A survey of lower bounds in arithmetic circuit complexity

https://github.com/dasarpmar/lowerbounds-survey/
Version 9.0.3
SHA: 7dc9b26bf88d2a63a7d5e9bf7243eb00284a76a8

Ramprasad Saptharishi
Tata Institute of Fundamental Research
ramprasad@tifr.res.in

Preface

Arithmetic circuit complexity has seen a flurry of activity recently with respect to lower bounds. There suddenly seems to be some optimism proving explicit circuit lower bounds in the near future. Besides the question of lower bounds, there has also been tremendous progress on polynomial identity testing and polynomial reconstruction as well.

In 2014, I was a part of two surveys on arithmetic circuit lower bounds. The first one [KS14a] was with Neeraj Kayal, and was a part of a volume dedicated to Somenath Biswas' 60th Birthday Celebrations. This survey was a comprehensive article of almost all known lower bound proofs at that time. Soon after the survey was written, there were more lower bounds proved for homogeneous depth four circuits. The second survey [Sap14] appeared in the Bulletin of the EATCS and this focused on those lower bounds for homogeneous depth four circuits (among some other results).

Instead of writing a new survey every time there are a fresh set of lower bounds, a better idea was to have one expanding survey that is kept up to date with the current state of the art. Much like an application, that keeps getting updated and new releases. Also, this would be greatly accelerated if the community could contribute by looking for bugs, adding more content, changing presentation etc. The natural answer to all this was to do this the way software applications are built, and I chose github as that is the most popular platform for this.

This survey would be present on https://github.com/dasarpmar/lowerbounds-survey and anyone is welcome to contribute to it. The github repository also has a wiki to assist people who are new to git and/or github.

What to expect from this article

Most of the proofs in this article are complete and self-contained. However, as one would expect in the more delicate proofs, there would eventually involve a fair amount of calculation and setting of parameters. There might be proofs where this last technical calculation is avoided, but the hope is to make the presentation insightful enough so that it would enable any student to do the calculations (him/her)self.

Also, quite a lot of the proofs presented here are slightly different from the original proofs. The reason for the deviation would almost always be for more clarify and intuition. However, this process might also make the parameters involved a little weaker than in the original statements. We shall try to ensure that such losses do not change the overall strength of the theorem by much, and if they do we shall mention that explicitly.

Why do we need this?

> So why are super polynomial lower bounds still not proved? Maybe it's because not enough people are working on it. – Ran Raz (in [Raz10b])

I strongly believe that the above statement really hits the nail on the head. Fortunately, over the last few years we have seen such a phenomenal activity in arithmetic circuit lower bounds and an increased optimism that we can indeed soon prove super-polynomial circuit lower bounds. In fact, a lot of the recent lower bounds have come really close to this goal. The hope of this survey is that this would assist people familiarize with the known lower bounds and develop the necessary tools. As a student, the surveys of [SY10, CKW11] were immensely helpful and this is an attempt to give back to the community.

Ramprasad Saptharishi
Version 9.0.3

Contributors to this article

Content

- Ramprasad Saptharishi (Tel Aviv University)

  `ramprasad@cmi.ac.in`

- Suryajith Chillara (Chennai Mathematical Institute)

  `suryajith@cmi.ac.in`

  Parts of Chapter 7

- Mrinal Kumar (Rutgers University)

  `mrinal.kumar@rutgers.edu`

  Chapter 26

- Anamay Tengse (Tata Institute of Fundamental Research)

  `tengse.anamay@tifr.res.in`

  Chapter 15

Corrections, minor edits

- V Vinay (Limberlink Technologies)

  `vinay@jed-i.in`

- Noam Mazor (Tel Aviv University)

- Amir Shpilka (Tel Aviv University)

  `shpilka@post.tau.ac.il`

- Anamay Tengse (Tata Institute of Fundamental Research)

  `tengse.anamay@tifr.res.in`

- Suhail Sherif (Tata Institute of Fundamental Research)

  `suhail.sherif@gmail.com`

- `Christian Engels`

  `https://github.com/Narfinger`

- Prateek Dwivedi

  `https://github.com/prateekdwv`

- Prerona Chatterjee

  `http://www.tifr.res.in/~prerona.chatterjee/`

- `anag004@github`

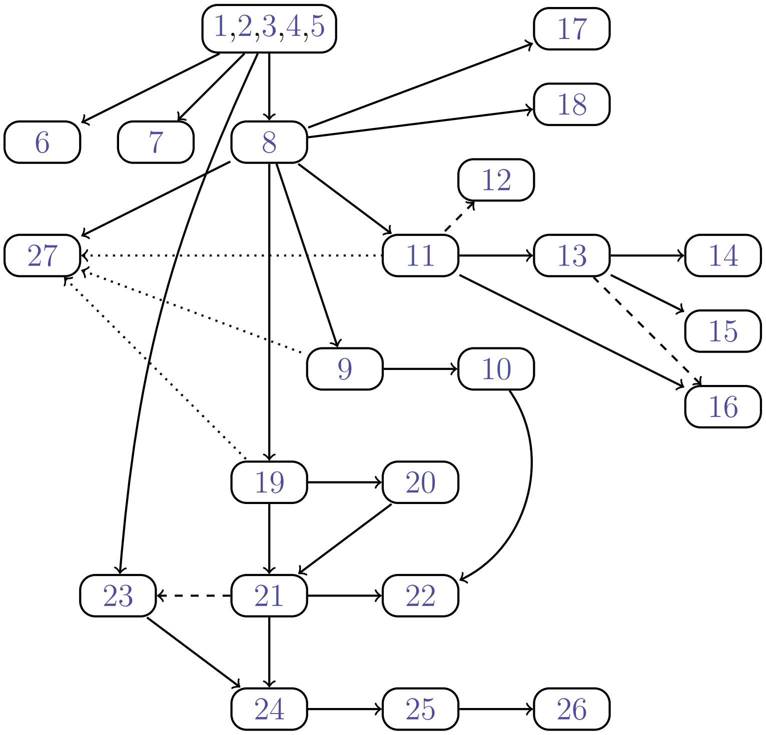
  `https://github.com/anag004`

- Srikanth Srinivasan

  `AarhusUniversity`

Contents

Chapter Dependencies



⟶ : Dependency

⇢ : Not a dependency, but recommended

⋯→ : Familiarity would be helpful

Chapter 1

Introduction

“What is the best way to compute a given polynomial $f(x_1, \ldots, x_n)$ from basic operations such as + and ×?” This is the main motivating problem in the field of arithmetic circuit complexity. The notion of *complexity* of a polynomial is measured via the size of the smallest arithmetic circuit computing it. Arithmetic circuits provide a robust model of computation for polynomials. Formally, these are directed acyclic graphs with a unique sink vertex, where internal nodes are labelled by + and × and each source node labelled with either a variable or a field constant. Each + gate computes the sum of the polynomials computed at its children, and × gates the product. The unique sink vertex is called the root or the output gate, and the polynomial computed by that gate is the polynomial computed by the circuit.

There are several interesting questions that can be asked about arithmetic circuits, and polynomials that they compute. One category of problems are of the form, “Is there an explicit polynomial $f(x_1, \ldots, x_n)$ that require (perhaps restricted) arithmetic circuits of size $2^{\Omega(n)}$ to compute them?”, or questions about proving lower bounds. Another category of problems are of the form, “Is the given circuit computing the 0 polynomial?”, which is also called ‘Polynomial Identity Testing (PIT)’. Yet another class of questions are of the form “Given oracle access to a circuit, can you write down the polynomial computed by this circuit?”, which are also called ‘polynomial reconstruction’. Several of these problems have very strong connections between each other despite being of very different flavours. Formal connections between PIT and lower bounds have been shown by [KI04, Agr05]. Further, strong lower bounds for restricted models have often been succeeded by reconstruction algorithms (at least on average). In this article we shall mainly be looking at lower bounds. For more on reconstruction and PIT questions, the author is invited to read other excellent surveys such as [SY10, CKW11].

1.1 Arithmetic complexity classes

In the seminal paper of [Val79], Valiant defined two classes of polynomials which we now call VP and VNP.

Definition 1.1. *The class* VP *is defined as the set of all polynomial* $f(x_1, \ldots, x_n)$ *with* $\deg(f) = n^{O(1)}$ *that can be computed by an arithmetic circuit of size* $s = n^{O(1)}$.

The class VNP *is defined as the set of all polynomial* $f(x_1, \ldots, x_n)$ *such that there exists a* $g(x_1, \ldots, x_n, y_1, \ldots, y_m) \in \mathsf{VP}$ *with* $m = n^{O(1)}$ *such that*

$$f(x_1, \ldots, x_n) \quad = \quad \sum_{y_1=0}^{1} \cdots \sum_{y_m=0}^{1} g(x_1, \ldots, x_n, y_1, \ldots, y_m).$$

$\Diamond$

The class VP is synonymous to what we understand as *efficiently computable* polynomials. The class VNP, whose definition is similar to the boolean class NP, is in some sense a notion of what deem as *explicit*.

Fact 1.2. *Let* $f(x_1, \ldots, x_n)$ *be a polynomial such that* $\deg(f) = n^{O(1)}$ *and given any exponent vector* $e_1, \ldots, e_n$*, the coefficient of the monomial* $x_1^{e_1} \ldots x_n^{e_n}$ *in* f *can be computed in polynomial time. Then,* $f \in \mathsf{VNP}$.

For example, consider the permanent of a symbolic $n \times n$ matrix. In fact, [Val79] showed that the symbolic $n \times n$ permanent is in some sense complete for the class VNP. Further, he also showed that the determinant of a symbolic $n \times n$ matrix is (almost) complete for the class VP. Separating the determinant and the permanent is the Holy Grail in the field of arithmetic circuit complexity.

Remark. Note that the above fact merely gives a sufficient condition for a polynomial to be in VNP. There are examples of polynomials f where computing the coefficient of a given monomial is believed to be very hard but $f \in \mathsf{VNP}$.[1] In this article however, all the polynomials we shall be dealing with would have this property that the coefficient of a given monomial can be efficiently computed. For more about completeness classes in arithmetic complexity, [BCS97] is a wonderful text.

[1] For example, consider the n^2 variate multilinear polynomial f such that the coefficient $x_{11}^{e_{11}} \ldots x_{nn}^{e_{nn}}$ is the permanent of the $n \times n$ matrix $((e_{ij}))_{i,j}$. Turns out $f \in \mathsf{VNP}$. In fact, a necessary and sufficient condition is that the coefficient of a given monomial can be computed in $\#\mathsf{P}/\operatorname{poly}$.

1.2 Prior lower bounds

Proving lower bounds is generally considered challenging, in most models of computation. For general circuits, the best lower bound we have for an explicit polynomial is by [BS83] who prove an $\Omega(n \log n)$ lower bound. For the subclass of arithmetic formulas, [Kal85] has shown a $\Omega(n^{3/2})$ lower bound. On the other hand, we know by standard counting methods that most n-variate degree d polynomials require circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.

To gain better understanding of computation by arithmetic circuits, researchers focused on proving lower bounds for restricted models of computation. One very natural restriction is the depth of the circuit. Proving lower bounds for depth two circuits are trivial. For general depth three circuits, the best lower bound we have is by [SW01] who present an $\Omega(n^2)$ lower bound. Exponential lower bounds are known with additional restrictions like *homogeneity* [NW97], *multilinearity* [Raz09, RY09], over finite fields [GR00, GK98], *monotonicity* [JS82] etc.

For multilinear models, more is known for even larger depth. [Raz09] showed an $n^{\Omega(\log n)}$ lower bound for the class of multilinear formulas. [RY09] extended those techniques to show an $2^{n^{\Omega(1/\Delta)}}$ lower bound for multilinear formulas of depth Δ.

1.3 Relevance of shallow circuits for "VP vs VNP"

The study of lower bounds for shallow circuits is not just an attempt to simplify the problem and gain insight on the larger goal. The class of shallow arithmetic circuits are surprisingly powerful, unlike the boolean case. Shallow circuits in the arithmetic world almost capture the entire computational power of unrestricted circuits!

There has been a long series of results that simulate a general arithmetic circuit C by a *shallow* circuit of size comparable to the size of C. This task simulating a circuit but another not-too-large circuit of small depth is called *depth reduction*. The first result in this regard is by [VSBR83] who proved the following.

Theorem 1.3 ([VSBR83]). *Let f be an n-variate degree d polynomial computed by an arithmetic circuit C of size s. Then, f can be equivalently computed by a homogeneous circuit C' of depth $O(\log d)$ with unbounded fan-in $+$ and $\times$ gates and size $s' = (nds)^{O(1)}$.*

The above theorem allows us to focus on just homogeneous circuits of $O(\log d)$ depth and attempt lower bounds for this model. Any super-polynomial lower bound for the class of $O(\log d)$ depth circuits automatically yields a super-polynomial lower bound for general circuits.

However, if we really hope to prove much stronger lower bounds for Perm_n like say $2^{\Omega(n)}$, maybe we can afford to incur a slightly larger blow-up in size to obtain an even shallower circuit. This line was first pursued by [AV08], and subsequently strengthened by [Koi12] and [Tav15] to yield the following result.

Theorem 1.4 ([AV08, Koi12, Tav15])**.** *Let f be an n-variate degree d polynomial computed by an arithmetic circuit of size s. Then f can be computed by a homogeneous $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ circuit of size $s' \leq s^{O(\sqrt{d})}$*

More generally, for any $0 \leq r \leq d$, there is a homogeneous $\Sigma\Pi^{[O(d/r)]}\Sigma\Pi^{[r]}$ circuit of top fan-in at most $s^{O(d/t)}$ computing f.

Recall that a $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ circuit computes a polynomial of the form

$$f \quad = \quad \sum_{i=1}^{s} Q_{i1} \dots Q_{ia}, \quad \text{where } a = O(\sqrt{d}) \text{ and } \deg Q_{ij} \leq \sqrt{d}.$$

In other words, if we can prove a lower bound of $n^{\omega(\sqrt{d})}$ for the class of $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ circuits, we would have a super-polynomial lower bound for the class of general arithmetic circuits! In fact, the model of depth 4 circuits seem so central in that almost all known lower bounds for other restricted models proceed by proving a suitable lower bound for a depth 4 analogue. Several examples of this may be seen in [KS14a].

The first breakthrough was obtained by [GKKS14] who showed an $2^{\Omega(\sqrt{d})}$ lower bound for such circuits computing the symbolic $d \times d$ determinant or permanent. Subsequently, there was a flurry of activity towards achieving the goal of proving $n^{\omega(\sqrt{d})}$ lower bounds [KSS14, FLMS15, KS14b], and this is where we currently stand.

Theorem 1.5. *There is an explicit homogeneous n-variate degree d polynomial f that can be computed by a homogeneous depth 4 circuit of size $n^{O(1)}$ but any $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ computing it requires top fanin $s = n^{\Omega(\sqrt{d})}$.*

If we could change the $n^{\Omega(\sqrt{d})}$ to $n^{\omega(\sqrt{d})}$ in the above theorem (of course, the polynomial f cannot then have a small arithmetic circuit computing it), we would have proved a super-polynomial lower bound for general arithmetic circuits! The following is the simplest formulation of a lower bound of shallow circuit that would imply lower bounds for general circuits.

Open Problem 1.1

Find an explicit n-variate degree d polynomial f such that any expression of the form

$$f \quad = \quad (Q_1)^{\sqrt{d}} + \cdots + (Q_s)^{\sqrt{d}} \quad , \quad \deg(Q_i) \leq \sqrt{d} \text{ for all } i$$

must have $s = n^{\omega(\sqrt{d})}$.

Subsequent to this line of work, several researchers addressed the task of proving lower bounds for homogeneous depth 4 circuits without any restriction on the fan-ins. It is worth noting that a lower bound for homogeneous depth 4 circuits must be on the total size and not the top fan-in, as otherwise one could just compute the polynomial f in a single gate of the bottom two layers.

What to expect from this article

This article is intended to be a rolling survey of (almost) all known lower bounds in arithmetic circuit complexity. Most of the proofs in this article are complete and self-contained. However, as one would expect in the more delicate proofs, there would eventually involve a fair amount of calculation and setting of parameters. There might be proofs where this last technical calculation is avoided, but the hope is to make the presentation insightful enough so that it would enable any student to do the calculations (him/her)self.

Also, quite a lot of the proofs presented here are slightly different from the original proofs. The reason for the deviation would almost always be for more clarify and intuition. However, this process might also make the parameters involved a little weaker than in the original statements. We shall try to ensure that such losses do not change the overall strength of the theorem by much, and if they do we shall mention that explicitly.

Disclaimer: At many points in the survey we may use words like ‘recently’. This may have been true at the time of writing but perhaps not any more. :-)

What is not (yet) covered in this survey

There are surely a few notable lower bounds that are not (yet) discussed in the survey. A few that come to mind are the super-linear lower bound of Raz, Shpilka and Yehudayoff for syntactic multilinear circuits [RSY08], a recent improvement of this by Alon, Kumar and Volk [AKV18] and some more recent lower bounds involving shifted partial derivatives that are yet to be included.

Besides lower bounds, there are many approaches to prove lower bounds such as Raz's lower bound approach via elusive functions [Raz10a], or Valiant's rigidity approach, or the sum-of-squares question of [HWY10], or the Real τ conjecture, Geometric Complexity Theory etc.

Perhaps some of these gaps shall be filled in the near[2] future. Perhaps.

[2] :-)

Part I

Preliminaries

Chapter 2

Notation and Preliminaries

We first explain some notation that shall be used throughout this article.

- In most cases, the degree of the polynomial shall be denoted by d and the number of variables shall be denoted by n. In some cases, n would refer to a parameter that would determine the number of variables (though perhaps not exactly).
- Almost all the polynomials that we shall be studying would be multilinear. In multilinear polynomials, we shall identify a monomial with the *set of variables* that it is a product of. This would allow us to abuse notation and say $x_i \in m$ to mean that x_i divides the monomial m, or to say $m_1 \cap m_2$ to refer to the $\gcd(m_1, m_2)$. We shall also use the notation $m \in f$ to mean that the polynomial m has a non-zero coefficient for the monomial m.
- We shall use $[n]$ to denote $\{1, \ldots, n\}$ and we shall use boldface letters such as $\mathbf{x}$ to denote sets of variables. Further, $\mathbf{x}_{[n]}$ shall refer to a set of variables $\{x_1, \ldots, x_n\}$. However, if the number of variables is understood, we shall drop the subscript.

2.1 Models of computation

As mentioned earlier, the most robust model of computation that are studied are arithmetic circuits, which are formally defined as follows.

Definition 2.1 (Arithmetic circuits and formulas)**.** *An arithmetic circuit is a directed acyclic graph with a unique sink vertex called the* root*. The source vertices are labelled by either formal variables or field constants, and each internal node of the graph is labelled by either* $+$

or $\times$. Nodes compute formal polynomials in the input variables in the natural way. Further, edges entering a + nodes also might have field constants on them to allow the + to compute an $\mathbb{F}$-linear combination of the children (rather than just a sum).

The polynomial computed by the circuit is defined as the polynomial computed by the root.

If the underlying graph is a tree *instead of a general acyclic graph, the circuit is called a* formula*.* ◊

Another model of computation that is studied often is the model *algebraic branching programs*, defined as follows.

Definition 2.2 (Algebraic Branching Program)**.** *An algebraic branching program (ABP) is a layered graph with a unique source vertex (that we shall call s) and a unique sink vertex (that we shall call t). All edges are from layer i to layer $i+1$, and each edge is labelled by a linear polynomial. The polynomial computed by the ABP is defined as*

$$f \quad = \quad \sum_{\gamma : s \rightsquigarrow t} \mathrm{wt}(\gamma)$$

where for every path γ from $s \rightsquigarrow t$, the weight $\mathrm{wt}(\gamma)$ *is defined as the product of the labels over the edges in γ.* ◊

The width of the ABP is defined as the maximum number of vertices in any layer, and the depth is defined as the length of the longest path from s to t. The polynomial computed by an ABP is captured by the *iterated matrix multiplication* polynomial that we shall soon see.

It is easy to observe that any arithmetic formula of size s can be simulated by an algebraic branching program of size $\mathrm{poly}(s)$, and any algebraic branching program of size s can be simulated of an arithmetic circuit of size $\mathrm{poly}(s)$. It is a major open problem to show a separation between any of these.

$$\text{Formulas} \quad \subseteq \quad \text{ABPs} \quad \subseteq \quad \text{Circuits}$$

Open Problem 2.1

Show a super-polynomial separation between any of the models – formulas, ABPs and *circuits.*

2.1.1 Constant depth circuits

We shall be dealing a lot with constant depth circuits. Normally, the root is assumed to be a + gate[1] and the circuit is assumed to consist of alternating layers of + and × gates. A layer of + gates are called Σ layers, and a layer of × gates are called Π layers. Thus, a depth two circuit consist of the form

$$f \quad = \quad \sum_{i=1}^{s} \prod_{j=1}^{d} x_{ij}$$

is a $\Sigma\Pi$ circuit.

Fact 2.3. *Any arithmetic circuit of depth Δ and size s, can be simulated by an arithmetic formula of depth Δ and size $s' \leq s^{\Delta}$.*

Thus for constant depth circuits (where $\Delta = O(1)$), we may assume that we are dealing with formulas without much loss of generality.

It would also be useful to keep track of the *fan-in* of the gates in a certain layer (especially of multiplication gates). We shall use superscripts to denote this. For example, an $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuits computes a polynomial of the form

$$f \quad = \quad \sum_{i} \prod_{j=1}^{a} Q_{ij}$$

where each Q_{ij} is a polynomial of degree at most b.

It would also be useful to consider special layers of multiplication gates that multiply the same polynomial several times, rather than multiplying several different polynomials together. Since such gates simply raise the input to a certain power, these would be called *exponentiation* gates. A layer of exponentiation will be denoted by $\wedge$ [2] and, for example, a

[1] The reason for this is that often the polynomial computed by the circuits would be irreducible, and hence would be silly to have a × gate as a root.

[2] To say a little on the choice of notation, it was introduced in [GKKS16] and the first choice was to use ^, but looked rather ugly to write it as say Σ^Σ. Hence, $\Sigma\wedge\Sigma$ was chosen instead.

$\Sigma\wedge\Sigma$ circuit computes a polynomial of the form

$$f = \sum_{i=1}^{s} \ell_i^d$$

where each ℓ_i is a linear polynomial.

Exercise 2.1 *Show that any constant width ABP can be simulated by a polynomial sized formula.*

2.2 Polynomials of interest

There are a few polynomials that are the usual suspects while proving lower bounds. The polynomials that we would be dealing with in this article are defined below.

The determinant and the permanent families

The determinant of an $n \times n$ symbolic matrix shall be denoted by Det_n and is defined as

$$\mathsf{Det}_n = \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) x_{1,\sigma(1)} \dots x_{n,\sigma(n)}.$$

The permanent of an $n \times n$ symbolic matrix shall be denoted by Perm_n and is defined as

$$\mathsf{Perm}_n = \sum_{\sigma \in S_n} x_{1,\sigma(1)} \dots x_{n,\sigma(n)}.$$

Both of these polynomials are of degree n and over n^2 variables. We know that Det_n can be computed by a polynomial sized arithmetic circuit and it is widely believed that the permanent requires circuits of size $2^{\Omega(n)}$.

The Nisan-Wigderson polynomial families

Definition 2.4 (Nisan-Wigderson Polynomials)**.** *Let n, m, d be arbitrary parameters with m being a power of a prime, and $n, d \leq m$. Since m is a power of a prime, let us identify the set $[m]$ with the field $\mathbb{F}_m$ of m elements. Note that since $n \leq m$, we have that $[n] \subseteq \mathbb{F}_m$. The*

Nisan-Wigderson polynomial with parameters n, m, d, denoted by $\mathrm{NW}_{n,m,d}$ is defined as

$$\mathrm{NW}_{n,m,d}(\mathbf{x}) \quad = \quad \sum_{\substack{p(t)\in\mathbb{F}_m[t]\\ \deg(p)\leq d}} x_{1,p(1)} \dots x_{n,p(n)}.$$

That is, for every univariate polynomial $p(t) \in \mathbb{F}_m[t]$ of degree at most d, we add one monomials that encodes the 'graph' of p on the points $[n]$. This is a polynomial of degree n over mn variables. ♢

This monomials of this polynomial satisfy a very useful "low pairwise-intersection" property.

Lemma 2.5. *Let m_1 and m_2 be any two distinct monomials in $\mathrm{NW}_{n,m,d}(\mathbf{x})$. Then, there are at most d variables that divide both m_1 and m_2.*

Proof. Let m_1 and m_2 correspond to univariates $p_1(t), p_2(t) \in \mathbb{F}_m[t]$ of degree at most d. Then if x_{ij} divides m_1, then $p_1(i) = j$, similarly for m_2. But since p_1 and p_2 are two distinct polynomials of degree at most d, they can agree in at most d evaluations. Thus, there can be at most d variables that divide both m_1 and m_2. □

For most generic choices of the parameters, the polynomial $\mathrm{NW}_{n,m,d}$ is believed to require circuits of exponential size to compute them.

