma} \subseteq \mathsf{VBP}$ [BDI21] .

We will talk about some important de-bordering proofs in the next section.

6.2 De-bordering Simple Models

In this section, we will discuss known de-bordering results of restricted models like product of sum of univariates and ARO. Some of these facts will be crucially used in our main de-bordering result as well.

Polynomials approximated by ΠΣ can be easily de-bordered [BIZ18, Proposition A.12]. In fact, it is the only constructive de-bordering result known so far. We extend it to show that same holds for polynomials approximated by ΠΣ∧ circuits. In fact, we start it by showing a much more general theorem.

Let $\mathscr{C}$ and $\mathscr{D}$ be two classes in $\mathbb{F}[x]$. Consider the bloated-class $(\mathscr{C}/\mathscr{C}) \cdot (\mathscr{D}/\mathscr{D})$, which has elements of the form $(g_1/g_2) \cdot (h_1/h_2)$, where $g_i \in \mathscr{C}$ and $h_i \in \mathscr{D}$ $(g_2 h_2 \neq 0)$. One can also similarly define its border (which will be an element in $\mathbb{F}(x)$). Here is an important observation.

Lemma 6.2.1 $\overline{(\mathscr{C}/\mathscr{C}) \cdot (\mathscr{D}/\mathscr{D})} = (\overline{\mathscr{C}}/\overline{\mathscr{C}}) \cdot (\overline{\mathscr{D}}/\overline{\mathscr{D}})$.

Proof. **To show** $\subseteq$**:** Suppose $(g_1/g_2)\cdot(h_1/h_2) = f + \epsilon \cdot Q$, where $Q \in \mathbb{F}(x, \epsilon)$, and $f \in \mathbb{F}(x)$. Let $\mathrm{val}_\epsilon(g_i) =: a_i$ and $\mathrm{val}_\epsilon(h_i) =: b_i$. Denote, $g_i =: \epsilon^{a_i} \cdot \tilde{g}_i$, similarly $\tilde{h}_i$. Further, assume $\tilde{g}_i =: \hat{g}_i + \epsilon \cdot \hat{g}_i'$; similarly for $\tilde{h}_i$, we define $\hat{h}_i \in \mathbb{F}[x]$. Note that $\hat{g}_i \in \overline{\mathscr{C}}$, similarly $\hat{h}_i \in \overline{\mathscr{D}}$.

So, LHS $= \epsilon^{a_1 - a_2 + b_1 - b_2} \cdot (\tilde{g}_1/\tilde{g}_2) \cdot (\tilde{h}_1/\tilde{h}_2)$. Since, limit of LHS, $\lim_{\epsilon \to 0}$ LHS, is well-defined, so $a_1 + b_1 - a_2 - b_2 \geq 0$. If it is ≥ 1, the limit in RHS is 0, and so $f = 0$. If $a_1 + b_1 - a_2 - b_2 = 0$, then

$$f = (\hat{g}_1/\hat{g}_2) \cdot (\hat{h}_1/\hat{h}_2) \in (\overline{\mathscr{C}}/\overline{\mathscr{C}}) \cdot (\overline{\mathscr{D}}/\overline{\mathscr{D}}) .$$

To show $\supseteq$**:** Suppose, $g_1/g_2 \in \overline{\mathscr{C}}/\overline{\mathscr{C}}$, for $g_i \in \overline{\mathscr{C}}$, and $h_1/h_2 \in \overline{\mathscr{D}}/\overline{\mathscr{D}}$, where $h_i \in \overline{\mathscr{D}}$. Let $\hat{g}_i = g_i + \epsilon \cdot R_i$, and $\hat{h}_i = h_i + \epsilon \cdot S_i$. Here, $\hat{g}_i$ and $\hat{h}_i$ are over $\mathbb{F}(\epsilon)$. Then,

$$(g_1/g_2) \cdot (h_1/h_2) = \lim_{\epsilon \to 0} (\hat{g}_1/\hat{g}_2) \cdot (\hat{h}_1/\hat{h}_2) \in \overline{(\mathscr{C}/\mathscr{C}) \cdot (\mathscr{D}/\mathscr{D})} .$$

□

Now, we show an important de-bordering result on ΠΣ∧ circuits.

Lemma 6.2.2 (De-bordering ΠΣ∧) *Consider a polynomial $f \in \mathbb{F}[x]$ which is approximated by ΠΣ∧ of size s over $\mathbb{F}(\epsilon)$. Then there exists a ΠΣ∧ (hence an ARO) of size s which exactly computes $f(x)$.*

Proof. We will show that

$$\overline{\Pi\Sigma\wedge} = \Pi\Sigma\wedge \subseteq \text{ARO}\,.$$

From Lemma 6.2.1 (and its proof), it follows that $\overline{\Pi\Sigma\wedge} \subseteq \Pi(\overline{\Sigma\wedge})$. However, we note that $\overline{\Sigma\wedge} = \Sigma\wedge$ and it does not change the size (as it can not increase the sparsity). Therefore, the size does not increase and further it is an ARO. Thus, the conclusion follows. □

Next we show that polynomials approximated by ARO can be easily de-bordered. To the best of our knowledge the following lemma was sketched in [For16]; also implicitly in [GKS17].

Lemma 6.2.3 (De-bordering ARO) *Consider a polynomial $f \in \mathbb{F}[x]$ which is approximated by ARO of size s in $\mathbb{F}(\epsilon)[x]$. Then, there exists an ARO of size s which exactly computes $f(x)$.*

Proof. By definition, there exists a polynomial $g = f + \epsilon Q$ computable by width w ARO over $\mathbb{F}(\epsilon)$. Note that $w \leq s$. In this proof, we will use the partial derivative matrix. With respect to any-order-prefix $y \subset x$, consider the partial derivative matrix $N(g)$. Using Lemma 2.6.5 and 2.6.6, we know that

$$\text{rank}_{\mathbb{F}(\epsilon)}(N(g)) \leq w\,.$$

This means the determinant of *any* $(w+1)\times(w+1)$ minor of $N(g)$ is identically zero. One can see that the entries of the minor are coefficients of monomials of g which are in $\mathbb{F}[\epsilon][x \setminus y]$. Thus, the determinant polynomial will remain zero even under the limit of $\epsilon = 0$. Since, $\lim_{\epsilon\to 0} g = f$, each minor (under limit) captures the partial derivative matrix of f of corresponding rows and columns. Thus, we get $\text{rank}_{\mathbb{F}}(N(f)) \leq w$. Lemma 2.6.6 shows that there exists an ARO, of width w over $\mathbb{F}$, which *exactly* computes f. □

An obvious consequence of Lemma 2.6.11 and Lemma 6.2.3 is the following de-bordering result.

Lemma 6.2.4 (De-bordering $\Sigma\wedge\Sigma\wedge$) *Consider a polynomial $f \in \mathbb{F}[x]$ which is approximated by $\Sigma\wedge\Sigma\wedge$ of size s over $\mathbb{F}(\epsilon)$, and syntactic degree D. Then there exists an* ARO *of size $O(sn^2D^2)$ which exactly computes $f(x)$.*

6.3 Border Depth-3 Circuits

Since, depth-2 circuits $(\Sigma\Pi, \Pi\Sigma)$ are closed under approximation, it is natural to study border of depth-3 circuits. Again, it is not hard to show that $\overline{\Pi\Sigma\Pi} = \Pi\Sigma\Pi$, which leaves us to understand $\overline{\Sigma\Pi\Sigma}$ circuits. Kumar [Kum20] showed that border depth-3 fanin-2 circuits are 'universal'; i.e. $\Sigma^{[2]}\Pi^{[D]}\Sigma$, over $\mathbb{C}(\epsilon)$, can approximate *any* homogeneous d-degree, n-variate polynomial [1]; though this expression requires an exceedingly large $D = \exp(n, d)$. It is easy to see that the same proof works for even *non*-homogeneous polynomials. For brevity, we state and prove it. We remark that our upper bound is *slightly* better than the bound achieved in [Kum20], because we use the best known Waring rank upper bound for a generic form [BT15].

1: This is not true in the classical sense. The inner product polynomial $\sum_{i=1}^{n} x_i y_i$, cannot be written as a depth-3 circuit of fan-in $n-1$; for a proof, see Theorem 2.9.1.

Theorem 6.3.1 *Let $P \in \mathbb{C}[x]$, be any n-variate, degree-d polynomial. Then, there is a $\Sigma^{[2]}\Pi\Sigma$ circuit $C \in \mathbb{C}(\epsilon)[x]$, of size $O(\binom{n+d}{d-1})$, such that*

$$C(x, \epsilon) \;=\; P + \epsilon \cdot Q\,,$$

where $Q \in \mathbb{C}[\epsilon, x]$.

Proof. Let $\tilde{P}$ be the *homogenized* version of P, i.e.,

$$\tilde{P} \;=\; x_0^d \cdot P\left(\frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0}\right)\,.$$

In the above, x_0 is a new variable. By definition, $\tilde{P}(x)$ is a homogeneous polynomial of degree d in $n+1$ variables.

Let $\mathrm{WR}(\tilde{P}) = m$. By [BT15][2],

2: In [BT15], they showed it over $\mathbb{R}$; but it is easy to see that the rank over $\mathbb{R}$ may be significantly *larger* than that over $\mathbb{C}$, because of the *complexification*. So trivially, the bound holds over $\mathbb{C}$.

$$m \;\leq\; 2 \cdot \left\lceil \frac{1}{n+1} \cdot \binom{n+d}{d} \right\rceil .$$

By definition, there are linear forms ℓ_i, such that

$$\tilde{P} \;=\; \sum_{i=1}^{m} \ell_i^d \,.$$

Now, consider

$$\begin{aligned} A(x_0, x_1, \ldots, x_n) \;&:=\; \prod_{i=1}^{m} (1 + \ell_i^d) \\ &= \prod_{i=1}^{m} \prod_{j=1}^{d} (\alpha_j + \ell_i) \,, \end{aligned}$$

where $\alpha_j \in \mathbb{C}$. Note that

$$A \;=\; 1 + \tilde{P} + B \,,$$

where $\deg(B) \geq 2d$. Now, replace each x_i, by $\epsilon \cdot x_i$, for a new variable ϵ, to get that

$$\prod_{i=1}^{m} \prod_{j=1}^{d} (\alpha_j + \epsilon \cdot \ell_i) \;=\; 1 + \epsilon^d \cdot \tilde{P} + \epsilon^{2d} \cdot R(x, \epsilon) \,.$$

Divide by ϵ^d, and rearrange to get

$$\begin{aligned} \tilde{P} + \epsilon^d \cdot R(x, \epsilon) \;&=\; -\epsilon^{-d} + \epsilon^{-d} \cdot \prod_{i=1}^{m} \prod_{j=1}^{d} (\alpha_j + \epsilon \cdot \ell_i) \\ &\in\; \Sigma^{[2]} \Pi^{[md]} \Sigma \,. \end{aligned}$$

Finally, substitute $x_0 = 1$; note that the top-fanin remains 2. Also, it is easy to see that

$$md \leq O\left(\frac{d}{n} \cdot \binom{n+d}{d}\right) = O\left(\frac{(n+d)!}{(d-1)! \cdot (n+1)!} \cdot \frac{n+1}{n}\right) = O\left(\binom{n+d}{d-1}\right) ,$$

as desired! □

6.4 Border Depth-3 Circuits: A Geometric View

In [Gup+16], the authors (implicitly) proposed a geometric method to approach Valiant's conjecture, via determining equations for *chow embedding* of the *secant* varieties; this was observed and explicitly mentioned in [Mul17]. For formal definitions and the statement, see below.

Notation. Let W be a complex vector space, and $X \subset \mathbb{P}W$ be an algebraic variety. $\mathbb{P}V$ denotes the projective space, and we denote $[v]$ as a corresponding point. Finally, we denote by S^dW, the space of polynomials of degree d on W^*

Secant variety. Now, define

$$\sigma_k^0(X) \;=\; \bigcup_{p_1,\ldots,p_k \in X} \langle p_1, \ldots, p_k \rangle \subset \mathbb{P}W,$$

where $\langle p_1, \ldots, p_k \rangle$ denotes the projective plane spanned by $p_1, \ldots, p_k$. Define the k-th secant variety of X to be

$$\sigma_k(X) \;=\; \overline{\sigma_k^0(X)} \subset \mathbb{P}W,$$

where the overline denotes closure in the Zariski topology.

Chow variety and Chow rank. Once we have defined the secant varieties, we come to one important variety, called the *Chow variety*. Informally, if one specializes to group of diagonal matrices and takes the orbit closure, one obtains the chow variety $\mathsf{Ch}_d(W) \subset \mathbb{P}S^dW$. Formally,

$$\mathsf{Ch}_d(W) := \{[z] \in \mathbb{P}S^dW \mid z = w_1 \ldots w_d, \text{ for } w_j \in W\}\,.$$

Therefore, one can define the *Chow rank* of a homogeneous polynomial f of degree d, denoted $\mathrm{rank}_{\mathsf{Ch}}(f)$, to be the minimum k such that $f = \sum_{i=1}^{k} \prod_{j=1}^{d} \ell_{ij}$, where ℓ_{ij} are linear forms. Often in the literature, $\mathrm{rank}_{\mathsf{Ch}}(f) = k$ is equivalently expressed as: the smallest k such that f (as a point) is in $\sigma_k^0(\mathsf{Ch}_d(W))$ (= set of points with $\mathsf{Ch}_d(W)$-rank at most k).

Moving to the border setting, one defines *Chow border rank*, $\overline{\text{rank}}_{\text{Ch}}(f)$, as the border analogue of the Chow rank. In other words,

$$\overline{\text{rank}}_{\text{Ch}}(f) = k \iff f \in \sigma_k(\text{Ch}_d(W))$$

where $\sigma_k(\text{Ch}_d(W))$ is the Zariski closure in $\mathbb{P}S^dW$ of $\sigma_k^0(\text{Ch}_d(W))$. For details, refer to [Lan15; Lan17]. These two ranks happen to exactly coincide with the depth-3 respectively border depth-3 homogeneous circuits of f, with the smallest fanin k.

The following theorem is the geometric rephrasing of the result that appeared in [Gup+16].

Theorem 6.4.1 ([Gup+16; Lan15]) *If for all but a finite number of m, for all r, n, with $r \cdot n = 2^{\omega(\sqrt{m}\log m)}$, one has*

$$[\ell^{n-m} \cdot \text{perm}_m] \notin \sigma_r(Ch_n(\mathbb{C}^{m^2+1}))\ ,$$

then, VP $\neq$ VNP *holds.*

From an algebraic complexity perspective, we are usually interested in the *non-homogeneous* setting. Moreover, Kumar's expression [Kum20, Section 3.1] is *strictly* non-homogeneous. However, with suitable padding (& $W = \mathbb{C}^{n+1}$), Kumar's result translates into geometric terms:

> *For any degree-d homogeneous polynomial f, there exists a linear form ℓ such that $[\ell^{m-d}f] \in \sigma_2(Ch_m(W))$, or equivalently, $\overline{\text{rank}}_{Ch}(\ell^{m-d}f) = 2$, where $m = \exp(n, d)$.*

Here is an important question which we will try to answer in this chapter.

Characterize Chow border rank-2 polynomials

Characterize $[\ell^{m-d}f] \in \sigma_2(\text{Ch}_m(W))$, when $m = \text{poly}(n, d)$. More simply put, find a 'nice' class $\mathscr{C}$, such that polynomial-sized $\overline{\Sigma^{[2]}\Pi\Sigma}$ circuits are in $\mathscr{C}$.

6.4.1 Our results

The universality result of border depth-3 fanin-2 circuits makes it imperative to study $\overline{\Sigma^{[2]}\Pi^{[d]}\Sigma}$, for $d = \text{poly}(n)$, and

understand its computational power. To start with, are polynomials in this class even 'explicit' (i.e. the coefficients are efficiently computable)? If yes, is $\overline{\Sigma^{[2]}\Pi^{[d]}\Sigma} \subseteq \mathsf{VNP}$? (See [GMQ16; Edi18] for more general questions in the same spirit.)

To our surprise, we show that the class is very explicit; in fact every polynomial in this class has a small ABP. We remark that our proof *does not* reveal the polynomial dependence on the ϵ-degree. However, this positive result could be thought as a 'baby step' towards $\overline{\mathsf{VP}} = \mathsf{VP}$. We assume the field $\mathbb{F}$ characteristic to be $= 0$, or large enough. For a detailed statement, see Theorem 6.5.1.

Theorem 6.4.2 (De-bordering depth-3 circuits) *For any constant k, $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \mathsf{VBP}$, i.e. any polynomial in the border of constant top-fanin size-s depth-3 circuits, can also be computed by a* $\mathrm{poly}(s)$ *size algebraic branching program (ABP).*

It is not hard to extend our results to constant top-fanin depth-4 circuits, see [DDS21b]. But we donot include the results in this thesis.

Remarks. 1. When $k = 1$, we have a better bound, because $\overline{\Pi\Sigma} = \Pi\Sigma$ [BIZ18, Proposition A.12], see Lemma 6.2.2.

2. The size of the ABP turns out to be $s^{\exp(k)}$. It is an interesting open question whether $f \in \overline{\Sigma^{[k]}\Pi\Sigma}$ has a subexponential ABP when $k = \Theta(\log s)$.

3. $\overline{\Sigma^{[k]}\Pi\Sigma}$ is the *orbit closure* of k-sparse polynomials [MS21, Theorem 1.31]. Separating the orbit and its closure of certain classes is the key difficulty in GCT. Theorem 6.4.2 is one of the first such results to demystify orbit closures (of constant-sparse polynomials).

6.4.2 Proof idea of Theorem 6.4.2

In this section, we sketch the proof of Theorem 6.4.2. The proof is analytic, based on induction on the top fan-in. It uses *logarithmic derivative*, and the power-series expansion, that emerges from the following *inverse* identity: $(1-x_1)^{-1} = \sum_{i\geq 0} x_1^i$; we call the technique as DiDIL (**Di** = Divide, **D**=Derive, **I** = Induct, **L** = Limit). We *essentially* reduce to the well-known 'wedge' model (as fractions, with unbounded top-fanin) and then 'interpolate' it.

The road-map. The base case when the top fan-in $k = 1$, i.e., we have a single product of affine linear forms, and we are interested in its border. It is not hard to see that the polynomial in the border is also just a product of appropriate affine forms; for details, refer to section 6.5). Now, suppose we have a depth-3 circuit of top fan-in 2, $g(x, \epsilon) = T_1 + T_2$, where each T_i is a product of affine linear forms. The goal is to somehow reduce this to the case of single summand.

Before moving forward, we remark that some ideas described below, can even be 'formally' incorrect! Nonetheless, this sketch is "morally'" correct and, the eventual road-map insinuates the strength of the DiDIL-technique.

For simplicity, let us assume that each linear form has a non-zero constant term (for instance by a random translation of the variables). Moreover, every variable x_i is replaced by $x_i \cdot z$ for a new variable z; this variable z is the 'degree counter' that helps to keep track of the degree of the polynomials involved. Now, dividing both sides by T_1, we get $g/T_1 = 1 + T_2/T_1$, and taking derivatives with respect to the variable z, we get

$$\partial_z(g/T_1) \;=\; \partial_z(T_2/T_1)\,.$$

This has reduced the number of summands on the right-hand side to 1, although each summand has become more complicated now, and we have no control on what happens as $\epsilon \to 0$!

Since, T_1 is invertible in the power series ring in z, T_2/T_1 is well-defined as well. Moreover, $\lim_{\epsilon\to 0} T_1$ exists (well *not really*, but formally a proper ϵ-scaling of it does, which suffices since derivative wrt z does not affect the ϵ-scaling!) and is non-zero. From this it follows that after some truncation wrt high degree z monomials, $\lim_{\epsilon\to 0} \partial_z(T_2/T_1)$ exists, and has a nice relation to the original limit of g; see Claim 6.5.2!

Lastly, and crucially, the following expression can be computed by a not-too-complicated circuit structure:

$$\partial_z(T_2/T_1) \;\bmod\; z^d \;=\; (T_2/T_1) \cdot \mathsf{dlog}(T_2/T_1) \;\bmod\; z^d\,.$$

Interestingly, the circuit form is *closed* under this operation of dividing, taking derivatives and taking limits! Note that, the dlog operator distributes the product gate into summation giving $\mathsf{dlog}(T_2/T_1) = \sum \mathsf{dlog}(\Sigma)$, where Σ denotes linear

polynomials. Further, we observe that

$$\mathrm{dlog}(\Sigma) \;=\; \Sigma/\Sigma \in \Sigma\wedge\Sigma\,,$$

the depth-3 powering circuits, over some 'nice' ring. The idea is to expand $1/\ell$, where ℓ is a linear polynomial, as the sum of powers of linear terms using the inverse identity:

$$1/(1-a\cdot z) \;\equiv\; 1 + a\cdot z + \cdots + a^{d-1}\cdot z^{d-1} \mod z^d\,.$$

When there is a single remaining summand, the border of the more general structure is easy-to-compute, and can be shown to have an algebraic branching program (ABP) of not too large size. For details, we refer to Claim 6.5.4. For a constant k (& even general bounded depth-4 circuits), the above idea can be extended with some additional clever division and computation.