The Iterated-Matrix-Multiplication polynomial

For parameters n and d, the Iterated-Matrix-Multiplication polynomial, denoted by $\mathrm{IMM}_{n,d}$, is defined as follows

$$\mathrm{IMM}_{n,d} \quad = \quad \sum_{1\leq i_1,\dots,i_d\leq n} x^{(1)}_{1,i_1} x^{(2)}_{i_1,i_2} \dots x^{(d-1)}_{i_{d-2},i_{d-1}} x^{(d)}_{i_{d-1},1}.$$

An equivalent way of defining the polynomial as the $(1,1)$-th entry of the product of d generic $n \times n$ matrices:

$$\mathrm{IMM}_{n,d} \quad = \quad \left(\begin{bmatrix} x^{(1)}_{11} & \dots & x^{(1)}_{1n} \\ \vdots & \ddots & \vdots \\ x^{(1)}_{n1} & \dots & x^{(1)}_{nn} \end{bmatrix} \cdots \begin{bmatrix} x^{(d)}_{11} & \dots & x^{(d)}_{1n} \\ \vdots & \ddots & \vdots \\ x^{(d)}_{n1} & \dots & x^{(d)}_{nn} \end{bmatrix} \right)_{(1,1)}.$$

It is often useful to think of this as the polynomial computed by a *generic algebraic branching program* of width n and depth n (where the edge connecting vertex i of layer ℓ to vertex j of layer $\ell+1$ has weight $x_{ij}^{(\ell)}$).

This is a polynomial of degree d and over $n^2(d-2)+2n$ variables[3]. Further, since the polynomial corresponds to a generic algebraic branching program, $\mathrm{IMM}_{n,d}$ can be computed by an arithmetic circuit of size $\mathrm{poly}(n,d)$.

[3] Only the first row of the first matrix, and the first column of the last matrix participates in the $(1,1)$ entry of the product

Chapter 3

Algebraic complexity classes

Valiant [Val79] defined two algebraic complexity classes that can be thought of as analogues of the boolean classes P and NP. This chapter focuses on their definitions, and some important properties related to the polynomial families Det_n and Perm_n.

3.1 Definition of classes

Recall that P is[1] the class of boolean functions that can be computed by circuits of polynomial size. As any boolean function can be expressed as a multilinear[2] polynomial, an analogue of this in the arithmetic world could be multilinear polynomials $f(x_1, \ldots, x_n)$ that can be computed by arithmetic circuits of size $\mathrm{poly}(n)$. However, unlike in the boolean world, the polynomial x^2 is not equal to the polynomial x as we are dealing with formal polynomials. Valiant's definition of VP was the class of the class of "low degree" polynomials that can be computed by circuits of small size.

Definition 3.1 (Valiant's P). *The class* VP *refers to the set of polynomials* $f(x_1, \ldots, x_n)$ *of* degree $\mathrm{poly}(n)$ *that can be computed by arithmetic circuits of size* $\mathrm{poly}(n)$*.* $\Diamond$

In the literature, one also encounters classes such as VBP and VF that correspond to polynomials computed by polynomial-sized ABP and formulas respectively. These are subclasses of VP by definition.

[1] If one is to be more precise, this is P/poly or non-uniform P. But in this article, we shall be interested only in the non-uniform versions since we mainly deal with circuit sizes.

[2] A polynomial where the degree in any variable is bounded by 1.

More on the low-degree restriction

But should the analogue of VP not be the class of polynomials that are computable by poly(n)-sized arithmetic circuits, including polynomials of very large degree? We can indeed compute polynomials of very large degree, such as a circuit that is a chain of s multiplication gates thus computing a polynomial of degree exp(s) (by repeated squaring). Let us first take a moment to understand why the additional restriction of "low-degree" in the above definition of VP was imposed. There are several intuitive reasons for this, and this "restricted" definition also yields beautiful structural results. This discussion is from an answer by Joshua Grochow on `cstheory.SE` [Gro] to shed more light on the above definition.

1. Every boolean function can be expressed as a multilinear polynomial. Multilinear polynomials are, of course, polynomials of "low degree".

2. Much of the earlier work was based on understanding formula size. In the case of arithmetic formulas, the degree cannot be more than the size of the formula. If a polynomial is computed by a poly(n) sized formula, then its degree must be bounded by poly(n) too.

3. Most interesting polynomials, such as Det_n or Perm_n, are in fact of low degree. Once we choose to deal with only polynomials of low degree, the above definition does not have any restriction on the circuit used to compute it (besides its size). It is certainly possible that intermediate computations of the circuit could involve very large degree polynomials.

   However, as we shall soon see, Strassen's result shows that such high degree computations may be eliminated with just an $O(\deg^2)$ increase in size. Dealing with low-degree circuits also makes the class robust under this transformation. Eliminating division gates also only incurs an $O(\deg^2)$ increase in size.

4. Without this restriction, one cannot hope to show that Det_n or Perm_n is "complete" for such classes, or show the numerous structural results such as depth reduction that we have.

Having said that, there is also a notion of "degree" in a boolean circuit that is defined syntactically as follows:

Degree of all leaves is 1.

For any OR gate, the degree is the maximum of the degree of its children.

For any AND gate, the degree is the sum of the degree of its children.

The class of boolean functions that can be computed by poly-sized poly-degree circuits coincide with a class called LOGCFL or SAC^1. With this notion, one might say that VP is really an analogue of SAC^1.

We now move on to the arithmetic analogue of the class NP. Recall that the class NP may be defined as the set of all boolean functions $f(x_1, \dots, x_n)$ such that there is some $g(x_1, \dots, x_n, y_1, \dots, y_m)$ with $m = \text{poly}(n)$ and

$$f(x_1, \dots, x_n) \quad = \quad \bigvee_{\mathbf{a} \in \{0,1\}^m} g(x_1, \dots, x_n, a_1, \dots, a_m).$$

Valiant's NP is defined analogously by replacing the OR by a sum.

Definition 3.2 (Valiant's NP)**.** *The class* VNP *is defined to be the set of polynomials* $f(x_1, \dots, x_n)$ *such that there is some* $g(x_1, \dots, g_n, y_1, \dots, y_m) \in \mathsf{VP}$ *with* $m = \text{poly}(n)$ *and*

$$f(x_1, \dots, x_n) \quad = \quad \sum_{\mathbf{a} \in \{0,1\}^m} g(x_1, \dots, x_n, a_1, \dots, a_m).$$

$\Diamond$

It follows from definitions that $\mathsf{VF} \subseteq \mathsf{VBP} \subseteq \mathsf{VP} \subseteq \mathsf{VNP}$. We do not know if any of the containments is strict (although it is widely believed that all of them are).

3.2 Some properties of these classes

We shall state a few properties of these classes here. For a more extensive treatment, Bürgisser's book [Bür00] has a comprehensive study of these classes and a lot of the proofs presented in this chapter are based on the description in his book.

Valiant [Val79] presented a very useful sufficient condition to show that a given polynomial is in VNP.

Theorem 3.3 (Valiant's Criterion)**.** *Let* $f = \sum_{\mathbf{e}} c(\mathbf{e}) \cdot x_1^{e_1} \dots x_n^{e_n}$*. Suppose the function* φ *that takes as input the exponent vector* $\mathbf{e} = (e_1, \dots, e_n)$ *and outputs the coefficient* $c(\mathbf{e})$ *is in the class* $\#\mathsf{P}/\text{poly}$*, then* $f \in \mathsf{VNP}$*.*

Thus in particular, if we compute the coefficient for a given monomial in polynomial time, then the polynomial is in VNP.

Corollary 3.4. *The polynomials* Perm_n *and* $\mathrm{NW}_{n,m,d}$ *are in* VNP*.*

As stated in Definition 3.2, the polynomial $g(x_1, \ldots, x_n, y_1, \ldots, y_m) \in \mathsf{VP}$. However, a subsequent paper of Valiant [Val82] showed that we may assume without loss of generality that $g \in \mathsf{VF}$, that is g is computable by a small formula. This is similar to the fact that counting solutions of 3-CNF instance, which is a formula, is as hard as counting solutions of any polynomial sized boolean circuit. We state this result here, and a proof may be found in Bürgisser's book [Bür00]. Malod and Portier [MP08] presented an alternate proof via *proof-trees* which is perhaps more instructive.

Theorem 3.5. *For any* $f(x_1, \ldots, x_n) \in \mathsf{VNP}$*, there is a* $g(x_1, \ldots, x_n, y_1, \ldots, y_m)$ *that can be computed by a* poly(n) *sized* formula *such that*

$$f(x_1, \ldots, x_n) \quad = \sum_{\mathbf{a} \in \{0,1\}^m} g(x_1, \ldots, x_n, a_1, \ldots, a_m).$$

3.3 Reductions and completeness

For polynomials, the most natural form of reductions are via *projections*. We shall say that a polynomial f *reduces* to g via projections if g may be obtained by substituting variables of f to other variables or field constants. Under such reductions, do we have natural complete polynomials for the classes VP and VNP? Valiant [Val79] showed that the Det_n and Perm_n are (almost) complete for the classes VP and VNP respectively. We shall see the proof of this in this section.

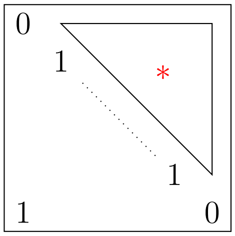
Theorem 3.6 ([Val79])**.** *If f is a polynomial computed by an ABP of size s, then f reduces to* Det_n *via projections for* $n = \mathrm{poly}(s)$*.*

Theorem 3.7 ([Val79])**.** *Over any field $\mathbb{F}$ of characteristic not equal to* 2*, the polynomial* Perm_n *is* complete *for the class* VNP *under projections.*

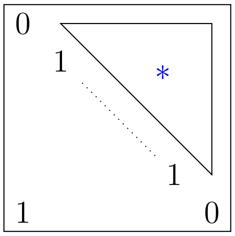
3.3.1 Warm-up: Formulas to permanents

Let us first show that any polynomial sized arithmetic formula can be expressed as permanent of a small matrix. In fact, all matrices that we shall be building this way will have the following structure, and this would be important. (Here $*$, $*$ denote upper triangle entry of the matrix of f, g respectively.)

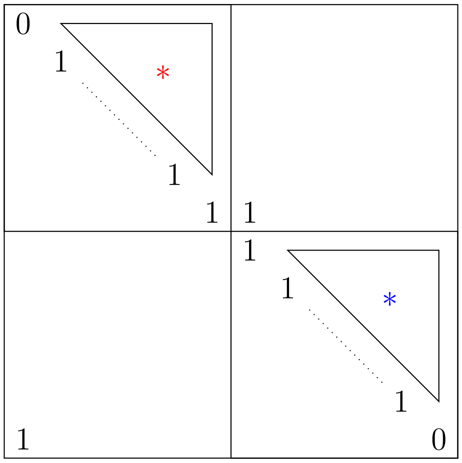
$f \quad = \quad \mathsf{Perm}$



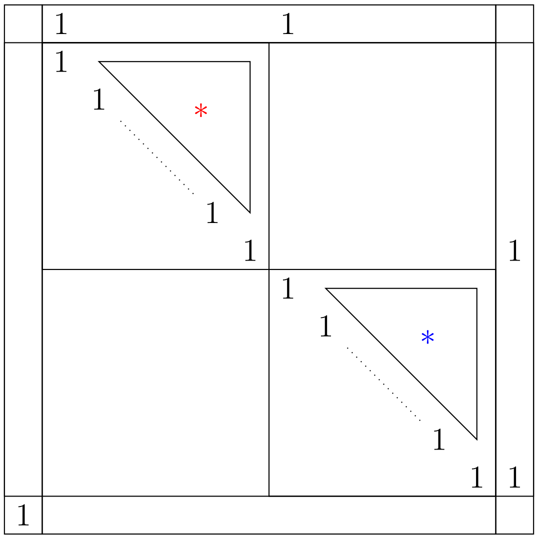
$g \quad = \quad \mathsf{Perm}$



Multiplication



Addition



We leave it to as an exercise to check that this indeed works. This may seem bizarre at first but there is indeed a method to this. To understand that method, we must realise that there is a graph theoretic way to understand Det_n and Perm_n.

Graph theoretic representation of Det_n and Perm_n

Let us think of an $n \times n$ symbolic matrix as an adjacency matrix of a directed graph G with the edge from i to j having weight x_{ij}. Then every monomial of Det_n or Perm_n corresponds to a permutation σ, and the corresponding edges in the graph G form a *cycle cover* i.e., a partition of the vertices of G into disjoint cycles. The weight of a cycle-cover shall be defined as the product of weights of the edges constituting the cycles, and the sign of the cycle cover shall be the sign of the permutation σ. This allows us to write Det_n and Perm_n as

$$\begin{aligned} \det(G) &= \sum_{C \in \text{CycleCovers}(G)} wt(C) \cdot \text{sign}(C), \\ \text{perm}(G) &= \sum_{C \in \text{CycleCovers}(G)} wt(C). \end{aligned}$$

3.3.2 ABPs reduce to Det_n

We shall now show that for any f computable by an ABP, there is a matrix A each of whose entries are either variables of constants such that $\det(A) = f$. Since any ABP is a projection of the polynomial $\text{IMM}_{n,d}$, we shall show that we can construct a matrix A such that $\det(A) = \text{IMM}_{n,d}$.

Consider the graph G underlying the ABP that corresponds to the polynomial $\text{IMM}_{n,d}$. Let s and t be the unique source and sink vertices respectively. Every path from $s \rightsquigarrow t$ corresponds to a single monomial of $\text{IMM}_{n,d}$ of degree d. We shall now modify the graph slightly such that each such $s \rightsquigarrow t$ path would map to a unique cycle cover:

> To the graph G, add an edge with weight 1 from t to s.
>
> Further, for all nodes except s and t, add a self-loop of weight 1.

Let A be the adjacency matrix of this new graph G'. The claim is that $\det(A)$ is either $\text{IMM}_{n,d}$ or $-\text{IMM}_{n,d}$. To see this observe that all cycle covers of G' must consist of a single $s \rightsquigarrow t$ path that loops back to s via the edge $t \to s$ that we added, and self-loops on all excluded vertices. Further, since all $s \rightsquigarrow t$ paths in G were of the same length, it is easy to check that all the cycle covers have the same sign. If the sign is negative, we can change the weight of the $t \rightsquigarrow s$ edge to -1. Thus $\mathrm{IMM}_{n,d}$ does indeed reduce to Det_m for $m = \mathrm{poly}(n)$. □(Theorem 3.6)

Exercise 3.1 *Look back at the construction in Subsection 3.3.1 to convince yourself that the permanents are indeed $f \cdot g$ and $f + g$.*

Proving the same for determinants requires handling signs. Modify the construction appropriately for determinants.

Hint: *Convert a formula to an ABP with the property that all $s \rightsquigarrow t$ paths have the same length?*

3.3.3 Det_n can be computed by ABPs

A result that is often stated, but not proven explicitly, is the existence of a polynomial sized circuit for Det_n. This is often attributed by Berkowitz [Ber84]. In fact, Det_n has a polynomial sized ABP and this construction is due to Mahajan and Vinay [MV97] based on *clow sequences.* The construction is really neat and we shall describe the ABP explicitly here.

$$\mathsf{Det}_n \quad = \quad \begin{vmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nn} \end{vmatrix}$$

If the symbolic matrix on the RHS is the adjacency matrix of an n-vertex graph, then the determinant is just the sum of weighted signed cycle-covers of the graph. A natural approach to compute this via an ABP is to somehow compute each cycle cover on one path of the ABP. Unfortunately, if one were to try the naïve approach of building a cycle cover over many layers, to decide what our next vertex should be, we are forced to remember the entire partial construction thus far. This ends up yielding an ABP with super-polynomial width (the width intuitively corresponds to the memory required).

The key insight of Mahajan and Vinay was to relax the notion of cycle covers to something weaker that can be built with less memory, to what they called *clow sequences.*

Definition 3.8 (Clow Sequences)**.** *Label the vertices of the graph as* $1, \ldots, n$*. A clow of length* ℓ *is a closed walk on the graph* G *of length* ℓ *such as* $v_1, \ldots, v_\ell, v_1$ *such that* $v_1 < v_i$ *for all* $i = 2, \ldots, \ell$*. We shall also refer to* v_1 *as the* head of the clow*.*

In other words, the head of a clow is the smallest vertex of the walk, and the head does not repeat in a clow (although other vertices can).

A clow sequence is a sequence of clows $(C_1, \ldots, C_r)$ *that additionally satisfy* $\mathrm{head}(C_1) < \cdots < \mathrm{head}(C_r)$*.*

The length of a clow sequence is the sum of the lengths of the clows that it comprises of. The weight of a clow sequence is just the product of weights of the edges it comprises of. Also, the sign of a clow sequence of length ℓ *that comprises of* r *clows is* $(-1)^{\ell + r}$*.* ♢

Any cycle cover is, of course, a clow sequence. Further, the sign of the cycle cover matches the above definition of the sign of a clow sequence. But there are many clow sequences that visit some vertices multiple times, and hence are not cycle covers. However, Mahajan and Vinay show that the sum of signed-weights of all clow sequences also yields the determinant.

Lemma 3.9 ([MV97])**.** *If* A_G *is the adjacency matrix of a graph* G*, then*

$$\begin{aligned} \det(A_G) \quad &= \sum_{C \in \mathrm{CycleCover(G)}} wt(C) \cdot \mathrm{sign}(C) \quad = \quad \det(A_G) \\ &= \sum_{C \in \mathrm{ClowSequence(G)}} wt(C) \cdot \mathrm{sign}(C). \end{aligned}$$

They prove this by showing that the set of clow sequences that are not cycle covers can be partitioned into pairs (C_1, C_2) such that $wt(C_1) = wt(C_2)$ and $\mathrm{sign}(C_1) = -\mathrm{sign}(C_2)$. We shall see an explicit proof of this shortly, but first let us see why this yields an ABP.

The ABP consists of $n+1$ layers labelled as layer $1, \ldots, n+1$. Besides layer 1 and $n+1$, every other layer ℓ consists of $\Theta(n^2)$ nodes that we shall label as $v_{i,j}^{(\ell)}$ for $1 \leq i \leq j \leq n$. It is best to think of i as representing the *head of the current clow*, and j as the *current vertex in the clow*. The length of the partial clow sequence constructed so far is captured by the layer index ℓ.

The first layer consists of a single vertex that we shall call $s = v_{1,1}^{(1)}$ (to maintain the same notation) and the last layer consists of a single vertex that we shall call t. The edges between layers, and the weights are defined as follows:

1. For each node $v_{i,j}^{(\ell)}$ in layer $\ell \in [n]$, there is an edge to vertex $v_{i,k}^{(\ell+1)}$ for every $k > i$. The weight of this edge is x_{jk}.

(This is like adding vertex $k > i$ to our current clow by taking edge x_{jk}. The head continues to be i, and the current vertex is now k.)

2. For each node $v_{i,j}^{(\ell)}$ in layer $\ell \in [n]$, there is an edge to vertex $v_{k,k}^{(\ell+1)}$ for every $k > i$. The weight of this edge is $(-x_{ji})$.

   (This is like ending the current clow by taking edge x_{ji} back to the head, and starting a new clow with head as k. Thus, the head of the current clow is k, and the current vertex is also k. In this process, we increased the number of clows in the sequence by 1 and hence the weight being $(-x_{ji})$ accounts for the sign change as well.)

3. For the last layer, each node $v_{i,j}^{(n)}$ has an edge to t with weight $(-x_{ji})$.

   (This just corresponds to ending the last clow in our sequence.)

Summarizing this as a theorem, we have:

Theorem 3.10 ([MV97])**.** *The polynomial* Det_n *can be computed by an ABP of size* $O(n^3)$ *over any field* $\mathbb{F}$*. Thus, in particular,* Det_n *can be computed by a arithmetic circuit of size* $\mathrm{poly}(n)$*.*

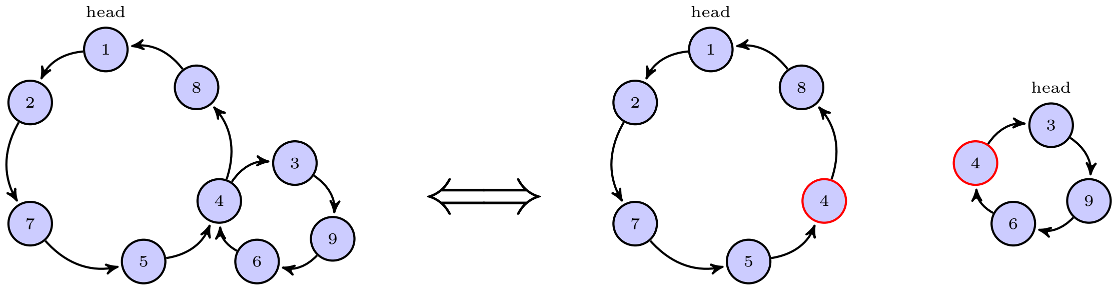
Proof of Lemma 3.9. Consider a clow sequence $C = (C_1, \ldots, C_r)$ of length n, ordered so that $\mathrm{head}(C_1) < \cdots < \mathrm{head}(C_r)$. If C is *not* a cycle-cover, then some vertex must be repeated in C. Starting from the last clow and proceeding backwards, let i be an index such that $(C_{i+1}, \ldots, C_r)$ is a union of disjoint cycles but $(C_i, \ldots, C_r)$ is not, and let $C_i = (v_1, \ldots, v_k)$. Let j be the first index that makes the vertex v_j show that $(C_i, \ldots, C_r)$ not be a union of disjoint cycles. Then, exactly one of the two situations must occur:

Case 1: $v_j = v_{j'}$ for some $j' < j$,

Case 2: v_j occurs in one of the cycles $C_{i+1}, \ldots, C_r$.

In the first case, the vertices $v_{j'+1}, \ldots, v_j$ are all distinct since v_j was the first occurrence of a repeated node. Define a new clow sequence $\tilde{C}$ obtained by decomposing the clow C_i with two clows $((v_1, \ldots, v_{j'}, v_{j+1}, \ldots, v_k), (v_{j'}, v_{j'+1}, \ldots, v_j))$. Note that the second clow $(v_{j'}, \ldots, v_j)$ is an honest-to-god cycle that does not intersect with any of the cycles $C_{i+1}, \ldots, C_r$. This transformation converts the clow sequence C with r clows to a clow sequence C' of $r + 1$ clows and hence $\mathrm{sign}(C) = -\mathrm{sign}(C')$.

In the second case, we have $v_j \in C_i$ also present in $C_{i'}$ for some $i' > i$. Here we shall apply the inverse operation of combining the clows C_i and C_i' at the vertex j. Formally, let us rotate $C_{i'}$ cyclically so that $C_{i'} = (v_1', \ldots, v_k')$ with $v_1' = v_j$. The new clow sequence C' shall be constructed by replacing the clows C_i and $C_{i'}$ by a new clow $(v_1, \ldots, v_j, v_2', \ldots, v_k', v_{j+1}, \ldots)$. Since every vertex in $C_{i'}$ is greater than $\text{head}(C_{i'})$ which is greater than $\text{head}(C_i)$, this process does indeed result in a valid clow. Once again, $\text{sign}(C) = -\text{sign}(C')$ as the number of clows in the sequence has reduced by exactly one.



$(1, 2, 7, 5, 4, 3, 9, 6, 4, 8, 1)$

$(1, 2, 7, 5, 4, 8, 1), (3, 9, 6, 4, 3)$

It is not hard to see that the two operations in the two different cases are exact inverses of each other. Thus, this establishes a matching among all clow sequences that are not cycle covers, with every matched pair having opposing signs. Thus, the overall contribution of clow sequences that are not cycle covers is zero. □

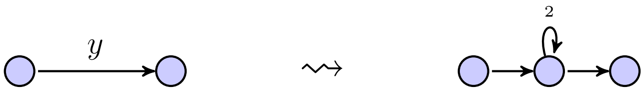
3.3.4 Completeness of the permanent

The VNP completeness for the permanent is trickier, and uses a very clever gadget. The proof described here is a modification of the proof in Bürgisser's book [Bür00]. This is the simplest proof that I am aware of currently.

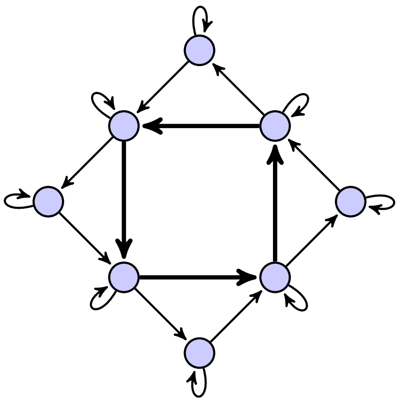
Let $f(x_1, \ldots, x_n) = \sum_{\mathbf{a}} g(x_1, \ldots, x_n, a_1, \ldots, a_m)$ be in VNP with $g(\mathbf{x}, \mathbf{y})$ computable by a formula of size s (Theorem 3.5). Like in the previous section, we can construct a graph G (with weights being either scalars or variables in $\mathbf{x}$ or $\mathbf{y}$) such that the sum of weighted cycle covers is equal to $g(\mathbf{x}, \mathbf{y})$. The goal is to now compute $\sum_{\mathbf{a}} g(\mathbf{x}, \mathbf{a})$.

Let us consider a simpler case, where there is a variable $y \in \mathbf{y}$ such that there is just one edge in $e_y \in G$ with weight y. Can we transform the graph G locally to compute

$g' = g_{(y=0)} + g_{(y=1)}$? Note that since y occurs only once in G, we have that $g = y \cdot g_1 + g_0$ where g_1 and g_0 are independent of y. Thus, the polynomial g' can be written as $g_1 + 2g_0$. One way to compute this is to transform the graph G so that any cycle-cover of G that includes the edge e_y has the same weight as before, but every cycle-cover that does not include the edge e_y has its weight multiplied by 2. This can be achieved by splitting the edge e_y in the middle with a new vertex v with a self-loop of weight 2:



Clearly, any cycle-cover that uses the edge e_y has the same weight, and all other cycle-covers are forced to take the self-loop around the added vertex of weight 2. This allows us to handle graphs G where every variable $y \in \mathbf{y}$ occurs only once in G. The complication arises because there could be multiple edges that has the label y. We want a way by which we can say that all cycle covers that choose *any* of the y-edges have the same weight, but cycle-covers that do not pick any y-edge have weight multiplied by 2. The following is a gadget that has similar properties, called *rosette*. The diagram below represents the 4-rosette.

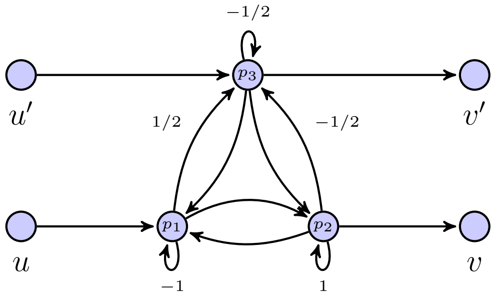


The thick edges in the above picture shall be called *connector edges* (these shall play the role of the y-edges). Note that the rosette has the following two properties:

1. For any non-empty subset S of the connector edges, there is exactly one cycle-cover of the rosette that includes exactly the set S of the connector-edges.

2. There are exactly two cycle-covers of the rosette that do not include any connector edge.

Thus, if we could somehow "glue" the connector edges with our y-edges, we would be done. This is achieved by yet another gadget that we shall call the *glue gadget*. The following is the description of the glue gadget that glues edges (u, v) and (u', v') by adding three additional vertices.



The adjacency matrix between the nodes p_1, p_2 and p_3 is

$$A = \begin{bmatrix} -1 & 1 & 1/2 \\ 1 & 1 & -1/2 \\ 1 & 1 & -1/2 \end{bmatrix}.$$

Claim 3.11. *Let (u, v) and (u', v') be two edges of a graph G, and let G' be the graph with the glue gadget between them as described above. Then $\operatorname{perm}(G')$ equals the sum of all weighted cycle covers of G that either include both (u, v) and (u', v') in it or neither.*

Proof. If both edges (u, v) and (u', v') are taken in the cycle cover, this is realized in G' as $(\ldots u, p_1, p_2, v \ldots)(\ldots u', p_3, v' \ldots)$, which has the same weight.

If neither of the edges (u, v) and (u', v') are taken in the cycle cover, then the total contribution of all cycle covers of p_1, p_2 and p_3 is $\operatorname{perm}(A) = 1$.

If the edge (u, v) is taken but (u', v') is not, then the edge (u, v) can be realized in G' as either $\{(\ldots u, p_1, p_2, v \ldots)(p_3)\}$ or $\{(\ldots u, p_1, p_3, p_2, v \ldots)\}$. The total contribution is therefore zero.

Similarly, if the edge (u', v') is taken but not (u, v), then this can be realized in G' as $\{(\ldots u', p_3, v' \ldots), (p_1, p_2)\}$ and $\{(\ldots u', p_3, v' \ldots)(p_1)(p_2)\}$. Again, the net contribution is zero. □

Now with these two gadgets, we are done. For every variable $y_i \in \mathbf{y}$, let $e_{i,1}, \ldots, e_{i,r_i}$ be the edges labelled with y. Let R_i be a r_i-rosette disjoint from the graph G The graph G' is built as follows:

Take a disjoint union of G with one r_i-rosette for each $i = 1, \ldots, m$.

Glue each edge labelled with y_i with the r_i connector edges in the r_i-rosette.

It should be easy to see that $\text{perm}(G) = \sum_{\mathbf{a}} g(\mathbf{x}, \mathbf{a})$. This completes the proof of the VNP-completeness of Perm_n. □(Theorem 3.7)

Note that we needed to divide by 2 in the glue gadget. This is why we require the characteristic of the field to be different from two for the above proof to work.

Exercise 3.2 *For any matrix A, we shall use the notation that $A[1|2]$ refers to the submatrix obtained by removing row 1 and column 2.*

Show that, for a three-vertex glue gadget*, it suffices to find a 3×3 matrix A that satisfies the following three properties:*

- $\text{perm}(A) = \text{perm}(A[2,3|1,3]) = 1.$
- $\text{perm}(A[3|3]) = \text{perm}(A[2|1]) = \text{perm}(A[1|3]) = \text{perm}(A[2|3]) = 0.$

Come up with an alternate construction of a matrix A as a glue gadget*. Also prove that any such construction must involve entries of A with its denominator divisible by 2, and show that replacing* perm *by* det *above would yield no solution.*

Chapter 4

Some estimates for binomial coefficients

Throughout this article, we would be seeing several binomial coefficients. The following estimates would allow us to get a better handle on the growth of such terms.

We shall use log to refer to $\log_2$ and ln to refer to the natural logarithm.

Definition 4.1 (Entropy function)**.** *The binary entropy function $H_2 : [0,1] \to [0,1]$ is defined as*

$$H_2(p) \quad = \quad -p\log_2(p) \quad - \quad (1-p)\log_2(1-p).$$

The entropy function with respect to the natural logarithm is refer to as H and

$$H(p) \quad = \quad -p\ln(p) \quad - \quad (1-p)\ln(1-p).$$

$\Diamond$

Proposition 4.2. *For any $0 < p < 1$, we have $p \ln \frac{1}{p} \leq H(p) \leq p \ln \frac{1}{p} + p$.*

Proposition 4.3 (Stirling's Approximation)**.**

$$\lim_{n\to\infty} \frac{n!}{\left(\frac{n}{e}\right)^n \sqrt{2\pi n}} \quad = \quad 1.$$

Proposition 4.4. *For any $n \geq k \geq 0$,*

$$\left(\frac{n}{k}\right)^k \quad \leq \quad \binom{n}{k} \quad \leq \quad \left(\frac{ne}{k}\right)^k.$$

Proposition 4.5. *For any constants α, β,*

$$\log \binom{\alpha n}{\beta n} \quad = \quad H_2\left(\frac{\beta}{\alpha}\right) \cdot \alpha n - O(\log n).$$

In particular, if $\beta = \alpha/2$, then $\binom{\alpha n}{\beta n} = 2^{\alpha n} / \operatorname{poly}(n)$.

For the more recent lower bounds, we would encounter several delicate ratios of binomial coefficients. The following lemma would help us simplify several such expressions and get a better handle on the growth.

Lemma 4.6. *[GKKS14, Lemma 6] For any $a, b = O(\sqrt{n})$, then*

$$\frac{(n+a)!}{(n-b)!} \quad = \quad n^{a+b} \cdot \operatorname{poly}(n).$$

We shall be using the above lemma very often in the lower bounds. One particular instantiation that shall also appear frequently shall be the following lemma.

Lemma 4.7. *Let n and ℓ be parameters such that $\ell = \frac{n}{2}(1-\varepsilon)$ for some $\varepsilon = o(1)$. For any a, b such that $a, b = O(\sqrt{n})$,*

$$\binom{n-a}{\ell-b} \quad = \quad \binom{n}{\ell} \cdot 2^{-a} \cdot (1+\varepsilon)^{a-2b} \cdot \exp(O(b \cdot \varepsilon^2)).$$

Proof. The proof of the above lemma would repeated use Lemma 4.6.

$$\begin{aligned}
\frac{\binom{n-a}{\ell-b}}{\binom{n}{\ell}} &= \frac{(n-a)!}{n!} \cdot \frac{\ell!}{(\ell-b)!} \cdot \frac{(n-\ell)!}{(n-\ell-a+b)!} \\
&\overset{\text{poly}}{\approx} \frac{1}{n^a} \cdot \ell^b \cdot \frac{(n-\ell)^a}{(n-\ell)^b} \\
&= \frac{\left(\frac{n}{2}\right)^a (1+\varepsilon)^a}{n^a} \cdot \frac{(1-\varepsilon)^b}{(1+\varepsilon)^b} \\
&= 2^{-a} \cdot (1+\varepsilon)^{a-2b} \cdot \exp(O(b \cdot \varepsilon^2)).
\end{aligned}$$

□

Chapter 5

Structural Results

This chapter shall be devoted to looking at some structural results on arithmetic circuits. This would help us understand the relevance of shallow circuits in the context of proving lower bounds for arithmetic circuits of arbitrary depth.

5.1 Homogenization

Suppose we have an n-variate degree d polynomial computed by an arithmetic circuit C. How large can the degree of intermediate computations be? Potentially, intermediate computations can involve very high degree terms which somehow cancel each other at the root. However, the following lemma of Strassen shows that we may assume without much loss of generality that arithmetic circuits never compute polynomials of degree more than the output.

Definition 5.1 (Homogeneous circuits)**.** *A circuit C is said to be* homogeneous *if every gate in the circuit computes a homogeneous polynomial.* $\diamondsuit$

Lemma 5.2 (Homogenization)**.** *Let f be an n-variate degree d polynomial computed by a circuit C of size s. Then, for every $0 \leq i \leq d$, there is a* homogeneous arithmetic circuit *C'_i, of size at most $O(sd^2)$, that computes the degree i homogeneous polynomial in f (the sum of all degree i monomials in f).*

Proof. Assume without loss of generality that the circuit C has all gates with fan-in at most 2. For every gate $g \in C$, define $(d+1)$ gates $g^{(0)}, \ldots, g^{(d)}$; we shall construct a new circuit C' such that $g^{(i)}$ computes the degree i homogeneous component of the polynomial computed at g. If g has children h_1 and h_2, then C' would have the following connections depending on the type of the gate g:

$$\begin{array}{rclcl} g = h_1 + h_2 & \implies & g^{(i)} & = & h_1^{(i)} \quad + \quad h_2^{(i)} \quad \text{for all } i \\ g = h_1 \times h_2 & \implies & g^{(i)} & = & \sum_{j=0}^{i} h_1^{(j)} h_2^{(i-j)} \quad \text{for all } i \end{array}$$

It is easy to check that the size of the circuit C' is at most $O(sd^2)$, and computes all the homogeneous components of f. □

Thus, for arithmetic circuits, we can assume without much loss of generality that we are working with a homogeneous circuit.

Remark. *For the class of arithmetic formulas, it is not clear if we can homogeneous without loss of generality. If we were to apply the above lemma to an arbitrary arithmetic formula, the resulting object is a homogeneous circuit and not a formula. It is unclear if any formula can be homogenized without loss of generality. The same is the case even for circuits of a fixed depth, as the above construction doubles the depth of the circuit.*

However, the class of ABPs can also be assumed to be homogeneous without loss of generality. We leave this as an exercise. ◊

Exercise 5.1 *Prove a similar homogenization lemma for algebraic branching programs.*

5.2 Interpolation

A very useful technique in arithmetic complexity is interpolation. Suppose we have a circuit C that computes a polynomial $P(x_1, \ldots, x_n, y)$ and say $\deg_y P = d$. We can interpret the polynomial P as an element of $(\mathbb{F}[\mathbf{x}])[y]$, that is we can think of P as a univariate polynomial in y with coefficients being elements of $\mathbb{F}[\mathbf{x}]$.

$$P(x_1, \ldots, x_n, y) \quad = \quad P_0(x_1, \ldots, x_n) + P_1(x_1, \ldots, x_n)y + \cdots + P_d(x_1, \ldots, x_n)y^d$$

Given that we can compute P by a circuit C, can we also compute one of the coefficients $P_i(x_1, \ldots, x_n)$ by a small circuit? The answer is 'Yes' as long as we are working over a large enough field.

Lemma 5.3 (Interpolation)**.** *Let $P(x_1, \ldots, x_n, y)$ be a polynomial with $\deg_y P \leq d$. For any set of distinct scalars $\alpha_0, \ldots, \alpha_d \in \mathbb{F}$ and for every $i \in \{0, \ldots, d\}$, each of the coefficient* *polynomials $P_i(x_1, \ldots, x_n)$ defined above can be expressed as a linear combination of*

$$\{P(x_1, \ldots, x_n, \alpha_0), \ldots, P(x_1, \ldots, x_n, \alpha_d)\}.$$

Thus in particular, if P is computed by a size s circuit, then each P_i is computable by a size $s(d+1)$ circuit. Furthermore, if P is computable by a size s circuit from some class $\mathcal{C}$, then each P_i is computable by a size $s(d+1)$ circuit from the class $\Sigma\mathcal{C}$ (which are circuits with a $+$ gate as a root with $\mathcal{C}$-circuits as its children).

Proof. For this proof we shall use $P(\alpha_i)$ as a short-hand for $P(x_1, \ldots, x_n, \alpha_i)$. Then we have the following matrix identity

$$\begin{bmatrix} 1 & \alpha_0 & \cdots & \alpha_0^d \\ 1 & \alpha_1 & \cdots & \alpha_1^d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_d & \cdots & \alpha_d^d \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_d \end{bmatrix} = \begin{bmatrix} P(\alpha_0) \\ P(\alpha_1) \\ \vdots \\ P(\alpha_d) \end{bmatrix}$$

Observe that the matrix on the left is a Vandermonde matrix, and hence is invertible. This implies that there is some matrix $((\beta_{ij}))$, its inverse, such that

$$\begin{bmatrix} P_0 \\ P_1 \\ \vdots \\ P_d \end{bmatrix} = \begin{bmatrix} \beta_{00} & \beta_{01} & \cdots & \beta_{0d} \\ \beta_{10} & \beta_{11} & \cdots & \beta_{1d} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{d0} & \beta_{d1} & \cdots & \beta_{dd} \end{bmatrix} \begin{bmatrix} P(\alpha_0) \\ P(\alpha_1) \\ \vdots \\ P(\alpha_d) \end{bmatrix}$$

In particular, for each $i \in \{0, \ldots, d\}$,

$$P_i(x_1, \ldots, x_n) = \beta_0 P(x_1, \ldots, x_n, \alpha_0) + \cdots + \beta_d P(x_1, \ldots, x_n, \alpha_d).$$

The size bounds and structure claimed is clear from the above expression. □

The use of interpolation is one of the most important methods when dealing with arithmetic circuits. We would many applications of it later in this article. We just give a couple of concrete applications for now. The first is the computation of partial derivatives in a single variable. We leave this as an easy exercise.

Exercise 5.2 [Computing partial derivatives] *Suppose $P(x_1, \ldots, x_n, y)$ be a polynomial computed by a size s circuit with $\deg_y P \leq d$. Show that for any i, the partial derivative $\frac{\partial^i P}{\partial y^i}$ can be computed by a circuit of size at most $s(d+1)$.*

What about computing mixed partials such as $\frac{\partial^n P}{\partial x_1 \cdots \partial x_n}$?

Another important application is the task of computing homogeneous components.

5.2.1 Computing homogeneous components

Suppose we have a size s circuit C that is computing a polynomial P of degree d. We have already seen in Lemma 5.2 that we can construct each of the homogeneous components of P by a circuit C' of size at most sd^2. Unfortunately, the construction described in Section 5.1 results in a circuit C' even if we started off with a formula computing P. Furthermore, if P was computable by a constant depth circuit, the construction results in an non-constant depth circuit. However, the construction in Lemma 5.2 not just computes the homogeneous components of P but it computes them via a *homogeneous* circuit. If the goal is to just compute the homogeneous components (by a possibly non-homogeneous circuit), one can use interpolation. Furthermore, this method would preserve constant depth as well.

Let us use $\mathrm{Hom}_i(P)$ to denote the homogeneous part of degree i, and let $d = \deg(P)$.

$$P(x_1, \ldots, x_n) \quad = \quad \mathrm{Hom}_0(P) + \cdots + \mathrm{Hom}_d(P)$$

Let y be a new variable and consider the polynomial $P'(x_1, \ldots, x_n, y) := P(yx_1, \ldots, yx_n)$. Then observe that

$$P'(x_1, \ldots, x_n) \quad = \quad \mathrm{Hom}_0(P) + y\,\mathrm{Hom}_1(P) + \cdots + y^d\,\mathrm{Hom}_d(P).$$

In other words, each homogeneous component of P is a coefficient of some y^i of $P'(x_1, \ldots, x_n, y)$. Thus, Lemma 5.3 tells us that we can use a linear combination of evaluations of P' to compute these coefficients.

We summarize this as a lemma.

Lemma 5.4 (Homogeneous component computations)**.** *Let $P(x_1, \ldots, x_n)$ be a polynomial with $\deg P = d$. For any set of distinct scalars $\alpha_0, \ldots, \alpha_d \in \mathbb{F}$ and for every $i \in \{0, \ldots, d\}$, each of the homogeneous components $\mathrm{Hom}_i(P)$ defined above can be expressed as a linear* *combination of*

$$\{P(\alpha_0 x_1, \ldots, \alpha_0 x_n), \ldots, P(\alpha_d x_1, \ldots, \alpha_d x_n)\}\,.$$

Thus in particular, if P is computed by a size s circuit, then each $\mathrm{Hom}_i(P)$ *is computable by a size $s(d+1)$ circuit. Furthermore, if P is computable by a size s circuit from some class $\mathcal{C}$, then each* $\mathrm{Hom}_i(P)$ *is computable by a size $s(d+1)$ circuit from the class $\Sigma\mathcal{C}$ (which are circuits with a $+$ gate as a root with $\mathcal{C}$-circuits as its children).*

It is important to note that for interpolation, we need to be able to find sufficiently many distinct elements in the base field, and this is possible only if the base field is large enough.

Exercise 5.3 *Assume the underlying field is large enough. Find a polynomial sized constant depth circuit that computes* Sym_d*, the elementary symmetric polynomial of degree d, defined as*

$$\mathsf{Sym}_d(x_1, \ldots, x_n) \quad = \quad \sum_{1 \leq i_1 < i_2 < \cdots < i_d \leq n} x_{i_1} \cdots x_{i_d}.$$

Exercise 5.4 *Assume the underlying field is large enough. Find a polynomial sized constant depth circuit that computes* complete homogeneous symmetric polynomial *defined as*

$$\sigma_d(x_1, \ldots, x_n) \quad = \quad \sum_{m \in \{\text{deg. } d \text{ monomials}\}} m.$$

For example, $\sigma_2(x_1, x_2, x_3) = x_1^2 + x_2^2 + x_3^2 + x_1x_2 + x_2x_3 + x_1x_3$.

5.3 Depth reduction

The phenomenon of simulating an arbitrary arithmetic circuit by a *shallow* arithmetic circuit is called *depth reduction.* Arithmetic circuits exhibit some remarkable depth reduction results, and we shall go over these in this section.

5.3.1 Depth reduction for arithmetic formulas

The depth reduction for formulas is quite easy to describe. This would also serve as step towards understanding the depth reduction for arithmetic circuits. The following depth reduction is due to Brent [Bre74].

Lemma 5.5 ([Bre74]). *Let f be an n-variate degree d polynomial computed by an arithmetic formula Φ of size s. Then, f can also be computed by a formula Φ' of size $s' = \mathrm{poly}(s, n, d)$ and depth $O(\log s)$.*

Proof. Assume without loss of generality that Φ is a formula of fan-in 2. Starting from the root, walk down to the leaves by always taking the child with a larger sub-tree under it. Consider the first node in this path v such that the size of the formula rooted at v is smaller than $\frac{2s}{3}$. Let Φ_v refer to the sub-formula rooted at v. By the choice of the path from the root, we have

$$\frac{s}{3} \quad \leq \quad |\Phi_v| \quad < \quad \frac{2s}{3}.$$

Let $\hat{\Phi}_v$ denote the formula where the sub-formula at v is replaced by a fresh variable y. Since we are dealing with formulas, $\hat{\Phi}_v$ is a linear polynomial in the variable y. Hence,

$$\begin{aligned} \hat{\Phi}_v(y) &= A \cdot y \quad + \quad B \\ \text{and,} \quad \Phi &= A \cdot \Phi_v \quad + \quad B \end{aligned}$$

for some polynomials A and B. But we can compute both A and B from $\hat{\Phi}_v(y)$ as

$$\begin{aligned} A &= \hat{\Phi}_v(1) - \hat{\Phi}_v(0) \\ B &= \hat{\Phi}_v(0) \end{aligned}$$

Thus,

$$f \quad = \quad (\hat{\Phi}_v(1) - \hat{\Phi}_v(0)) \cdot \Phi_v \quad + \quad \hat{\Phi}_v(0)$$

All the formulas (i.e. $\hat{\Phi}_v(1)$, $\hat{\Phi}_v(0)$, and Φ_v) in the above equation have size at most $\frac{2s}{3}$. Thus, by recursively applying this process on each of these sub-formulas, we obtain

$$\mathrm{Depth}(s) \quad = \quad \mathrm{Depth}(2s/3) \quad + \quad 3$$

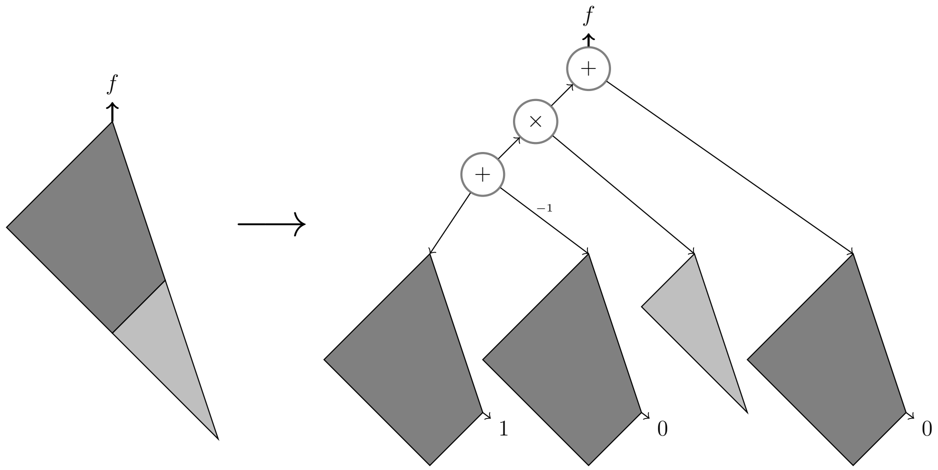


Figure 5.1: Depth reduction for formulas

$$\begin{aligned} \implies \text{Depth}(s) &= O(\log s) \\ \text{Size}(s) &\le 4 \cdot \text{Size}(2s/3) \quad + \quad O(1) \\ \implies \text{Size}(s) &= \text{poly}(s). \end{aligned}$$

□

5.3.2 Depth reduction for arithmetic circuits

The key point in the above depth reduction was that for any node v, the formulas Φ_v and $\hat{\Phi}_v(y)$ were disjoint. This however is not the case for general arithmetic circuits. Thus, it is not clear if we can find a node in the circuit such that the subcircuit under it has size between $s/3$ and $2s/3$. However, we do not really need to make the subcircuits have size drop by a constant factor, but any parameter dropping by a constant factor would be fine. One parameter that we could work with instead is the *degree*.

Applying Brent's reduction with degree

By Lemma 5.2, we may assume that we have a homogeneous circuit Φ if size s computing a homogeneous n-variate polynomial f of degree d. Using a similar argument as in the proof of Lemma 5.5, we can find a node $v \in \Phi$ such that $\frac{d}{3} < \deg(v) \le \frac{2d}{3}$. However, we cannot quite write f as $A \cdot \Phi_v + B$ as we are now dealing with a circuit and there could be multiple paths from the root leading to v.

Consider the set of all nodes of such intermediate degree as $\mathcal{F}$:

$$\mathcal{F} = \left\{ v \in \Phi \ : \ \frac{d}{3} < \deg(v) \leq \frac{2d}{3} \right\}$$

Instead of expressing f using a single $v \in \mathcal{F}$ as in Lemma 5.5, we shall express f as a function of *all* nodes in $\mathcal{F}$.