6.4.3 Limitation of standard techniques

In this section, we briefly discuss the standard techniques for the upper bounds, in the border sense, and point out why they fail to yield our results.

Why known upper bound techniques fail? One of the most obvious way to de-border restricted classes is to essentially show a polynomial ϵ-degree bound and interpolate. In general, the bound is known to be exponential [Bür20, Theorem 5.7] which crucially uses [LL89, Proposition 1]. This proposition essentially shows the existence of an irreducible curve C, whose degree is bounded in terms of the degree of the affine variety, that we are interested in. The degree is in general exponentially upper bounded by the size [BCS13, Theorem 8.48]. Unless and until, one improves these bounds for varieties induced by specific models (which seems hard), one should not expect to improve the ϵ-degree bound, and thus the interpolation trick seems useless.

As mentioned before, $\overline{\Sigma\wedge\Sigma}$-circuits could be de-bordered using the duality trick [Sax08] (see Lemma 2.6.7) to make it an $\overline{\mathrm{ARO}}$ and finally using Nisan's characterization giving $\overline{\mathrm{ARO}} = \mathrm{ARO}$ [Nis91; For16; GKS17] (Lemma 6.2.3). But this trick is directly inapplicable to our models with the Π-gate,

due to large Waring rank & ROABP-width, as one could expect 2^d-blowup in the top fanin while converting Π-gate to $\wedge$. We also remark that the duality trick was made *field independent* in [For14, Lemma 8.6.4]. In fact, very recently, [BDI21, Theorem 4.3] gave an *improved* duality trick with no size blowup, independent of degree and number of variables.

Moreover, all the non-trivial current upper bound methods, for $\overline{\text{limit}}$, seem to need an auxiliary linear space, which even for $\overline{\Sigma^{[2]}\Pi\Sigma}$ is not clear, due to the possibility of heavy cancellation of ϵ-powers. To elaborate, one of the major bottleneck is that individually $\lim_{\epsilon\to 0} T_i$, for $i \in [2]$ *may not* exist, however, $\lim_{\epsilon\to 0}(T_1 + T_2)$ does exist, where $T_i \in \Pi\Sigma$ (over $\mathbb{F}(\epsilon)$). For e.g.,

$$T_1 \;:=\; \epsilon^{-1}(x+\epsilon^2 y)y, \text{ and } T_2 \;:=\; -\epsilon^{-1}(y+\epsilon x)x\,.$$

No generic tool is available to 'capture' such cancellations, and may even suggest a non-linear algebraic approach to tackle the problem.

Furthermore, [SSS13] explicitly classified certain factor polynomials to solve non-border $\Sigma^{[2]}\Pi\Sigma\wedge$ PIT. This factoring-based idea seems to fail miserably when we study factoring mod $\langle\epsilon^M\rangle$. In fact, in that case, we get non-unique, usually exponentially-many, factorization. For e.g.,

$$x^2 \;\equiv\; (x - a\cdot\epsilon^{M/2})\cdot(x + a\cdot\epsilon^{M/2}) \mod \langle\epsilon^M\rangle\,,$$

for all $a \in \mathbb{F}$. In this case, there are, in fact, infinitely many factorizations. Moreover,

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon^M}\cdot\Big(x^2 - (x - a\cdot\epsilon^{M/2})\cdot(x + a\cdot\epsilon^{M/2})\Big) \;=\; a^2\,.$$

Therefore, infinitely many factorizations may give infinitely many limits. To top it all, Kumar's result [Kum20] hinted a possible hardness of border-depth-3 (top-fanin-2). In that sense, ours is a very non-linear algebraic proof for restricted models which successfully opens up a possibility of finding non-representation-theoretic, and elementary, upper bounds.

6.5 Proof of Theorem 6.4.2

Before getting into the proof, we discuss the bloated model on which we will induct.

Definition 6.5.1 (Bloated model) *We call a circuit $\mathscr{C} \in \mathsf{Gen}(k,s)$, in the fractional ring $\mathsf{R}(x)$, with parameter k and size s, if it computes $f \in \mathsf{R}(x)$, where*

$$f = \sum_{i\in[k]} T_i, \text{ such that } T_i = (U_i/V_i)\cdot(P_i/Q_i),$$

with $U_i, V_i, P_i, Q_i \in \mathsf{R}[x]$ such that $U_i, V_i \in \Pi\Sigma$ and $P_i, Q_i \in \Sigma\wedge\Sigma$.

Further, $\mathsf{size}(\mathscr{C}) = \sum_{i\in[k]} \mathsf{size}(T_i)$, and, finally,

$$\mathsf{size}(T_i) = \mathsf{size}(U_i) + \mathsf{size}(V_i) + \mathsf{size}(P_i) + \mathsf{size}(Q_i)\,.$$

It is easy to see that size-s $\Sigma^{[k]}\Pi\Sigma$ lies in $\mathsf{Gen}(k,s)$, which will be our general model of induction. Here is the main de-bordering theorem for depth-3 circuits.

Theorem 6.5.1 (De-bordering $\overline{\Sigma^{[k]}\Pi\Sigma}$) *Let $f(x) \in \mathbb{F}[x_1,\ldots,x_n]$, such that f can be computed by a $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuit of size s. Then f is also computable by an ABP (over $\mathbb{F}$), of size $s^{O(k\cdot 7^k)}$.*

Proof. We will use DiDIL technique as discussed in paragraph 6.4.2. The $k = 1$ case is obvious, as $\overline{\Pi\Sigma} = \Pi\Sigma$ and trivially it has a small ABP. Further, as discussed before, $k = 2$ is already non-trivial. Eventually it involves de-bordering $\overline{\mathsf{Gen}(1,s)}$; as DiDIL technique reduces the $k = 2$ problem to $\overline{\mathsf{Gen}(1,s)}$ and then we interpolate.

❑ **Base step: De-bordering $\overline{\mathsf{Gen}(1,s)}$**

Let $g(x,\epsilon) \in R(x,\epsilon)$ be approximating $f \in R(x)$; here R is a commutative ring (the ring will be clear later in the next few paragraphs). We also assume the *syntactic degree* bound, of the denominator and numerator computing g to be d. Here is the de-bordering result.

Claim 6.5.1 $\overline{\mathsf{Gen}(1,s)} \in \mathsf{ABP}/\mathsf{ABP}$, of size $O(sd^2n^2)$, while the syntactic degree blows up to $O(nd)$.

Proof. Using Definition 6.5.1,

$$g(x,\epsilon) \;=:\; (U(x,\epsilon)/V(x,\epsilon))\cdot P(x,\epsilon)/Q(x,\epsilon) \;=\; f(x)+\epsilon\cdot S(x,\epsilon)\;,$$

where $U,V,P,Q \in \mathbb{R}(\epsilon)[x]$ such that $U,V \in \Pi\Sigma, P,Q \in \Sigma\wedge\Sigma$. Let $a_1 := \mathrm{val}_\epsilon(U)$, $a_2 := \mathrm{val}_\epsilon(V)$, $b_1 := \mathrm{val}_\epsilon(P)$ and $b_2 := \mathrm{val}_\epsilon(Q)$. Extracting the maximum ϵ-power, we get

$$f + \epsilon\cdot S \;=\; \epsilon^{(a_1-a_2)+(b_1-b_2)}\cdot\left(\tilde{U}/\tilde{V}\right)\cdot\left(\tilde{P}/\tilde{Q}\right)\;,$$

where $\tilde{U},\tilde{V},\tilde{P},\widetilde{Q} \in R(\epsilon)[x]$, and their valuations wrt. ϵ are zero i.e. $\lim_{\epsilon\to 0}\tilde{U}$ exists (similarly for $\tilde{V},\tilde{P},\tilde{Q}$). Since, LHS is well-defined at $\epsilon=0$, it must happen that $(a_1-a_2)+(b_1-b_2)\geq 0$. If $(a_1-a_2)+(b_1-b_2)\geq 1$, then $f=0$, and we have trivially de-bordered. Therefore, we can assume $(a_1-a_2)+(b_1-b_2)=0$ which implies that

$$\begin{aligned} f \;&=\; (\lim_{\epsilon\to 0}\tilde{U}/\lim_{\epsilon\to 0}\tilde{V})\cdot(\lim_{\epsilon\to 0}\tilde{P}/\lim_{\epsilon\to 0}\tilde{Q}) \\ &\in\; (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO}) \\ &\subseteq\; \mathrm{ABP}/\mathrm{ABP}\;. \end{aligned}$$

We have used the fact that $\tilde{U},\tilde{V}\in\Pi\Sigma$ and $\tilde{P},\widetilde{Q}\in\Sigma\wedge\Sigma$ of size at most s, over $R(\epsilon)$. Further, by Lemma 6.2.2 and Lemma 6.2.4, we know that $\overline{\Pi\Sigma}=\Pi\Sigma$ and $\overline{\Sigma\wedge\Sigma}\subseteq\mathrm{ARO}$; therefore f is computable by a ratio of two ABPs of size at most $O(s\cdot d^2n^2)$, and the degree gets blown up to atmost $O(nd)$. The last simply follows from Lemma 2.6.7. □

❑ Bloat out: Reducing $\overline{\Sigma^{[k]}\Pi\Sigma}$ to de-bordering $\overline{\mathrm{Gen}(k-1,\cdot)}$

Let $f_0 := f$ be an arbitrary polynomial in $\overline{\Sigma^{[k]}\Pi\Sigma}$, approximated by $g_0\in\mathbb{F}(\epsilon)[x]$, computed by a depth-3 circuit $\overline{C}$ of size s over $\mathbb{F}(\epsilon)$, i.e.,

$$g_0 \;:=\; f_0+\epsilon\cdot S_0\;.$$

Further, assume that $\deg(f_0) < d_0 := d \leq s$; we keep the parameter d separately, to optimize the complexity later. Here, we also stress that one could think of the degree to be the syntactic degree as well. Then, $g_0 =: \sum_{i\in[k]} T_{i,0}$, such that $T_{i,0}$ is computable by a $\Pi\Sigma$-circuit of size at most s over $\mathbb{F}(\epsilon)$. Moreover, define

$$U_{i,0} := T_{i,0}, V_{i,0} := P_{i,0} := Q_{i,0} = 1\;,$$

as the base input case of $\mathrm{Gen}(1,\cdot)$. As explained in the preliminaries, we do a safe division and derivation for reduction.

Φ *homomorphism.* To ensure invertibility and facilitate derivation, we define a homomorphism

$$\Phi : \mathbb{F}(\epsilon)[x] \to \mathbb{F}(\epsilon)[x,z], \quad \text{such that } x_i \mapsto z \cdot x_i + \alpha_i ,$$

where α_i are *random* elements in $\mathbb{F}$. Essentially, it suffices to ensure that $\Phi(T_{i,0})|_{x=a} = T_{i,0}(a) \neq 0$ for all $i \in [k]$. We will be working with different ring $\mathscr{R}_i(x)$, at i-th step of induction, with $\mathscr{R}_0 := \mathbb{F}[z]/\langle z^d \rangle$; here think of the z-variable as 'cost-free'. The map Φ can be thought of as a 'shift & scale' map. In a way, choosing random z and then shifting and scaling it back gives the original f. So, our target is to prove the size upper bound for $\Phi(f_0)$ in the ring $\mathscr{R}(x)$, and thereby prove the upperbound for f_0.

Di*vide and* ***de****rive.* Let $\nu_{i,0} := \mathrm{val}_z(\Phi(T_{i,0}))$. By Φ-map, $\nu_{i,0} \geq 0$, for each $i \in [k]$. Further, wrt ϵ-valuation, assume that

$$\Phi(T_{i,0}) =: \epsilon^{a_{i,0}} \cdot \tilde{T}_{i,0} ,$$

where $\tilde{T}_{i,0} =: t_{i,0} + \epsilon \cdot \tilde{t}_{i,0}(x,z,\epsilon)$, i.e., $t_{i,0} = \tilde{T}_{i,0}|_{\epsilon=0}$. Note that, $\nu_{i,0} = \mathrm{val}_z(\tilde{T}_{i,0})$. Without loss of generality, assume that

$$\min_{i\in[k]} \mathrm{val}_z(\tilde{T}_{i,0}) = \nu_{k,0} ,$$

i.e. the minimum is wrt k, otherwise we can rearrange. Then, we divide $\Phi(g_0)$ by $\tilde{T}_{k,0}$, and derive wrt z:

$$\begin{aligned}
\Phi(f_0)/\tilde{T}_{k,0} + \epsilon \cdot \Phi(S_0)/\tilde{T}_{k,0} &= \epsilon^{a_{k,0}} + \sum_{i=1}^{k-1} \Phi(T_{i,0})/\tilde{T}_{k,0} \quad [\textbf{Di}\text{vide}] \\
\implies \partial_z\left(\Phi(f_0)/\tilde{T}_{k,0}\right) + \epsilon\partial_z\left(\Phi(S_0)/\tilde{T}_{k,0}\right) &= \sum_{i=1}^{k-1} \partial_z\left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \quad [\textbf{D}\text{erive}] \\
&= \sum_{i=1}^{k-1} \left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \cdot \mathrm{dlog}\left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \\
&\qquad\qquad (6.1) \\
&=: g_1 .
\end{aligned}$$

Definability. Let $\mathscr{R}_1 := \mathbb{F}[z]/\langle z^{d_1}\rangle$, and $d_1 := d_0 - \nu_{k,0} - 1$. For $i \in [k-1]$, define

$$T_{i,1} := (\Phi(T_{i,0})/\tilde{T}_{k,0}) \cdot \mathrm{dlog}(\Phi(T_{i,0})/\tilde{T}_{k,0}), \text{ and } f_1 := \partial_z\left(\Phi(f_0)/t_{k,0}\right) .$$

Here is an important claim.

Claim 6.5.2 g_1 approximates f_1 correctly, i.e. $\lim_{\epsilon\to 0} g_1 = f_1$, where g_1 (respectively f_1) are well-defined in $\mathscr{R}_1(\epsilon, x)$ (respectively $\mathscr{R}_1(x)$).

Proof. As we divide by the minimum valuation, by Lemma 2.2.2, we have

$$\begin{aligned} \mathrm{val}_z(\Phi(T_{i,0})/\tilde{T}_{k,0}) \geq 0 &\implies \Phi(T_{i,0})/\tilde{T}_{k,0} \in \mathbb{F}(x,\epsilon)[[z]] \\ &\implies T_{i,1} \in \mathbb{F}(x,\epsilon)[[z]] \,. \end{aligned}$$

Note that,

$$\mathrm{val}_z(\Phi(f_0) + \epsilon \cdot S_0) = \mathrm{val}_z(\sum_{i\in[k]} \Phi(T_{i,0})) \geq \nu_{k,0} \,.$$

Setting $\epsilon = 0$, implies that $\mathrm{val}_z(\Phi(f_0)) \geq \nu_{k,0}$, and hence, $\Phi(f_0)/\tilde{T}_{k,0} \in \mathbb{F}(x,\epsilon)[[z]]$ (by Lemma 2.2.2). Moreover,

$$(\Phi(f_0)/\tilde{T}_{k,0})|_{\epsilon=0} = \Phi(f_0)/t_{k,0} \in \mathbb{F}(x,z) \,.$$

Combining these it follows that

$$\Phi(f_0)/t_{k,0} \in \mathbb{F}(x)[[z]] \implies f_1 \in \mathbb{F}(x)[[z]] \,.$$

Once we know that each $T_{i,1}$ and f_1 are well-defined power-series, we claim that (Equation 6.1) holds mod $z^{d_0-\nu_{k,0}-1}$. Note that, $\Phi(f_0) + \epsilon \cdot \Phi(S_0) = \sum_{i\in[k]} T_i$, holds mod z^d. Thus after dividing by the minimum valuation element (with z-valuation $\nu_{k,0}$), it holds mod $z^{d_0-\nu_{k,0}}$; finally after differentiation it must hold mod $z^{d_0-\nu_{k,0}-1}$.

Further, as $\lim_{\epsilon\to 0} \tilde{T}_{k,0}$ exists, we must have $\partial_z(\Phi(f_0)/t_{k,0}) = \lim_{\epsilon\to 0} g_1$; i.e. g_1 approximates f_1 correctly, in $\mathscr{R}_1(x)$. □

However, we stress that we also think of these as elements in $\mathbb{F}(x,z,\epsilon)$, with z-degree being 'kept track of' (which could be $> d$). All these different 'lenses' of looking and computing will be important later.

Now what with the lower fanin? The main claim now is to show that– 1) $f_1 \in \overline{\mathrm{Gen}(k-1,\cdot)}$, and 2) assuming we know $\overline{\mathrm{Gen}(k-1,\cdot)}$ has small ABP/ABP, how to lift it for f_0 (we will show how to generally reduce fanin in the next few paragraphs).

To show that, we will show that each $T_{i,1}$ has small $(\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$-circuit in $\mathscr{R}_1(x,\epsilon)$ and then we will interpolate. Once the degree of z is maintained to be *small*, this interpolation would not be costly, which will finally achieve our goal; as polynomially many sum of ratios of ABPs is still a ratio of small ABPs. We remark that these two steps are needed in the general reduction as well, and thus once we show the general inductive reduction, we will illustrate these steps.

❑ Inductive step (j-th step): Reducing $\overline{\mathrm{Gen}(k-j,\cdot)}$ to $\overline{\mathrm{Gen}(k-j-1,\cdot)}$

Suppose, we are at the j-th ($j \geq 1$) step. Our induction hypothesis assumes–

1. $\sum_{i\in[k-j]} T_{i,j} =: g_j$, in $\mathscr{R}_j(x,\epsilon)$, such that it approximates f_j correctly, where $f_j \in \mathscr{R}_j(x)$, where $\mathscr{R}_j := \mathbb{F}[z]/\langle z^{d_j}\rangle$.
2. Here, $T_{i,j} =: (U_{i,j}/V_{i,j})\cdot(P_{i,j}/Q_{i,j})$, where

   $$U_{i,j}, V_{i,j} \in \Pi\Sigma \text{ and } P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma, \text{ each in } \mathscr{R}_j(\epsilon)[x]\ .$$

   Each can be thought as an element in $\mathbb{F}(x,z,\epsilon)\bigcap\mathbb{F}(x,\epsilon)[[z]]$ as well. Assume that the syntactic degree of each denominator and numerator of $T_{i,j}$ is bounded by D_j.
3. $\nu_{i,j} := \mathrm{val}_z(T_{i,j}) \geq 0$, for $i \in [k-j]$. Wlog, assume that $\min_i \nu_{i,j} = \nu_{k-j,j}$. Moreover, $U_{i,j}|_{z=0} \in \mathbb{F}(\epsilon)\backslash\{0\}$ (similarly for $V_{i,j}$).

We proceed, like the $j=0$-th step done above, without applying any new homomorphism. Similar to that reduction, we divide and derive to reduce the fanin further by 1.

***Di**vide and **D**erive.* Let $T_{k-j,j} =: \epsilon^{a_{k-j,j}}\cdot\tilde{T}_{k-j,j}$, where $\tilde{T}_{k-j,j} =: (t_{k-j,j}+\epsilon\cdot\tilde{t}_{k-j,j})$ is not divisible by ϵ. Divide $g_j =: f_j+\epsilon\cdot S_j$, by $\tilde{T}_{k-j,j}$, to get:

$$
\begin{aligned}
f_j/\tilde{T}_{k-j,j} + \epsilon \cdot S_j/\tilde{T}_{k-j,j} &= \epsilon^{a_{k-j,j}} + \sum_{i=1}^{k-j-1} T_{i,j}/\tilde{T}_{k-j,j} \\
\implies \partial_z\left(f_j/\tilde{T}_{k-j,j}\right) + \epsilon \cdot \partial_z\left(S_j/\tilde{T}_{k-j,j}\right) &= \sum_{i=1}^{k-j-1} \partial_z\left(T_{i,j}/\tilde{T}_{k-j,j}\right) \\
&= \sum_{i=1}^{k-j-1} \left(T_{i,j}/\tilde{T}_{k-j,j}\right) \cdot \mathrm{dlog}\left(T_{i,j}/\tilde{T}_{k-j,j}\right) \qquad (6.2) \\
&=: g_{j+1}\,.
\end{aligned}
$$

Definability. Let $\mathscr{R}_{j+1} := \mathbb{F}[z]/\langle z^{d_{j+1}}\rangle$, where $d_{j+1} := d_j - v_{k-j,j} - 1$. For $i \in [k-j-1]$, define

$$T_{i,j+1} := \left(T_{i,j}/\tilde{T}_{k-j,j}\right)\cdot \mathrm{dlog}\left(T_{i,j}/\tilde{T}_{k-j,j}\right)\,, \text{ and } f_{j+1} := \partial_z(f_j/t_{k-j,j})\,.$$

Claim 6.5.3 (**I**nduction hypotheses) (i) g_{j+1} (respectively f_{j+1}) are well-defined in $\mathscr{R}_{j+1}(x,\epsilon)$ (respectively , $\mathscr{R}_{j+1}(x)$).