Claim 5.6. *If $\mathcal{F} = \{v_1, \ldots, v_s\}$, then f may be written as*

$$f \quad = \quad \sum_{i,j} A_{ij} \cdot \Phi_{v_i} \Phi_{v_j} \quad + \quad \sum_i B_i \cdot \Phi_{v_i} \tag{5.7}$$

where $\deg(A_{ij}), \deg(B_i) \leq \frac{2d}{3}$ for all i, j. Moreover, each A_{ij} and B_i's may be computed by an arithmetic circuit of size at most $O(s)$.

Proof. From the circuit Φ, construct the circuit Φ' that is obtained by removing the incoming edges of every $v_i \in \mathcal{F}$ thereby making these nodes as leaves as well. Then, Φ' computes a polynomial $f'(x_1, \ldots x_n, v_1, \ldots, v_s)$ satisfying

$$f \quad = \quad f'(x_1, \ldots, x_n, \Phi_{v_1}, \ldots, \Phi_{v_s}).$$

Because of the degree of each v_i, we easily obtain that the degree of f' in the v_i variables must be at most 2, and each coefficient A_{ij} or B_i cannot have degree more than $2d/3$. Further, obtaining the A_{ij}'s and B_i's from Φ' is a simple exercise. □

Since every polynomial appearing in (5.7) is computable by size at most $\mathrm{poly}(s)$ and has degree at most $2d/3$, we may apply induction as earlier. Thus,

$$\begin{aligned} \mathrm{Depth}(d) &= \mathrm{Depth}(2d/3) \quad + 2 \\ \Longrightarrow \mathrm{Depth}(d) &= O(\log d) \end{aligned}$$

Unfortunately, the size of the resulting circuit could be as large as $s^{O(\log d)}$. This reduction is along the lines of Hyafil [Hya79].

Notice that in this reduction, the final circuit we obtain is in fact an arithmetic formula of size $s^{O(\log d)}$ and depth $O(\log d)$ (assuming that addition gates can have unbounded fan-in; with bounded fan-in addition gates, the depth would be $O(\log d \cdot \log s)$). Valiant, Skyum Berkowitz and Rackoff [VSBR83] showed that we can attain a similar depth reduction to $O(\log d)$ depth while keeping the size polynomial.

Depth reduction of [VSBR83]

This section shall be devoted to the proof of the remarkable theorem of Valiant, Skyum, Berkowitz and Rackoff.[1]

Theorem 5.8 ([VSBR83, AJMV98])**.** *Let f be an n-variate degree d polynomial computed by an arithmetic circuit Φ of size s. Then there is an arithmetic circuit Φ' computing f and has size $s' = \mathrm{poly}(s, n, d)$ and depth $O(\log d)$.*

We may assume without loss of generality that Φ is a homogeneous circuit. We will also assume that all multiplication gates in Φ have fan-in at most 2, and that the degree of the right child of any multiplication gate is at least as large as the degree of the its left child. Such circuits are known to as *right heavy* circuits. For any gate u in Φ, we denote by $[u]$ the polynomial computed at gate u. It will also denote a gate label in the new depth reduced circuit.

We need the following definition of gate quotients.

Definition 5.9. *For any pair of gates u, v, the polynomial $[u : v]$ is defined as follows:*

1. *If u and v are the same nodes, then $[u : v] = 1$.*

2. *If u is a leaf, and $u \neq v$, then $[u : v] = 0$.*

3. *If $u = u_1 + u_2$, then $[u : v] = [u_1 : v] + [u_2 : v]$.*

4. *If $u = u_1 \times u_2$, then $[u : v] = [u_1] \cdot [u_2 : v]$.* ♢

It is easy to see that $[u : v]$ is a homogeneous polynomial of degree $\deg(u) - \deg(v)$.

Some Intuition

For an arithmetic formula Φ with u as root, and any other node v, we can write $\Phi = [u]$ as $A \cdot [v] + B$ for some polynomials A and B. We would like to denote $[u : v]$ to denote the quotient A. In a circuit things get complicated due to multiple paths from the u to v.

A minimal computation in a circuit is formalized by the notion of a *proof-tree*. A proof-tree is a sub-circuit T such that, 1. the root is in T. 2. if a multiplication gate v is in T, then so are both its children. And, 3. if an addition gate v is in T, then exactly one child of v is also in T.

[1] The proof described here follows the structure of a subsequent result [AJMV98], and not the original proof, although both proofs are quite similar.

Every proof-tree computes a monomial, and the polynomial computed by the circuit is just the sum over all proof-trees. The technical issue in defining the gate quotient stems from the fact that a node v could occur multiple times in a proof-tree, and the number of occurrences could also vary with different proof-trees. To consistently define the gate quotient, we need to be careful which of these v's we are referring to. We do this by defining the right-most path in the proof-tree as a canonical path, and replacing the unique occurrence of v on this canonical path by a leaf labelled 1 (and if v does not occur on this path, then this proof-tree does not contribute to $[u : v]$). In fact, it can be seen that Definition 5.9 is precisely this notion although stated algebraically, and this was the perspective used in [AJMV98]. Although it provides more intuition, we will not use the notion of proof-trees any further to prove Theorem 5.8.

A possible alternate definition is to interpret u as a polynomial in v, and take the first-order partial derivative (as described in [SY10]). In the case when $\deg(v) > \deg(u)/2$, this notion coincides with the above definition of $[u : v]$ (as v cannot occur more than once in any proof-tree). However, some of key properties (especially Lemma 5.12 that we shall soon see) are not true in this setting unless similar degree restrictions are placed on the pair of nodes. While we could reprove Theorem 5.8 in this way, we need to be careful about these degree restrictions.

This proof described here can be thought of as a hybrid of [VSBR83] and [AJMV98]. The circuit is built top-down along the lines of [AJMV98] but using the language of [VSBR83].

Definition 5.10 (Frontier). *For any parameter m, define the* frontier at degree m *as*

$$\mathcal{F}_m \quad = \quad \{v \ : \ \deg(v) \geq m \ , \ \deg(v_L), \deg(v_R) < m\}$$

That is, $\mathcal{F}_m$ are the deepest nodes in the circuit that have degree at least m. ◊

Note that from the above definition, all frontier nodes are multiplication gates (since we are working with a homogeneous circuit). Also, a frontier forms a maximal anti-chain.

Observation 5.11. *If a node v does not occur in the sub-circuit rooted at u, then $[u : v] = 0$. In particular, if $u, v \in \mathcal{F}_m$ for some m, and $u \neq v$, then $[u : v] = 0$.*

The following is the key lemma for the depth reduction.

Lemma 5.12. *Suppose Φ is a homogeneous, right-heavy circuit. Let m be a parameter such* *that* $\deg(u) \geq m$. *Then,*

$$[u] \quad = \quad \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w] \tag{5.13}$$

Also, if u, v *are nodes such that* $\deg(u) \geq m > \deg(v)$, *then*

$$[u : v] \quad = \quad \sum_{w \in \mathcal{F}_m} [u : w][w : v] \tag{5.14}$$

Proof. The proof would be by induction on the depth of u.

1. The base case would be when $\deg(u) \geq m$ but all its children have degree less than m, that is $u \in \mathcal{F}_m$. Such a node u has to be a × gate. Hence,

$$\sum_{w \in \mathcal{F}_m} [u : w] \cdot [w] \quad = \quad [u : u] \cdot [u] \quad + \quad \sum_{\substack{w \in \mathcal{F}_m \\ w \neq u}} [u : w] \cdot [w] \quad = \quad [u] \quad + \quad 0$$

and

$$\sum_{w \in \mathcal{F}_m} [u : w] \cdot [w : v] \quad = \quad [u : u] \cdot [u : v] \quad + \quad \sum_{\substack{w \in \mathcal{F}_m \\ w \neq u}} [u : w] \cdot [w : v] \quad = \quad [u : v] \quad + \quad 0$$

as $[u : w] = 0$ by Observation 5.11.

2. If $u = u_1 + u_2$,

$$\begin{aligned} [u] &= [u_1] + [u_2] \\ &= \sum_{w \in \mathcal{F}_m} \left([u_1 : w] \cdot [w] + [u_2 : w] \cdot [w] \right) \quad \text{(inductive hypothesis)} \\ &= \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w] \quad \text{(Definition 5.9.3)} \end{aligned}$$

and

$$\begin{aligned} [u : v] &= [u_1 : v] + [u_2 : v] \\ &= \sum_{w \in \mathcal{F}_m} \left([u_1 : w] \cdot [w : v] + [u_2 : w] \cdot [w : v] \right) \quad \text{(inductive hypothesis)} \end{aligned}$$

$$= \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w : v] \quad \text{(Definition 5.9.3)}$$

3. If $u = u_1 \times u_2$ with $\deg(u_2) \geq m$,

$$\begin{aligned} [u] &= [u_1] \cdot [u_2] \\ &= [u_1] \cdot \left(\sum_{w \in \mathcal{F}_m} [u_2 : w] \cdot [w] \right) \quad \text{(inductive hypothesis)} \\ &= \sum_{w \in \mathcal{F}_m} ([u_1] \cdot [u_2 : w]) \cdot [w] = \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w] \quad \text{(Definition 5.9.4)} \end{aligned}$$

$$\begin{aligned} [u : v] &= [u_1] \cdot [u_2 : v] \\ &= [u_1] \cdot \left(\sum_{w \in \mathcal{F}_m} [u_2 : w] \cdot [w : v] \right) \quad \text{(inductive hypothesis)} \\ &= \sum_{w \in \mathcal{F}_m} ([u_1] \cdot [u_2 : w]) \cdot [w : v] = \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w : v] \quad \text{(Definition 5.9.4)} \end{aligned}$$

□

Now we are ready to write down the depth reduced circuit. As mentioned earlier, the original proof of [VSBR83] follows a bottom-up approach, but it would be more useful to us to take a top-down approach (as in [AJMV98]) to obtain some additional structural properties that we would require.

Theorem 5.15 (Theorem 5.8 restated)**.** *Let Φ be a homogeneous, right-heavy circuit of size s computing an n-variate degree d polynomial. Then, there is a circuit Φ' of size* $\mathrm{poly}(s)$ *with the following properties:*

1. *For every pair of nodes $u, v \in \Phi$, there are nodes in Φ' computing $[u]$ and $[u : v]$.*
2. *For every multiplication gate in Φ', all its children have at most half its degree.*

Proof. We shall recursively compute each node $[u]$ and $[u : v]$ from nodes of lower degree.

For any node $u \in \Phi$, let $\mathcal{F}(u) = \mathcal{F}_m$, where $m = \deg(u)/2$. Thus by (5.13),

$$[u] = \sum_{w \in \mathcal{F}(u)} [u : w] \cdot [w]$$

$$= \sum_{w \in \mathcal{F}(u)} [u : w] \cdot [w_L] \cdot [w_R]$$

By our choice of m, all the terms on the RHS have degree at most $\deg(u)/2$. Also, $[u]$ is an addition gate with fan-in s and the multiplication gates feeding into it have fan-in 3.

For any pair of nodes $u, v \in \Phi$, let $\mathcal{F}(u, v) = \mathcal{F}_m$, where $m = (\deg(u) + \deg(v))/2$. By (5.14),

$$\begin{aligned} [u : v] &= \sum_{w \in \mathcal{F}(u,v)} [u : w] \cdot [w : v] \\ &= \sum_{w \in \mathcal{F}(u,v)} [u : w] \cdot [w_L] \cdot [w_R : v] \end{aligned}$$

Again by the choice of m, the degree of $[u : w]$ and the degree of $[w_R : v]$ is at most $(\deg(u) - \deg(v))/2$. The degree of $[w_L]$ however could be as large as $\deg(u) - \deg(v)$. Nevertheless, we can use the above expansion once more to write it as

$$\begin{aligned} [u : v] &= \sum_{w \in \mathcal{F}(u,v)} [u : w] \cdot [w_L] \cdot [w_R : v] \\ &= \sum_{w \in \mathcal{F}(u,v)} [u : w] \left(\sum_{p \in \mathcal{F}(w_L)} [w_L : p] \cdot [p_L] \cdot [p_R] \right) \cdot [w_R : v] \\ &= \sum_{w \in \mathcal{F}(u,v)} \sum_{p \in \mathcal{F}(w_L)} [u : w] \cdot [w_L : p] \cdot [p_L] \cdot [p_R] \cdot [w_R : v] \end{aligned}$$

Now all the terms on the RHS have degree at most $(\deg(u) - \deg(v))/2$ as required. Also, $[u : v]$ is an addition gate with fan-in s^2 and the multiplication gates feeding into it have fan-in 5.

Eventually we shall reach a case where $\deg(u) \leq 1$ or $\deg([u : v]) \leq 1$. These are just linear polynomials over n variables and shall be explicitly computed in Φ'.

Starting with the output gate, it is clear how these steps can be used to build a depth reduced circuit in a top-down fashion. □

Observe that the proof also shows that all addition gates in Φ' have fan-in at most s^2 and all multiplication gates have fan-in at most 5. This completes the list of properties we seek from the depth reduced circuit.

5.3.3 Reduction to depth four circuits

One of the consequence of a depth reduction such as Theorem 5.8 is that proving lower bounds for general circuits is reduced to the task of proving lower bounds for $O(\log d)$ depth circuits.

Corollary 5.16. *If f is an n-variate degree d polynomial that requires super-polynomial (in n and d) size circuits of $O(\log d)$ depth to compute it, then any general arithmetic circuit computing f must also be of super-polynomial size.*

But, optimistically, we would expect that the *right* lower bound must be truly exponential, and not merely super-polynomial. Keeping that in mind, a depth reduction even with a slightly super-polynomial blow-up might be useful in this regard. This line was first pursued by Agrawal and Vinay [AV08], and the result was subsequently strengthened by Koiran [Koi12] and Tavenas [Tav15].

Theorem 5.17 ([AV08, Koi12, Tav15])**.** *Let f be an n-variate degree d polynomial computed by a size s arithmetic circuit. Then for any $0 < t \leq d$, f can be equivalently computed by a homogeneous $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of top fan-in $s^{O(d/t)}$ and size $s^{O(t+d/t)}$.*

If we were to optimize the size of the final depth four circuit, then we should choose $t = \sqrt{d}$ to get a $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of size $s^{O(\sqrt{d})}$. Note that this implies that if we could prove a lower bound of $n^{\omega(\sqrt{d})}$ for such $\Sigma\Pi\Sigma\Pi^{[\sqrt{d}]}$ circuits, then we would have proved a lower bound for general circuits! In fact, in the recent past, we have come pretty close to the required threshold and we shall see them in the later chapters.

In this section, we shall see a proof of Theorem 5.17 but this is not the original proof of Tavenas [Tav15]. We shall see an alternate proof by [SV14], which I find more insightful.

Proof of Theorem 5.17. Let C be the $O(\log d)$ depth circuit computing f obtained from Theorem 5.8 applied on the size s circuit computing f. Let s' be the size of C. If g is a polynomial computed at any intermediate node of C, then from the structure of C (Theorem 5.15) we have a homogeneous expression

$$g = \sum_{i=1}^{s'} g_{i1} \cdot g_{i2} \cdot g_{i3} \cdot g_{i4} \cdot g_{i5} \qquad (5.18)$$

where each g_{ij} is computed by a node in C as well, and $\deg(g_{ij}) \leq \deg(g)/2$. If we look at

(5.18) for f, then the RHS is a $\Sigma\Pi\Sigma\Pi^{[d/2]}$ circuit of top fan-in s' computing f. To obtain a $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit eventually, we shall do the following natural process.

> For each summand $g_{i1}\dots g_{ir}$ in the RHS, if the largest degree g_{ij} has degree more than t, expand that g_{ij} in-place using (5.18).
>
> Repeat this process until all g_{ij}'s on the RHS have degree at most t.

Note that in each iteration of the above procedure, we increase the top fan-in by a multiplicative factor of s', and what we gain is that some the terms in the RHS would now have smaller degrees. If we could show that the in $O(d/t)$ iterations all terms on the RHS have degree at most t, then we would have obtained an $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of top fanin $s'^{O(d/t)}$ computing f.

To bound the number of iterations, let us count the number of terms of degree more than $t/8$ in each term. Note that since we would always maintain homogeneity, the number of terms of degree $t/8$ or more in any summand is at most $8d/t$. Thus, it suffices to show that each iteration increases the number of terms of degree $t/8$ by at least one.

Note that in (5.18), if $\deg(g) = d'$ then the largest degree term of any summand on the RHS is at least $d'/5$ (since the sum of the degrees of the five terms must add up to d'). Also, the largest degree term can have degree at most $d'/2$. Hence there must be at least $d'/2$ degree contributed by the other four factors in each term. This implies that the second largest factor in each summand has degree at least $d'/8$. Therefore, as long as we are expanding factors using (5.18) of degree more than $t/8$, we are guaranteed that each new term has at least one more factor of degree more than $t/8$. As argued earlier, we can never have more than $8d/t$ such terms in any summand and this bounds the number of iterations by $8d/t$.

Thus, when the above procedure stops, we have an $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit of top fan-in $s'^{O(8d/t)} = s^{O(d/t)}$. Observing that any polynomial of degree t can have at most n^t monomials, we get that the size of the circuit overall is at most $s^{O(t+d/t)}$. □

Thus, proving a "good enough" top fanin (or size) lower bound for the class of $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit would suffice for proving lower bounds for general circuits. We would be using this fact quite a lot so we state this explicitly as a corollary.

Corollary 5.19. *If f is an n-variate degree d polynomial that requires homogeneous $\Sigma\Pi\Sigma\Pi^{[t]}$ circuits of top fan-in $n^{\omega(d/t)}$ to compute it, then f requires general arithmetic circuits of size $n^{\omega(1)}$ to compute it.*

Exercise 5.5 *Define the product-depth of any circuit to be the maximum number of multiplication gates encountered on any root-to-leaf path.*

Show that for any polynomial n-variate degree d polynomial f that can be computed by a size s arithmetic circuit, there is a circuit Φ' of size $s^{O(d^{1/\Delta})}$ and product depth Δ computing f.

Reduction to depth three

There have been some further depth reductions results.

Theorem 5.20 ([GKKS16])**.** *If f is an n-variate degree d polynomial in $\mathbb{Q}[\mathbf{x}]$ that can be computed by an arithmetic circuit of size s, then it can be equivalently computed by a depth three circuit of size $s^{O(\sqrt{d})}$.*

We shall defer this theorem to later in the interest of presenting more insight and intuition. They would be better placed after we have seen a few of the recent lower bounds for restricted depth four circuits (but those who are impatient can find it in Section 23.2). We now proceed to see some lower bounds.

5.3.4 Depth reduction for formulas, again

In this section we shall revisit Theorem 5.17 when it is applied to formulas. Do the resulting depth four circuits have any additional structure when we start from a homogeneous formula? The alternate proof of Saptharishi and Vinay [SV14] provides some insight into this.

We would need the following lemma that was present in the survey of Shpilka and Yehudayoff [SY10] and also in the result of Hrubeš and Yehudayoff [HY11] that we shall see a proof of in Chapter 13 as Lemma 13.3 and Lemma 13.4.

Lemma 5.21 (Lemma 13.4 stated without proof)**.** *Let Φ be a homogeneous formula of size s computing a polynomial p of degree d. Then p can be written as a sum of $(s+1)$* log-product *polynomials, that is,*

$$p \quad = \quad f_1 + \cdots + f_r \qquad \textit{with } r \leq s+1 \tag{5.22}$$

where for each $i \in [r]$, we have $f_i = f_{i1} \cdots f_{i\ell}$ satisfying

- *each f_{ij} is homogeneous, and $\sum_j \deg(f_{ij}) = d$,*

- $(1/3)^j \cdot d \leq \deg(f_{ij}) \leq (2/3)^j \cdot d$,
- $f_{i\ell} = 1$.

In particular, each f_i factors into $\Omega(\log d)$ non-trivial factors of geometrically decreasing degrees.

Furthermore, if Φ was a multilinear formula to begin with, then so is the expression on the RHS. And if Φ was set-multilinear, then so is the expression on the RHS. In other words, it says that a homogeneous formula has a homogeneous depth-4 formula of the form $\Sigma^{[s+1]}\Pi^{[\log d]}\Sigma\Pi^{[2d/3]}$. We can now describe the mild extension of Theorem 5.17 applied for homogeneous formulas.

Theorem 5.23 (Theorem 5.17 for homogeneous formulas)**.** *Let f be a homogeneous n-variate degree d polynomial computed by a size s homogeneous formula. Then for any $0 < t \leq d$, f can be equivalently computed by a homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[t]}$ formula of top fan-in $s^{10(d/t)}$ where*

$$a \geq \frac{1}{10}\left(\frac{d}{t}\right)\log t.$$

What this means is that even though the degree of all polynomials computed by the bottom two levels of the $\Sigma\Pi\Sigma\Pi$ circuit have degree bounded by t, each summand is a product of much more than d/t factors. Another way to view this is that Theorem 5.17 gives a $\Sigma\Pi\Sigma\Pi$ circuit of size $s^{O(d/t)}$ where the maximum bottom degree and *average bottom degree* are both bounded by $O(t)$. Whereas in the above theorem we have maximum bottom degree bounded by t but *average bottom degree* bounded by $(t/\log t)$.

Proof. The proof is exactly along the lines of Theorem 5.17 but instead of using the 5-product expression in (5.18), we shall use the log-product expression of (5.22). The key point again is that in every such summand, there are two factors of degree at least $d/9$. Therefore, the proof of Theorem 5.17 proceeds verbatim and the number of iterations is bounded by $9(d/t)$. This gives the $\Sigma\Pi\Sigma\Pi^{[t]}$ formula of size at most $(s+1)^{9(d/t)} \leq s^{10(d/t)}$. The only thing left to check is that we do indeed have $\Omega((d\log t)/t)$ non-trivial factors in each summand. We leave this as an exercise with a hint. □

Exercise 5.6 *Complete the proof of Theorem 5.23 by showing that the number of non-trivial factors in any summand of the resulting $\Sigma\Pi\Sigma\Pi^{[t]}$ circuit has $\Omega((d\log t)/t)$ non-*

trivial factors.

Hint: *Show that in* (5.22)*, in any summand, if we only consider the factors of degree at most t, the sum of their degrees is at most $2t$. Use that to say each term must involve $\Omega(d/t)$ expansions via* (5.22) *thus yielding $\Omega((d \log t)/t)$ factors.*

This additional structure may be useful in the quest for proving homogeneous formula lower bounds but at the moment it seems unclear. We shall however see one application of this additional structure later in Chapter 16.

Part II

Classical lower bounds

Chapter 6

Lower bounds for general circuits and formulas

Despite several attempts by various researchers to prove lower bounds for arithmetic circuits or formulas, we only have very mild lower bounds for general circuits or formulas thus far. In this chapter, we shall look at the two modest lower bounds for general circuits and formulas.

6.1 Lower bounds for general circuits

The only super-linear lower bound we currently know for general arithmetic circuits is the following result of Baur and Strassen [BS83].

Theorem 6.1 ([BS83])**.** *Any fan-in 2 circuit that computes the polynomial $f = x_1^{d+1} + \cdots + x_n^{d+1}$ has $\Omega(n \log d)$ edges.*

6.1.1 An exploitable weakness

Without loss of generality, let us assume that the circuit is a fan-in 2 circuit. This would allow us to talk in terms of the number of nodes instead of edges.

Each gate of the circuit Φ computes a local operation on the two children. To formalize this, define a new variable y_g for every gate $g \in \Phi$. Further, for every gate g define a quadratic equation Q_g as

$$Q_g = \begin{cases} y_g - (y_{g_1} + y_{g_2}) & \text{if } g = g_1 + g_2 \\ y_g - (y_{g_1} \cdot y_{g_2}) & \text{if } g = g_1 \cdot g_2. \end{cases}$$

Further if y_o corresponds to the output gate, then the system of equations

$$\{Q_g = 0 \ : \ g \in \Phi\} \quad \cup \quad \{y_o = 1\}$$

completely characterize the computations of Φ that results in an output of 1.

The same can also be extended for *multi-output* circuits that compute several polynomials simultaneously. In such cases, the set of equations

$$\{Q_g = 0 \ : \ g \in \Phi\} \quad \cup \quad \{y_{o_i} = 1 \ : \ i = 1, \ldots, n\}$$

completely characterize computations that result in an output of all ones. The following classical theorem allows us to bound the number of common roots to a system of polynomial equations.

Theorem 6.2 (Bézout's theorem)**.** *Let $g_1, \ldots, g_r \in \mathbb{F}[X]$ and $\deg(g_i) = d_i$ such that the number of common roots of $g_1 = \cdots = g_r = 0$ is finite. Then, the number of common roots (counted with multiplicities) is bounded by $\prod d_i$.*

Thus in particular, if we have a circuit Φ of size s that *simultaneously* computes $\{x_1^d, \ldots, x_n^d\}$, then we have d^n inputs that evaluate to all ones (where each x_i must be a d-th root of unity). Hence, Bézout's theorem implies that

$$2^s \quad \geq \quad d^n \qquad \Longrightarrow \qquad s \quad = \quad \Omega(n \log d).$$

Observe that $\{x_1^d, \ldots, x_n^d\}$ are all first-order derivatives of $f = x_1^{d+1} + \cdots + x_n^{d+1}$ (with suitable scaling). A natural question here is the following — if f can be computed an arithmetic circuit of size s, what is the size required to compute all first-order partial derivatives of f simultaneously? The naïve approach of computing each derivative separately results in a circuit of size $O(s \cdot n)$. Baur and Strassen [BS83] show that we can save a factor of n.

Lemma 6.3 ([BS83])**.** *Let Φ be an arithmetic circuit of size s and fan-in 2 that computes a polynomial $f \in \mathbb{F}[X]$. Then, there is a multi-output circuit of size $O(s)$ computing all first order derivatives of f.*

Note that this immediately implies that any circuit computing $f = x_1^{d+1} + \cdots + x_n^{d+1}$ requires size $\Omega(n \log d)$ as claimed by Theorem 6.1.

6.1.2 Computing all first order derivatives simultaneously

Since we are working with fan-in 2 circuits, the number of edges is at most twice the size. Hence let s denote the number of edges in the circuit Φ. We shall prove by induction that all first order derivatives of Φ can be computed by a circuit of size at most $5s$. Pick a non-leaf node v in the circuit Φ closest to the leaves with both its children being variables, and let x_1 and x_2 are the variables feeding into v. In other words, $v = x_1 \odot x_2$ where $\odot$ is either $+$ or $\times$.

Let Φ' be the circuit obtained by deleting the two edges feeding into v, and replacing v by a new variable. Hence, Φ' computes a polynomial $f' \in \mathbb{F}[X \cup \{v\}]$ and has at most $(s-2)$ edges. By induction on the size, we can assume that there is a circuit $\mathbb{D}(\Phi')$ consisting of at most $5(s-2)$ edges that computes all the first order derivatives of f'.

Observe that since $f' \mid_{(v=x_1 \odot x_2)} = f(\mathbf{x})$, we have that

$$\frac{\partial f}{\partial x_i} = \left(\frac{\partial f'}{\partial x_i}\right)_{v=x_1\odot x_2} + \left(\frac{\partial f'}{\partial v}\right)_{v=x_1\odot x_2} \left(\frac{\partial(x_1 \odot x_2)}{\partial x_i}\right).$$

Hence, if $v = x_1 + x_2$ then

$$\begin{aligned}
\frac{\partial f}{\partial x_1} &= \left(\frac{\partial f'}{\partial x_1}\right)_{v=x_1+x_2} + \left(\frac{\partial f'}{\partial v}\right)_{v=x_1+x_2} \\
\frac{\partial f}{\partial x_2} &= \left(\frac{\partial f'}{\partial x_2}\right)_{v=x_1+x_2} + \left(\frac{\partial f'}{\partial v}\right)_{v=x_1+x_2} \\
\frac{\partial f}{\partial x_i} &= \left(\frac{\partial f'}{\partial x_i}\right)_{v=x_1+x_2} \qquad \text{for } i > 2.
\end{aligned}$$

If $v = x_1 \cdot x_2$, then

$$\begin{aligned}
\frac{\partial f}{\partial x_1} &= \left(\frac{\partial f'}{\partial x_1}\right)_{v=x_1\cdot x_2} + \left(\frac{\partial f'}{\partial v}\right)_{v=x_1\cdot x_2} \cdot x_2 \\
\frac{\partial f}{\partial x_2} &= \left(\frac{\partial f'}{\partial x_2}\right)_{v=x_1\cdot x_2} + \left(\frac{\partial f'}{\partial v}\right)_{v=x_1\cdot x_2} \cdot x_1 \\
\frac{\partial f}{\partial x_i} &= \left(\frac{\partial f'}{\partial x_i}\right)_{v=x_1\cdot x_2} \qquad \text{for } i > 2.
\end{aligned}$$

Hence, by adding at most 10 additional edges (see Figure 6.1, with the additional edges marked in red) to $\mathbb{D}(\Phi')$, we can construct $\mathbb{D}(\Phi)$ with at most $5s$ edges. □(Lemma 6.3)

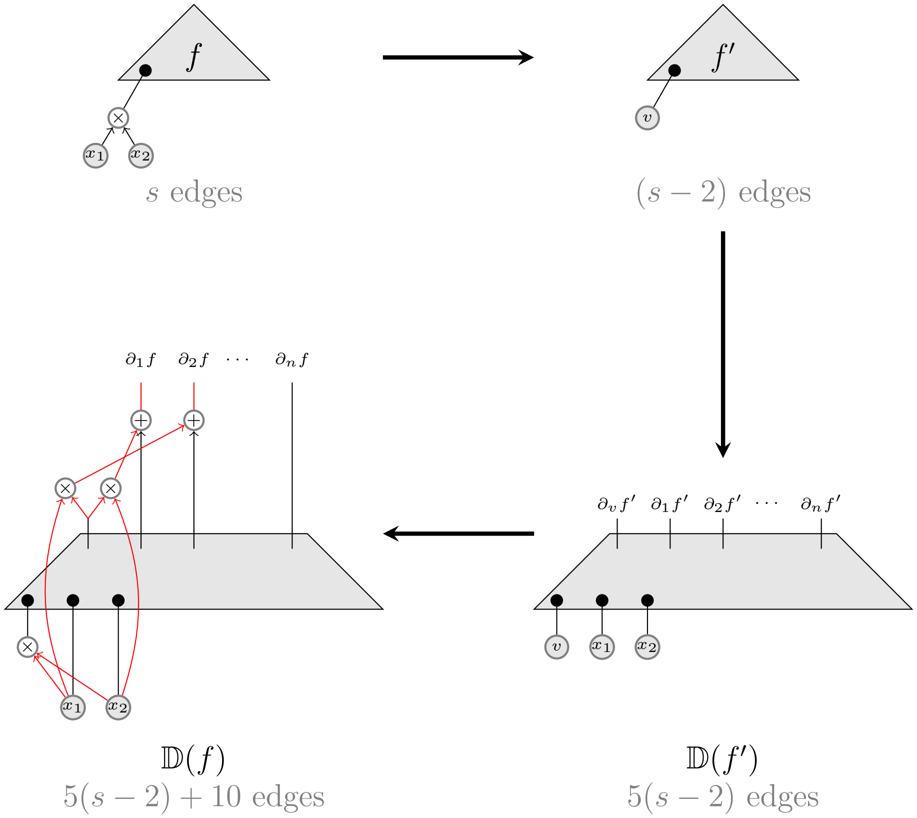


Figure 6.1: Baur-Strassen pictorially for $v = x_1 \times x_2$

6.2 Lower bounds for formulas

This section would be devoted to the proof of Kalorkoti's lower bound [Kal85] for formulas computing Det_n or Perm_n.

Theorem 6.4 ([Kal85]). *Any arithmetic formula computing* Perm_n *(or* Det_n*) requires* $\Omega(n^3)$ *size.*

Although this is an $\Omega(n^{3/2})$ lower bound, where n is the number of variables, the same technique can also give an $\Omega(n^2)$ lower bound for the following polynomial.

Theorem 6.5 ([Kal85]). *Any arithmetic formula computing* $\sum_{i=1}^{n}\sum_{j=1}^{n} x_i^j y_j$ *requires* $\Omega(n^2)$ *size.*

In this section, we shall prove the lower bound for Det_n and Perm_n and leave Theorem 6.5 as an exercise.

The exploitable weakness in this setting is again to use the fact that the polynomials computed at intermediate gates share many polynomial dependencies.

Definition 6.6 (Algebraic independence)**.** *A set of polynomials $\{f_1, \ldots, f_m\}$ is said to be* algebraically independent *if there is no non-trivial polynomial $H(z_1, \ldots, z_m)$ such that $H(f_1, \ldots, f_m) = 0$.*

The size of the largest algebraically independent subset of $\mathbf{f} = \{f_1, \ldots, f_m\}$ is called the transcendence degree *(denoted by* $\mathrm{trdeg}(f)$*).* ◊

The proof of Kalorkoti's theorem proceeds by defining a *complexity measure* using the above notion of algebraic independence.

The Measure: For any subset of variables $Y \subseteq X$, we can write a polynomial $f \in \mathbb{F}[X]$ of the form $f = \sum_{i=1}^{s} f_i \cdot m_i$ where m_i's are distinct monomials in the variables in Y, and $f_i \in F[X \setminus Y]$. We shall denote by $\mathrm{td}_Y(f)$ the transcendence degree of $\{f_1, \ldots, f_s\}$

Fix a partition of variables $X = X_1 \sqcup \cdots \sqcup X_r$. For any polynomial $f \in \mathbb{F}[X]$, define the map $\Gamma^{[\mathrm{Kal}]} : \mathbb{F}[X] \to \mathbb{Z}_{\geq 0}$ as

$$\Gamma^{[\mathrm{Kal}]}(f) \quad = \quad \sum_{i=1}^{r} \mathrm{td}_{X_i}(f).$$

The lower bound proceeds in two natural steps:

1. Show that $\Gamma^{[\mathrm{Kal}]}(f)$ is *small* whenever f is computable by a *small* formula.

2. Show that $\Gamma^{[\mathrm{Kal}]}(\mathsf{Det}_n)$ is *large.*

6.2.1 Upper bounding $\Gamma^{[\mathrm{Kal}]}$ for a formula

Lemma 6.7. *Let f be computed by a fan-in two formula Φ of size s. Then for any partition of variables $X = X_1 \sqcup \cdots \sqcup X_r$, we have $\Gamma^{[\mathrm{Kal}]}(f) = O(s)$.*

Proof. For any node $v \in \Phi$, let $\textsc{Leaf}(v)$ denote the leaves of the subtree rooted at v and let $\textsc{Leaf}_{X_i}(v)$ denote the leaves of the subtree rooted at v that are in the set X_i. Since the underlying graph of Φ is a tree, it follows that the size of Φ is bounded by a twice the number of leaves. For each set X_i, we shall show that $\mathrm{td}_{X_i}(f) = O(|\textsc{Leaf}_{X_i}(\Phi)|)$, which would prove the required bound.

Fix an arbitrary set $Y = X_i$. Since the goal is to just get a bound on $\textsc{Leaf}_Y(\Phi)$, we shall modify the formula Φ by introducing new leaf variables that compute polynomial in

$\mathbb{F}[X \setminus Y]$. This does not affect the size of $\textsc{Leaf}_Y(\Phi)$. This in some sense is allowing Φ to "freely" compute any polynomial on variables outside Y, as we are only interested in how many times the Y variables are used in the computation of Φ.

We modify Φ by introducing a new type of gate $\boxtimes$ that takes a node g and two leaves ℓ_1 and ℓ_2 and computes $\boxtimes(g, \ell_1, \ell_0) = (\ell_1 \cdot g) + \ell_0$. We shall always ensure that when we introduce new leaf nodes, they only hold polynomials $\mathbb{F}[X \setminus Y]$.

Define the following three sets of nodes:

$$\begin{aligned} V_0 &= \{v \in \Phi \ : \ |\textsc{Leaf}_Y(v)| = 0\} \\ V_1 &= \{v \in \Phi \ : \ |\textsc{Leaf}_Y(v)| = 1 \quad \text{and} \quad |\textsc{Leaf}_Y(\textsc{Parent}(v))| \geq 2\} \\ V_2 &= \{v \in \Phi \ : \ |\textsc{Leaf}_Y(v)| \geq 2\}. \end{aligned}$$

Each node $v \in V_0$ computes a polynomial in $f_v \in \mathbb{F}[X \setminus Y]$, and we shall replace the subtree at v by a leaf computing the polynomial f_v.

Similarly, any node $v \in V_1$ computes a polynomial of the form $f_1 \cdot y_v + f_0$ for some $y_v \in Y$ and $f_0, f_1 \in \mathbb{F}[X \setminus Y]$. We shall create two new leaf nodes ℓ_0, ℓ_1 computing f_0 and f_1 respectively, and replace the gate v by a $\boxtimes$ gate with inputs y_v, ℓ_1, ℓ_0 so that it computes $\boxtimes(y_v, \ell_1, \ell_0) = (f_1 \cdot y_v) + f_0$

Hence, the formula Φ now reduces to a smaller formula Φ_Y with leaves being the nodes in V_0 and V_1 (and nodes in V_2 are unaffected). Furthermore, all new leaves that are added compute polynomials in $\mathbb{F}[X \setminus Y]$ and hence $\textsc{Leaf}_Y$ is unchanged. We would like to show that the size of the reduced formula, which is at most twice the number of its leaves, is $O(|\textsc{Leaf}_Y(\Phi)|)$.

Observation 6.8. $|V_1| \leq |\textsc{Leaf}_Y(\Phi)|$.

Proof. Each node in V_1 has a distinct leaf labelled with a variable in Y. Hence, $|V_1|$ is bounded by the number of leaves labelled with a variable in Y. □ (Obs)

This shows that the V_1 leaves are not too many. Unfortunately, we cannot immediately bound the number of V_0 leaves, since we could have a long chain of V_2 nodes each with one sibling being a V_0 leaf. The following observation would show how we can eliminate such long chains.

Observation 6.9. *Let u be an arbitrary node, and v be another node in the subtree rooted at u with $\textsc{Leaf}_Y(u) = \textsc{Leaf}_Y(v)$. Then the polynomial g_u computed at u and the polynomial g_v computed at v are related, i.e., $g_u = f_1 g_v + f_0$ for some $f_1, f_0 \in \mathbb{F}[X \setminus Y]$.*

Proof. If $\textsc{Leaf}_Y(u) = \textsc{Leaf}_Y(v)$, then every node on the path from u to v must have a V_0 leaf as the other child. The observation follows as all these nodes are $+$ or $\times$ gates. $\square$ (Obs)

Using the above observation, we shall remove the need for V_0 nodes completely by adding new $\mathbb{F}[X \setminus Y]$ leaves and $\boxtimes$ gates. Formally, if a V_0 node computing f was connected to a $+$ node that was computing $f + g$, then we can replace the V_0 node by a new leaf ℓ_f computing f and replace that $+$ node with $\boxtimes(g, 1, \ell_f) = (g \cdot 1) + f$. Similarly, if a V_0 node computing f was incident on a $\times$ node that was computing $f \cdot g$, then we can replace the V_0 node by a new leaf ℓ_f computing f and replace the $\times$ gate with a $\boxtimes(g, \ell_f, 0) = g \cdot f + 0$.

Furthermore, using Observation 6.9, we can now contract any two nodes u and v with $\textsc{Leaf}_Y(u) = \textsc{Leaf}_Y(v)$, we can replace the entire chain from u to v by an appropriate $\boxtimes$ node. This ensures that no $\boxtimes$ node is connected to another $\boxtimes$ node. Overall, Φ has been transformed in the following ways:

1. all V_0 nodes are replaced by leaves,
2. all V_1 are replaced by $\boxtimes$ nodes with three leaves incident on it,
3. no $\boxtimes$ node is incident on another $\boxtimes$ node, and hence its parent is a V_2 node,
4. distinct vertices $u, v \in V_2$ satisfy $\textsc{Leaf}_Y(u) \neq \textsc{Leaf}_Y(v)$,
5. leaves in Y are untouched,
6. all new leaves compute polynomials only in $\mathbb{F}[X \setminus Y]$.

(2) implies that the number of $|V_1|$ nodes in $\hat{\Phi}_Y$ is at most $|\textsc{Leaf}_Y(\Phi)|$. Also (4) implies the number of V_2 nodes is at most $|\textsc{Leaf}_Y(\Phi)| - 1$. Therefore, the size of the augmented formula $\hat{\Phi}_Y$ is at most $2\,|\textsc{Leaf}_Y(\Phi)|$.

Suppose Φ computes a polynomial f, which can be written as $f = \sum_{i=1}^{t} f_i \cdot m_i$ with $f_i \in \mathbb{F}[X \setminus Y]$ and m_i's being distinct monomials in Y. Since $\hat{\Phi}_Y$ also computes f, each f_i is a polynomial combination of the leaves of $\hat{\Phi}_Y$ that compute polynomials in $\mathbb{F}[X \setminus Y]$. Since $\hat{\Phi}_Y$ consists of at most $2\,|\textsc{Leaf}_Y(\Phi)|$ augmented nodes, we have that $\mathrm{td}_Y(f) \leq 4\,|\textsc{Leaf}_Y(\Phi)|$. Therefore,

$$\mathrm{td}_Y(f) \quad = \quad \mathrm{trdeg}\,\{f_i \;:\; i \in [t]\} \quad \leq \quad 4\,|\textsc{Leaf}_Y(\Phi)|$$

Hence,

$$\Gamma^{[\mathrm{Kal}]}(\Phi) = \sum_{i=1}^{r} \mathrm{td}_{X_i}(f_i) \leq 4\left(\sum_{i=1}^{r} |\textsc{Leaf}_{X_i}|\right) = O(s). \qquad \square$$

6.2.2 Lower bounding $\Gamma^{[\mathrm{Kal}]}(\mathsf{Det}_n)$

Lemma 6.10. *Let* $X = X_1 \sqcup \cdots \sqcup X_n$ *be the partition as defined by* $X_t = \{x_{ij} \ : \ i - j \equiv t \bmod n\}$. *Then,* $\Gamma^{[\mathrm{Kal}]}(\mathsf{Det}_n) = \Omega(n^3)$.

Proof. By symmetry, it is easy to see that $\mathrm{td}_{X_i}(\mathsf{Det}_n)$ is the same for all i. Hence, it suffices to show that $\mathrm{td}_Y(\mathsf{Det}_n) = \Omega(n^2)$ for $Y = X_n = \{x_{11}, \ldots, x_{nn}\}$.

To see this, observe that the determinant consists of the monomials $\left(\frac{x_{11}\ldots x_{nn}}{x_{ii}x_{jj}}\right) \cdot x_{ij}x_{ji}$ for every $i \neq j$. Hence, $\mathrm{td}_Y(\mathsf{Det}_n) \geq \mathrm{trdeg}\,\{x_{ij}x_{ji} \ : \ i \neq j\} = \Omega(n^2)$. Therefore, $\Gamma^{[\mathrm{Kal}]}(\mathsf{Det}_n) = \Omega(n^3)$. □

The proof of Theorem 6.4 follows from Lemma 6.7 and Lemma 6.10.

Exercise 6.1 *Prove an* $\Omega(n^2)$ *lower bound for* $\Gamma^{[\mathrm{Kal}]}(f)$ *where* $f = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i^j y_j$ *for an appropriate partition.*

Chapter 7

Determinantal Complexity Lower Bounds

Recall that any polynomial f that is computable by an ABP of polynomial size can be written as a projection of the determinant (Theorem 3.6). Thus, a direct way to prove lower bounds is to find an explicit polynomial that requires super-polynomially large determinants to compute it. Hence, for a polynomial f, it is natural to ask "If f is to be written as a projection of an $m \times m$ determinant, how large should m be?" This is called the *determinantal complexity* of f.

Definition 7.1. *Let f be an n-variate polynomial. The determinantal complexity of $f(\mathbf{x})$, denoted by* $\mathrm{DetComp}(f)$ *is the smallest m such that there is an $m \times m$ matrix $A(\mathbf{x})$ with each entry being an linear function in $\mathbf{x}$ such that $f = \det(A(\mathbf{x}))$.* $\Diamond$

By Theorem 3.6, if we could show that $\mathrm{DetComp}(\mathsf{Perm}_n) = n^{\omega(1)}$, then this immediately would imply super-polynomial circuit lower bounds.

Some prior work

In 1986, Von zur Gathen [vzG86] proved that $\mathrm{DetComp}(\mathsf{Perm}_n) \geq \left(\sqrt{\frac{8}{7}}\right) n$. Subsequently, Cai [Cai90] and Meshulam [Mes89] independently improved the lower bound to $\sqrt{2}n$.

In 2004, Mignon and Ressayre[MR04] came up with a new idea of using second order derivatives and proved the first super-linear lower bound of $\mathrm{DetComp}(\mathsf{Perm}_n) \geq \frac{n^2}{2}$ over fields of characteristic zero. Following that, Cai, Chen and Li [CCL08] extended the result of Mignon and Ressayre to all fields of characteristic $\neq 2$.

In this chapter, we shall see the proof of Mignon and Ressayre's result. We first describe some intuition for the approach of Mignon and Ressayre of using the *Hessian.*

7.1 Why Hessian?

As in all the previous lower bounds, we would like to identify a certain "weakness of the model", and here we are attempting to identify some properties of the determinant that are not shared by the permanent. The following is the key question that led Mignon and Ressayre towards their approach.

> Consider a matrix X_0 for which $\mathsf{Det}_n(X_0) = 0$. What are the *perturbations* of X_0 that continue to keep Det_n to be zero?
>
> What is the answer to the same question for Perm_n?

Small perturbation around points where the function stays constant is really just the tangent plane at that point. Their main observation was that the answer to the tangent planes at zeros of Det_n and Perm_n look very different.

In the case of Det_n, since $\mathsf{Det}_n(X_0) = 0$, there is some non-zero vector $\mathbf{v}$ in the kernel. Now consider the space vector space $\mathcal{V}$ of matrices that also have $\mathbf{v}$ in their kernel. Then, clearly $\mathsf{Det}_n(M + X_0)$ is also zero for every $M \in \mathcal{V}$. Notice that $\mathcal{V}$ is a space of dimension at least $n^2 - n$ as enforcing $M(\mathbf{v}) = \mathbf{0}$ adds just n homogeneous constraints on n^2 variables. Hence, around any zero of Det_n is a *huge* subspace in which Det_n continues to stay zero. Such a statement, intuitively, should be unlikely to hold for a generic matrix X_0 with $\mathsf{Perm}_n(X_0) = 0$. The question now is if we can convert this geometric statement into a measure.

Let us attempt to formalize this intuition. For an arbitrary n-variate function f, let us look at the *zero surface* of all points with $f(X) = 0$ and let X_0 be one such point on the surface. The tangent of the surface at the point X_0 is a hyperplane whose normal is specified by the *gradient* of f defined as

$$\nabla f(X_0) \quad = \quad (\partial_1(f), \ldots, \partial_n(f))\,(X_0)$$

If it so turns out that, $f(\lambda V + X_0) = 0$ for all $\lambda \in \mathbb{F}$, and hence V must also lie on the tangent plane. This therefore also implies that V must be perpendicular to $\nabla f(X_0)$ as well.

Note that this *does not* imply that the gradient at $V + X_0$ must be equal to $\nabla f(X_0)$ but at the very least they must both be perpendicular to V.

Now suppose we have a vector space $\mathcal{V}$ of dimension $n - r$ such that for every $V \in \mathcal{V}$ we have $f(V + X_0) = 0$. For any $V \in \mathbb{F}^n$ let $G_V = \nabla f(X_0 + V)$, the gradient of f at $X_0 + V$. By the discussion above, we know that $G_V \perp \mathcal{V}$ for every $V \in \mathcal{V}$. Therefore,

$$\operatorname{rank}\{G_V \ : \ V \in \mathcal{V}\} \quad \leq \quad r.$$

If we have $n - r \geq r$, then this implies that there must be at least $(n - 2r)$ *directions* in which $\nabla f(X_0)$ does not change. This can be precisely captured by the *Hessian* of f, denoted by $\mathrm{Hess}(f)$, defined as follows:

$$\mathrm{Hess}(f)(X_0) \quad := \quad \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix} (X_0).$$

In other words, the i-th row of the Hessian corresponds to the derivative of ∇f with respect to x_i. Since there are $n - 2r$ directions in which the gradient does not change, this implies that the rank of the Hessian at X_0 is at most $2r$. This is formalized in the following lemma.

Lemma 7.2. *Let f be an n-variate polynomial and let $f(X_0) = 0$ for some $X_0 \in \mathbb{F}^n$. If there is a space $\mathcal{V}$ such that $f(X_0 + V) = 0$ for all $V \in \mathcal{V}$ and if $\dim \mathcal{V} = n - r$, then*

$$\operatorname{rank}(\mathrm{Hess}(f)(X_0)) \quad \leq \quad 2r.$$

Proof. Consider the $\mathcal{H} : \mathcal{V} \to \mathbb{F}^n$ that maps $M \in \mathcal{V}$ to $\partial_M(\nabla f)(X_0)$, the coordinate-wise directional derivative of ∇f in the direction M evaluated at X_0. That is,

$$\partial_M(\nabla f)(X_0) \quad = \quad \left(\left(\frac{\partial G_1}{\partial M}\right)(X_0), \ldots, \left(\frac{\partial G_n}{\partial M}\right)(X_0)\right) \in \mathbb{F}^n$$

where $G = (G_1, \ldots, G_n) = \nabla f$.