(ii) g_{j+1} approximates f_{j+1} correctly, i.e., $\lim_{\epsilon\to 0} g_{j+1} = f_{j+1}$.

Proof. Remember, f_j and $T_{i,j}$'s are elements in $\mathbb{F}(x,z,\epsilon)$ which also belong to $\mathbb{F}(x,\epsilon)[[z]]$. After dividing by the minimum valuation, by similar argument as in Claim 6.5.2, it follows that $T_{i,j+1}$ and f_{j+1} are elements in $\mathbb{F}(x,z,\epsilon)\bigcap\mathbb{F}(x,\epsilon)[[z]]$, proving the second part of induction-hypothesis-(2). In fact, trivially $v_{i,j+1} \geq 0$, for $i \in [k-j-1]$ proving induction-hypothesis-(3).

Similarly, Equation 6.2 holds in $\mathscr{R}_{j+1}(\epsilon,x)$, or equivalently mod $z^{d_{j+1}}$; this is because of the division by z-valuation of $v_{k-j,j}$ and then differentiation, showing induction-hypothesis-(1). So, Equation 6.2 being computed mod $z^{d_{j+1}}$ is indeed valid. We also mention that using similar argument as in Claim 6.5.2, $f_{j+1} \in \mathbb{F}(x)[[z]]$.

Finally, as f_{j+1} exists, it is obvious to see that $\lim_{\epsilon\to 0} g_{j+1} = f_{j+1}$. □

Invertibility of ΠΣ*-circuits.* Before going into the size analysis, we want to remark that the dlog computation plays a crucial role here and the invertibility of the ΠΣ-circuits are crucial

for our arguments to go through. The action $\mathsf{dlog}(\Sigma\wedge\Sigma) \in (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, is of polynomial size (Lemma 2.6.9).

What is the action on $\Pi\Sigma$*?* As dlog distributes the product *additively*, so it suffices to work with $\mathsf{dlog}(\Pi\Sigma)$; and we show that $\mathsf{dlog}(\Pi\Sigma) \in \Sigma\wedge\Sigma$, is of polynomial size. For the time being, assume these hold. Then, we simplify

$$\frac{T_{i,j}}{\tilde{T}_{k-j,j}} \;=\; \epsilon^{-a_{k-j,j}} \cdot \frac{(U_{i,j}\cdot V_{k-j,j})}{(V_{i,j}\cdot U_{k-j,j})} \cdot \frac{(P_{i,j}\cdot Q_{k-j,j})}{(Q_{i,j}\cdot P_{k-j,j})}\,,$$

and its dlog. Therefore, one can define $U_{i,j+1} := \epsilon^{-a_{k-j,j}} \cdot U_{i,j} \cdot V_{k-j,j}$; similarly $V_{i,j+1} := V_{i,j}\cdot U_{k-j,j}$. We stress that dlog computation will produce $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, which will further multiply with $P's$ and Q's; it will be clear after the lemma. This directly means:

$$U_{i,j+1}|_{z=0}, V_{i,j+1}|_{z=0} \;\in\; \mathbb{F}(\epsilon)\setminus\{0\}\,.$$

This proves the second part of induction-hypothesis-(3).

The overall size blowup. Finally, we show the main step: how to use dlog which is the crux of our reduction. We assume that at the j-th step, $\mathsf{size}(T_{i,j}) \le s_j$ and by assumption $s_0 \le s$.

Claim 6.5.4 (Size blowup from DiDIL)

$$T_{1,k-1} \in (\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \;\in\; \mathscr{R}_{k-1}(x,\epsilon)\,,$$

of size $s^{O(k7^k)}$. It is computed as an element in $\mathbb{F}(\epsilon,x,z)$, with syntactic degree (in x,z) $d^{O(k)}$.

Proof. Steps $j=0$ vs $j>0$ are slightly different because of the homomorphism Φ. However, the main idea of using dlog and expand it as a power-series is the same, which eventually shows that $\mathsf{dlog}(\Pi\Sigma) \in \Sigma\wedge\Sigma$ with a controlled blowup.

For $j=0$, we want to study dlog's effect on $\Phi(T_{i,0})/\tilde{T}_{k,0}$. As dlog distributes over product, and thus it suffices to study $\mathsf{dlog}(\ell)$, where $\ell \in \mathscr{R}(\epsilon)[x]$. However, by the property of Φ, each ℓ must be of the form $\ell = A - zB$, where $A \in \mathbb{F}(\epsilon)\backslash\{0\}$ and $B \in \mathbb{F}(\epsilon)[x]$. Using the power series expansion, we have the

following, in $\mathscr{R}_1(x, \epsilon)$:

$$\mathrm{dlog}(\ell) = -\frac{\partial_z (A - z \cdot B)}{A\,(1 - z \cdot B/A)} = -\frac{B}{A} \cdot \sum_{j=0}^{d_1 - 1} \left(\frac{z \cdot B}{A}\right)^j . \tag{6.3}$$

Note,(B/A) and $(-z \cdot B/A)^j$ have a trivial $\wedge\Sigma$ circuits, each of size $O(s)$. For all j use Lemma 2.6.10 on $(B/A) \cdot (-z \cdot B/A)^j$ to obtain an equivalent $\Sigma\wedge\Sigma$ of size $O(j \cdot d \cdot s)$. Re-indexing gives us the final $\Sigma\wedge\Sigma$ circuit for $\mathrm{dlog}(\ell)$ of size $O(d^3 \cdot s)$. We use the fact that $d_1 \leq d_0 = d$. Here the syntactic degree blowsup to $O(d^2)$.

For $j > 0$, the above equation holds in $\mathscr{R}_j(x)$. However, as mentioned before, the degree could be D_j (possibly $> d_j$) of the corresponding A and B. Thus, the overall size after the power-series expansion would be $O(D_j^2 d\mathrm{size}(\ell))$ [here again we use that $d_j \leq d$].

The effect of dlog on $\Sigma\wedge\Sigma$ is, naturally, more straightforward. This is because Lemma 2.6.9 shows that dlog is closed under differentiation. Using Lemma 2.6.9, we obtain $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ circuit for $\mathrm{dlog}(P_{i,j})$ of size $O\left(D_j^2 \cdot s_j\right)$. Similar claim can be made for $\mathrm{dlog}(Q_{i,j})$. Also, $\mathrm{dlog}(U_{i,j} \cdot V_{k-j,j}) \in \sum \mathrm{dlog}(\Sigma)$, which could be computed using the above Equation. Thus,

$$\begin{aligned} \mathrm{dlog}(T_{i,j}/\tilde{T}_{k-j,j}) \;&\in\; \mathrm{dlog}(\Pi\Sigma/\Pi\Sigma) \pm \Sigma^{[4]}\mathrm{dlog}(\Sigma\wedge\Sigma) \\ &\subseteq\; \Sigma\wedge\Sigma + \Sigma^{[4]}\,(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \;=\; (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)\ . \end{aligned}$$

Here, $\Sigma^{[4]}$ means sum of 4-many expressions. The first containment is by linearization. The steps and the size blowups are as follows:

1. Express $\mathrm{dlog}(\Pi\Sigma/\Pi\Sigma)$ as a single $\Sigma\wedge\Sigma$-expression of size $O(D_j^2 d_j s_j)$, by summing up the $\Sigma\wedge\Sigma$-expressions obtained from $\mathrm{dlog}(\Sigma)$.
2. Next, there are 4-many $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expressions of size $O(D_j^2 s_j)$ as there are 4-many P's and Q's.
3. Additionally, the syntactic degree of each denominator and numerator of $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ grows up to $O(D_j)$.
4. Next, we club $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ expressions (4 of them) to express it as a single $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ expression using Lemma 2.6.9, with size blowup of $O(D_j^{12} s_j^4)$.

5. Finally, add the single $\Sigma\wedge\Sigma$ expression of size $O(D_j^3 s_j)$, and degree $O(dD_j)$, to get $O(s_j^5 D_j^{16} d)$ size representation.

Also, we need to multiply with $T_{i,j}/\tilde{T}_{k-j,j}$ which is of the form $(\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, where each $\Sigma\wedge\Sigma$ is basically the product of two $\Sigma\wedge\Sigma$ expressions of size s_j, and syntactic degree D_j and clubbed together, owing a blowup of $O(D_j s_j^2)$. Hence, multiplying this $(\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$-expression with the $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expression obtained from dlog-compuation, gives a size blowup of

$$s_{j+1} \;:=\; s_j^7 D_j^{O(1)} d\,.$$

As mentioned before, the main blowup of syntactic degree in the dlog computation could be $O(dD_j)$ and clearing expressions and multiplying the without dlog expression increases the syntactic degree only by a constant multiple. Therefore, $D_{j+1} := O(dD_j) \implies D_j = d^{O(j)}$. Hence,

$$s_{j+1} = s_j^7 \cdot d^{O(j)} \implies s_j \le (sd)^{O(j\cdot 7^j)}\,.$$

In particular, $s_{k-1} \le s^{O(k\cdot 7^k)}$; here we used that $d \le s$. This calculation quantitatively establishes induction-hypothesis-(2). □

Roadmap to trace back f_0. The above claim established that $g_{k-1} \in \mathrm{Gen}(1,\cdot)$ and approximates f_{k-1} correctly. We also know that $\overline{\mathrm{Gen}(1,\cdot)} \in \mathrm{ABP}/\mathrm{ABP}$, from Claim 6.5.1. Whence, g_{k-1} having $s^{O(k7^k)}$ size bloated-circuit implies: it can be computed as a ratio of ABPs with size $s^{O(k7^k)} \cdot D_{k-1}^2 \cdot n^2 = s^{O(k7^k)}$, and syntactic degree $n \cdot D_{k-1} = d^{O(k)}$. Now, we recursively 'lift' this quantity, via interpolation, to recover in order, $f_{k-2}, f_{k-3}, \ldots, f_0$; which we originally wanted.

Interpolation: To integrate and limit. As mentioned above, we will interpolate recursively. We know $f_{k-1} = \partial_z(f_{k-2}/t_{2,k-2})$ has a ABP/ABP circuit in $\mathbb{F}(x,z)$, i.e. each denominator and numerator is being computed in $\mathbb{F}[x,z]$, and size bounded by $\mathcal{S}_{k-1} := s^{O(k7^k)}$. Here is an important claim about the size of f_{k-2} (we denote it by $\mathcal{S}_{k-2}$).

Claim 6.5.5 (Tracing back one step) f_{k-2} can be expressed

as

$$f_{k-2} \;=\; \sum_{i=0}^{d_{k-2}-1} (\text{ABP}/\text{ABP})\; z^i \;,$$

of size $s^{O(k7^k)}$ and syntactic degree $d^{O(k)}$.

Proof. Let the degree of f_{k-1} (both denominator and numerator) be bounded by $D'_{k-1} := d^{O(k)}$, and further, we know that keeping information (of the power series) till mod $z^{d_{k-1}}$ suffices. While computing it, it may happen that the valuation of each denominator and numerator is > 0, i.e. it is of the form $z^{e_1} \cdot (\text{ABP})/z^{e_2} \cdot (\text{ABP})$ (e_1, e_2 being valuations wrt z). It must happen that $e_1 \geq e_2$, if it is indeed a power series in z; the e_i's are bounded by D'_{k-1}.

Furthermore, these ABPs (after dividing by z-power) have similar size as z is considered free [think of them being computed in $\mathbb{F}(z)[x]$]. Therefore, (ABP/ABP) can be expressed as $\sum_{i=0}^{d_{k-1}-1} C_{i,k-1} \cdot z^i$, by using the inverse identity:

$$1/(1-z) \;\equiv\; 1 + \ldots + z^{d_{k-1}-1} \bmod\; z^{d_{k-1}} \;.$$

Here, each $C_{i,k-1}$ has an (ABP/ABP) of size at most $O(\mathcal{S}_{k-1} \cdot {D'_{k-1}}^2)$; for details, see Lemma 2.6.4.

Once we get $f_{k-1} = \sum_{i=0}^{d_{k-1}-1} C_{i,k-1} z^i$, definite-integration implies:

$$f_{k-2}/t_{2,k-2} - f_{k-2}/t_{2,k-2}|_{z=0} \;\equiv\; \sum_{i=1}^{d_{k-1}} (C_{i,k-1}/i) \cdot z^i \;\bmod\; z^{d_{k-1}+1} \;.$$

The final trick is to get $f_{k-2}/t_{2,k-2}|_{z=0}$, and 'reach' f_{k-2}. As, $f_{k-2}/t_{2,k-2} \in \mathbb{F}(x)[[z]]$, substituting $z = 0$, yields an element in $\mathbb{F}(x)$. Recall the identity:

$$\begin{aligned} f_{k-2}/t_{2,k-2}|_{z=0} \;&=\; \lim_{\epsilon \to 0} (T_{1,k-2}/\tilde{T}_{2,k-2}|_{z=0} + \epsilon^{a_{2,k-2}}) \\ &\in\; \lim_{\epsilon \to 0} (\mathbb{F}(\epsilon) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) + \epsilon^{a_{2,k-2}}) \;. \end{aligned}$$

Since, $\mathbb{F}(\epsilon) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) + \epsilon^{a_{2,k-2}} \in (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, in $\mathbb{F}(\epsilon)(x)$, we know that the limit exists and is (ARO/ARO), and of course, trivially,

$$(\text{ARO}/\text{ARO}) \;\subseteq\; (\text{ABP}/\text{ABP}) \;.$$

Further, the syntactic degree is $d^{O(k)}$, and the size is $s_{k-1} \cdot d^{O(k)}$.

Thus, from the above equation, it follows:

$$f_{k-2}/t_{2,k-2} \;=\; f_{k-2}/t_{2,k-2}|_{z=0} + \sum_{i=1}^{d_{k-1}} (C_{i,k-1}/i)\cdot z^i \;\in\; \sum_{i=0}^{d_{k-1}} (\mathrm{ABP}/\mathrm{ABP})\cdot z^i\,,$$

of size $d_{k-1}\cdot \mathcal{S}_{k-1}D_{k-1}^{'2} + s_{k-1}\cdot d^{O(k)}$, and degree $D'_{k-1}+d^{O(k)}$. Lastly,

$$\begin{aligned} t_{2,k-2} &\in \lim_{\epsilon\to 0} (\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \\ &\subseteq (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO})\,. \end{aligned}$$

Thus, it has size s_{k-2}, by previous claims, and degree bound D_{k-2}. Moreover, we know that $\mathrm{val}_z(t_{2,k-2}) \geq \nu_{2,k-2} = d_{k-2} - d_{k-1} - 1$. Thus, multiply $t_{2,k-2}$ and truncate it till $d_{k-2}-1$. This gives us the blowup: size $\mathcal{S}_{k-2} = d_{k-1}\cdot \mathcal{S}_{k-1}D_{k-1}^{'2} + s_{k-1}\cdot d^{O(k)}$ and degree $D'_{k-2} = D'_{k-1} + d^{O(k)}$.

So, we get: f_{k-2} has $\sum_{i=0}^{d_{k-2}-1}(\mathrm{ABP}/\mathrm{ABP})z^i$ of size $\mathcal{S}_{k-2} = s^{O(k7^k)}$ and degree $D'_{k-2} = d^{O(k)}$. □

The $z=0$-evaluation. To trace back further, we imitate the step as above; and get f_j one by one. But we first need a claim about the $z=0$ evaluation of $f_j/t_{k-j,j}$.

Claim 6.5.6 (For definite integration)

$$f_j/t_{k-j,j}|_{z=0} \;\in\; \mathrm{ARO}/\mathrm{ARO} \;\subseteq\; \mathrm{ABP}/\mathrm{ABP}$$

of size $s^{O(k7^k)}$.

Proof. Note that, $g_j/\tilde{T}_{k-j,j} = \sum_{i\in[k-j]} T_{i,j}/\tilde{T}_{k-j,j} \in \mathbb{F}(x)[[z,\epsilon]]$, as the valuation wrt z, respectively ϵ, is non-negative. Therefore,

$$\begin{aligned} \left(\frac{f_j}{t_{k-j,j}}\right)\Big|_{z=0} &= \lim_{\epsilon\to 0} \sum_{i\in[k-j]} \left(\frac{T_{i,j}}{\tilde{T}_{k-j,j}}\right)\Big|_{z=0} \\ &= \lim_{\epsilon\to 0} \sum_{i\in[k-j]} \left(\epsilon^{-a_{k-j,j}}\cdot \frac{U_{i,j}\cdot V_{k-j,j}}{U_{k-j,j}\cdot V_{i,j}}\cdot \frac{P_{i,j}\cdot Q_{k-j,j}}{P_{k-j,j}\cdot Q_{i,j}}\right)\Big|_{z=0} \\ &\in \lim_{\epsilon\to 0} \sum_{i\in[k-j]} \left(\mathbb{F}(\epsilon)\cdot \frac{\Sigma\wedge\Sigma}{\Sigma\wedge\Sigma}\right) = \lim_{\epsilon\to 0}\left(\frac{\Sigma\wedge\Sigma}{\Sigma\wedge\Sigma}\right) \subseteq \left(\frac{\mathrm{ARO}}{\mathrm{ARO}}\right)\,. \end{aligned}$$

Here we crucially used induction-hypothesis-(3) part: each $U_{i,j}, V_{i,j}$ at $z=0$, is an element in $\mathbb{F}(\epsilon)$. Also, we used that $\Sigma\wedge\Sigma$

is *closed* under constant-fold multiplication (Lemma 2.6.10). Finally, we take the limit to conclude that

$$\overline{(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)} \subseteq (\mathrm{ARO}/\mathrm{ARO}) .$$

To show the upper bound on the ABP size, let us denote the size$(f_j/t_{k-j,j}|_{z=0}) =: S'_j$, and the syntactic degree D'_j. We claim that

$$S'_j = O(s_j^{O(k-j)} \cdot {D'_j}^2 n^2) .$$

Because, we have a sum of $k-j$ many $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ expressions each of size s_j (the syntactic degrees are also bounded by s_j); $\Sigma\wedge\Sigma$ is closed under multiplication (Lemma 2.6.10) and $\Sigma\wedge\Sigma$ to ARO conversion introduces exponent 2 in the degree (Lemma 2.6.11). Each time the syntactic degree blowup is only a constant multiple, thus $D'_j := d^{O(k)}$ (which is $\leq s^{O(k)}$). Therefore,

$$S'_j = s^{O(k-j)\cdot j7^j} = s^{O(j(k-j)7^j)} = s^{O(k7^k)} .$$

Here, we again use the fact that $\max_{j\in[k-1]} j(k-j)7^j = (k-1)7^{k-1}$ (see Lemma 2.2.1). This finishes the proof. □

Size blowup. Suppose the size of the ABP, comptuing f_j is $\mathcal{S}_j$; thus we need to estimate $\mathcal{S}_0$.

We *remark* that we do not need to eliminate division at each tracing-back-step (which we did to obtain f_{k-2}). Since, once we have $\sum_{i=0}^{d_j-1}(\mathrm{ABP}/\mathrm{ABP})\cdot z^i$, it is easy to integrate (wrt z) without any blowup as we already have all the (ABP/ABP)'s in hand (they are z-free). The main size blowup (= S'_j) happens due to $z=0$ computation which we calculated above (Claim 6.5.6). Thus, the final recurrence is

$$\mathcal{S}_j = \mathcal{S}_{j+1} + S'_j .$$

This gives $\mathcal{S}_0 = s^{O(k7^k)}$, which is the size of $\Phi(f)$, in the ring $\mathbb{F}(z,x)$, being computed as an (ABP/ABP).

Finally, plugging 'random' z, shifting-and-scaling, gives us f; represented as an (ABP/ABP) of similar size. At the final stage, we eliminate the division-gate which gives us f represented as an ABP of size $s^{O(k7^k)}$. □

Remark. Our proof de-bordered Gen(k,s), and that too for any field of characteristic $= 0$ or $\geq d$.

6.6 Discussion

Depth-3 circuits with constant top fan-in themselves seem like fairly 'simple' models of computation, but have been objects of intense study in algebraic complexity. The prior known de-bordering techniques are mostly rank-based, or, very simple; they do not appear to extend to the border of these circuits, at least in obvious ways. In this work, we introduce new ideas and techniques which help bridge this gap in understanding. The method is quite powerful, which almost converts a Π-gate to an $\wedge$-gate! It is not too-wishful to expect some more applications of DiDIL, on de-bordering a more general circuit class, and related problems. It will be particularly nice to see if DiDIL can be *expedited*.

In particular, the current method gives $\exp(k)$, in the exponent, since DiDIL works *linearly*, and at each step, there is a *multiplicative* blowup. If the same process can be done in a divide-and-conquer fashion, the same method will give $\text{poly}(k)$, in the exponent. However, we *agree* that our current analysis might not be very *optimized*, and thus, it might still be possible to get 4^k or 5^k, instead of 7^k. But it clearly does not give better than $\exp(k)$, and hence, we do not get into messy optimizations!

Border depth-3 PIT 7

"The timeless in you is aware of life's timelessness. And knows that yesterday is but today's memory and tomorrow is today's dream."

— Khalil Gibran, The Prophet.

Abstract. In this chapter, we give the *first* quasipolynomial-time blackbox identity test for constant fanin border depth-3 circuits ($\overline{\Sigma^{[k]}\Pi\Sigma}$, for constant k). Constructing a set of points $H \subset \mathbb{C}^n$, of small bit complexity, that is guaranteed to be a hitting set for polynomials that can be *infinitesimally approximated* by small algebraic circuits over the complex numbers, is an interesting complexity theoretic question. Forbes and Shpilka [FS18] introduced the concept of *robust hitting set* and gave a PSPACE algorithm for constructing a small hitting set for $\overline{\mathsf{VP}}$. Very recently, [LST21] gave a sub-exponential PIT for any constant-depth circuits, which can in fact be extended to the border paradigm [AF21], giving a similar time PIT.

Since, no sub-exponential derandomization of PIT for general ABPs is known, this is really "independent" of the debordering result, presented in Chapter 6. This result also has a similar high level strategy as Chapter 6, and uses DiDIL-technique, though thee are additional technical difficulties, which some additional care, and *multi-stage combination* of hitting set points.

This chapter is based on the very small part from the first half of the article titled *Demystifying the border of depth-3 algebraic circuits*, which is a joint work with Prateek Dwivedi and Nitin Saxena, that appeared in FOCS 2021 [DDS21b], and *invited* in the special SICOMP issue on FOCS'21.

7.1 Border PIT

Before going into the details, let us formally define what we mean by border PIT.

Definition 7.1.1 (Hitting set for border classes) *$\mathscr{H}$ is a hitting set for a class $\overline{\mathscr{C}}$, if $g(x,\epsilon) \in \mathscr{C}_{\mathbb{F}(\epsilon)}$, approximates a* nonzero *polynomial $f(x) \in \overline{\mathscr{C}}$, then $\exists a \in \mathscr{H}$ such that $g(a,\epsilon) \notin$*

$\epsilon \cdot \mathbb{F}[\epsilon]$, *i.e.* $f(a) \neq 0$.

Note that, as $\mathscr{H}$ will also 'hit' polynomials of class $\mathscr{C}$, construction of hitting set for the border classes (we call it 'border PIT') is a natural, and possibly a different avenue to derandomize PIT. Here, we emphasize that $a \in \mathbb{F}^n$ such that $g(a, \epsilon) \neq 0$, *may not* hit the limit polynomial f since $g(a, \epsilon)$ might still lie in $\epsilon \cdot \mathbb{F}[\epsilon]$; because f could have really high complexity compared to g. Intrinsically, this property makes it harder to construct an explicit hitting set for $\overline{\mathrm{VP}}$. For an explicit example, see the margin note.

We also remark that there is no 'whitebox' setting in the border, and thus we cannot really talk about 't-time algorithm'; rather we would only be using the term 't-time hitting set', since the given circuit after evaluating on $a \in \mathbb{F}^n$, may require *arbitrarily* high-precision in $\mathbb{F}(\epsilon)$.

Prior known border PITs. Mulmuley [Mul17] asked the question of constructing an efficient hitting set for $\overline{\mathrm{VP}}$. Forbes and Shpilka [FS18] gave a PSPACE algorithm over the field $\mathbb{C}$. In [Guo+19], the authors extended this result to *any* field. A very few better hitting set constructions are known for the restricted border classes, e.g.,

1. polynomial-time hitting set for $\overline{\Pi\Sigma} = \Pi\Sigma$ [BT88; KS01],
2. quasipolynomial-time hitting set for all the three respective classes $\overline{\Sigma\wedge\Sigma} \subseteq \overline{\mathrm{ARO}} \subseteq \overline{\mathrm{ROABP}}$ [FS13a; Agr+15; Gur+17],
3. polynomial-time hitting set for the border of a restricted sum of log-variate ROABPs [BS21].

Why care about border PIT? PIT for $\overline{\mathrm{VP}}$ has a lot of applications in the context of borderline geometry and computational complexity, as observed by Mulmuley [Mul12]. For e.g., Noether's Normalization Lemma (NNL); it is a fundamental result in algebraic geometry where the computational problem of constructing explicit *normalization map* reduces to constructing small size hitting set of $\overline{\mathrm{VP}}$ [Mul17; FS13b]. Close connection between certain formulation of derandomization of NNL, and the problem of showing explicit circuit lower bounds, is also known [Mul17; Muk16].

The second motivation comes from the hope to find an explicit 'robust' hitting set for VP [FS18]; this is a hitting set $\mathscr{H}$ such that after an adequate normalization, there will be a point in $\mathscr{H}$ on which f evaluates to (say) 1. This notion overcomes the discrepancy between a hitting set for VP, and a hitting set for $\overline{\text{VP}}$ [FS18; MS21] [1]. We know that a small robust hitting set exists [CW01], but an explicit PSPACE construction was given in [FS18]. It is not at all clear whether the efficient hitting sets known for restricted depth-3 circuits are robust or not.

1: To understand the discrepancy, consider the example in Chapter 1: $h_\epsilon(x,y) = \frac{1}{3\epsilon}((x+\epsilon y)^3 - x^3)$. As argued before, $\lim_{\epsilon\to 0} h_\epsilon = h := x^2 y$. Now consider $(x,y) = (0,1)$. Note that, $h_\epsilon((0,1)) = \epsilon^2/3 \neq 0$; but in the limit we get zero. However, the limit polynomial *is not* a zero polynomial.

7.2 Our Border PIT Results

We continue our study on $\overline{\Sigma^{[k]}\Pi^{[d]}\Sigma}$ circuits, and ask for an efficient hitting set. Already, a polynomial-time hitting set is known for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ [SS11; SS12; Agr+16]. But, the border class seems to be more powerful, and the known hitting sets seem to fail. However, using our structural understanding and the analytic DiDIL technique, we are able to completely quasi-derandomize the class. For the detailed statement, see Theorem 7.3.1.

Theorem 7.2.1 (Quasi-derandomizing depth-3 [DDS21b]) *There exists an explicit quasipolynomial-time ($s^{O(\log\log s)}$) hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits of size s and constant k.*

Remarks. 1. For $k = 1$, as $f \in \overline{\Pi\Sigma} = \Pi\Sigma$, there is an explicit polynomial-time hitting set. In particular, $f(z, z^2, \ldots, z^n) \neq 0$, where z is a new variable!

2. Our technique *necessarily* blows up the size to $s^{\exp(k)\cdot\log\log s}$. Therefore, it would be interesting to design a *subexponential*-time algorithm when $k = \Theta(\log s)$; or polynomial-time for $k = O(1)$.

3. We can not directly use the de-bordering result of Theorem 6.4.2 and try to find efficient hitting set, as we do not know explicit good hitting set for general ABPs.

4. One can extend this technique to construct quasipolynomial-time hitting set for depth-4 classes: $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ and $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$, when k and δ are constants, [DDS21b]. However, we have not include these results in this thesis.

The log-variate regime. In recent developments [AGS19; KST19; Guo+19; DST21] low-variate polynomials, even in highly restricted models, have gained a lot of clout for their general implications in the context of derandomization and hardness results. A slightly *non-trivial* hitting set for trivariate $\Sigma\Pi\Sigma\wedge$-circuits [AGS19] would in fact imply quasipolynomial-time PIT for general circuits (optimized to polynomial-time in [Guo+19] with a hardness hypothesis). This motivation has pushed researchers to work on log-variate regime and design efficient PITs. Some of the most important PITs in the log-variate regime are:

1. In [FGS18], the authors showed a poly(s)-time black-box identity test for $n = O(\log s)$ variate size-s circuits that have poly(s)-dimensional partial derivative space; e.g., log-variate depth-3 diagonal circuits.
2. Very recently, Bisht and Saxena [BS21] gave the first poly(s)-time blackbox PIT for sum of constant-many, size-s, $O(\log s)$-variate constant-width ROABPs (and its border).

We remark that non-trivial border-PIT in the low-variate bootstraps to non-trivial PIT for $\overline{\mathrm{VP}}$ as well [AGS19; Guo+19]. Motivated thus, we try to derandomize log-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits. Unfortunately, direct application of Theorem 7.2.1 fails to give a polynomial-time PIT. Surprisingly, adapting techniques from [FGS18] to extend the existing result (Theorem 7.4.1), combined with our DiDIL-technique, we prove the following. For details, see Theorem 7.4.2.

Theorem 7.2.2 (Derandomizing log-variate depth-3 circuits [DDS21b]) *There exists an explicit* poly(s)*-time hitting set for $n = O(\log s)$ variate, size-s, $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits, for constant k.*

7.2.1 The 100-foot view of the proofs

The PIT results also have a similar high level strategy as of Theorem 6.4.2, although there are additional technical difficulties which need some care at every stage. At the core, the idea is really "primal", and depends on the following.

Observation

If a bivariate polynomial $G(X,Z) \neq 0$, then either its derivative $\partial_Z G(X,Z) \neq 0$, or its constant-term $G(X,0) \neq 0$ (note: $G(X,0) = G \mod Z$).

So, if $G(a,0) \neq 0$ or $\partial_Z G(b,Z) \neq 0$, then the union-set $\{a,b\}$ hits $G(X,Z)$, i.e., either $G(a,Z) \neq 0$ or $G(b,Z) \neq 0$.

Therefore, after the DiDIL-technique, we convert the Π-gate to $\wedge$-gate (with unbounded fanin), where efficient hitting sets are known. Moreover, in the log-variate regime, the classical hitting set proof is *rank*-based [FGS18], and thus can be extended to the border setting as well.

7.2.2 Why known PIT techniques fail?

Once we understand an interesting upperbound for $\overline{\Sigma^{[k]}\Pi\Sigma}$, it is natural to look for efficient derandomization. However, as we do not know any efficient PIT for ABPs, known techniques would not yield an efficient PIT for the same. Further, in a nutshell—(i) limited (almost non-existent) understanding of linear/algebraic dependence under limit, (ii) exponential upper bound on ϵ, and (iii) not-good-enough understanding of restricted border classes make it really hard to come up with an efficient hitting set. We elaborate these points below.

Dvir and Shpilka [DS07] gave a rank-based approach to design the first quasipolynomial-time algorithm for $\Sigma^{[k]}\Pi\Sigma$. A series of works [KS09; SS11; SS12; SS13] finally gave a $s^{O(k)}$-time algorithm for the same. Their techniques depend on either generalizing Chinese remaindering (CR) via ideal-matching or certifying paths, or via efficient variable-reduction, to obtain a good enough rank-bound on the multiplication ($\Pi\Sigma$) terms. Most of these approaches required a linear space, but the possibility of exponential ϵ-powers and non-trivial cancellations make these methods fail miserably in the limit. Similar obstructions also hold for [MS21; ST21; BG21] which give efficient hitting sets for the orbit of sparse polynomials (which is in fact *dense* in $\Sigma\Pi\Sigma$). In particular, Medini and Shpilka [MS21] gave PIT for the orbits of variable disjoint monomials (see [MS21, Definition 1.29]), under the affine group, but not the closure of it. Thus, they do not even give a subexponential PIT for $\overline{\Sigma^{[2]}\Pi\Sigma}$.

Recently, Guo [Guo21] gave an s^{δ^k}-time PIT, for non-SG $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits, by constructing explicit variety evasive subspace families; but to apply this idea to border PIT, one has to devise a radical-ideal based PIT idea. Currently, this does not work in the border, as $\epsilon \mod \langle \epsilon^M \rangle$ has an exponentially high nilpotency. Since radical$\langle \epsilon^M \rangle = \langle \epsilon \rangle$, it 'kills' the necessary information unless we can show a polynomial upper bound on M.

Finally, [Agr+16] came up with *faithful* map by using Jacobian + certifying path technique, which is more about algebraic rank rather than linear-rank. However, it is not at all clear how it behaves wrt $\lim_{\epsilon \to 0}$. For e.g. $f_1 = x_1 + \epsilon^M \cdot x_2$, and $f_2 = x_1$, where M is arbitrary large. Note that the underlying Jacobian $J(f_1, f_2) = \epsilon^M$ is nonzero; but it flips to zero in the limit. This makes the whole Jacobian machinery collapse in the border setting; as it cannot possibly give a variable reduction for the border model. (E.g., one needs to keep both x_1 and x_2 above.)

In Chapter 4, we gave a quasipolynomial-time hitting set for exact $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits, when k and δ are constant. This result is dependent on the Jacobian technique, which fails under taking the limit, as mentioned above. However, a polynomial-time whitebox PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits was shown using DiDI-technique (Divide, Derive and Induct). This cannot be directly used because there was no ϵ (i.e. without limit) and $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ has only blackbox access. Further, Theorem 6.4.2 gives an ABP, where DiDI-technique cannot be directly applied. Therefore, our DiDIL-technique can be thought of as a *strict* generalization of the DiDI-technique, first introduced in [DDS21a], which now applies to uncharted borders.

In a recent breakthrough result, Limaye, Srinivasan and Tavenas [LST21] showed the first *super*polynomial lower bound for constant-depth circuits. Their lower bound result, together with the 'hardness vs randomness' tradeoff result of [CKS18] gives the first deterministic *subexponential*-time blackbox PIT algorithm for general constant-depth circuits. Interestingly, these methods can be adapted in the border setting as well [AF21]. However, compared to their algorithms, our hitting sets are significantly faster!

7.3 Quasi-derandomizing $\overline{\Sigma^{[k]}\Pi\Sigma}$ Circuits

Induction step of DiDIL is important to give any meaningful upper bound of circuit complexity. However, hitting set construction demands less— each inductive step of fanin reduction must preserve nonzeroness. Eventually, we exploit this to give an efficient hitting set construction for $\overline{\Sigma^{[k]}\Pi\Sigma}$, and in the process of reducing the top fanin analyse the bloated model $\mathsf{Gen}(k,\cdot)$.

Theorem 7.3.1 (Efficient hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$) *There exists an explicit quasipolynomial-time ($s^{O(k\cdot 7^k\cdot \log\log s)}$) hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits of size s and constant k.*

Proof. The basic reduction strategy is same as section 6.5. Let $f_0 := f$ be an arbitrary polynomial in $\overline{\Sigma^{[k]}\Pi\Sigma}$, approximated by $g_0 \in \mathbb{F}(\epsilon)[x]$, computed by a depth-3 circuit $\overline{C}$ of size s over $\mathbb{F}(\epsilon)$, i.e. $g_0 := f_0 + \epsilon\cdot S_0$. Further, assume that $\deg(f_0) < d_0 := d \le s$. Let $g_0 =: \sum_{i\in[k]} T_{i,0}$, such that $T_{i,0}$ is computable by a $\Pi\Sigma$-circuit of size atmost s over $\mathbb{F}(\epsilon)$. As before, define $\mathscr{R}_0 := \mathbb{F}[z]/\langle z^d\rangle$. Thus, $f_0 + \epsilon\cdot S_0 = \sum_{i\in[k]} T_{i,0}$, holds in $\mathscr{R}_0(x,\epsilon)$.

Define $U_{i,0} := T_{i,0}$ and $V_{i,0} := P_{i,0} := Q_{i,0} = 1$ to set the input instance of $\mathsf{Gen}(k,s)$. Of course, we assume that each $T_{i,0} \neq 0$ (otherwise it is a smaller fanin than k).

Φ homomorphism. To ensure invertiblity and facilitate derivation, we define the same Φ as in section 6.5, i.e. $\Phi : \mathbb{F}(\epsilon)[x] \to \mathbb{F}(\epsilon)[x,z]$ such that $x_i \mapsto z\cdot x_i + \alpha_i$. For the upper bound proof, we took $\alpha_i \in \mathbb{F}$ to be *random*; but for the PIT purpose, we cannot work with a random shift. The purpose of shifting was to ensure the invertibility, i.e., $\mathbb{F}(\epsilon) \ni T_{i,0}(a) \neq 0$; that is easy to ensure since over any field,

$$\prod_i T_{i,0}(y,y^2,\dots,y^n) = \prod_{\ell | T_{i,0}} \ell(y,y^2,\dots,y^n) \neq 0\,.$$

Trivially, $\deg(\prod_i T_{i,0}) \le s$, therefore, $a = (i,i^2,\dots,i^n)$, for some $i \in [s]$ works, as a hitting point, to ensure the invertibility! In the proof, we will work with every such a (s-many), and

for the right-value, nonzeroness will be preserved, which suffices.

❑ 0-**th step: Reduction from** k **to** $k-1$

We will use the same notation as in section 6.5. We know that g_1 approximates f_1 correctly in the ring $\mathscr{R}_1(x,\epsilon)$. Rewriting the same, we have

$$f_0 + \epsilon \cdot S_0 = \sum_{i\in[k]} T_{i,0}\,, \quad \text{in } \mathscr{R}_0(x,\epsilon) \tag{7.1}$$

$$\implies f_1 + \epsilon \cdot S_1 = \sum_{i\in[k-1]} T_{i,1}\,, \quad \text{in } \mathscr{R}_1(x,\epsilon)\,. \tag{7.2}$$

Here, define $T_{i,1} := (\Phi(T_{i,0})/\tilde{T}_{k,0}) \cdot \mathrm{dlog}(\Phi(T_{i,0})/\tilde{T}_{k,0})$, for $i \in [k-1]$ and $f_1 := \partial_z\big(\Phi(f_0)/t_{k,0}\big)$, same as before. Also, we will consider $T_{i,1}$ as an element of $\mathbb{F}(x,z,\epsilon)$ and keep track of $\deg(z)$.

The "iff" condition. Note that the equality in Equation 7.1 in $\mathscr{R}_1(\epsilon,x)$ is only "one-sided". Whereas, to reduce identity testing, we need a necessary and sufficient condition: If $f_0 \neq 0$, we *would like* to claim that $f_1 \neq 0$ (over $\mathscr{R}_1(x)$). However, it may not be directly true because of the loss of z-free terms of f_0, due to differentiation. Note that $f_1 \neq 0$ implies $\mathrm{val}_z(f_1) < d =: d_1$. Further, $f_1 = 0$, in $R_1(x)$, implies–

either, (1) $\Phi(f_0)/t_{k,0}$ is z-free. This implies $\Phi(f_0)/t_{k,0} \in \mathbb{F}(x)$, which further implies it is in $\mathbb{F}$, because z-free implies x-free, by substituting $z = 0$, by the definition of Φ. Also, note that $f_0, t_{k,0} \neq 0$ implies $\Phi(f_0)/t_{k,0}$ is a *nonzero* element in $\mathbb{F}$. Thus, it suffices to check whether $\Phi(f_0)|_{z=0} = f_0(a)$, is nonzero or not.

or, (2) $\partial_z(\Phi(f_0)/t_{k,0}) = z^{d_1}\cdot p$, where $p \in \mathbb{F}(z,x)$ such that $\mathrm{val}_z(p) \geq 0$. By simple power series expansion, one can conclude that $p \in \mathbb{F}(x)[[z]]$ (Lemma 2.2.2). Hence,

$$\Phi(f_0)/t_{k,0} = z^{d_1+1}\cdot\tilde{p}, \text{where } \tilde{p} \in \mathbb{F}(x)[[z]] \implies \mathrm{val}_z(\Phi(f_0)) \geq d\,,$$

a contradiction. Here we used the simple fact that differentiation decreases the valuation by 1.

Conversely, it is obvious that $f_0 = 0$ implies $f_1 = 0$. Thus, we have proved the following:

$$f_0 \neq 0 \text{ in } \mathbb{F}[x] \iff f_1 \neq 0 \text{ in } \mathscr{R}_1(x), \quad \text{or} \quad 0 \neq \Phi(f_0)|_{z=0} \in \mathbb{F}.$$

Recall, Claim 6.5.4 shows that $T_{i,1} \in (\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ with a polynomial blowup. Therefore, subject to $z = 0$ test, we have reduced the identity testing problem to $k-1$. We will recurse over this until we reach $k = 1$.