We know that $(\nabla f)(X_0)$ is perpendicular to $\mathcal{V}$ but we claim that $\partial_M(\nabla f)(X_0)$ is also perpendicular to $\mathcal{V}$. This is because $(\nabla f)(X_0 + \varepsilon M) - (\nabla f)(X_0) \in \mathcal{V}^\perp$ for every $\varepsilon \in \mathbb{F}$, and therefore we must have $\partial_M(\nabla f)(X_0) \in \mathcal{V}^\perp$.

Furthermore, note that $\mathcal{H}$ is linear map, just by the definition of directional derivatives. Thus $\mathcal{H}$ is a linear map from a space $\mathcal{V}$ of dimension at least $n - r$ to a $\mathcal{V}^\perp$ of dimension at most r. Therefore, the kernel of $\mathcal{H}$ has dimension at least $n - 2r$. Since $\mathcal{H}$ is just the restriction of $\text{Hess}(f)(X_0)$ to the space $\mathcal{V}$, it follows that $\text{rank}(\text{Hess}(f)(X_0)) \leq 2r$. □

Corollary 7.3. *For any X_0 such that $\mathsf{Det}_n(X_0) = 0$, we have*

$$\square \qquad \text{rank}\left(\text{Hess}(\mathsf{Det}_n)(X_0)\right) \quad \leq \quad 2n.$$

Thus the rank of the Hessian at a zero of the polynomials certainly seems like a good complexity measure for determinantal complexity.

7.2 The lower bound of Mignon-Ressayre

The main theorem of this section would be the following.

Theorem 7.4 ([MR04])**.** *Over any characteristic zero field,* $\text{DetComp}(\mathsf{Perm}_n) \geq n^2/2$*.*

The rest of this section would be dedicated to the proof of this theorem. Throughout this section, we shall assume that $\mathbb{F}$ is a characteristic zero field. Let $\mathsf{Perm}_n = \mathsf{Det}_m(A(\mathbf{x}))$ where $A(\mathbf{x})$ is an $m \times m$ matrix consisting of linear polynomials. The goal is to show that $m \geq n^2/2$. As mentioned earlier, the complexity measure would the rank of the Hessian at a carefully chosen matrix X_0. The proof is immediate from the following two lemmas.

Lemma 7.5 (Upper bound for determinant)**.** *Let $X_0 \in \mathbb{F}^{n^2}$ such that $\mathsf{Det}_m(A(X_0)) = 0$. Then,*

$$\text{rank}\left(\text{Hess}(\mathsf{Det}_m(AX))(X_0)\right) \quad \leq \quad 2m.$$

Lemma 7.6 (Lower bound for permanent)**.** *There exists $X_0 \in \mathbb{F}^{n^2}$ such that $\mathsf{Perm}_n(X_0) = 0$ and*

$$\text{rank}\left(\text{Hess}(\mathsf{Perm}_n(X_0)\right) \quad = \quad n^2.$$

The proof of Lemma 7.5 is quite an easy adaptation of Lemma 7.2, and the proof of Lemma 7.6 is a just a (slightly tedious) calculation.

Proof of Lemma 7.5. Let $A(X - X_0) = L(X) + B_0$ where $L(X)$ is a matrix of *linear forms* (no constant term) and B_0 a matrix of constants. Since $\mathsf{Det}_m(A(X_0)) = 0$, we must have that $\mathsf{Det}_m(B_0) = 0$. Let $\mathbf{v} \in \mathbb{F}^{m^2}$ be a vector such that $B\mathbf{v} = \mathbf{0}$. Consider the vector space $\mathcal{V}$ defined by

$$\mathcal{V} \quad := \quad \left\{ M \in \mathbb{F}^{n^2} \ : \ L(M)\, \mathbf{v} = \mathbf{0} \right\}.$$

As earlier, we have $\dim \mathcal{V} \geq n^2 - m$ and $\mathsf{Det}_m(A(M + X_0)) = 0$ for all $M \in \mathcal{V}$. Hence, applying Lemma 7.2 to $f(X) = \mathsf{Det}_m(A(X))$, we have

$$\square \qquad \operatorname{rank}\Big(\operatorname{Hess}(\mathsf{Det}_m(AX))(X_0)\Big) \quad \leq \quad 2m.$$

Proof of Lemma 7.6. Mignon and Ressayre [MR04] use the following matrix for X_0:

$$X_0 \quad = \quad \begin{bmatrix} 1-d & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Since the proof is just a calculation, we only mention the main steps of the proof and leave it as an exercise to fill in the details.

Claim 7.7. *For X_0 as defined above, the matrix $\operatorname{Hess}(\mathsf{Perm}_n)(X_0)$ can be expressed as*

$$\operatorname{Hess}(\mathsf{Perm}_n)(X_0) \quad = \quad (d-3)! \cdot \begin{bmatrix} \mathbf{0} & P & P & \cdots & P \\ P & \mathbf{0} & Q & \cdots & Q \\ P & Q & \mathbf{0} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & Q \\ P & Q & \cdots & Q & \mathbf{0} \end{bmatrix}_{n^2 \times n^2}$$

where

$$P = (d-2) \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix}_{n \times n} \quad \text{and} \quad Q = \begin{bmatrix} 0 & d-2 & d-2 & \cdots & d-2 \\ d-2 & 0 & -2 & \cdots & -2 \\ d-2 & -2 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -2 \\ d-2 & -2 & \cdots & -2 & 0 \end{bmatrix}_{n \times n}.$$

Claim 7.8. *For any pair of invertible $n \times n$ matrices P and Q, the matrix*

$$\begin{bmatrix} \mathbf{0} & P & P & \cdots & P \\ P & \mathbf{0} & Q & \cdots & Q \\ P & Q & \mathbf{0} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & Q \\ P & Q & \cdots & Q & \mathbf{0} \end{bmatrix} \quad \textit{is also invertible.}$$

From these two claims, it follows that $\mathrm{Hess}(\mathsf{Perm}_n)(X_0)$ is invertible and hence has rank n^2. □

Theorem 7.4 follows from Lemma 7.5 and Lemma 7.6. $\square_{\text{(Theorem 7.4)}}$

Chapter 8

Some simple lower bounds

8.1 "Natural" proof strategies

The lower bounds presented in Chapter 6 proceeded by first identifying a *weakness* of the model, and exploiting it in an explicit manner. More concretely, Section 6.2 presents a promising strategy that could be adopted to prove lower bounds for various models of arithmetic circuits. The crux of the lower bound was the construction of a good map Γ that assigned a number to every polynomial. The map $\Gamma^{[\mathrm{Kal}]}$ was useful to show a lower bound in the sense that any f computable by a *small* formula had *small* $\Gamma^{[\mathrm{Kal}]}(f)$. In fact, all subsequent lower bounds in arithmetic circuit complexity have more or less followed a similar template of a "natural proof". More concretely, all the subsequent lower bounds we shall see would essentially follow the outlined plan.

> **Step 1 (normal forms)** For every circuit in the circuit class $\mathcal{C}$ of interest, express the polynomial computed as a *small sum of simple building blocks*.

For example, every $\Sigma\Pi\Sigma$ circuit is a *small* sum of *products of linear polynomials* which are the building blocks here. In this case, the circuit model naturally admits such a representation but we shall see other examples with very different representations as sum of building blocks.

> **Step 2 (complexity measure)** Construct a map $\Gamma : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{Z}_{\geq 0}$ that is *sub-additive* i.e. $\Gamma(f_1 + f_2) \leq \Gamma(f_1) + \Gamma(f_2)$.

In most cases, $\Gamma(f)$ is the rank of a large matrix whose entries are linear functions in the coefficients of f. In such cases, we immediately get that Γ is sub-additive.

The strength of the choice of Γ is determined by the next step.

> **Step 3 (potential usefulness)** Show that if B is a *simple building block*, then $\Gamma(B)$ is *small.* Further, check if $\Gamma(f)$ for a *random polynomial* f is large (potentially).

This would suggest that if any f with large $\Gamma(f)$ is to be written as a sum of $B_1 + \cdots + B_s$, then sub-additivity and the fact that $\Gamma(B_i)$ is small for each i and $\Gamma(f)$ is large immediately imply that s must be large. This implies that the complexity measure Γ does indeed have a potential to prove a lower bound for the class. The next step is just to replace the *random polynomial* by an explicit polynomial.

> **Step 4 (explicit lower bound)** Find an explicit polynomial f for which $\Gamma(f)$ is large.

These are usually the steps taken in almost all the known arithmetic circuit lower bound proofs. The main ingenuity lies in constructing a useful complexity measure, which is really to design Γ so that it is small on the *building blocks*.

Of course, there could potentially be lower bound proofs that do not follow the road-map outlined. For instance, it could be possible that Γ is not small for a random polynomial, but specifically tailored in a way to make Γ large for the Perm_n. Or perhaps Γ need not even be sub-additive and maybe there is a very different way to argue that all polynomial in the circuit class have small Γ. However, this has been the road-map for almost all lower bounds so far (barring very few exceptions). As a warmup, we first present some very simple applications of the above plan to prove lower bounds for some very simple subclasses of arithmetic circuits in the next section. We then move on to more sophisticated proofs of lower bounds for less restricted subclasses of circuits.

Let us start with the simplest complete[1] class of arithmetic circuits – depth-2 circuits or $\Sigma\Pi$ circuits.

8.1.1 Lower bounds for $\Sigma\Pi$ circuits

Any $\Sigma\Pi$ circuit of size s computes a polynomial $f = m_1 + \cdots + m_s$ where each m_i is a monomial multiplied by a field constant. Therefore, any polynomial computed by a *small*

[1] in the sense that any polynomial can be computed in this model albeit of large size

$\Sigma\Pi$ circuit must have a *small* number of monomials. Hence, it is obvious that any polynomial that has many monomials require large $\Sigma\Pi$ circuits.

This can be readily rephrased in the language of the outline described last section by defining $\Gamma(f)$ to simply be the number of monomials present in f. Hence, $\Gamma(f) \leq s$ for any f computed by a $\Sigma\Pi$ circuit of size s. Of course, even a polynomial like $f = (x_1+x_2+\cdots+x_n)^n$ have $\Gamma(f) = n^{\Omega(n)}$ giving the lower bound.

8.1.2 Lower bounds for $\Sigma\wedge\Sigma$ circuits

A $\Sigma\wedge\Sigma$ circuit of size s computes a polynomial of the form $f = \ell_1^{d_1} + \cdots + \ell_s^{d_s}$ where each ℓ_i is a linear polynomial over the n variables.[2]

Clearly as even a single ℓ^d could have exponentially many monomials, the Γ defined above cannot work in this setting. Nevertheless, we shall try to design a similar map to ensure that $\Gamma(f)$ is *small* whenever f is computable by a *small* $\Sigma\wedge\Sigma$ circuit.

In this setting, the *building blocks* are terms of the form ℓ^d. The goal would be to construct a *sub-additive* measure Γ such that $\Gamma(\ell^d)$ is *small.* Here is the key observation to guide us towards a good choice of Γ.

Observation 8.1. *Any k-th order partial derivative of ℓ^d is a constant multiple of ℓ^{d-k}.*

Hence, if $\partial^{=k}(f)$ denotes the set of k-th order partial derivatives of f, then the space spanned by $\partial^{=k}(\ell^d)$ has dimension 1. This naturally leads us to define Γ exploiting this weakness.

$$\Gamma_k(f) \quad \overset{\text{def}}{=} \quad \dim\left(\partial^{=k}(f)\right)$$

It is straightforward to check that Γ_k is indeed sub-additive and hence $\Gamma_k(f) \leq s$ whenever f is computable by a $\Sigma\wedge\Sigma$ circuit of size s. For a random polynomial f, we should be expecting $\Gamma_k(f)$ to be $\binom{n+k}{k}$ as there is unlikely to be any linear dependencies among the partial derivatives. Hence, all that needs to be done is to find an explicit polynomial with large Γ_k.

If we consider Det_n or Perm_n, then any partial derivative of order k is just an $(n-k) \times (n-k)$ minor. Also, these minors consist of disjoint sets of monomials and hence are linearly independent. Hence, $\Gamma_k(\mathsf{Det}_n) = \binom{n}{k}^2$. Choosing $k = n/2$, we immediately get that any $\Sigma\wedge\Sigma$

[2] such circuits are also called *diagonal depth-3 circuits* in the literature

circuit computing Det_n or Perm_n must be of size $2^{\Omega(n)}$.

8.1.3 Low-rank $\Sigma\Pi\Sigma$

A slight generalization of $\Sigma\wedge\Sigma$ circuits is a *rank-r $\Sigma\Pi\Sigma$ circuit* that computes a polynomial of the form

$$f \quad = \quad T_1 \; + \; \ldots \; + \; T_s$$

where each $T_i = \ell_{i1} \ldots \ell_{id}$ is a product of linear polynomials such that $\dim \{\ell_{i1}, \ldots, \ell_{id}\} \leq r$.

Thus, $\Sigma\wedge\Sigma$ is a rank-1 $\Sigma\Pi\Sigma$ circuit, and a similar partial-derivative technique for lower bounds works here as well.

In the setting where r is much smaller than the number of variables n, each T_i is essentially an r-variate polynomial masquerading as an n-variate polynomial using an affine transformation. In particular, the set of n first order derivatives of T have rank at most r. This yields the following observation.

Observation 8.2. *Let $T = \ell_1 \ldots \ell_d$ with $\dim \{\ell_1, \ldots, \ell_d\} \leq r$. Then for any k, we have*

$$\Gamma_k(T) \quad \overset{def}{=} \quad \dim\left(\partial^{=k}(T)\right) \quad \leq \quad \min\left(\binom{r+k}{k}, \binom{d}{k}\right)$$

Thus once again by sub-additivity, for any polynomial f computable by a rank-r $\Sigma\Pi\Sigma$ circuit of size s, we have $\Gamma_k(f) \leq s \cdot \binom{r+k}{r}$. Note that a random polynomial is expected to have $\Gamma_k(f)$ close to $\binom{n+k}{k}$, which could be much larger for $r \ll n$. We already saw that $\Gamma_k(\mathsf{Det}_n) = \binom{n}{k}^2$. This immediately gives the following lower bound, the proof of which we leave as an exercise to the interested reader.

Theorem 8.3. *Let $r \leq n^{2-\delta}$ for some constant $\delta > 0$. For $k = \varepsilon n^{\delta}$, where $\varepsilon > 0$ is sufficiently small, we have*

$$\frac{\binom{n}{k}^2}{\binom{r+k}{k}} \quad = \quad \exp\left(\Omega(n^{\delta})\right).$$

Hence, any rank-r $\Sigma\Pi\Sigma$ circuit computing Det_n or Perm_n must have size $\exp\left(\Omega(n^{\delta})\right)$. □

This technique of using the rank of partial derivatives was introduced by Nisan and Wigderson [NW97] to prove lower bounds for *homogeneous depth-3 circuits* (which also follows as a corollary of Theorem 8.3). The survey of Chen, Kayal and Wigderson [CKW11] give a comprehensive exposition of the power of the *partial derivative method*.

In the examples we saw above, Step 1 of constructing the normal forms were obtained from just the model of computation. We conclude this chapter with a more non-trivial example of a lower bound where the building blocks are constructed differently.

8.2 Lower bounds for monotone circuits

We shall now see a slight generalization of a lower bound by Jerrum and Snir [JS82]. To motivate our presentation here, let us first assume that the underlying field is $\mathbb{R}$, the field of real numbers. A monotone circuit over $\mathbb{R}$ is a circuit having $+, \times$ gates in which all the field constants are *non-negative* real numbers. Such a circuit can compute any polynomial f over $\mathbb{R}$ all of whose coefficients are nonnegative real numbers, such as for example the permanent. It is then natural to ask whether there are small monotone circuits over $\mathbb{R}$ computing the permanent. Jerrum and Snir [JS82] obtained an exponential lower bound on the size of monotone circuits over $\mathbb{R}$ computing the permanent. Note that this definition of monotone circuits is valid only over $\mathbb{R}$ (actually more generally over ordered fields but not over say finite fields) and such circuits can only compute polynomials with non-negative coefficients. Here we will present Jerrum and Snir's argument in a slightly more generalized form such that the circuit model makes sense over any field $\mathbb{F}$ and is complete, i.e. can compute any polynomial over $\mathbb{F}$.[3] Let us first explain the motivation behind the generalized circuit model that we present here. Observe that in any monotone circuit over $\mathbb{R}$, there is no cancellation as there are no negative coefficients. Formally, for a node v in our circuits let us denote by f_v the polynomial computed at that node. For a polynomial f let us denote by $\mathrm{Mon}(f)$ the set of monomials having a nonzero coefficient in the polynomial f.

1. If $w = u + v$ then
$$\mathrm{Mon}(f_w) = \mathrm{Mon}(f_u) \cup \mathrm{Mon}(f_v).$$

2. If $w = u \times v$ then
$$\mathrm{Mon}(f_w) = \mathrm{Mon}(f_u) \cdot \mathrm{Mon}(f_v) \overset{\text{def}}{=} \{m_1 \cdot m_2 \ : \ m_1 \in \mathrm{Mon}(f_u), m_2 \in \mathrm{Mon}(f_v)\}.$$

This means that for any node v in a monotone circuit over $\mathbb{R}$ one can determine $\mathrm{Mon}(f_v)$

[3] This generalization was told to me by Neeraj Kayal

in a very syntactic manner starting from the leaf nodes. Let us make precise this syntactic computation that we have in mind.

Definition 8.4 (Formal Monomials.)**.** *Let Φ be an arithmetic circuit. The* formal monomials *at any node $v \in \Phi$, which shall be denoted by* FM(v)*, shall be inductively defined as follows:*

> *If v is a leaf labelled by a variable x_i, then $\mathrm{FM}(v) = \{x_i\}$. If it is labelled by a constant, then $\mathrm{FM}(v) = \{1\}$.*
>
> *If $v = v_1 + v_2$, then $\mathrm{FM}(v) = \mathrm{FM}(v_1) \cup \mathrm{FM}(v_2)$.*
>
> *If $v = v_1 \times v_2$, then*

$$\begin{aligned} \mathrm{FM}(v) &= \mathrm{FM}(v_1) \cdot \mathrm{FM}(v_2) \\ &\overset{def}{=} \{m_1 \cdot m_2 \ : \ m_1 \in \mathrm{FM}(v_1), m_2 \in \mathrm{FM}(v_2)\}\,. \end{aligned}$$

◊

Note that for any node v in any circuit we have $\mathrm{Mon}(f_v) \subseteq \mathrm{FM}(v)$ but in a monotone circuit over $\mathbb{R}$ this containment is in fact an equality at every node. This motivates our definition of a slightly more general notion of a monotone circuit as follows.

Definition 8.5 (Monotone circuits)**.** *A circuit C is said to be* syntactically monotone *(simply monotone for short) if $\mathrm{Mon}(f_v) = \mathrm{FM}(v)$ for every node v in C.* ◊

The main theorem of this section is the following:

Theorem 8.6 ([JS82])**.** *Over any field $\mathbb{F}$, any syntactically monotone circuit C computing Det_n or Perm_n must have size at least $2^{\Omega(n)}$.*

The proof of this theorem is relatively short assuming the following structural result (which is present in standard depth-reduction proofs [VSBR83, AJMV98]).

Lemma 8.7. *Let f be a degree d polynomial computed by a monotone circuit of size s. Then, f can be written of the form $f = \sum_{i=1}^{s} f_i \cdot g_i$ where the f_i's and g_i's satisfy the following properties.*

1. *For each $i \in [s]$, we have $\frac{d}{3} < \deg g_i \leq \frac{2d}{3}$.*
2. *For each i, we have $\mathrm{FM}(f_i) \cdot \mathrm{FM}(g_i) \subseteq \mathrm{FM}(f)$.*

Sketch of Proof. The proof of this Lemma is just an application of (5.13) with $t = d/3$:

$$f \quad = \quad [\text{root}] \quad = \quad \sum_{v \in \mathcal{F}_t} [\text{root} : v] \cdot [v]$$

It is easy to observe that $[\text{root} : v]$ and $[v]$ are polynomials of degree between $d/3$ and $2d/3$. Further, it can also be seen that the above equation preserves monotonicity if the original circuit was monotone. □

Exercise 8.1 *Show that the process of homogenization and depth reduction via Theorem 5.8 and Theorem 5.17 on a monotone circuit results in a monotone circuit as well.*

The complexity measure $\Gamma(f)$ in this case is just the number of monomials in f, but it is the above *normal form* that is crucial in the lower bound. Although Theorem 8.6 gives a lower bound for Det_n and Perm_n, we shall give a simpler lower bound for a different polynomial and leave proving a lower bound for Det_n and Perm_n

Theorem 8.8. *Any monotone circuit* Φ *computing the polynomial* $\mathrm{NW}_{n,n,n/10}$ *must have size* $n^{\Omega(n)}$.

Proof. Let us assume that Φ is a size s monotone circuit that computes $f = \mathrm{NW}_{n,n,n/10}$. Then by Lemma 8.7,

$$f \quad = \quad \sum_{i=1}^{s} f_i \cdot g_i$$

with the appropriate degree bounds. Suppose f_i had at least two non-zero monomials m_1 and m_2, for any monomial $m \in g_i$ we would have $m_1 \cdot m \in \mathrm{FM}(f)$ and $m_2 \cdot m \in \mathrm{FM}(f)$. But since g_i is a polynomial of degree at least $n/3$, this implies the monomials $m_1 \cdot m$ and $m_2 \cdot m$ are two distinct monomials that intersect in at least $n/3$ places. But this contradicts the key property of the NW family (Lemma 2.5) which is that no two monomials intersect in more than $n/10$ places. Thus, each of the f_i's and g_i's must in fact be monomials and hence $\Gamma(f_i \cdot g_i) \leq 1$ for each i. This then immediately forces

$$\Gamma(\mathrm{NW}_{n,n,n/10}) \quad = \quad n^{\Omega(n)} \quad \leq \quad \sum_{i=1}^{s} \Gamma(f_i \cdot g_i) \quad \leq \quad s$$

□

Exercise 8.2 *Using the normal form provided by Lemma 8.7 to prove a $2^{\Omega(n)}$ lower bound for Det_n and Perm_n.*

We shall now proceed towards some more involved lower bounds.

Part III

Partial Derivative Spaces

Chapter 9

Lower bounds for depth-3 circuits

In this chapter, we shall see the lower bound of Shpilka and Wigderson [SW01] for non-homogeneous depth-3 circuits over arbitrary fields. The main theorem of this section would be the following *quadratic* lower bound.

Theorem 9.1 ([SW01]). *Any $\Sigma\Pi\Sigma$ circuit that computes the polynomial Sym_d, for $d = n/100$ must have at least $\Omega(n^2)$ wires.*

In fact, Sym_d can indeed be computed by a $O(n^2)$-sized $\Sigma\Pi\Sigma$ circuit over a characteristic zero field (Exercise 5.3) and hence the above result is tight for Sym_d. Until recently, this was the best lower bound we knew for the class of general $\Sigma\Pi\Sigma$ circuits but a very recent result of Kayal, Saha and Tavenas [KST16] has improved this to an almost cubic lower bound (for a different explicit family of polynomials) which we shall see at a later point. This chapter however shall focus on the proof of the above theorem.

9.1 Lower bounds for hom. $\Sigma\Pi\Sigma$ circuits [NW97]

Let us first consider the following restricted question. Say we want to compute an n-variate degree d polynomial f using a $\Sigma\Pi\Sigma$ circuit, but under the restriction that all intermediate computations have degree at most d. The class of such circuits are denoted by $\Sigma\Pi^{[d]}\Sigma$ circuits. Can we prove lower bounds for this class first?

Indeed, and in fact we have already seen how to in Chapter 8. But it would be good to recall the method again as we would be using this heavily.

Theorem 9.2 ([NW97]). *Any $\Sigma\Pi^{[d]}\Sigma$ circuit that computes Sym_d must have size $\frac{\binom{n}{d/2}}{2^d}$.*

Thus, if $n = 3d$, this gives an exponential lower bound.

Proof. For a polynomial f, define the *dimension of k-th order partial derivatives*, denoted by $\Gamma_k^{[\mathrm{NW}]}(f)$, as follows

$$\Gamma_k^{[\mathrm{NW}]}(f) \quad = \quad \dim \partial^{=k}(f)$$

Claim 9.3. *If $f = \ell_1 \cdots \ell_d$ where each ℓ_i is a linear polynomial, then $\Gamma_k^{[\mathrm{NW}]}(f) \leq \binom{d}{k}$. Thus, if $f = \ell_{11} \cdots \ell_{1d} + \cdots + \ell_{s1} \cdots \ell_{sd}$, then $\Gamma_k^{[\mathrm{NW}]}(f) \leq s \cdot \binom{d}{k}$.*

A fact that we would need here but won't prove is that Sym_d has large space of partial derivatives.