❑ Induction step

Assume that we are at the end of j-th step ($j \geq 1$). Our inductive hypothesis assumes the following invariants:

1. $\sum_{i\in[k-j]} T_{i,j} = f_j + \epsilon \cdot S_j$ in $\mathscr{R}_j(\epsilon, x)$, where $T_{i,j} \neq 0$ and $\mathscr{R}_j := \mathbb{F}[z]/\langle z^{d_j}\rangle$.
2. Each $T_{i,j} = (U_{i,j}/V_{i,j}) \cdot (P_{i,j}/Q_{i,j})$ where $U_{i,j}, V_{i,j} \in \Pi\Sigma$ and $P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma$.
3. $\mathrm{val}_z(T_{i,j}) \geq 0$, for all $i \in [k-j]$. Moreover, $U_{i,j}|_{z=0} \in \mathbb{F}(\epsilon)\backslash\{0\}$ (similarly $V_{i,j}$).
4. $f_0 \neq 0$ iff: $f_j \neq 0$ in $\mathscr{R}_j(x)$, or $\bigvee_{i=1}^{j-1} \left(f_i/t_{k-i,i}|_{z=0} \neq 0, \text{ in } \mathbb{F}(x)\right)$.

Reducing the problem to $k-j-1$. We will follow the $j = 0$ case, *without* applying any homomorphism. Again, this reduction step is exactly the same as before, which yields: $f_j + \epsilon \cdot S_j = \sum_{i\in[k-j]} T_{i,j}$, in $\mathscr{R}_j(x,\epsilon) \implies$

$$f_{j+1} + \epsilon \cdot S_{j+1} = \sum_{i\in[k-j-1]} T_{i,j+1}, \text{ in } \mathscr{R}_{j+1}(x,\epsilon). \tag{7.3}$$

Here, $T_{i,j+1} := \left(T_{i,j}/\tilde{T}_{k-j,j}\right)\cdot \mathrm{dlog}(T_{i,j}/\tilde{T}_{k-j,j})$, and $f_{j+1} := \partial_z(f_j/t_{k-j,j})$, as before.

It remains to show that, all the invariants assumed are still satisfied for $j+1$. The first 3 invariants are already shown in section 6.5. The 4-th invariant is the iff condition to be shown below.

The "iff" condition in the induction. The above Equation 7.3 pioneers to reduce from $k-j$-summands to $k-j-1$. But we want an 'iff' condition to efficiently reduce the identity testing. If $f_{j+1} \neq 0$, then $\mathrm{val}_z(f_{j+1}) < d_{j+1}$. Further, $f_{j+1} = 0$, in $R_{j+1}(x)$ implies–

either, (1) $f_j/t_{k-j,j}$ is z-free, i.e. $f_j/t_{k-j,j} \in \mathbb{F}(x)$. Now, if indeed $f_0 \neq 0$, then $t_{k-j,j}$ as well as f_j *must be nonzero* in $\mathbb{F}(z,x)$, by induction hypothesis (assuming they are nonzero in $\mathscr{R}_j(x)$). We will eventually show that $f_j/t_{k-j,j}|_{z=0}$, has a small (ARO/ARO) circuit; which helps us to construct a quasi-polynomial size hitting set using Theorem 2.7.7.

or, (2) $\partial_z(f_j/t_{k-j,j}) = z^{d_{j+1}} \cdot p$, where $p \in \mathbb{F}(z,x)$ s.t. $\mathrm{val}_z(p) \geq 0$. By simple power series expansion, one concludes that $p \in \mathbb{F}(x)[[z]]$ (Lemma 2.2.2). Hence,

$$\begin{aligned} & \frac{f_j}{t_{k-j,j}} \in z^{d_{j+1}+1} \cdot \tilde{p}, \text{ where } \tilde{p} \in \mathbb{F}(x)[[z]], \\ \implies & \mathrm{val}_z(f_j) \geq d_j, \\ \implies & f_j = 0, \text{ in } \mathscr{R}_j(x). \end{aligned}$$

Conversely,

$$\begin{aligned} f_j = 0, \text{ in } \mathscr{R}_j(x) \implies & \mathrm{val}_z(f_j/\tilde{T}_{k-j,j}) \geq d_j - \nu_{k-j,j} \\ \implies & \mathrm{val}_z(\partial_z(f_j/\tilde{T}_{k-j,j})) \geq d_j - \nu_{k-j,j} - 1 = d_{j+1} \\ \implies & \partial_z(f_j/\tilde{T}_{k-j,j}) = 0, \text{ in } \mathscr{R}_{j+1}(\epsilon, x). \end{aligned}$$

Fixing $\epsilon = 0$, we deduce that

$$f_{j+1} = \partial_z(f_j/t_{k-j,j}) = 0.$$

Thus, we have proved that $f_j \neq 0$ in $\mathscr{R}_j(x)$ iff

$$f_{j+1} \neq 0 \text{ in } R_{j+1}(x), \text{ or }, \quad 0 \neq (f_j/t_{k-j,j})|_{z=0} \in \mathbb{F}(x).$$

This concludes the proof of the 4-th invariant.

Note: In the above substitution ($z = 0$), ($\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$) maybe undefined by directly evaluating at numerator and denominator, i.e. $= 0/0$. But we can keep track of the z degree of numerator and denominator, which will be polynomially bounded as seen in Claim 6.5.4. We can interpolate and cancel the z-powers to get the ratio.

❑ Constructing the hitting set

The above discussion has reduced the problem of testing $\Phi(f)$, to testing f_{k-1}, or $f_j/t_{k-j,j}|_{z=0}$, for $j \in [k-2]$. We know that $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO})$, of size $s^{O(k7^k)}$, from Claim 6.5.4. We obtain the hitting set of $\Pi\Sigma$ circuit from Theorem 2.7.5, and for $\Sigma\wedge\Sigma$ circuit, we obtain the hitting set from Theorem 2.7.7 (due to Lemma 2.6.11). Finally, we combine the two hitting sets using Lemma 2.7.3, and use the fact that the syntactic degree is bounded by $s^{O(k)}$, to obtain a hitting set $\mathscr{H}_{k-1}$, of size $s^{O(k7^k \log\log s)}$.

However, it remains to show the following –

1. efficient hitting set for $f_j/t_{k-j,j}|_{z=0}$, for $j \in [k-2]$, and most importantly,
2. how to translate these hitting sets to that of $\Phi(f)$.

Recall: Claim 6.5.6 shows that $f_k/t_{k-j,j}|_{z=0} \in \text{ARO}/\text{ARO}$, of size $s^{O(k7^k)}$ (over $\mathbb{F}(x)$). Thus, it has a hitting set $\mathscr{H}_j$ of size $s^{O(k7^k \log\log s)}$ (Theorem 2.7.7).

To translate the hitting set, we need a small property which will bridge the gap of lifting the hitting set to f_0.

Claim 7.3.1 (Fix x) For $b \in \mathbb{F}^n$, if the following two things hold:

(i) $f_{j+1}|_{x=b} \neq 0$, in $\mathscr{R}_{j+1}$, and

(ii) $\text{val}_z(\tilde{T}_{k-j,j}|_{x=b}) = \nu_{k-j,j}$,

then, over the ring $\mathscr{R}_j$,

$$f_j|_{x=b} \neq 0 .$$

Proof. Suppose the hypothesis holds, and $f_j|_{x=b} = 0$, over $\mathscr{R}_j$. Then,

$$\text{val}_z\left(\left(\frac{f_j}{\tilde{T}_{k-j,j}}\right)\Big|_{x=b}\right) \geq d_j - \nu_{k-j,j} \implies \text{val}_z(\partial_z\left(\left(\frac{f_j}{\tilde{T}_{k-j,j}}\right)\Big|_{x=b}\right) \geq d_{j+1}.$$

The last condition implies that $\partial_z(f_j/\tilde{T}_{k-j,j})|_{x=b} = 0$, in $\mathscr{R}_{j+1}(x)$. Fixing $\epsilon = 0$ we deduce $f_{j+1}|_{x=b} = 0$. This is a contradiction! □

Finally, we have already shown in section 6.5 that $\tilde{T}_{k-j,j} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, and $t_{k-j,j} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\text{ARO}/\text{ARO})$, of size $s^{O(k7^k)}$, which is similar to f_{k-1}. Note: val_z of a $\Sigma\wedge\Sigma$ again reduces to a $\Sigma\wedge\Sigma$ PIT question.

Joining the dots: The final hitting set. We now have all the ingredients to construct the hitting set for $\Phi(f_0)$. We know $\mathscr{H}_{k-1}$ works for f_{k-1} (as well as $t_{2,k-2}$, because they both are of the same size and belong to $(\Pi\Sigma/\Pi\Sigma)\cdot(\text{ARO}/\text{ARO})$). This lifts to f_{k-2}. But from the 4-th invariant, we know that $\mathscr{H}_{k-2}$ works

for the $z = 0$ part. Eventually, lifting this using Claim 7.3.1, the final hitting set (in x) will be

$$\mathscr{H} := \bigcup_{j\in[k-1]} \mathscr{H}_j .$$

We remark that we do not need extra hitting set for each $t_{k-j,j}$, because it is already covered by $\mathscr{H}_{k-1}$. We have also kept track of $\deg(z)$ which is bounded by $s^{O(k)}$. We use a trivial hitting set for z which *does not* change the size. Thus, we have successfully constructed a $s^{O(k7^k \log\log s)}$-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$. □

Remark. This is a PIT for $\overline{\mathrm{Gen}(k,s)}$, and that too for any field of characteristic $= 0$ or $\geq d$.

7.4 Border PIT for log-variate Depth-3 Circuits

In this section, we prove Theorem 7.2.2. This proof is dependent on adapting and extending [FGS18] proof, by showing that there is a $\mathrm{poly}(s)$-time hitting set for log-variate $\overline{\Sigma\wedge\Sigma}$-circuits.

Theorem 7.4.1 (Derandomizing log-variate $\overline{\Sigma\wedge\Sigma}$) *There is a* $\mathrm{poly}(s)$*-time hitting set for* $n = O(\log s)$ *variate* $\overline{\Sigma\wedge\Sigma}$*-circuits of size* s.

Proof sketch. Let $g = f + \epsilon \cdot Q$, such that $g \in \Sigma\wedge\Sigma$, over $\mathbb{F}(\epsilon)$, approximates $f \in \overline{\Sigma\wedge\Sigma}$. The idea is the same as [FGS18]—

1. show that f has $\mathrm{poly}(s,d)$ partial derivative space, and,
2. low partial derivative space implies low cone-size monomials,
3. One can extract low cone-size monomials efficiently,
4. number of low cone-size monomials is $\mathrm{poly}(sd)$-many.

We remark that (2) is direct from [For14, Corollary 4.14] (with origins in [FS13b]); see Theorem 2.6.1. (4) is also directly taken from [FGS18, Lemma 5] once we assume (1); for the full statement we refer to Lemma 2.6.2.

To show (1), we know that g has $\text{poly}(s,d)$ partial-derivative space over $\mathbb{F}(\epsilon)$. Denote

$$V_\epsilon := \left\langle \frac{\partial g}{\partial x^a} \mid a < \infty \right\rangle_{\mathbb{F}(\epsilon)}, \text{ and } V := \left\langle \frac{\partial f}{\partial x^a} \mid a < \infty \right\rangle_{\mathbb{F}}.$$

Consider the matrix M_ϵ, where we index the rows by ∂_{x^a}, while columns are indexed by monomials (say supporting g), and the entries are the operator-values. Suppose, $\dim(V_\epsilon) =: r \leq \text{poly}(s,d)$ (because of $\Sigma\wedge\Sigma$). That means, any $(r+1)$-many polynomials $\frac{\partial g}{\partial x^a}$ are linearly dependent. In other words, determinant of any $(r+1)\times(r+1)$ minor of M_ϵ is 0. Note that $\lim_{\epsilon\to 0} M_\epsilon = M$, the corresponding partial-derivative matrix for f. Crucially, the zeroness of the determinant of any $(r+1)\times(r+1)$ minor of M_ϵ translates to the corresponding $(r+1)\times(r+1)$ submatrix of M as well [one can also think of det as a "continuous" function, yielding this property]. In particular, $\dim(V) \leq r \leq \text{poly}(s,d)$.

Finally, to show (3), we note that the coefficient extraction lemma [FGS18, Lemma 4] also holds over $\mathbb{F}(\epsilon)$. Thus, given the circuit of g, we can decide whether the coefficient of $m =: x^a$ is zero or not, in $\text{poly}(\text{cs}(m), s, d)$-time; see Lemma 2.6.3. Note: the coefficient is an arbitrary element in $\mathbb{F}(\epsilon)$; however we are only interested in its nonzeroness, which is merely 'unit-cost' for us.

We only extract monomials with cone-size $\text{poly}(s,d)$ (property (2)), and there are only $\text{poly}(s,d)$ many such monomials. Therefore, we have a $\text{poly}(s)$-time hitting set for $\overline{\Sigma\wedge\Sigma}$. □

Once we have Theorem 7.4.1, we argue that this polynomial-time hitting set can be used to give a polynomial-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$. We restate Theorem 7.2.2 with proper complexity below.

Theorem 7.4.2 (Efficient hitting set for log-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$) *There exists an explicit $s^{O(k7^k)}$-time hitting set for $n = O(\log s)$ variate, size-s, $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits.*

Proof sketch. We proceed similarly as in section 7.3, with same notations. The reduction and branching out remains exactly the same; in the end, we get that $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma)\cdot(\text{ARO}/\text{ARO})$. Crucially, observe that this ARO is not a generic

polynomial-sized ARO; these AROs are de-bordered log-variate $\overline{\Sigma\wedge\Sigma}$ circuits.

From Theorem 7.4.1, we know that there is a $s^{O(k7^k)}$-time hitting set (because of the size blowup, as seen in section 6.5). Using Lemma 2.7.3, it is easy to combine this hitting set with $\Pi\Sigma$-hitting.

Moreover, $t_{k-j,j}$ are also of the form $(\Pi\Sigma/\Pi\Sigma)\cdot(\text{ARO}/\text{ARO})$, where again these AROs are de-bordered log-variate $\overline{\Sigma\wedge\Sigma}$ circuits, and $s^{O(k7^k)}$-time hitting set exists. Therefore, take the union of the hitting sets (as before), each of size $s^{O(k7^k)}$. This gives the final hitting set, which is again $s^{O(k7^k)}$-time constructible! □

7.5 Discussion

As mentioned before, [Agr+16] came up with *faithful* map by using Jacobian + certifying path technique, which is more about algebraic rank rather than linear-rank. This unified all the derandomization results for bounded depth-3 circuits. However, it is not at all clear how it behaves wrt $\lim_{\epsilon\to 0}$, mainly because the Jacobian might just be a *nonzero* polynomial in the ideal $\langle\epsilon\rangle_{\mathbb{F}[\epsilon,x]}$, but clearly, it flips to zero in the limit. This makes the whole Jacobian machinery collapse in the approximative setting.

However, the bigger question might be whether there is an 'approximate' version of the Jacobian criterion, which preserves the independence, even in the limit. For now, it is not at all clear if such a criterion exists, and if so, whether it could be of any computational help, to expedite the state-of-the-art PIT algorithms. Nevertheless, we finish by asking the following meta-question.

Meta-question on algebraic independence

Can we define *approximative algebraic dependence*, which is coherent with the Jacobian polynomial, or, some version of it?

Approximative τ-conjectures and their consequences

8

"Thoreau wrote, "Simplify! Simplify!" And, indeed, simplification is one mark of real genius."

— Dan Ariely, Predictably Irrational.

Abstract. In this chapter, we study the SOS-τ-conjecture, respectively, the SOC-τ-conjecture, in the *border* or *approximative sense*. The SOS- and SOC-hardness, defined in Chapter 3, can also be extended in the border or approximative complexity-theoretic sense, which would eventually strengthen the lower bound and PIT consequences. Furthermore, one can also conjecture similar τ-conjectures, and show similar big-ticket consequences, in GCT.

This chapter is based on the extended version of the published article in CSR'21, titled *Real tau-Conjecture for sum-of-squares: A unified approach to lower bound and derandomization*, which is currently under review [Dut21].

8.1 Border-SOS-τ-conjecture and VNP $\not\subseteq \overline{\text{VP}}$

Definition 8.1.1 (Approximative SOS and border-support-sum size $\overline{S}_R(f)$) *Let R be a ring. An n-variate polynomial $f(x) \in R[x]$ is* approximated as a (weighted) SOS, *if there exists an integer $M \geq 0$ such that*

$$f(x) = \lim_{\epsilon \to 0} \frac{1}{\epsilon^M} \sum_{i=1}^{s} c_i f_i^2(x, \epsilon), \tag{8.1}$$

for some top-fanin *s, where $f_i \in R[x, \epsilon]$ and $c_i \in R[\epsilon]$.*

The size in the representation of f in (Equation 8.1) is the border support-sum, *the sum of the support size (or sparsity) of the polynomials f_i over $R[\epsilon]$. The* border-support-sum size *of f, is defined as the minimum border-support-sum of f, denoted by $\overline{S}_R(f)$, or simply $\overline{S}(f)$, when the ring R is clear from the context.*

Note that, by definition, $\bar{S}_R(f) \leq S_R(f)$. In particular, when f is univariate and has sparsity, $\|f\|_0 = d+1$, over any field $R = \mathbb{F}$, of characteristic $\neq 2$, similar bounds hold:

$$\sqrt{d+1} \;\leq\; \bar{S}(f) \;\leq\; S(f) \;\leq\; 2d+2\,. \tag{8.2}$$

We can now conjecture the following, which is *stronger* than Conjecture 3.1.2. Further, we show that proving a τ-conjecture in the border-SOS setting, is enough to separate $\overline{\mathsf{VP}}$ from VNP!

Conjecture 8.1.1 (Border-SOS-τ-conjecture) Consider any non-zero polynomial $f(x) \in \mathbb{R}[x]$. Then, there exists a positive constant $c > 0$ such that the number of distinct real roots of f is at most $c \cdot \bar{S}_{\mathbb{R}}(f)$.

Trivially, Border-SOS-τ-conjecture implies SOS-τ-conjecture. We point out that the decomposition lemma (Lemma 3.3.4) works for approximative circuits as well. This lemma plays the pivotal role to establish a connection between approximative SOS-τ-conjecture and the general circuit hardness, in the border sense.

Lemma 8.1.1 (Border Sum-of-product-of-2) *Let $f(x)$ be an n-variate, homogeneous, degree d polynomial approximated by a circuit C of size s, over $\mathbb{F}(\epsilon)$. Then, there exist polynomials $f_{ij} \in \mathbb{F}[x,\epsilon]$ s.t.*

$$C(x,\epsilon) \;=\; \sum_{i=1}^{s} f_{i1}\cdot f_{i2}\,, \quad \textit{with the following properties:} \tag{8.3}$$

(1) $d/3 \leq \deg_x(f_{i1}), \deg(f_{i2}) \leq 2d/3$, $\forall\, i \in [s]$, and (2) $\deg_x(f_{i1}) + \deg_x(f_{i2}) = d$, $\forall\, i \in [s]$.

The proof is essentially the same proof with frontier $m = d/3$ [Sap21], except that the working field is $\mathbb{F}(\epsilon)$.

Main result. We come to our main result in this section. We show how to connect the number of roots of a univariate polynomial of degree-d, approximated by an SOS-model to a multivariate polynomial that has approximative circuits of exponential size, trivially implying it is not in $\overline{\mathsf{VP}}$, but its explicitness ensures it to be in VNP.

Theorem 8.1.2 *Conjecture 8.1.1 implies that* $\text{VNP}_{\mathbb{C}}$ *is exponentially harder than* $\overline{\text{VP}_{\mathbb{C}}}$.

Proof sketch. The proof is similar to the proof of Theorem 3.3.9. We define $P_{n,k}$ with the similar parameters as in that proof. As f_d is explicit, so is $P_{n,k}$. Therefore $(P_{n,k})_n \in \text{VNP}$.

To show that $(P_{n,k})_n \notin \overline{\text{VP}}$, We will argue that $\overline{\text{size}}(P_{n,k}) \geq d^{1/7} = 2^{\Omega(kn)}$, which trivially implies that $(P_{n,k})_n \notin \overline{\text{VP}}$.

The proof is again via contradiction. If $\overline{\text{size}}(P_{n,k}) \leq d^{1/7}$, then there is a circuit $C(y,\epsilon) \in \mathbb{F}(\epsilon)[x]$ of size $d^{1/7}$, and a $M \geq 0$, such that

$$C(y,\epsilon) \;=\; \epsilon^M P_{n,k} + \epsilon^{M+1} Q(y,\epsilon)\,.$$

By Lemma 8.1.1, there exist polynomials $Q_i(y,\epsilon)$ such that

$$C(y,\epsilon) \;=\; \sum_{i=1}^{s} c_i Q_i(y,\epsilon)^2\,,$$

where $s = O(d^{1/7}\cdot n^2)$, and $\deg_y(Q_i) \leq 2n/3$.

If we apply the inverse multilinear Kronecker map $\psi_{n,k}$ to the polynomials Q_i, we get

$$\epsilon^M f_d + \epsilon^{M+1}\psi_{n,k}(Q) = \sum_{i=1}^{s} c_i\, g_i^2\,,$$

where $g_i(x) = \psi_{n,k}(Q_i(y))$. Note that, $\text{sp}(g_i) \leq \text{sp}(Q_i)$, over $\mathbb{F}(\epsilon)$. For the sparsity of Q_i, we use the general bound (Equation 2.1). That is, $\text{sp}(Q_i) \leq \binom{kn+2n/3}{2n/3}$, for all $i \in [s]$. Thus, by definition,

$$\bar{S}(f_d) \;\leq\; s\cdot\binom{kn+2n/3}{2n/3}\,.$$

The same calculation as in the proof of Theorem 3.3.9 shows that $\bar{S}(f_d) = o(d)$. This is a contradiction; this is because the coefficients of f_d satisfies the Kurtz condition implying f_d has all distinct real roots, then Conjecture 8.1.1 and Lemma 8.3.2 imply that

$$\bar{S}_{\mathbb{R}}(f_d) \;\geq\; \Omega(d) \;\Longrightarrow\; \bar{S}_{\mathbb{C}}(f_d) \;\geq\; \Omega(d)\,.$$

□

Similarly, one can lift the approximative hardness of a univariate polynomial of degree d in the SOS-model (with ϵ parameter) to a multivariate polynomial that has approximative circuits of super-polynomial size, implying it is not in $\overline{\mathsf{VP}}$, but its explicitness ensures it to be in VNP. We state the theorem without proving it.

Theorem 8.1.3 *If there exists an approximative SOS-hard explicit family (f_d) with hardness parameter $\epsilon = \epsilon(d) = \omega\left(\sqrt{\frac{\log\log d}{\log d}}\right)$, then* $\mathsf{VNP} \not\subseteq \overline{\mathsf{VP}}$.

8.2 Border-SOC-hardness and Efficient Hitting Set for $\overline{\mathsf{VP}}$

In this section, we introduce Border-SOC-τ-conjecture and show its intrinsic connection to construct efficient hitting sets for $\overline{\mathsf{VP}}$. Though the existence of a polynomial size hitting set is known due to [HS80a], the best complexity bound known for constructing a hitting set for $\overline{\mathsf{VP}}$ is PSPACE [FS18; Guo+19]. The main difficulty comes from certifying that the set that has been constructed is indeed a hitting set. Very recently, Kumar, Saptharishi, and Solomon [KSS19] showed that the hardness of constant-variate polynomials in the approximative sense suffices to construct an HSG for $\overline{\mathsf{VP}}$.

Theorem 8.2.1 *[KSS19, Theorem 1.6] Let P be a k-variate polynomial in $\mathbb{F}[x]$ of degree d such that* $\mathrm{coef}(P)$ *can be computed in time* $\mathrm{poly}(d)$. *Further, suppose $\overline{\mathrm{size}}(P) > s^{10k+2}\, d$, for some parameter s. Then there is a* $\mathrm{poly}(s)$ *size hitting set for $\overline{\mathscr{C}(s,s,s)}$.*

Next, we define the approximative SOC-model and its complexity measure.

Definition 8.2.1 (Approximative SOC and border-support-union size $\overline{U}_R(f,s)$) *Let R be a ring. An n-variate polynomial $f(x) \in R[x]$ is* approximated as a SOC, *if there exists an*

integer $M \geq 0$, such that

$$f(x) \;=\; \lim_{\epsilon \to 0} \frac{1}{\epsilon^M} \sum_{i=1}^{s} c_i f_i^3(x, \epsilon)\,, \tag{8.4}$$

for some top-fanin *s, where $f_i \in R[x, \epsilon]$ and $c_i \in R[\epsilon]$.*

The size *of the representation of f in (Equation 8.4) is the size of the support-union over $R[\epsilon]$, i.e. $|\bigcup_{i=1}^{s} \text{supp}(f_i)|$, where $\text{supp}(f_i)$ denotes the set of monomials with a nonzero coefficient in f_i. The* border support-union size *of f with respect to s, denoted $\overline{U}_R(f, s)$, is defined as the minimum border support-union size when f is written as in (Equation 8.4).*

Note that, by definition, $\overline{U}_R(f, s) \leq U_R(f, s)$. In particular, when f is a univariate polynomial, and has sparsity $\text{sp}(f) = d+1$, similar bounds hold over any field $R = \mathbb{F}$, of characteristic $\neq 2, 3$:

$$(d+1)^{1/3} \;\leq\; \overline{U}(f, s) \;\leq\; U(f, s) \;\leq\; d + 1\,. \tag{8.5}$$

Thus, it follows that for s large enough, $\overline{U}(f, s)$ is *small*. However, it is unclear whether this is true when $s = o(\sqrt{d})$. We call a polynomial family approximative SOS-hard, if its border support-union size attains the trivial upper bound.

Definition 8.2.2 (Approximative SOC-hardness) *A polynomial family $(f_d(x))_d$ is* approximative SOC-hard, *if there is a constant $0 < \varepsilon < 1/2$ such that $\overline{U}_{\mathbb{F}}(f_d, d^\varepsilon) = \Omega(d)$.*

Once we have defined approximative SOC-hardness, we conjecture a *stronger* postulate.

Conjecture 8.2.1 (Border-SOC-τ-conjecture) Consider any non-zero polynomial $f \in \mathbb{R}[x]$. Then, there exist positive constants $\varepsilon < 1/2$, and c such that the number of distinct real roots of f is at most $c \cdot \overline{U}_{\mathbb{R}}(f, d^\varepsilon)$.

The main ingredient is a SOC decomposition in the approximative sense. This decomposition is very similar to [DST21, Lemma 16], except that the working field is $\mathbb{F}(\epsilon)$.

Lemma 8.2.2 (Approximative SOC decomposition) *There exists a constant c, such that for any n-variate polynomial $p \in \mathbb{F}[x]$ of degree d that can be approximated by a circuit of*

size s, we have a representation

$$\epsilon^M p + \epsilon^{M+1} q(x,\epsilon) = \sum_{i=1}^{(sd)^c} q_i^3, \tag{8.6}$$

where $q_i \in \mathbb{F}[\epsilon][x]$, for all $i \in [(sd)^c]$, such that

1. *$deg(q_i) < \frac{4}{11} d$,*
2. *q_i has a circuit of size $\mathrm{poly}(s,d)$ over $\mathbb{F}(\epsilon)$.*

Assuming the above conjecture, it is not hard to construct an explicit and efficient hitting set for $\overline{\mathrm{VP}}$. It essentially uses the above lemma and Theorem 8.2.1. The proof goes along the lines of [DST21, Theorem 11].

Theorem 8.2.3 (Hitting set for $\overline{\mathrm{VP}}$) *Border-SOC-τ-conjecture implies a polynomial-time hitting set for $\overline{\mathrm{VP}}$.*

8.3 Border-SOS Structures

In this section, we prove Lemma 3.9.2 in the border setting. The proof is slightly more subtle, because we would not be able to take $\sqrt{c_i}$, the weights, inside the square-root that easily. However, the following lemma allows us to do so.

Lemma 8.3.1 *Let $f \in \mathbb{C}[x]$, such that $\overline{S}_{\mathbb{C}}(f) = s$. Then, $\exists m_1, \ldots, m_s, M \in \mathbb{Z}_{\geq 0}$ and polynomials $S, f_i \in \mathbb{C}[x,\epsilon]$, for $i \in [s]$, such that*

$$\epsilon^M \cdot f + \epsilon^{M+1} \cdot S = \sum_{i\in[s]} \epsilon^{m_i} \cdot f_i^2,$$

where $\sum_{i\in[s]} \mathrm{sp}_{\mathbb{C}(\epsilon)}(f_i) = s$.

Proof. $\overline{S}_{\mathbb{C}}(f) = s \implies \exists c_i \in \mathbb{F}[\epsilon]$ *, and $f_i, S \in \mathbb{C}[x,\epsilon]$, for $i \in [s]$, such that

$$\epsilon^M \cdot f + \epsilon^{M+1} \cdot S = \sum_{i\in[s]} c_i \cdot f_i^2 .$$

If, for some i, c_i is a perfect square of a polynomial in $\mathbb{F}[\epsilon]$, we will replace f_i^2 with $\epsilon^0 \cdot (\sqrt{c_i} \cdot f_i)^2$. Note that, $\mathrm{sp}(\sqrt{c_i} \cdot f_i) = \mathrm{sp}(f_i)$, over $\mathbb{F}(\epsilon)$.

* Remember: the underlying field is $\mathbb{F}(\epsilon)$, but we can always clear out the denominators such that the minimum ϵ-power does not change.

If c_i is not a square, let $c_i = \epsilon^{m_i} \cdot p_i$, where $p_i \in \mathbb{F}[\epsilon]$, such that $\epsilon \nmid p_i$. It is not hard to show that $\sqrt{p_i} \in \mathbb{F}[[\epsilon]]$, by simple binomial expansion. Moreover, let $q_i := \sqrt{p_i} \mod \epsilon^{M+1}$. It suffices to look at mod ϵ^{M+1}, since we are interested in the coefficient of ϵ^M. Therefore, by using the above trick, one could replace $c_i \cdot f_i^2$, by $\epsilon^{m_i} \cdot (q_i \cdot f_i)^2$. Note that, in this process the S polynomial would change, but the coefficient of ϵ^M remains f. Furthermore, $\mathrm{sp}(f_i) = \mathrm{sp}(q_i \cdot f_i)$. This finishes the proof. □

Using the above lemma, one could prove the border version of Lemma 3.9.2.

Lemma 8.3.2 *$\bar{S}_{\mathbb{R}}(\mathfrak{R}(f)) \leq 2 \cdot \bar{S}_{\mathbb{C}}(f)$, for any $f \in \mathbb{C}[x]$.*

Proof. The proof is almost similar to Lemma 3.9.2, once we have proved Lemma 8.3.1. However, we still present the whole proof for brevity.

Suppose, $\bar{S}_{\mathbb{C}}(f) = s$. Then, $\exists m_1, \ldots, m_s, M \in \mathbb{Z}_{\geq 0}$ and polynomials $S, f_i \in \mathbb{C}[x, \epsilon]$, for $i \in [s]$, such that

$$\epsilon^M \cdot f + \epsilon^{M+1} \cdot S = \sum_{i \in [s]} \epsilon^{m_i} \cdot f_i^2 ,$$

where $\sum_{i \in [s]} \mathrm{sp}_{\mathbb{F}(\epsilon)}(f_i) = s$. Therefore,

$$\begin{aligned} \epsilon^M \cdot \mathfrak{R}(f) + \epsilon^{M+1} \cdot \mathfrak{R}(S) &= \sum_{i=1}^{s} \epsilon^{m_i} \mathfrak{R}(f_i^2) \\ &= \sum_{i=1}^{s} \epsilon^{m_i} \cdot \mathfrak{R}(\mathfrak{R}(f_i) + \iota \cdot \mathfrak{I}(f_i))^2 \\ &= \sum_{i=1}^{s} \epsilon^{m_i} \cdot (\mathfrak{R}(f_i)^2 - \mathfrak{I}(f_i)^2) . \end{aligned}$$

In the above, by $\mathfrak{R}(S)$, for an $\sum s_i x^i = S \in \mathbb{C}[x, \epsilon]$, we mean that $\sum \mathfrak{R}(s_i) x^i$, i.e., $\mathfrak{R}(s_i) \in \mathbb{R}[\epsilon]$. The last expression implies that

$$\begin{aligned} \bar{S}_{\mathbb{R}}(\mathfrak{R}(f)) &\leq \sum_{i=1}^{s} \mathrm{sp}_{\mathbb{R}(\epsilon)}(\mathfrak{R}(f_i)) + \sum_{i=1}^{s} \mathrm{sp}_{\mathbb{R}(\epsilon)}(\mathfrak{I}(f_i)) \\ &\leq \sum_{i=1}^{s} 2 \cdot \mathrm{sp}_{\mathbb{C}(\epsilon)}(f_i) \\ &= 2 \cdot \bar{S}_{\mathbb{C}}(f) . \end{aligned}$$

□

8.4 Discussion

This work effectively establishes that studying the number of real roots of univariate polynomials for approximative sum-of-squares representation (respectively cubes) is fecund. In fact, proving a strong upper bound suffices to solve major open problems in GCT.

Here are some immediate questions of interest which require rigorous investigation.

1. Does Border-SOS-τ-conjecture solve PIT completely? The current proof technique fails to reduce from cubes to squares.
2. Prove the upper bound on the number of real roots for the polynomials, approximated by *sum of constantly* many squares. Currently, we only know it for $s = 2$, since Theorem 3.8.1 can be extended in the border setting as well, due to the closure property, which follows from factoring, and using the fact that $\overline{\Sigma\Pi} = \Sigma\Pi$.

Future Direction in GCT | 9

"The power of mathematics is often to change one thing into another, to change geometry into language."

— Marcus du Sautoy, FRS, University of Oxford.

In this thesis, we study and solve some interesting GCT questions, using non-geometric tools. Essentially, we investigate the significance of roots (τ-conjecture) and non-roots (PIT) in the approximative setting. More concretely, Chapter 6 studies qualitative power of approximations in the restricted setting (Q3). On the other hand, Chapter 7 studies non-roots, and removes randomness in the restricted setting (Q4). Finally, Chapter 8 establishes that studying the number of real roots of univariate polynomials, *approximated* by sum-of-squares representation (respectively cubes), is fruitful, and leads to strong lower bound results (Q2). Moreover, the approximative setting (GCT) brings us close to solving $\mathsf{P} \neq \mathsf{NP}$, answering Q1.

Of course, we are far away from understanding the magic and super-structures that GCT has to offer. More likely, it would be deep, compelling. A subgroup even claims it to be "the only game in town" (not surprisingly, a claim disputed by the fans of rival idea!). Below, we discuss some obvious steps and questions to take towards advancing the current state-of-the-art.

9.1 Quest for More De-bordering Results

In Chapter 6, we talked about the DiDIL-technique introduced in [DDS21b], and successfully de-bordered $\overline{\Sigma^{[k]}\Pi\Sigma}$. In the same work [DDS21b], we have also extended our results to restricted depth-4 circuits. This opens a variety of questions which would enrich border-complexity theory.

Open Problems 9.1

1. Does $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \Sigma\Pi\Sigma$, or $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \mathsf{VF}$, i.e. does it have a small formula?
2. Can we de-border $\overline{\Sigma \wedge \Sigma\Pi^{[\delta]}}$, or $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$, for constant k and δ? In [DDS21b], we have already designed a quasipolynomial-time PIT for the same.
3. Can we de-border $\overline{\Sigma^{[2]}\Pi\Sigma\wedge^{[2]}}$? i.e. the bottom-layer has variable mixing.

Recent developments. In [DDS21b], we asked whether the following is true: $\mathsf{VBP} \neq \overline{\Sigma^{[k]}\Pi\Sigma}$? After our work, the breakthrough lower bound result by Limaye, Srinivasan & Tavenas [LST21] showed a *super*polynomial separation between depth-3 (unbounded fanin) circuits and IMM (Iterated Matrix Multiplication), with different parameters. IMM can be shown to have 'small' ABP. Since, the method is linear-rank-based, it can be extended to the border as well, establishing a superpolynomial separation between VBP and $\overline{\Sigma^{[k]}\Pi\Sigma}$. Very recently, we showed an *exponential* gap between the two classes [DS22].

9.2 Quest for Efficient Border PITs

Here are a few interesting derandomization questions in the restricted border regime, which could lead to some involved technical developments.

Open Problems 9.2

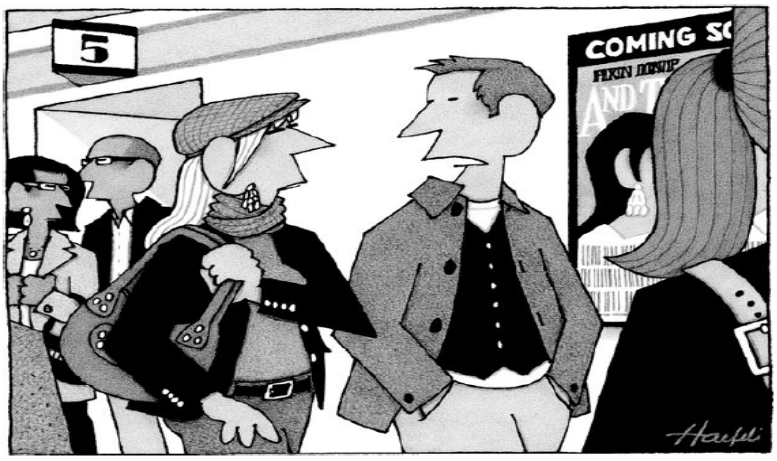
1. Can we improve the current hitting set of $s^{\exp(k)\cdot\log\log s}$ to $s^{O(\mathrm{poly}(k)\cdot\log\log s)}$, or even a $\mathrm{poly}(s)$-time hitting set? The current technique seems to blowup the exponent.
2. Can we find a polynomial time hitting set for the border class $\overline{\Sigma^{[2]}\Pi\Sigma\wedge^{[2]}}$?

9.3 Approximative τ-conjectures

As far as the literature is concerned, we are not aware of approximative τ-conjectures and their implications. In that regard, this thesis introduces such conjectures and show interesting connection with GCT. Here are some questions that should be explored.

Open Problems 9.3

1. Does Border-SOC-τ-conjecture hold for a 'generic' polynomial f (say, over $\mathbb{Q}$)?
2. Are border τ-conjectures *strictly stronger* than the classical τ-conjectures?
3. Is there an approximative version of the classical Blum-Shub-Smale τ-conjecture, that is useful in the GCT sense?



"I can't decide if that was bad in a good way, good in a good way, good in a bad way, or bad in a bad way."

10 Conclusion

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

— Douglas Adams, The Long Dark Tea-Time of the Soul.

In this thesis, we looked at some central questions in the field of Algebraic Complexity and Geometric Complexity Theory. The division has been quite intentional: including GCT results as a subpart of the Algebraic Complexity, would not do full justice to the GCT program. We have already mentioned the strengths and ambitions of the program in the introduction.

In the first part, the thesis provides some important insights of proving strong lower bounds and derandomizing PIT, by studying possibly the simplest models of computation. It further attempts to gain a better understanding of hitting sets for restricted algebraic models. Proving $\mathsf{VP} \neq \mathsf{VNP} \implies \mathsf{PIT} \in \mathsf{P}$, is an exciting open problem in the area of algebraic hardness versus randomness.

In the second part, we provide a new technique for de-bordering restricted border circuit classes and constructing hitting sets for the same. Proving $\overline{\mathsf{VP}} = \mathsf{VP}$, or separating the two, is probably the most exciting open problem in the de-bordering paradigm.

I will end the thesis by writing one of my favorite poems by Emily Dickinson. She takes an abstract feeling or idea – in this case, hope – and likens it to something physical, visible, and tangible – here, a singing bird. Hope, for Dickinson, sings its wordless tune and never stops singing it: nothing can faze it.

"Hope" is the thing with feathers -
That perches in the soul -
And sings the tune without the words -
And never stops - at all -
And sweetest – in the Gale – is heard –
And sore must be the storm –
That could abash the little Bird
That kept so many warm –
I've heard it in the chillest land –
And on the strangest Sea –
Yet – never – in Extremity,
It asked a crumb – of me.

— Emily Dickinson, 'Hope' is the thing with feathers.

Bibliography

Here are the references in alphbetical order.

[Aar16] Scott Aaronson. "$P \stackrel{?}{=} NP$." In: *Open problems in mathematics*. Springer, 2016, pp. 1–122 (cited on pages 2, 9, 15).

[Agr05] Manindra Agrawal. "Proving lower bounds via pseudo-random generators." In: 25^{th} *International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05)*. Springer. 2005, pp. 92–105 (cited on page ix).

[Agr20] Manindra Agrawal. Private Communication. 2020 (cited on page 99).

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. "Bootstrapping variables in algebraic circuits." In: *Proceedings of the National Academy of Sciences* 116.17 (2019). Preliminary version in the Proceedings of the 50^{th} Annual ACM Symposium on Theory of Computing (STOC'18), pp. 8107–8118 (cited on pages 20, 69, 70, 168).

[Agr+15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. "Hitting-sets for ROABP and sum of set-multilinear circuits." In: *SIAM Journal on Computing* 44.3 (2015), pp. 669–697 (cited on pages 122, 166).

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. "PRIMES is in P." In: *Annals of mathematics* (2004), pp. 781–793 (cited on page 8).

[Agr+16] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. "Jacobian hits circuits: Hitting sets, lower bounds for depth-D occur-k formulas and depth-3 transcendence degree-k circuits." In: *SIAM Journal on Computing* 45.4 (2016). Preliminary version in 44^{th} Symposium on Theory of Computing, 2018 (STOC'12), pp. 1533–1562 (cited on pages 51–53, 118–120, 122, 167, 170, 178).