Claim 9.4.

$$\Gamma_k^{[\mathrm{NW}]}(\mathsf{Sym}_d) = \min\left(\binom{n}{d-k}, \binom{n}{k}\right)$$

Hence, for $k = d/2$, we get

$$\Gamma_k^{[\mathrm{NW}]}(\mathsf{Sym}_d) = \binom{n}{d/2}.$$

The theorem follows directly from these two claims. □

9.2 Handling *few* high degree gates

In the last section we saw a way to prove lower bounds for $\Sigma\Pi\Sigma$ circuits were the degree of each product of linear forms was bounded by d. Suppose we had a $\Sigma\Pi\Sigma$ circuit C such that all but say two of the products of linear forms have degree bounded by d. That is,

$$C = C' + T_1 + T_2$$

where $C' \in \Sigma\Pi^{[d]}\Sigma$ and T_1, T_2 is a product of linear forms of degree possibly larger than d. Can we prove lower bounds for such circuits as well?

Key Idea: Replace some variables by linear functions in other variables to make T_1 and T_2 equal to zero.

Say T_1 had one of the linear polynomials as $x_1 + 3x_2 + 5x_3 - 4$, then we shall replace x_1 by $-(3x_2 + 5x_3 - 4)$. The result is that T_1 now becomes zero. What happens to T_2? Note that T_2 after this substitution still remains a product of linear polynomials, and its degree cannot increase in this process. However, it may be the case that T_2 was $(x_1 + 3x_2 + 5x_3 - 7)^{2d}$ and under our substitution of x_1 this reduces to a constant. But this is still good because the goal is to eliminate all high degree T_is completely (either by making them zero, or reducing them to constants). This was the key idea of Shpilka and Wigderson [SW01] and we shall formalize this as a lemma.

Lemma 9.5. *Let $C = T_1 + \cdots + T_s$ be a $\Sigma\Pi\Sigma$ and suppose r of the T_is have degree greater than d. Then by taking an* affine projection of co-dimension at most r, *that is by setting at most r variables to linear functions in the remaining variables, the resulting circuit $C' = T'_1 + \cdots T'_{s'}$ is a $\Sigma\Pi^{[d]}\Sigma$ circuit with $s' \leq s$.*

In order to prove a lower bound for $\Sigma\Pi^{[d]}\Sigma$ circuits, we needed to find a polynomial f for which $\dim \partial^{=k}(f)$ is large. The strategy to prove lower bounds for $\Sigma\Pi\Sigma$ circuits with r or fewer *high degree* T_is is now clear:

> Find a polynomial g such that for every g' that is an *affine projection of co-dimension* r on g (that is, obtained from g by setting at most r variables to linear functions in the remaining), we have that $\dim \partial^{=k}(g')$ is large.

9.3 Shpilka and Wigderson's lower bound for $\Sigma\Pi\Sigma$ circuits

Shpilka and Wigderson [SW01] show that Sym_d not only has a large $\dim \partial^{=k}$, it also has large $\dim \partial^{=k}$ even after affine projections of co-dimension $d/100$.

Theorem 9.6 ([SW01])**.** *Consider the polynomial Sym_d for any $d < \frac{n}{100}$. If g is an affine projection of Sym_d of co-dimension $r < \frac{d}{100}$, then*

$$\dim \partial^{=k}(g) \quad \geq \quad \min\left(\binom{n-2r}{k}, \binom{n-2r}{d-2r-k}\right).$$

Thus, if $k = (d-2r)/2$, then

$$\dim \partial^{=k}(g) \geq \binom{n-2r}{(d-2r)/2}.$$

We shall defer the proof of Theorem 9.6 to the end of this section but see why the above theorem implies Theorem 9.1.

Proof of Theorem 9.1. Consider the polynomial Sym_d for $d = \frac{n}{100}$. Suppose it is computable by a $\Sigma\Pi\Sigma$ circuit $C = T_1 + \cdots + T_s$ with at most $\frac{d^2}{100}$ wires. Let r be the number of T_is of degree more than d. Note that, given the bound on the number of wires, there cannot be more than $\frac{d}{100}$ T_is that have degree more than d. Hence, we know that $r \leq \frac{d}{100}$.

Now that $r \leq \frac{d}{100}$, by Lemma 9.5, there is an affine projection ρ of co-dimension at most r such that

$$\rho(\mathsf{Sym}_d) \quad = \quad T'_1 + \cdots + T'_s \quad \in \quad \Sigma\Pi^{[d]}\Sigma$$

with $\deg(T'_i) \leq d$ and $s \leq \frac{d^2}{100}$.

But then, on the one hand we know from Claim 9.3 that $\Gamma_k^{[\mathrm{NW}]}(\rho(\mathsf{Sym}_d))$ is at most $s \cdot \binom{d}{k}$ but on the other hand Theorem 9.6 states that $\Gamma_k^{[\mathrm{NW}]}(\rho(\mathsf{Sym}_d))$ is *large*. If we set the value of k right ($k = (d-2r)/2$ should work) we get a contradiction to the original assumption that $s < \frac{d^2}{100}$.

Hence, $s > \frac{d^2}{100} = \Omega(n^2)$ as claimed by the theorem. □

This entire discussion can be summarized in the following very general remark.

Remark 9.7. *Suppose we have a measure Γ to prove lower bounds for a degree d polynomial computed by a $\Sigma\Pi^{[D]}\Sigma$ circuit. If we can find an explicit polynomial f of degree d with the following properties,*

- *$\Gamma(f)$ is* large,
- *even after an affine restriction ρ of co-dimension r, we have $\Gamma(\rho(f))$ is* large

Then, we can get a $\Omega(Dr)$ lower bound for f. ◊

What we did above was choose the polynomial f as Sym_d with $D = d = \Omega(n)$ and using the fact that the partial derivative space remains large even after an affine restriction of co-dimension $r = \Omega(d)$, we get a $\Omega(d^2) = \Omega(n^2)$ lower bound for $\Sigma\Pi\Sigma$ circuits.

However, this remark is quite general and in principle, if we could prove a lower bound for say $\Sigma\Pi^{[d^2]}\Sigma$ circuits using some measure, then we could potentially extend this to give a cubic lower bound using the affine projection idea. In fact, Kayal, Saha and Tavenas [KST16] (which was subsequently strengthened by Balaji, Limaye and Srinivasan [BLS16]) do indeed show an *almost* cubic lower bound using a measure that we shall be discussed later.

Theorem 9.8 ([KST16]). *There is an explicit n-variate poylnomial f of degree d such that any $\Sigma\Pi\Sigma$ circuit computing f must require $\Omega(n^3)\operatorname{poly}\log(n)$ size.*

Much later in the survey[1], we shall see the proof of Balaji, Limaye and Srinivasan [BLS16] that, besides being a much simpler and modular proof of [KST16], also proves the lower bound for a polynomial computed by a depth-5 circuit.

9.3.1 Rough proof of Theorem 9.6

Let us order the variables as $x_1 \succ x_2 \succ \cdots \succ x_n$. Say we have an affine projection ρ of co-dimension r that sets the linear polynomials $\{\ell_1, \ldots, \ell_r\}$ to zero. Let x_{i_j} be the *highest* variables participating in each ℓ_j, that is,

$$\ell_j \quad = \quad x_{i_j} - \ell'_j.$$

Thus, applying this affine restriction is equivalent to replacing each x_{i_j} by ℓ'_j. To get a sense of what this does to Sym_d, let y_j be the *highest variable* participating in ℓ'_j.

Let us now look at Sym_d, paying attention to the variables x_{i_j}s and y_js. We may assume that $S = \{x_{i_1}, \ldots, x_{i_r}, y_1, \ldots, y_r\}$ are all distinct variables (why?).

$$\mathsf{Sym}_d(x_1, \ldots, x_n) \quad = \quad x_{i_1} \cdots x_{i_r} \cdot y_1 \cdots y_r \cdot \mathsf{Sym}_{d-2r}(\mathbf{x} \setminus S) \quad + \quad \text{other monomials}$$

Replacing each x_{i_j} by ℓ'_j introduces higher powers of $y_1, \ldots, y_r$.

Claim 9.9. *If we collect all monomials in $\rho(\mathsf{Sym}_d)$ that are divisible by $y_1^2 \cdots y_r^2$, it is precisely*

$$y_1^2 \cdots y_r^2 \cdot \mathsf{Sym}_{d-2r}(\mathbf{x} \setminus S).$$

[1] This isn't yet finished; will add this shortly

Exercise 9.1 *Prove this claim.*

That is, such monomials can *only* be generated by the first term in the above equation. Now, if we only choose to differentiate by monomials in $\mathbf{x} \setminus S$, the remaining monomials do not interfere with the monomials that are divisible by $y_1^2 \cdots y_r^2$. Therefore we get

$$\dim \partial^{=k}(\rho(\mathsf{Sym}_d)) \;\geq\; \dim \partial^{=k} \mathsf{Sym}_{d-2r}(\mathbf{x} \setminus S) \;=\; \min\left(\binom{n-2r}{d-2r-k}, \binom{n-2r}{k}\right). \quad \square$$

Chapter 10

Lower bounds for depth-3 circuits over finite fields

This chapter shall present the lower bound of Grigoriev and Karpinski [GK98] for $\Sigma\Pi\Sigma$ circuit computing Det_n over finite fields. A follow-up paper of Grigoriev and Razborov [GR00] extended the result over function fields, also including a weaker lower bound for the permanent, but we shall present a slightly different proof that works for the permanent as well.

Theorem 10.1. *[GK98] Any depth-3 circuit computing* Det_d *(or* Perm_d*) over a finite field* $\mathbb{F}_q$ *requires size* $2^{\Omega_q(d)}$.

We shall also prove a similar lower bound for a version of the elementary symmetric polynomial Sym_d.

Theorem 10.2. *Let* $n = d^2$*. Then, over any finite field* $\mathbb{F}_q$*, any depth-*3 *circuit computing the polynomial* $\mathsf{Sym}_{\leq d}$ *defined as*

$$\mathsf{Sym}_{\leq d} \quad \overset{def}{=} \quad \sum_{j \leq d} \mathsf{Sym}_j \quad = \quad \sum_{\substack{T \subset [n] \\ |T| \leq d}} \prod_{i \in T} x_i$$

must be of size $\exp(\Omega_q(d \log n))$.

To contrast this, over any field of at least $(n+1)$ elements, the polynomial $\mathsf{Sym}_{\leq d}$ can be computed by an $O(n^2)$ sized depth-3 circuit! This fact is attributed to Ben-Or but is a really nice exercise.

Exercise 10.1 *Let $\mathbb{F}$ be a field with at least $(n+1)$ elements. Show that, for every $d \leq n$, there is a $\Sigma\Pi\Sigma$ computing Sym_d of size $O(n^2)$.*
Hint: *Consider $(1 + tx_1) \cdots (1 + tx_n)$*

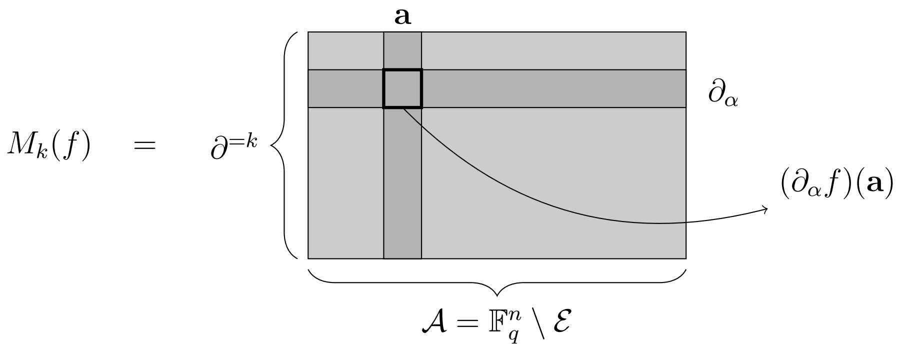
Main idea: We are working with the field $\mathbb{F}_q$ of q elements. Suppose $C = T_1 + \cdots + T_s$, where each T_i is a product of linear polynomials. Define $\mathrm{rank}(T_i)$ as in Subsection 8.1.3 to be the dimension of the set of linear polynomials that T_i is a product of.

In Subsection 8.1.3, we saw that the dimension of partial derivatives would handle *low rank* T_i's. As for the high rank T_i's, the fact that we are working over a finite field would become very useful. Since T_i is a product of at least r linearly independent linear polynomials, a random evaluation keeps T_i non-zero with probability at most $\left(1 - \frac{1}{q}\right)^r$. As q is a constant, we have that a random evaluation of a high rank T_i is almost always zero. Hence, in a sense, C can be "approximated" by just the low-rank components.

Grigoriev and Karpinski [GK98] formalize the above idea as a measure by combining the partial derivative technique seen in Subsection 8.1.3 with evaluations to show that Det_d cannot be approximated by a low-rank $\Sigma\Pi\Sigma$ circuit.

10.1 The complexity measure

For any polynomial $f \in \mathbb{F}_q[x_{11}, \ldots, x_{nn}]$, define the matrix $M_k(f)$ as follows — the columns of $M_k(f)$ are indexed by k-th order partial derivatives of f, and rows by elements of $\mathbb{F}_q^n$, with the entry being the evaluation of the partial derivative (column index) at the point (row index).


$M_k(f)$
=
$\partial^{=k}$
$\mathbf{a}$
∂_α
$(\partial_\alpha f)(\mathbf{a})$
$\mathcal{A} = \mathbb{F}_q^n \setminus \mathcal{E}$

The rank of $M_k(f)$ could be a possible choice of a complexity measure but it is not sure if $\mathrm{rank}(M_k(f))$ is small when f is even a single high rank term. However, Grigoriev and

Karpinski handle this by make a small modification to handle the high rank T_is. Instead, they look at the matrix $M_k(f)$ and remove a few *erroneous* evaluation points and use the rank of the resulting matrix. For any $\mathcal{A} \subseteq \mathbb{F}_q^n$, let us define $M_k(f;\mathcal{A})$ to be the matrix obtained from $M_k(f)$ by only taking the rows whose indices are in $\mathcal{A}$. Also, let $\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(f)$ denote $\mathrm{rank}(M_k(f;\mathcal{A}))$.

10.2 Upper-bounding $\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}$ for a depth-3 circuit

Our task here is to give an upper bound on the complexity measure for a $\Sigma\Pi\Sigma$-circuit of size s. We first see that the task reduces to upper bounding the measure for a single term via subadditivity. It follows from the linearity of the entries of the matrix.

Observation 10.3 (Sub-additivity). $\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(f+g) \quad \leq \quad \Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(f) + \Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(g)$.

Let us fix a threshold τ, and let $k = \tau/10q$ where q is the size of the finite field we are working over. The exact value of τ shall be fixed shortly depending on the polynomial we are proving a lower bound for. For Det_d, we shall choose $\tau = \Omega(d)$. For the elementary symmetric polynomial Sym_d, we shall choose $\tau = \Omega(d \log n)$.

We shall call a term $T = \ell_1 \cdots \ell_d$ to be of *low rank* if $\mathrm{rank}(T) \leq \tau$, and *large rank* otherwise. By the above observation, we need to upper-bound the measure $\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}$ for each term T, for a suitable choice of $\mathcal{A}$.

Low rank terms ($\mathrm{rank}(T) \leq \tau$)**:**

Suppose $T = \ell_1 \cdots \ell_d$ with $\{\ell_1, \ldots, \ell_r\}$ being a maximal independent set of linear polynomials (with $r \leq \tau$). Then T can be expressed as a linear combination of terms from the set $\{\ell_1^{e_1} \ldots \ell_r^{e_r} \ : \ e_i \leq d \quad \forall i \in [r]\}$. And since the matrix $M_k(f)$ depends only on evaluations in $\mathbb{F}_q^n$, we can use the relation that $x^q = x$ in $\mathbb{F}_q$ to express the evaluation of T on $\mathbb{F}_q^n$ as a linear combination of $\{\ell_1^{e_1} \ldots \ell_r^{e_r} \ : \ e_i < q \quad \forall i \in [r]\}$. Therefore, for any set $\mathcal{A} \subseteq \mathbb{F}^n$, we have that

$$\Gamma_{k;\mathcal{A}}^{[\mathrm{GK}]}(T) \quad \leq \quad \mathrm{rank}(M_k(f)) \quad \leq \quad q^r \quad \leq \quad q^\tau.$$

High rank terms ($\mathrm{rank}(T) > \tau$)**:**

Suppose $T = \ell_1 \ldots \ell_d$ whose rank is greater than τ, and let $\{\ell_1, \ldots, \ell_r\}$ be a maximal independent set. We want to use the fact that since T is a product of at least r independent linear polynomials, most evaluations would be zero. We shall be choosing our $\mathcal{A}$ to be the set where all k-th order partial derivatives evaluate to zero.

For each non-constant ℓ_i, we know that $\Pr_{\mathbf{a}}[\ell_i(\mathbf{a}) = 0] = 1/q$. Further, if the ℓ_is are linearly independent, then they *independently* evaluate to zero with probability $1/q$. Thus, on expectation, a random point $\mathbf{a}$ would evaluate to zero on $r/q > \tau/q$ of them. Since we chose $k = \tau/10q \ll r/q$, an application of Chernoff's bound shows that

$$\Pr_{\mathbf{a}}\left[\ \mathbf{a} \text{ evaluates to zero on at most } k \text{ of the factors of } T\right] \quad \leq \quad \exp(-\tau/8)$$

Let $\mathcal{E}_T$ be the set of $\mathbf{a}$s in the above event. Then every $\mathbf{a}$ outside $\mathcal{E}_T$ evaluates at least $(k+1)$ factors of T to zero. Thus, not only is T zero at $\mathbf{a}$ but so are all its k-th order partial derivatives.

Let $\mathcal{E} = \bigcup_{T \text{ of large rank}} \mathcal{E}_T$ and let $\mathcal{A} = \mathbb{F}_q^n \setminus \mathcal{E}$. Then, for any $T \in C$ that has large rank, the matrix $M_k(T;\mathcal{A})$ is simply the zero matrix and hence $\Gamma_{k;\mathcal{A}}^{[\mathrm{GK}]}(T) = 0$. This means that $\Gamma_{k;\mathcal{A}}^{[\mathrm{GK}]}$ is entirely contributed by the low-rank terms.

Lemma 10.4 (Upper bound on a small circuit)**.** *Let C be a depth-3 circuit over the field $\mathbb{F}_q$ of size at most s. Then, for any $\tau > 0$ and $k \leq \tau/10q$, there exists a set $\mathcal{E}$ of size at most $s \cdot \exp(-\tau/8) \cdot q^n$ such that*

$$\Gamma_{k;\mathcal{A}}^{[\mathrm{GK}]}(C) \quad \leq \quad s \cdot q^{\tau}$$

where $\mathcal{A} = \mathbb{F}_q^n \setminus \mathcal{E}$.

All that is left to do now is lower bound the measure for an explicit polynomial, and set the parameter τ appropriately. We shall first consider the polynomial $\mathsf{Sym}_{\leq d}$ which is a little simpler, and the look at Det_d and Perm_d.

10.3 Lower bound for Det_d and Perm_d

For the polynomials Det_d and Perm_d, the number of variables is $n = d^2$. The key technical lemma is to show that $\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(\mathsf{Det}_d)$ is large as long as $|\mathcal{A}| = (1 - o(1))q^{d^2}$.

Lemma 10.5. *For any set $\mathcal{A} \in \mathbb{F}_q^{d^2}$ such that $|\mathcal{A}| = (1 - o(1)) \cdot q^{d^2}$, we have that*

$$\Gamma_{k,\mathcal{A}}^{[\mathrm{GK}]}(\mathsf{Det}_d) \quad = \quad \binom{d}{k}^2$$

The same bound shall also hold for Perm_d. We shall defer this theorem for later and see how this would imply the proof of Theorem 10.1.

Proof of Theorem 10.1. Let $\tau = \alpha d$, for a constant $\alpha > 0$ that shall be chosen shortly, and $k = \tau/10q$. Assume that there is a $\Sigma\Pi\Sigma$ circuit of size $s < \exp(\tau/10) = \exp(\alpha d/10)$ that computes Det_d. Then, by Lemma 10.4, there is a set $\mathcal{A}$ of size $(1 - o(1))q^{d^2}$ such that

$$\Gamma^{[\mathrm{GK}]}_{k;\mathcal{A}}(\mathsf{Det}_d) \quad \leq \quad s \cdot q^{\alpha d}$$

On the other hand, by Lemma 10.5 we have

$$\Gamma^{[\mathrm{GK}]}_{k,\mathcal{A}}(\mathsf{Det}_d) \quad = \quad \binom{d}{k}^2 \quad = \quad \Omega\left(2^{2d \cdot H_2(\alpha/10q)}\right)$$

where $H_2(\gamma)$ is the binary entropy function (Definition 4.1). Together, this forces

$$\begin{aligned} s \cdot q^{\alpha d} \quad &\geq \quad \Omega\left(2^{2d \cdot H_2(\alpha/10q)}\right) \\ \implies \log s \quad &= \quad \Omega((2H_2(\alpha/10q) - \alpha \log q) \cdot d) \end{aligned}$$

The binary entropy function satisfies $H_2(\varepsilon) > \varepsilon \log_2(1/\varepsilon)$. Hence, we can always set α to be small enough constant to ensure that $2H_2(\alpha/10q) - \alpha \log q > 0$. Thus we get $s = \exp(\Omega_q(d))$. □

We only need to complete the proof of Lemma 10.5.

10.3.1 Proof of Lemma 10.5

We now wish to show that $M_k(\mathsf{Det}_d; \mathcal{A})$ has large rank. The original proof of Grigoriev and Karpinski is tailored specifically for the determinant, and does not extend directly to the permanent. The following argument is a proof communicated by Srikanth Srinivasan [Sri13] that involves an elegant trick that he attributes to [Kou08]. The following proof is presented for the determinant, but immediately extends to the permanent as well.

Note that if we were to just consider $M_k(\mathsf{Det}_d)$, it would have been easy to show that the rank is full by looking at just those evaluation points that keep exactly one $(d-k) \times (d-k)$ minor non-zero (set the main diagonal of the minor to ones, and every other entry to zero). Hence, $M_k(\mathsf{Det}_d)$ has the identity matrix *embedded inside* and hence must be full rank. However, we are missing a few of the evaluations (since a small set $\mathcal{E}$ of evaluations is removed) and we would still like to show that the matrix continues to have full column-rank.

Lemma 10.6. *Let $p(X)$ be a non-zero linear combination of $r \times r$ minors of the matrix $X = (\!(x_{ij})\!)$. Then,*

$$\Pr_{A \in \mathbb{F}_q^{d^2}}[p(A) \neq 0] \quad \geq \quad \frac{q-2}{q-1}.$$

This immediately implies that for every linear combinations of the columns of $M_k(\mathsf{Det}_d)$, a constant fraction of the coordinates have non-zero values. Since we are removing merely a set $\mathcal{E}$ of size $o(1) \cdot q^{d^2}$, there must continue to exist coordinates that are non-zero. In other words, no linear combination of columns of $M_k(\mathsf{Det}_d; \mathcal{A})$ results in the zero vector.

The proof of the above lemma would be an induction on the number of minors contributing to the linear combination. As a base case, we shall use a well-known fact about Det_d and Perm_d of random matrices.

Proposition 10.7. *If A is a random $d \times d$ matrix with entries from a fixed finite field $\mathbb{F}_q$,*

$$\Pr[\det(A) \neq 0] \quad \geq \quad \frac{q-2}{q-1}.$$

The proof of this is in fact a nice exercise, and we give a few hints at the end of this section. Let us begin with the proof of Lemma 10.6.