[ASS13] Manindra Agrawal, Chandan Saha, and Nitin Saxena. "Quasi-polynomial hitting-set for set-depth-Δ formulas." In: *Proceedings of the* 45^{th} *Annual ACM Symposium on Theory of Computing (STOC'13)*. 2013, pp. 321–330 (cited on page 119).

[AS09] Manindra Agrawal and Ramprasad SaptharishiI. "Classifying polynomials and identity testing." In: Indian Academy of Sciences. 2009 (cited on page 116).

[AV08] Manindra Agrawal and V Vinay. "Arithmetic Circuits: A Chasm at Depth Four." In: 49^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*. IEEE. 2008, pp. 67–75 (cited on pages x, 20, 60, 69, 70, 114–116).

[All+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. "Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing." In: *Proceedings of the* 50^{th} *Annual ACM Symposium on Theory of Computing (STOC'18)*. 2018, pp. 172–181 (cited on page 15).

[AW16] Eric Allender and Fengming Wang. "On the power of algebraic branching programs of width two." In: *Computational Complexity* 25.1 (2016). Preliminary version in 38^{th} International Colloquium on Automata, Languages, and Programming (ICALP'11), pp. 217–253 (cited on page 140).

[AC19] Josh Alman and Lijie Chen. "Efficient construction of rigid matrices using an NP oracle." In: 60^{th} *Annual Symposium on Foundations of Computer Science (FOCS'19)*. 2019, pp. 1034–1055 (cited on page 39).

[AP94] Noga Alon and Pavel Pudlak. "Superconcentrators of depths 2 and 3; odd levels help (rarely)." In: *Journal of Computer and System Sciences* 48.1 (1994), pp. 194–202 (cited on page 38).

[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). Preliminary version in the Proceedings of the 31^{st} Computational Complexity Conference (CCC'16), pp. 1–30 (cited on page 122).

[And20] Robert Andrews. "Algebraic hardness versus randomness in low characteristic." In: *Proceeding of the* 35^{th} *Computational Complexity Conference (CCC'20)* (2020) (cited on page 20).

[AF21] Robert Andrews and Michael A Forbes. "Ideals, Determinants, and Straightening: Proving and Using Lower Bounds for Polynomial Ideals." In: *Proceedings of the* 53^{rd} *Annual ACM Symposium on Theory of computing (STOC'21)*. (2021) (cited on pages 165, 170).

[BM22] Boaz Barak and Ankur Moitra. "Noisy tensor completion via the Sum-of-squares Hierarchy." In: *Mathematical Programming* 193(2) (2022). Preliminary version in the Proceedings of the 29^{th} Annual Conference on Learning Theory (COLT'16), pp. 513–548 (cited on page 60).

[BMS13] Malte Beecken, Johannes Mittmann, and Nitin Saxena. "Algebraic independence and blackbox identity testing." In: *Information and Computation* 222 (2013). Preliminary version in 38^{th} International Colloquium on Automata, Languages and Programming (ICALP'11), pp. 2–19 (cited on pages 52, 115, 118–122).

[BC92] Michael Ben-Or and Richard Cleve. "Computing Algebraic Formulas Using a Constant Number of Registers." In: *SIAM Journal on Computing* 21.1 (1992), pp. 54–58. DOI: 10.1137/0221006 (cited on page 140).

[BT88] Michael Ben-Or and Prasoon Tiwari. "A deterministic algorithm for sparse multivariate polynomial interpolation." In: *Proceedings of the* 20^{th} *Annual ACM Symposium on Theory of computing (STOC'88)*. 1988, pp. 301–309 (cited on pages 48, 166).

[BG21] Vishwas Bhargava and Sumanta Ghosh. "Improved Hitting Set for Orbit of ROABPs." In: 25^{th} *International Conference on Randomization and Computation (RANDOM'21)* (2021) (cited on page 169).

[BS21] Pranav Bisht and Nitin Saxena. "Blackbox identity testing for sum of special ROABPs and its border class." In: *Computational Complexity* 30.1 (2021), pp. 1–48 (cited on pages 122, 166, 168).

[BDI21] Markus Bläser, Julian Dörfler, and Christian Ikenmeyer. "On the complexity of evaluating highest weight vectors." In: *Proceedings of the* 36^{th} *Conference on Computational Complexity (CCC'21)* (2021) (cited on pages 47, 140, 150).

[Blä+21] Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. "On the Orbit Closure Containment Problem and Slice Rank of Tensors." In: *Proceedings of the* 32^{nd} *ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*. 2021, pp. 2565–2584 (cited on page 15).

[Blä+] Markus Bläser, Christian Ikenmeyer, Meena Mahajan, Anurag Pandey, and Nitin Saurabh. "Algebraic Branching Programs, Border Complexity, and Tangent Spaces." In: *Proceedings of the* 35^{th} *Computational Complexity Conference (CCC'20)*, 21:1–21:24 (cited on pages 21, 140).

[BP20] Markus Bläser and Anurag Pandey. "Polynomial Identity Testing for Low Degree Polynomials with Optimal Randomness." In: 24^{th} *International Conference on Randomization and Computation (RANDOM'20)*. Vol. 176. LIPIcs. 2020, 8:1–8:13 (cited on page 20).

[BT15] Grigoriy Blekherman and Zach Teitler. "On maximum, typical and generic ranks." In: *Mathematische Annalen* 362.3 (2015), pp. 1021–1031 (cited on page 143).

[Blu+00] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale. "Algebraic settings for the problem "P≠ NP?"" In: *The Collected Papers of Stephen Smale: Volume 3*. World Scientific, 2000, pp. 1540–1559 (cited on page 60).

[BSS89] Lenore Blum, Mike Shub, and Steve Smale. "On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines." In: *Bulletin (New Series) of the American Mathematical Society* 21.1 (1989), pp. 1–46 (cited on page 60).

[BC76] Allan Borodin and Stephen A. Cook. "On the Number of Additions to Compute Specific Polynomials." In: *SIAM Journal on Computing* 5.1 (1976), pp. 146–157 (cited on page 61).

[BIZ18] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. "On Algebraic Branching Programs of Small Width." In: Journal of the ACM 65.5 (2018). Preliminary version in the Proceedings of the 32^{nd} Conference on Computational Complexity (CCC'17), 32:1–32:29 (cited on pages 140, 141, 147).

[BB20] Irénée Briquel and Peter Bürgisser. "The real tau-conjecture is true on average." In: *Random Structures & Algorithms* (2020) (cited on page 64).

[Bür00] Peter Bürgisser. "Cook's versus Valiant's hypothesis." In: *Theoretical Computer Science* 235.1 (2000), pp. 71–88 (cited on pages 7, 36).

[Bür04] Peter Bürgisser. "The Complexity of Factors of Multivariate Polynomials." In: *Foundations of Computational Mathematics* 4.4 (2004), pp. 369–396 (cited on pages 9, 12, 140).

[Bür09] Peter Bürgisser. "On Defining Integers and Proving Arithmetic Circuit Lower Bounds." In: *Computational Complexity* 18.1 (2009). Preliminary version in the Proceedings of the 24^{th} Annual Symposium on Theoretical Aspects of Computer Science (STACS'07), pp. 81–103 (cited on pages 35, 36, 60).

[Bür13] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Vol. 7. Springer Science & Business Media, 2013 (cited on pages 5, 8, 34, 36).

[Bür20] Peter Bürgisser. "Correction To: The Complexity of Factors of Multivariate Polynomials." In: *Foundations of Computational Mathematics* 20.6 (2020), pp. 1667–1668. DOI: 10.1007/s10208-020-09477-6 (cited on pages 12, 140, 149).

[BCS13] Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic Complexity Theory*. Vol. 315. Springer Science & Business Media, 2013 (cited on pages 33, 38, 149).

[Bür+19] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. "Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes." In: *IEEE 60^{th} Annual Symposium on Foundations of Computer Science (FOCS'19)*. 2019, pp. 845–861 (cited on page 15).

[Bür+18] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. "Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory." In: *Proceedings of the 9^{th} Innovations in Theoretical Computer Science (ITCS'18)* (2018) (cited on page 15).

[BI13] Peter Bürgisser and Christian Ikenmeyer. "Explicit lower bounds via geometric complexity theory." In: *Proceedings of the 45^{th} Annual ACM Symposium on Theory of Computing*. 2013, pp. 141–150 (cited on page 15).

[CW01] Anthony Carbery and James Wright. "Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$." In: *Mathematical Research Letters* 8.3 (2001), pp. 233–248 (cited on page 167).

[CCG12] Enrico Carlini, Maria Virginia Catalisano, and Anthony V. Geramita. "The solution to the Waring problem for monomials and the sum of coprime monomials." In: *Journal of Algebra* 370 (2012), pp. 5–14 (cited on page 32).

[Cha+20] Prerona Chatterjee, Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. "On the Existence of Algebraically Natural Proofs." In: *Proceedings of the 61^{st} Annual IEEE Symposium on the Foundations of Computer Science (FOCS'20)* (2020) (cited on page 15).

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. *Partial derivatives in arithmetic complexity and beyond*. Now Publishers Inc, 2011 (cited on page 33).

[CKS18] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. "Hardness vs randomness for bounded depth arithmetic circuits." In: *Proceedings of the 33^{rd} Computational Complexity Conference (CCC'18)*. 2018 (cited on pages 120, 170).

[DL78] Richard A. Demillo and Richard J. Lipton. "A probabilistic remark on algebraic program testing." In: *Information Processing Letters* 7.4 (1978), pp. 193–195 (cited on pages 7, 19, 46).

[Dut21] Pranjal Dutta. "Real τ-Conjecture for Sum-of-Squares: A Unified Approach to Lower Bound and Derandomization." In: *Proceedings of the* 16^{th} *International Computer Science Symposium in Russia (CSR'21)*. Extended version is under review. Springer. 2021, pp. 78–101 (cited on pages xxi, 18, 59, 179).

[DDS21a] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. "Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits." In: *Proceedings of the* 36^{th} *Computational Complexity Conference (CCC'21)*. 2021, 11:1–11:27 (cited on pages xxi, 21, 115, 120, 129, 170).

[DDS21b] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. "Demystifying the border of depth-3 algebraic circuits." In: 62^{nd} *IEEE Annual Symposium on Foundations of Computer Science (FOCS'21)* (2021) (cited on pages xxi, 15, 22, 120, 139, 147, 165, 167, 168, 187, 188).

[Dut+21] Pranjal Dutta, Gorav Jindal, Anurag Pandey, and Amit Sinhababu. "Arithmetic circuit complexity of division and truncation." In: *Proceedings of the* 36^{th} *Computational Complexity Conference (CCC'21)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2021 (cited on page xxii).

[DR22] Pranjal Dutta and Mahesh Sreekumar Rajasree. "Algebraic Algorithms for Variants of Subset Sum." In: *Conference on Algorithms and Discrete Applied Mathematics (CALDAM'22)*. Springer. 2022, pp. 237–251 (cited on page xxii).

[DRS22] Pranjal Dutta, Mahesh Sreekumar Rajasree, and Santanu Sarkar. "On the hardness of monomial prediction and zero-sum distinguishers for Ascon." In: 12^{th} *International Workshop on Coding and Cryptography (WCC'22)* (2022) (cited on page xxii).

[DS20] Pranjal Dutta and Nitin Saxena. "Lower-bounding the sum of 4th-powers of univariates leads to derandomization and hardness." In: *Unpublished* (2020) (cited on page 92).

[DS22] Pranjal Dutta and Nitin Saxena. "Separated borders: Exponential-gap fanin-hierarchy theorem for approximative depth-3 circuits." In: 63^{rd} *IEEE Annual Symposium on Foundations of Computer Science (FOCS'22)* (2022) (cited on pages xxii, 188).

[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). Prelimary version in the Proceedings of the 50^{th} Annual ACM Symposium on Theory of Computing (STOC'18), 18:1–18:39 (cited on page xxii).

[DST20] Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. "Lower bounds on the sum of 25th-powers of univariates lead to complete derandomization of PIT." In: *ECCC*. Vol. 27. 2020, p. 39 (cited on page 92).

[DST21] Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. "A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization." In: 12^{th} *Innovations in Theoretical Computer Science Conference (ITCS'21)*. 2021. DOI: 10.4230/LIPIcs.ITCS.2021.23 (cited on pages xxi, 18, 59, 71, 88, 168, 183, 184).

[DGW19] Zeev Dvir, Alexander Golovnev, and Omri Weinstein. "Static data structure lower bounds imply rigidity." In: *Proceedings of the* 51^{st} *Annual ACM Symposium on Theory of Computing*. 2019, pp. 967–978 (cited on page 39).

[DS07] Zeev Dvir and Amir Shpilka. "Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits." In: *SIAM Journal on Computing* 36.5 (2007), pp. 1404–1434 (cited on pages 115, 119, 169).

[DSY10] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. "Hardness-randomness tradeoffs for bounded depth arithmetic circuits." In: *SIAM Journal on Computing* 39.4 (2010). Preliminary version in Proceedings of the 40^{th} Annual ACM Symposium on Theory of Computing (STOC'08), pp. 1279–1293 (cited on page 122).

[Edi18] Editors. "Open Problems." In: *Algebraic Methods, Simons Institute for the Theory of Computing* (2018). Emailed by Igor Carboni Oliveira (cited on page 147).

[Eis+08] Friedrich Eisenbrand, János Pach, Thomas Rothvoß, and Nir B Sopher. "Convexly independent subsets of the Minkowski sum of planar point sets." In: *The Electronic Journal of Combinatorics* 15.1 (2008), p. 8 (cited on page 113).

[For16] Michael Forbes. *Some concrete questions on the border complexity of polynomials. Presentation given at the Workshop on Algebraic Complexity Theory WACT 2016 in Tel Aviv*. 2016. URL: https://www.youtube.com/watch?v=1HMogQIHT6Q (cited on pages 142, 149).

[For15] Michael A Forbes. "Deterministic divisibility testing via shifted partial derivatives." In: *Proceedings of the* 56^{th} *Annual Symposium on Foundations of Computer Science (FOCS'15)*. IEEE. 2015, pp. 451–465 (cited on pages 21, 50, 115, 119, 121, 129).

[FSS14] Michael A Forbes, Ramprasad Saptharishi, and Amir Shpilka. "Hitting sets for multilinear read-once algebraic branching programs, in any order." In: *Proceedings of the* 46^{th} *Annual ACM Symposium on Theory of computing (STOC'14)*. 2014, pp. 867–875 (cited on pages 50, 122).

[FS13a] Michael A Forbes and Amir Shpilka. "Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs." In: 54^{th} *Annual Symposium on Foundations of Computer Science (FOCS'13)*. 2013, pp. 243–252 (cited on pages 43, 50, 119, 122, 166).

[FS13b] Michael A Forbes and Amir Shpilka. "Explicit noether normalization for simultaneous conjugation via polynomial identity testing." In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2013, pp. 527–542 (cited on pages 15, 166, 176).

[For14] Michael A. Forbes. "Polynomial identity testing of read-once oblivious algebraic branching programs." PhD thesis. Massachusetts Institute of Technology, Cambridge, MA, USA, 2014 (cited on pages 39, 46, 150, 176).

[FGS18] Michael A. Forbes, Sumanta Ghosh, and Nitin Saxena. "Towards Blackbox Identity Testing of Log-Variate Circuits." In: 45^{th} *International Colloquium on Automata, Languages, and Programming (ICALP'18)*. 2018 (cited on pages 39, 122, 168, 169, 176, 177).

[FS18] Michael A. Forbes and Amir Shpilka. "A PSPACE construction of a hitting set for the closure of small algebraic circuits." In: *Proceedings of the* 50^{th} *Annual ACM Symposium on Theory of Computing (STOC'18)* (2018), pp. 1180–1192 (cited on pages 165–167, 182).

[Fri93] Joel Friedman. "A note on matrix rigidity." In: *Combinatorica* 13.2 (1993), pp. 235–239 (cited on page 38).

[GK17] Ignacio Garcia-Marco and Pascal Koiran. "Lower bounds by Birkhoff interpolation." In: *Journal of Complexity* 39 (2017), pp. 38–50 (cited on page 69).

[GKT15] Ignacio Garcia-Marco, Pascal Koiran, and Sébastien Tavenas. "Log-concavity and lower bounds for arithmetic circuits." In: 40^{th} *International Symposium on Mathematical Foundations of Computer Science (MFCS'15)*. Springer. 2015, pp. 361–371 (cited on pages 60, 70, 71, 88).

[GS20] Abhibhav Garg and Nitin Saxena. "Special-case algorithms for blackbox radical membership, Nullstellensatz and transcendence degree." In: *Proceedings of the* 45^{th} *International Symposium on Symbolic and Algebraic Computation (ISSAC'20)*. 2020, pp. 186–193 (cited on page 121).

[Gar+16] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. "A deterministic polynomial time algorithm for non-commutative rational identity testing." In: *IEEE* 57^{th} *Annual Symposium on Foundations of Computer Science (FOCS'16)*. IEEE. 2016, pp. 109–117 (cited on pages 15, 122).

[Gho19] Sumanta Ghosh. "Low Variate Polynomials: Hitting Set and Bootstrapping." PhD thesis. Indian Institute of Technology Kanpur, 2019 (cited on page 39).

[Gri76] D Yu Grigoriev. *Using the notions of seperability and independence for proving the lower bounds on the circuit complexity. Notes of the Leningrad branch of the Steklov Mathematical Institute*. 1976 (cited on page 38).

[Gro13] Joshua Grochow. Stack exchange. 2013 (cited on page 5).

[Gro15] Joshua A Grochow. "Unifying known lower bounds via geometric complexity theory." In: *Computational Complexity* 24.2 (2015), pp. 393–475 (cited on page 15).

[Gro+17] Joshua A Grochow, Mrinal Kumar, Michael Saks, and Shubhangi Saraf. "Towards an algebraic natural proofs barrier via polynomial identity testing." In: *arXiv preprint arXiv:1701.01717* (2017) (cited on page 15).

[GMQ16] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. "Boundaries of VP and VNP." In: 43^{rd} *International Colloquium on Automata, Languages and Programming (ICALP'11)* (2016), 34:1–34:14 (cited on page 147).

[Gro12] Joshua Abraham Grochow. *Symmetry and equivalence relations in classical and geometric complexity theory*. PhD Thesis. The University of Chicago, 2012 (cited on pages 14, 15).

[Guo21] Zeyu Guo. "Variety Evasive Subspace Families." In: *Proceedings of the* 36^{th} *Computational Complexity Conference (CCC'21)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021 (cited on pages 117, 121, 170).

[Guo+19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. "Derandomization from Algebraic Hardness: Treading the Borders." In: *Proceedings of the* 60^{th} *IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)*. 2019, pp. 147–157 (cited on pages 47, 70, 90, 166, 168, 182).

[Gup14] Ankit Gupta. "Algebraic Geometric Techniques for Depth-4 PIT & Sylvester-Gallai Conjectures for Varieties." In: *Electronic Colloquium on Computational Complexity (ECCC)*. Vol. 21. 2014, p. 130 (cited on pages 117–119, 121, 122, 135).

[Gup+16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. "Arithmetic circuits: A chasm at depth three." In: vol. 45. 3. Prelimiary version in the Proceeding of the IEEE 54^{th} Annual Symposium on Foundations of Computer Science (FOCS'13). 2016, pp. 1064–1079 (cited on pages 69, 70, 114, 145, 146).

[GKS17] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. "Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs." In: *Theory of Computing* 13.2 (2017). Preliminary version in the Proceedings of the 31^{st} Computational Complexity Conference (CCC'16), pp. 1–21 (cited on pages 50, 122, 142, 149).

[Gur+17] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. "Deterministic identity testing for sum of read-once oblivious arithmetic branching programs." In: *Computational Complexity* 26.4 (2017). Preliminary version in the Proceedings of the 30^{th} IEE Computational Complexity Conference (CCC'15), pp. 835–880 (cited on pages 21, 122, 166).

[HS80a] Joos Heintz and Claus-Peter Schnorr. "Testing polynomials which are easy to compute." In: *Proceedings of the* 12^{th} *annual ACM Symposium on Theory of Computing (STOC'80)*. 1980, pp. 262–272 (cited on pages 20, 182).

[HS80b] Joos Heintz and Malte Sieveking. "Lower bounds for polynomials with algebraic coefficients." In: *Theoretical Computer Science* 11.3 (1980), pp. 321–330 (cited on page 34).

[Hru13] Pavel Hrubes. "On the Real τ-Conjecture and the Distribution of Complex Roots." In: *Theory of Computing* 9.1 (2013), pp. 403–411 (cited on page 64).

[Hru20] Pavel Hrubes. "On the distribution of runners on a circle." In: *European Journal of Combinatorics* 89 (2020), p. 103137 (cited on page 64).

[HWY11] Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. "Non-commutative circuits and the sum-of-squares problem." In: *Journal of the American Mathematical Society* 24.3 (2011). Preliminary version in the Proceedings of the 42^{th} Annual ACM Symposium on Theory of Computing (STOC'10), pp. 871–898 (cited on page 68).

[HL16] Jesko Hüttenhain and Pierre Lairez. "The boundary of the orbit of the 3-by-3 determinant polynomial." In: *Comptes Rendus Mathematique* 354.9 (2016), pp. 931–935 (cited on page 140).

[IQS18] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. "Non-commutative Edmonds' problem and matrix semi-invariants." In: *Computational Complexity* 26.3 (2018), pp. 717–763 (cited on page 15).

[JQS10] Maurice Jansen, Youming Qiao, and Jayalal Sarma. "Deterministic Black-Box Identity Testing π-Ordered Algebraic Branching Programs." In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010*. Vol. 8. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010, pp. 296–307 (cited on page 122).

[Joh90] David S Johnson. "A catalog of complexity classes." In: *Algorithms and complexity*. Elsevier, 1990, pp. 67–161 (cited on page 34).

[KI04] Valentine Kabanets and Russell Impagliazzo. "Derandomizing polynomial identity tests means proving circuit lower bounds." In: *Computational Complexity* 13.1-2 (2004). Prelimiary version in the Proceedings of the 35^{th} Annual ACM Symposium on Theory of Computing (STOC'03), pp. 1–46 (cited on pages ix, 8, 64, 70).

[Kar84] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming." In: *Proceedings of the sixteenth annual ACM Symposium on Theory of computing*. 1984, pp. 302–311 (cited on page 16).

[Kar+13] Zohar S Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. "Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in." In: *SIAM Journal on Computing* 42.6 (2013). Preliminary version in the Proceedings of the 42^{nd} ACM symposium on Theory of Computing (STOC'10), pp. 2114–2131 (cited on page 119).

[KS11] Zohar S Karnin and Amir Shpilka. "Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in." In: *Combinatorica* 31.3 (2011). Preliminary version in the Proceedings of the 23^{rd} Annual IEEE Conference on Computational Complexity (CCC'08), p. 333 (cited on page 119).

[Kay+15] Neeraj Kayal, Pascal Koiran, Timothée Pecatte, and Chandan Saha. "Lower bounds for sums of powers of low degree univariates." In: 42^{th} *International Colloquium on Automata, Languages, and Programming (ICALP'15)*. Springer. 2015, pp. 810–821 (cited on pages 69, 122).

[KNS20] Neeraj Kayal, Vineet Nair, and Chandan Saha. "Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth-three circuits." In: *ACM Transactions on Computation Theory* 12.1 (2020). Preliminary version in the Proceedings of the 33^{rd} Symposium on Theoretical Aspects of Computer Science (STACS'16), pp. 1–27 (cited on pages 42, 43).

[KS09] Neeraj Kayal and Shubhangi Saraf. "Blackbox polynomial identity testing for depth 3 circuits." In: *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE. 2009, pp. 198–207 (cited on pages 115, 169).

[KS07] Neeraj Kayal and Nitin Saxena. "Polynomial identity testing for depth 3 circuits." In: *Computational Complexity* 16.2 (2007). Preliminary version in the Proceedings of the 21^{st} Computational Complexity Conference (CCC'06), pp. 115–138 (cited on pages 115, 119, 121).

[Kha79] Leonid Genrikhovich Khachiyan. "A polynomial algorithm in linear programming." In: *Doklady Akademii Nauk*. Vol. 244. 5. Russian Academy of Sciences. 1979, pp. 1093–1096 (cited on page 16).

[KS01] Adam R Klivans and Daniel Spielman. "Randomness efficient identity testing of multivariate polynomials." In: *Proceedings of the 33^{rd} Annual ACM Symposium on Theory of computing (STOC'01)*. 2001, pp. 216–223 (cited on pages 48, 166).

[Koi11] Pascal Koiran. "Shallow circuits with high-powered inputs." In: *Innovations in Computer Science (ICS'11)*. 2011 (cited on pages 60, 69–71).

[Koi12] Pascal Koiran. "Arithmetic circuits: The chasm at depth four gets wider." In: *Theoretical Computer Science* 448 (2012), pp. 56–65 (cited on pages x, 60, 69, 70, 116).

[KPG18] Pascal Koiran, Timothée Pecatte, and Ignacio Garcia-Marco. "On the linear independence of shifted powers." In: *Journal of Complexity* 45 (2018), pp. 67–82 (cited on page 69).

[KP11] Pascal Koiran and Sylvain Perifel. "Interpolation in Valiant's theory." In: *Computational Complexity* 20.1 (2011), pp. 1–20 (cited on pages 36, 37).

[KPT15] Pascal Koiran, Natacha Portier, and Sébastien Tavenas. "A Wronskian approach to the real τ-conjecture." In: *Journal of Symbolic Computation* 68 (2015), pp. 195–214 (cited on page 122).

[Koi+15] Pascal Koiran, Natacha Portier, Sébastien Tavenas, and Stéphan Thomassé. "A τ-Conjecture for Newton Polygons." In: *Foundations of computational mathematics* 15.1 (2015), pp. 185–197 (cited on pages 70, 112, 113).

[Kro82] Leopold Kronecker. "Grundzüge einer arithmetischen Theorie der algebraischen Grössen.(Abdruck einer Festschrift zu Herrn EE Kummers Doctor-Jubiläum, 10. September 1881.)." In: *Journal für die reine und angewandte Mathematik* 92 (1882), pp. 1–122 (cited on page 29).

[Kum20] Mrinal Kumar. "On the Power of Border of Depth-3 Arithmetic Circuits." In: *ACM Transactions on Computation Theory* 12.1 (2020), 5:1–5:8 (cited on pages x, 53, 143, 146, 150).

[Kum+22] Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. "If VNP Is Hard, Then so Are Equations for It." In: *Proceedings of the 39th International Symposium on Theoretical Aspects of Computer Science (STACS'22)*. Vol. 219. LIPIcs. 2022, 44:1–44:13 (cited on page 15).

[KSS19] Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. *Derandomization from Algebraic Hardness: Treading the Borders*. ArXiv:1905.00091v1. 2019 (cited on page 182).

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. "Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits." In: *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*. SIAM. 2019, pp. 639–646 (cited on pages 20, 168).

[KS16] Mrinal Kumar and Shubhangi Saraf. "Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing." In: *Proceedings of the 31st Conference on Computational Complexity, CCC 2016*. Vol. 50. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, 35:1–35:29 (cited on pages 117, 119, 122).

[KS17] Mrinal Kumar and Shubhangi Saraf. "Arithmetic Circuits with Locally Low Algebraic Rank." In: *Theory of Computing* 13.1 (2017). Preliminary version in the Proceedings of the 31st Conference on Computational Complexity (CCC'16), pp. 1–33 (cited on pages 119, 120, 122).

[KV21] Mrinal Kumar and Ben Lee Volk. "Lower Bounds for Matrix Factorization." In: *Computational Complexity* 30.1 (2021). Prelimar version in the Proceedings of the 35th Computational Complexity Conference (CCC'20), p. 6 (cited on pages 81, 102, 103, 105, 107).

[Kur92] David C Kurtz. "A sufficient condition for all the roots of a polynomial to be real." In: *The american mathematical monthly* 99.3 (1992), pp. 259–263 (cited on pages 32, 33).

[LMP19] Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. "Non-commutative computations: lower bounds and polynomial identity testing." In: *Chicago Journal of Theoretical Computer Science* 2 (2019), pp. 1–19 (cited on page 122).

[Lan17] J. M. Landsberg. *Geometry and Complexity Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017 (cited on page 146).

[Lan15] Joseph M Landsberg. "Geometric complexity theory: an introduction for geometers." In: *Annali dell'universita'di Ferrara* 61.1 (2015), pp. 65–117 (cited on page 146).

[LO15] Joseph M. Landsberg and Giorgio Ottaviani. "New Lower Bounds for the Border Rank of Matrix Multiplication." In: *Theory of Computing* 11 (2015), pp. 285–298 (cited on page 15).

[Lan13] Serge Lang. *Algebraic number theory*. Vol. 110. Springer Science & Business Media, 2013 (cited on page 94).

[Las07] Jean B Lasserre. "A sum of squares approximation of nonnegative polynomials." In: *SIAM review* 49.4 (2007), pp. 651–669 (cited on page 60).

[Lau09] Monique Laurent. "Sums of squares, moment matrices and optimization over polynomials." In: *Emerging applications of algebraic geometry*. Springer, 2009, pp. 157–270 (cited on page 60).

[LL89] Thomas Lehmkuhl and Thomas Lickteig. "On the order of approximation in approximative triadic decompositions of tensors." In: *Theoretical Computer Science* 66.1 (1989), pp. 1–14 (cited on page 149).

[LST21] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. "Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits." In: *Proceedings of the 62nd Annual IEEE Symposium on the Foundations of Computer Science (FOCS'21)* (2021), p. 81 (cited on pages x, 20, 115, 120, 165, 170, 188).

[Lip94] Richard J Lipton. "Straight-line complexity and integer factorization." In: *International Algorithmic Number Theory Symposium*. Springer. 1994, pp. 71–79 (cited on page 35).

[Luc78] Edouard Lucas. "Théorie des fonctions numériques simplement périodiques." In: *American Journal of Mathematics* (1878), pp. 289–321 (cited on page 94).

[Mah14] Meena Mahajan. "Algebraic complexity classes." In: *Perspectives in Computational Complexity*. Springer, 2014, pp. 51–75 (cited on page 5).

[MH02] John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC press, 2002 (cited on page 35).

[MS21] Dori Medini and Amir Shpilka. "Hitting Sets and Reconstruction for Dense Orbits in VP_e and $\Sigma\Pi\Sigma$ Circuits." In: *Proceedings of the 36th Computational Complexity Confernece (CCC'21)* (2021) (cited on pages 147, 167, 169).

[Mit13] Johannes Mittmann. "Independence in algebraic complexity theory." PhD thesis. Universitäts-und Landesbibliothek Bonn, 2013 (cited on page 20).

[Muk16] Partha Mukhopadhyay. "Depth-4 identity testing and Noether's normalization lemma." In: *International Computer Science Symposium in Russia (CSR'16)*. Springer. 2016, pp. 309–323 (cited on pages 15, 119, 121, 166).

[Mul10] Ketan Mulmuley. "Geometric Complexity Theory VI: The flip via positivity." In: *arXiv preprint arXiv:0704.0229* (2010) (cited on page 140).

[Mul17] Ketan Mulmuley. "Geometric Complexity Theory V: Efficient algorithms for Noether normalization." In: *Journal of the American Mathematical Society* 30.1 (2017). Preliminary version in the IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS'12), pp. 225–309 (cited on pages x, 15, 145, 166).

[MS01] Ketan Mulmuley and Milind A. 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 (cited on pages x, 12, 13).

[Mul12] Ketan D Mulmuley. "The GCT program toward the P vs. NP problem." In: *Communications of the ACM* 55.6 (2012), pp. 98–107 (cited on pages x, 15, 94, 139, 140, 166).

[Mum95] David Mumford. *Algebraic geometry I: complex projective varieties*. Springer Science & Business Media, 1995 (cited on pages 12, 13).

[Nis91] Noam Nisan. "Lower bounds for non-commutative computation." In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC'91)*. 1991, pp. 410–418 (cited on pages 44, 140, 149).

[Ore22] ystein Ore. "Über höhere kongruenzen." In: *Norsk Mat. Forenings Skrifter* 1.7 (1922), p. 15 (cited on pages 7, 19, 46).

[Ost75] Alexander M Ostrowski. "On multiplication and factorization of polynomials, I. Lexicographic orderings and extreme aggregates of terms." In: *aequationes mathematicae* 13.3 (1975), pp. 201–228 (cited on page 112).

[PSS18] Anurag Pandey, Nitin Saxena, and Amit Sinhababu. "Algebraic independence over positive characteristic: New criterion and applications to locally low-algebraic-rank circuits." In: *Computational Complexity* 27.4 (2018). Preliminary version in the 41st International Symposium on Mathematical Foundations of Computer Science (MFCS'16), pp. 617–670 (cited on page 122).

[PS20] Shir Peleg and Amir Shpilka. "A generalized Sylvester-Gallai type theorem for quadratic polynomials." In: *Proceedings of the 35th Computational Complexity Conference (CCC'20)*. 2020 (cited on pages 115, 118, 119).

[PS21] Shir Peleg and Amir Shpilka. "Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials." In: 53rd *Annual ACM Symposium on Theory of computing (STOC'21)*. 2021 (cited on pages 115, 118, 120, 122).

[Pfi76] Albrecht Pfister. "Hilbert's seventeenth problem and related problems on definite forms." In: *Mathematical Developments Arising from Hilbert Problems, Proceedings of Symposia in Pure Mathematics, XXVIII.2.AMS*. Vol. 28. 1976, pp. 483–489 (cited on page 60).

[Pip77] Nicholas Pippenger. "Superconcentrators." In: *SIAM Journal on Computing* 6.2 (1977), pp. 298–304 (cited on page 38).

[Pud94] Pavel Pudlak. "Communication in bounded depth circuits." In: *Combinatorica* 14.2 (1994), pp. 203–216 (cited on pages 38, 102).

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. "Bounds for dispersers, extractors, and depth-two superconcentrators." In: *SIAM Journal on Discrete Mathematics* 13.1 (2000), pp. 2–24 (cited on page 38).

[Ram17] Srinivasa Ramanujan. "On the Expression of a Number in the Form ax^2 + by^2+ cz^2+ du^2." In: *Mathematical Proceedings of the Cambridge Philosophical Society*. Vol. 19. 1917, pp. 11–21 (cited on page 60).

[Ram20] C Ramya. "Recent Progress on Matrix Rigidity–A Survey." In: *arXiv preprint arXiv:2009.09460* (2020) (cited on page 39).

[Raz10] Ran Raz. "Elusive Functions and Lower Bounds for Arithmetic Circuits." In: *Theory of Computing* 6.1 (2010), pp. 135–177 (cited on pages 69, 71).

[RS05] Ran Raz and Amir Shpilka. "Deterministic polynomial identity testing in non-commutative models." In: *Computational Complexity* 14.1 (2005). Preliminary version in the Proceedings of the 19^{th} IEEE Annual Conference on Computational Complexity (CCC'04), pp. 1–19 (cited on pages 50, 122).

[Rez78] Bruce Reznick. "Extremal PSD forms with few terms." In: *Duke mathematical journal* 45.2 (1978), pp. 363–374 (cited on page 60).

[Ris85] Jean-Jacques Risler. "Additive complexity and zeros of real polynomials." In: *SIAM Journal on Computing* 14.1 (1985), pp. 178–183 (cited on page 61).

[SSS13] Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. "A case of depth-3 identity testing, sparse factorization and duality." In: *Computational Complexity* 22.1 (2013), pp. 39–69 (cited on pages 115, 116, 121, 150).

[ST21] Chandan Saha and Bhargav Thankey. "Hitting Sets for Orbits of Circuit Classes and Polynomial Families." In: 25^{th} *International Conference on Randomization and Computation (RANDOM'21)* (2021) (cited on page 169).

[Sap21] Ramprasad Saptharishi. *A survey of lower bounds in arithmetic circuit complexity*. Github survey. Version 8. 2021 (cited on pages 5, 49, 71, 77, 121, 180).

[SV18] Shubhangi Saraf and Ilya Volkovich. "Black-box identity testing of depth-4 multilinear circuits." In: *Combinatorica* 38.5 (2018). Preliminary version in the Proceedings of the 43^{rd} Annual ACM symposium on Theory of Computing (STOC'11), pp. 1205–1238 (cited on pages 119, 120).

[Sau12] Nitin Saurabh. *Algebraic models of computation*. MS Thesis, IMSc. 2012 (cited on page 72).

[Sax08] Nitin Saxena. "Diagonal circuit identity testing and lower bounds." In: *International Colloquium on Automata, Languages, and Programming (ICALP'08)*. Springer. 2008, pp. 60–71 (cited on pages 44, 50, 121, 149).

[Sax09] Nitin Saxena. "Progress on Polynomial Identity Testing." In: *Bulletin of the EATCS* 99 (2009), pp. 49–79 (cited on page 48).

[SS11] Nitin Saxena and Comandur Seshadhri. "An almost optimal rank bound for depth-3 identities." In: *SIAM Journal on computing* 40.1 (2011). Preliminary version in the Proceedings of the 24^{th} IEEE Conference on Computational Complexity (CCC'09), pp. 200–224 (cited on pages 115, 119, 167, 169).

[SS12] Nitin Saxena and Comandur Seshadhri. "Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter." In: *SIAM Journal on Computing* 41.5 (2012). Preliminary version in the 43^{rd} Annual ACM symposium on Theory of computing (STOC'11), pp. 1285–1298 (cited on pages 115, 119–121, 167, 169).

[SS13] Nitin Saxena and Comandur Seshadhri. "From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits." In: *Journal of the ACM* 60.5 (2013). Preliminary version in the 51^{st} Annual IEEE Symposium on Foundations of Computer Science (FOCS'10), pp. 1–33 (cited on pages 119, 169).

[Sch80] J. T. Schwartz. "Fast Probabilistic Algorithms for Verification of Polynomial Identities." In: *Journal of the ACM* 27.4 (Oct. 1980), pp. 701–717 (cited on pages 7, 19, 46).

[Sha79] Adi Shamir. "Factoring Numbers in O(log n) Arithmetic Steps." In: *Information Processing Letters* 8.1 (1979), pp. 28–31 (cited on page 35).

[Sha92] Adi Shamir. "IP = PSPACE." In: *Journal of the ACM* 39.4 (1992), pp. 869–877 (cited on page 8).

[SSS97] Mohammad Amin Shokrollahi, Daniel A Spielman, and Volker Stemann. "A remark on matrix rigidity." In: *Information Processing Letters* 64.6 (1997), pp. 283–285 (cited on page 38).

[Shp20] Amir Shpilka. "Sylvester-Gallai type theorems for quadratic polynomials." In: *Discrete Analysis* (2020). Prelimiary version in the Proceedings of the 51^{st} Annual ACM Symposium on Theory of Computing (STOC'19), p. 34 (cited on page 119).

[SY10] 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 (2010), pp. 207–388 (cited on pages 38, 71).

[SS95] Michael Shub and Steve Smale. "On the intractability of Hilbert's Nullstellensatz and an algebraic version of " NP≠ P?"" In: *Duke Mathematical* Journal 81.1 (1995), pp. 47–54 (cited on pages 59, 60).

[Sma98] Steve Smale. "Mathematical problems for the next century." In: *The mathematical intelligencer* 20.2 (1998), pp. 7–15 (cited on page 60).

[Str74] Volker Strassen. "Polynomials with rational coefficients which are hard to compute." In: *SIAM Journal on Computing* 3.2 (1974), pp. 128–149 (cited on page 33).

[Tav14] Sébastien Tavenas. "Bornes inferieures et superieures dans les circuits arithmetiques." PhD thesis. ENS Lyon, 2014 (cited on page 60).

[Tav15] Sébastien Tavenas. "Improved bounds for reduction to depth 4 and depth 3." In: *Information and Computation* 240 (2015), pp. 2–11 (cited on page 116).

[Tur48] Alan M Turing. "Rounding-off errors in matrix processes." In: *The Quarterly Journal of Mechanics and Applied Mathematics* 1.1 (1948), pp. 287–308 (cited on page 2).

[Val75] Leslie G Valiant. "On non-linear lower bounds in computational complexity." In: *Proceedings of the 7^{th} Annual ACM Symposium on Theory of Computing (STOC'75)*. 1975, pp. 45–53 (cited on page 38).

[Val77] Leslie G Valiant. "Graph-theoretic arguments in low-level complexity." In: *2^{nd} International Symposium on Mathematical Foundations of Computer Science (MFCS'77)*. 1977, pp. 162–176 (cited on page 38).

[Val79] Leslie G Valiant. "Completeness classes in algebra." In: *Proceedings of the 11^{th} Annual ACM Symposium on Theory of Computing (STOC'79)*. 1979, pp. 249–261 (cited on pages ix, 6, 36).

[Val+83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. "Fast Parallel Computation of Polynomials Using Few Processors." In: *SIAM Journal on Computing* 12.4 (1983), pp. 641–644 (cited on pages 5, 71, 72, 77).

[Vas04] Wolmer Vasconcelos. *Computational methods in commutative algebra and algebraic geometry*. Vol. 2. Springer Science & Business Media, 2004 (cited on page 121).

[Wag86] Klaus W Wagner. "The complexity of combinatorial problems with succinct input representation." In: *Acta informatica* 23.3 (1986), pp. 325–356 (cited on page 34).

[Zip79] Richard Zippel. "Probabilistic Algorithms for Sparse Polynomials." In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM'79)*. 1979, pp. 216–226 (cited on pages 7, 19, 46).