Proof of Lemma 10.6. If $p(X)$ is a scalar multiple of a single non-zero minor, then we already have the lemma from Proposition 10.7. Hence, let us assume that there are at least two distinct minors participating in the linear combination $p(X)$. Without loss of generality, assume that the first row occurs in some of the minors, and does not in others. That is,

$$\begin{aligned} p(X) \quad &= \quad \left(\sum_{i : \text{Row}_1 \in M_i} c_i M_i\right) \quad + \quad \left(\sum_{j : \text{Row}_1 \notin M_j} c_j M_j\right) \\ &= \quad (x_{11} M_1' + \cdots + x_{1d} M_d') \quad + \quad M'' \quad \text{(expanding along the first row).} \end{aligned}$$

To understand a random evaluation of $p(X)$, let us first set rows $2, \ldots, d$ to random values, and then setting row 1 to random values.

$$\begin{aligned} \Pr_A[p(A) \neq 0] \quad \geq \quad & \Pr[x_{11} M_1' + \cdots + x_{1d} M_d' + M'' \neq 0 \mid \text{some } M_i' \neq 0] \\ & \times \Pr[\text{some } M_i' \neq 0] \end{aligned}$$

Note that once we have set rows $2, \ldots, d$ to random values, $p(X)$ reduces to a linear polynomial in $\{x_{11}, \ldots, x_{1d}\}$. Further, a random evaluation of any non-constant linear polynomial is zero with probability exactly $\left(1-\frac{1}{q}\right)$. Hence,

$$\begin{aligned}\Pr_A[p(A)\neq 0] &\geq \Pr[x_{11}M_1'+\cdots+x_{1d}M_d'+M''\neq 0 \mid \text{some } M_i'\neq 0]\\ &\quad\times\Pr[\text{some } M_i'\neq 0]\\ &= \left(1-\frac{1}{q}\right)\cdot\Pr[\text{some } M_i'\neq 0].\end{aligned}$$

Now comes Koutis' Trick: the term $\left(1-\frac{1}{q}\right)\cdot\Pr[\text{some } M_i'\neq 0]$ is exactly the probability that $x_{11}M_1'+\cdots+x_{1d}M_d'$ is non-zero! Hence,

$$\begin{aligned}\Pr_A[p(A)\neq 0] &= \Pr[x_{11}M_1'+\cdots+x_{1d}M_d'+M''\neq 0]\\ &\geq \Pr[x_{11}M_1'+\cdots+x_{1d}M_d'\neq 0]\\ &= \Pr\left[\left(\sum_{i:\mathrm{Row}_1\in M_i} c_iM_i\right)\neq 0\right].\end{aligned}$$

which is just the linear combination obtained by only considering those minors that contain the first row. Repeating this process for other rows/columns until only one minor remains, we have

$$\Pr_A[p(A)\neq 0] \quad\geq\quad \Pr_A[\det(A)\neq 0] \quad=\quad \frac{q-2}{q-1} \quad \text{(by Proposition 10.7).} \qquad \square$$

Exercise 10.2 [Proving Proposition 10.7] *Let $\{q_n\}$ be a sequence of non-negative reals such that $\sum q_n$ converges to some $s<1$. Show that*

$$\prod_{i=1}^{\infty}(1-q_n) \quad\geq\quad 1-s.$$

Use this to infer that the probability that a random $n\times n$ matrix A over $\mathbb{F}_q$ is invertible, which is exactly

$$\left(1-\frac{1}{q}\right)\cdots\left(1-\frac{1}{q^n}\right),$$

is at least $1/4$.

Exercise 10.3 [Proving Proposition 10.7 for Perm_n] *Show that*

$$\begin{aligned}\Pr_{A\in\mathbb{F}_q^{n\times n}}[\mathsf{Perm}(A)=0] &\leq \frac{1}{q^n}+\frac{1}{q^{n-1}}+\cdots+\frac{1}{q}\\ &= \frac{1}{q-1}\left(1-\frac{1}{q^n}\right).\end{aligned}$$

Hint: Use Koutis' trick (Lemma 10.6).

10.4 Lower bound for $\mathsf{Sym}_{\leq d}$

The goal of this section would be to get a lower bound on $\Gamma^{[\mathrm{GK}]}_{k;\mathcal{A}}(\mathsf{Sym}_{\leq d})$, where $n=d^2$, when k is suitably chosen. In this case, we shall set $\tau=\alpha\cdot d\log n$ for a constant $\alpha>0$ that shall be chosen shortly, and $k=d/2$. The main lemma of this section would be the following.

Lemma 10.8. *Consider the polynomial* $\mathsf{Sym}_{\leq d}$ *with* $n=d^2$*. If* $k=d/2$ *and* $\mathcal{A}=\mathbb{F}_q^n\setminus\mathcal{E}$ *with* $|\mathcal{E}|\leq\exp(-\omega(d\log q))\cdot q^n$*, then*

$$\Gamma^{[\mathrm{GK}]}_{k;\mathcal{A}}(\mathsf{Sym}_{\leq d}) \quad\geq\quad \binom{n}{d/2}$$

We shall prove this lemma shortly but let us first see how this implies Theorem 10.2.

Proof of Theorem 10.2. Assume that the polynomial $\mathsf{Sym}_{\leq d}$ can be computed by a $\Sigma\Pi\Sigma$ circuit of size $s\leq\exp(\tau/10)=\exp((\alpha/10)\cdot d\log n)$. Then, by Lemma 10.4, there is a set $\mathcal{E}$ with $|\mathcal{E}|\leq\exp(-\Omega(d\log n))\cdot q^n\ll\exp(-\omega(d\log q))\cdot q^n$ such that

$$\Gamma^{[\mathrm{GK}]}_{k;\mathcal{A}}(\mathsf{Sym}_{\leq d}) \quad\leq\quad s\cdot q^{\alpha\cdot d\log n}$$

where $k=d/2$ and $\mathcal{A}=F_q^n\setminus\mathcal{E}$. On the other hand, from Lemma 10.8 we get

$$\Gamma^{[\mathrm{GK}]}_{k;\mathcal{A}}(\mathsf{Sym}_{\leq d}) \quad\geq\quad \binom{n}{d/2} \quad\geq\quad 2^{(d/4)\log d}$$

These two bounds force $s\geq\exp(\Omega_q(d\log d))=\exp(\Omega_q(d\log n))$. □

We only need to prove Lemma 10.8 to complete the proof.

10.4.1 Proof of Lemma 10.8

We have set $k = d/2$ and are hence studying the evaluations partial derivatives of $\mathsf{Sym}_{\leq d}$ of order $d/2$ on points in $\mathcal{A}$. Let us first focus on the partial derivatives as formal polynomials.

Consider the following matrix M where each rows is indexed by a partial derivative of order $k = d/2$, columns indexed by monomials of degree $d - d/2 = d/2$ and the corresponding entry being 1 if that monomial occurs in partial derivative of $\mathsf{Sym}_{\leq d}$ and zero otherwise. That is, each row of this matrix is a partial derivative of $\mathsf{Sym}_{\leq d}$ written down via its coefficients. Now notice that if we have a partial derivative corresponding to a subset S, a monomial corresponding to a subset T would be present in $\partial_S(\mathsf{Sym}_{\leq d})$ if and only if $|T| \leq d/2$ and $S \cap T = \emptyset$. These $\binom{n}{d/2} \times \binom{n}{\leq d/2}$ matrices have been well studied and are called *Disjointness Matrices*. The following lemma of Razborov [Raz87] shows that these matrices are full-rank.

Lemma 10.9 (Rank of disjointness matrices [Raz87]). *Let D be a $\binom{n}{d} \times \binom{n}{\leq d}$ matrix such that*

$$D(S,T) \quad = \quad \begin{cases} 1 & \text{if } S \cap T = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Then, over any field $\mathbb{F}$, the matrix D has rank $\binom{n}{d}$. □

Thus stated differently, this says that the $(d/2)$-th partial derivatives of $\mathsf{Sym}_{\leq d}$ are linearly independent as formal polynomials over any field $\mathbb{F}$. We would like to use this to infer that the matrix $M_k(f; \mathcal{A})$ is also full-rank as long as $\mathcal{A} = \mathbb{F}_q^n \setminus \mathcal{E}$ and $|\mathcal{E}| \leq \exp(-\omega(d \log q)) \cdot q^n$.

Consider any linear combination of $(d/2)$-th order partial derivatives of $\mathsf{Sym}_{\leq d}$. We know that this is a non-zero multilinear polynomial of degree $d/2$. The following exercise shows that non-zero multilinear polynomials of low-degree must have many non-zero evaluations.

Exercise 10.4 [Schwartz-Zippel over small fields] *Show that, over any field $\mathbb{F}_q$, any non-zero multilinear n-variate polynomial f of degree d evaluates to a non-zero value on at least q^{n-d} points of $\mathbb{F}_q^n$.*

Thus, any linear combination of partial derivatives must be non-zero on at least $q^{-d/2} \cdot q^n = \exp(-(d/2)\log q) \cdot q^n$ points. Since $|\mathcal{E}| < \exp(-\omega(d \log q) \cdot q^n$, we cannot have thrown away all the non-zero evaluations. Thus, for every linear combination of partial derivatives, there must exist some point $\mathbf{a} \in \mathcal{A}$ on which it evaluates to a non-zero value. In other words, no linear combination of rows of $M_k(\mathsf{Sym}_{\leq d}; \mathcal{A})$ yields the zero row and hence is full rank. $\square$

Part IV

Multilinear and non-commutative models

Chapter 11

The Partial Derivative Matrix

In this chapter, we shall look at a powerful technique introduced by Nisan [Nis91] that has been instrumental in many lower bound proofs and also in constructing polynomial identity tests. Nisan [Nis91] introduced the notion of the partial derivative matrix in the context of proving lower bounds for *non-commutative* ABPs, and we shall see that first.

11.1 Non-commutative models of computation

A non-commutative polynomials over many variables, denoted by $\mathbb{F}\{x_1, \ldots, x_n\}$, are formal polynomials over the variables wherein the variables do not commute. Hence, a polynomial $x_1x_2 - x_2x_1$ in $\mathbb{F}\{x_1, \ldots, x_n\}$ is a non-zero polynomial. They naturally can be added or multiplied but the order in which the variables are multiplied become important. Hence,

$$(x_1 + x_2)(x_1 + x_2) \quad = \quad x_1^2 + x_2x_1 + x_1x_2 + x_2^2 \quad \neq \quad x_1^2 + 2x_1x_2 + x_2^2$$

Each monomial is no longer identifiable by just an exponent vector but is rather a *word* on the set $\{x_1, \ldots, x_n\}$.

In this space, we can continue to talk about arithmetic circuits or algebraic branching programs where we always keep track of the order of variables multiplied. In arithmetic circuits or formulas, every $\times$ gate has labelled left and right children. In an algebraic branching program, the weight of a path from source to sink is the product of the edge weights *in the order from left to right.*

Nisan [Nis91] asked the question of whether we can prove lower bounds in this more restricted model of computation. In his paper, he introduced the complexity measure via the

Partial Derivative Matrix, and used it to not just prove lower bounds but exactly calculate the size of the smallest non-commutative ABP computing a homogeneous polynomial f.

Exercise 11.1 *Show that, given any non-commutative ABP of size s computing a homogeneous non-commutative polynomial of degree d, we can construct a homogeneous non-commutative ABP (edge weights are homogeneous linear forms) of size at most $s \cdot \mathrm{poly}(d)$ computing f.*

11.1.1 Partial derivative matrix for non-commutative ABPs

Definition 11.1 (Nisan's partial derivative matrix [Nis91]). *Let f be an n-variate homogeneous non-commutative polynomial of degree d. For any $i \in [d]$, the matrix $M_i(f)$ is defined follows:*

> *The matrix $M_i(f)$ has n^i rows and n^{d-i} columns, indexed by monomials (or words) of length i and $d-i$ respectively. The entry at (m_1, m_2) is the coefficient of the monomial (or word) $m_1 \cdot m_2$ in f.* ♢

Theorem 11.2 ([Nis91]). *For any n-variate homogeneous non-commutative polynomial f of degree d, the smallest non-commutative ABP that computes f must have size*

$$\mathrm{rank}(M_0(f)) + \mathrm{rank}(M_1(f)) + \cdots + \mathrm{rank}(M_{d-1}(f)).$$

The above is not an estimate; Nisan's result says that the sum of the ranks of the partial derivative matrix is *exactly* the size of the smallest ABP. The proof of this theorem is not hard, especially once you know what the answer is. We shall prove part of the proof to show that the sum of the ranks is a lower bound for the size of the smallest ABP, and leave the other direction as an exercise.

Proof. Let C be the smallest non-commutative ABP computing the polynomial f. We shall show that number of vertices in layer i is at least the rank of $M_i(f)$.

Suppose $v_1, \ldots, v_r$ are the vertices in the i-th layer, and let s be the unique source node and let t be the unique sink node. For each $i \in [r]$, let g_i be the non-commutative polynomial computed by the restricted ABP if we consider s as the source and v_i as the sink. Similarly, let h_i be the non-commutative polynomial computed by the restricted ABP with v_i as source and t as the sink. Then, $f = g_1 h_1 + \cdots + g_r h_r$. Since the ABP is homogeneous, each g_i

is a homogeneous non-commutative polynomial of degree i and each h_i is a homogeneous non-commutative polynomial of degree $d-i$. Now consider the matrix the $n^i \times r$ matrix G, with rows indexed by monomials (or words) of degree i and columns indexed by $[r]$, with the (m,i) entry being the coefficient of m in g_i. Similarly, let H be the $r \times n^{d-i}$ matrix, with rows indexed by $[r]$ and columns indexed by monomials (or words) of degree $d-i$, with the (j,m) entry being the coefficient of m in g_j.

Subclaim 11.3. *$M_i = G \cdot H$.*

With this claim, it follows that the rank of M is a lower bound for r. □

Exercise 11.2 *Complete the proof of Subclaim 11.3, and also show that the bound is tight to show the other direction of Theorem 11.2.*

11.1.2 An explicit hard polynomial

To complete the proof, we just need to construct an explicit polynomial for which one of the M_i's has large rank. A natural attempt is to make $M_{d/2}$ to be full-rank by making it something like the identity matrix. Indeed, if we choose the polynomial to be the *double polynomial* Doub_d defined as

$$\mathrm{Doub}_d \quad := \quad \sum_{w \in \{x_1,\ldots,x_n\}^{d/2}} \mathbf{x}_w \mathbf{x}_w$$

or the *Palindrome polynomial* Pal_d defined as

$$\mathrm{Pal}_d \quad := \quad \sum_{w \in \{x_1,\ldots,x_n\}^{d/2}} \mathbf{x}_w \mathbf{x}_{\mathrm{reverse}(w)},$$

then clearly $\mathrm{rank}(M_{d/2})$ is $n^{d/2}$ giving the required lower bound.

Theorem 11.4 ([Nis91])**.** *Any non-commutative ABP computing the polynomial* Doub_d *or* Pal_d *must have size $n^{\Omega(d)}$.*

As an added bonus, it is easy to see that Pal_d can in fact be computed by a non-commutative circuit of size $\mathrm{poly}(n,d)$. Thus, this in fact yields an exponential separation between non-commutative ABPs and non-commutative circuits.

An important point to also observe is that this lower bound implies that an analogue of the depth reduction of [VSBR83] is simply not possible in the non-commutative world.

11.2 Applications in the commutative world

There are some instances in the commutative world where computation behaves like a non-commutative computation. An example of this is the class of what are called *read-once oblivious algebraic branching programs (ROABP)*, first defined by Forbes and Shpilka [FS13].

Definition 11.5 (Read-once oblivious algebraic branching programs (ROABP) [FS13])**.** *An ABP (over commuting variables) is said to be an* read-once oblivious algebraic branching program (ROABP) in the order $(x_1, \ldots, x_n)$ *if it has the property that all edge weights between layer i and layer $i+1$ are* univariate polynomials *in x_i.* $\Diamond$

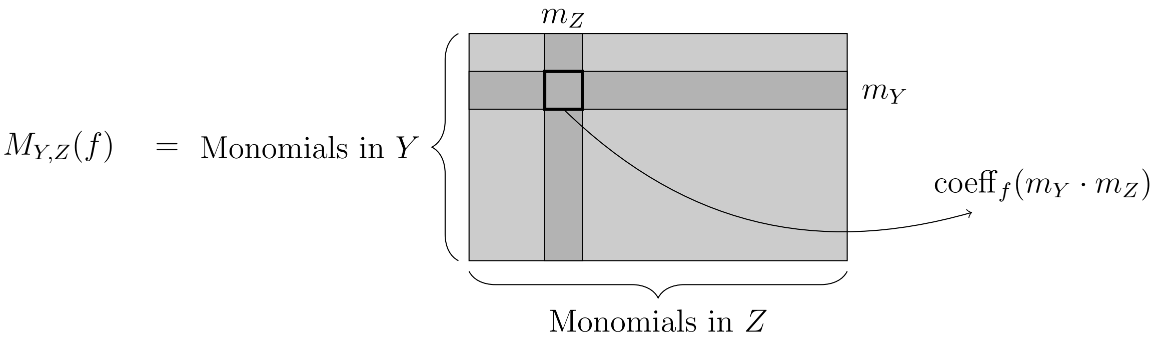
As seen in Chapter 2, if f is computable by an ROABP of width w, then f can be equivalently expressed as one entry of an iterated product of *univariate* matrices. That is,

$$f \quad = \quad (A_1(x_1) \cdots A_n(x_n))_{(1,1)}$$

where each $A_i(x_i)$ is a $w \times w$ matrix (where w is the width of the ABP) with entries as univariate polynomials in x_i.

In some sense, this is essentially a non-commutative computation that is masquerading as a commutative computation since the variables are multiplied in the same order. Thus, the partial derivative matrix that was used in the non-commutative ABP lower bound can also be used here. We shall abuse notation and use the following definition of the partial derivative matrix that is more useful for the commutative world.

Definition 11.6 (Partial derivative matrix for a partition)**.** *For any given partition of variables $X = Y \sqcup Z$, define the* partial derivative matrix $M_{Y,Z}(f)$ *to be the matrix described as follows — the rows are indexed by monomials in Y, columns indexed by monomials in Z, and the (i,j)-th entry of the matrix is the coefficient of the monomial $m_i(Y) \cdot m_j(Z)$ in f. Further, we shall call a polynomial f to be* full-rank *if $M_{Y,Z}(f)$ is full-rank.* $\Diamond$



The following observation is a an easy exercise.

Lemma 11.7. *Suppose a polynomial f is can be computed as an entry in a product of $w \times w$ univariates matrices in the order $(x_1, \ldots, x_n)$, that is*

$$f \quad = \quad (A_1(x_1) \cdots A_n(x_n))_{(1,1)} .$$

Then, for every $i \in [n]$, for the partition $Y = \{x_1, \ldots, x_i\}$ and $Z = X \setminus Y$ we have $\operatorname{rank}(M_{Y,Z}(f)) \leq w$.

Furthermore, the converse also holds.

Hence, the rank of the partial derivative matrix can be used to prove lower bounds for ROABPs.

Exercise 11.3 *What is the rank of the partial derivative matrices for the following polynomials?*

- *The elementary symmetric polynomials Sym_d, under any partition.*
- *Any $\Sigma\wedge\Sigma$ circuit of size s, under any partition.*
- *Det_n and Perm_n under the partition $X = Y \sqcup Z$ where Y the variables from the first $n/2$ rows.*
- *The polynomial $(x_1 + x_2) \cdots (x_{2n-1} + x_{2n})$ under the partition $X = Y \sqcup Z$ with $Y = \{x_1, \ldots, x_n\}$.*
- *The polynomial $(x_1 + x_2) \cdots (x_{2n-1} + x_{2n})$ under the partition $X = Y \sqcup Z$ with $Y = \{x_1, x_3, x_5, \ldots, x_{2n-1}\}$.*

11.2.1 An evaluation perspective

For this section, let us assume that we are working with a field that is large enough. The following is an alternative way to study the rank of the partial derivative matrix under some partition. It is essentially the same, but sometimes is easier to reason with and we shall see a few examples in this chapter.

Definition 11.8 (Evaluation dimension)**.** *Let $X = Y \sqcup Z$. The evaluation dimension of a polynomial f, with respect to $Y \sqcup Z$, denoted by $\operatorname{evalDim}_{Y,Z}(f)$, is defined as the rank of the* *following polynomials*

$$\mathrm{evalDim}_{Y,Z}(f) \quad = \quad \mathrm{rank}\left(\left\{f(\mathbf{a}, Z) \in \mathbb{F}[Z] \ : \ \mathbf{a} \in \mathbb{F}^{|Y|}\right\}\right).$$

In other words, $\mathrm{evalDim}_{Y,Z}(f)$ *of the space of partial evaluations of* f *by setting* Y *to arbitrary field constants.* $\Diamond$

The following lemma is easy to verify.

Lemma 11.9. *Over any field* $\mathbb{F}$*, we always have that* $\mathrm{evalDim}_{Y,Z}(f) \leq \mathrm{rank}(M_{Y,Z}(f))$*. If* $|\mathbb{F}| \geq \deg(f)$*, then*

$$\mathrm{evalDim}_{Y,Z}(f) \quad = \quad \mathrm{rank}(M_{Y,Z}(f)).$$

To illustrate why this perspective is convenient, we shall take one example from Exercise 11.3 and compute the evaluation dimension of a $\Sigma\wedge\Sigma$ circuit.

Claim 11.10. *Let* $f = \sum_{i=1}^{s} \ell_i^d$*. Then under any partition* $X = Y \sqcup Z$*, we have that* $\mathrm{evalDim}_{Y,Z}(f) \leq (d+1) \cdot s$*.*

Proof. It suffices to prove that the evaluation dimension of a single ℓ^d is at most $(d+1)$ and the lemma would follow due to sub-additivity. Say $\ell = (a_0 + a_1 x_1 + \cdots + a_n x_n)$. For any partition $X = Y \sqcup Z$, let $\ell_Y = \sum_{i \in Y} a_i x_i$ and $\ell_Z = \ell - \ell_Y$ so that $\ell = \ell_Y + \ell_Z$. Now if we take a partial evaluation of the Y variables to field constants, the resulting polynomial is

$$(\alpha + \ell_Z)^d \quad = \quad \alpha^d \ + \ \alpha^{d-1}\binom{d}{1}\ell_Z \ + \ \cdots \ + \ \ell_Z^d.$$

And as we change the evaluation, the only change in the above equation is the value of α. Hence it is clear that the rank of this space of polynomials is no more than $(d+1)$ as it is spanned by

$\square \qquad \left\{1, \ell_Z, \ldots, \ell_Z^d\right\}.$

Exercise 11.4 *Repeat Exercise 11.3 with evaluation dimension.*

Exercise 11.5 *Suppose you have a degree d polynomial f so that $\dim \partial^{\leq d}(f)$ is at most r, that is, there are at most r linearly independent partial derivatives of f. What can you say about* $\mathrm{evalDim}_{Y,Z}(f)$ *under any partition?*

What about the converse?

The notion of partial derivative matrix is very useful and in the next chapter we shall see how it can be used to prove lower bounds for multilinear models.

Chapter 12

Hardness amplification for non-commutative circuits

In Chapter 11, we saw exponential lower bounds for the non-commutative formulas and ABPs. As mentioned earlier, in the commutative world, this would have immediately yielded lower bounds for circuits as well due to the depth reduction results (Theorem 5.8). However this does not happen in the non-commutative world, and proving lower bounds for non-commutative circuits has remained elusive.

In this chapter, we shall see a beautiful result of Carmasino, Impagliazzo, Mihajlin and Lovett [CILM18] that show that even mildly super-linear lower bounds for a family of non-commutative polynomials can be *amplified* to obtain much stronger lower bounds.

Theorem 12.1 ([CILM18])**.** *Suppose there is some $\varepsilon > 0$ for which we have an explicit family of non-commuting polynomials $\{f_n\}$ (with f_n being an n-variate non-commuting polynomial of degree* $\mathrm{poly}(n)$*) such that f_n requires non-commutative circuits of size $n^{(\omega/2)+\varepsilon}$.*

Then, for every $c \geq 1$, there is an explicit family $\{g_n\}$ (with g_n being an n-variate polynomial of degree $\mathrm{poly}(n)$*) that requires non-commutative circuits of size n^c.*

In fact, the conclusion can be strengthened if the initial polynomial family $\{f_n\}$ is a constant degree family.

Theorem 12.2 ([CILM18])**.** *Suppose there is some $\varepsilon > 0$ for which we have an explicit family of non-commuting* constant degree *polynomials $\{f_n\}$ (with f_n being an n-variate non-commuting polynomial of degree d, which is a constant) such that f_n requires non-commutative circuits of size $n^{(\omega/2)+\varepsilon}$.*

Then, for some $\delta \geq 0$, there is an explicit family $\{g_n\}$ (with g_n being an n-variate polynomial of degree $\operatorname{poly}(n)$*) that requires non-commutative circuits of size* $\exp(n^{\delta})$*.*

In both the above theorems, ω refers to the exponent of matrix multiplication. For simplicity, we shall just work with this being replaced by 3. Hence, the above theorems say that if we can get lower bounds of $n^{1.5+\varepsilon}$, then this can be amplified.

12.1 Intuition

Both these theorems go via a clever *hardness-preserving variable reduction*. Formally, we will start with a polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ and transform this to a polynomial $g(y_1, \ldots, y_m) \in \mathbb{F}\langle y_1, \ldots, y_m \rangle$ where $m \ll n$. This transformation would be obtained by replacing each x_i by some $h_i(y_1, \ldots, y_m)$ for some relatively simple h_i's (in fact, they will just be monomials).

Clearly, if f has a small circuit, then so does g (since each of the h_i's are simple). The key point would be that a rough converse also would hold. That is, if someone provided a size s non-commutative circuit for g, then there is a circuit of size s' (which is not-much-larger-than s) that computes f. As a contrapositive, if f *cannot* be computed by circuits of size s, then g *cannot* be computed by circuits of size s'. Why is this a hardness amplification? Notice that $m \ll n$; hence s' as a function of m could be way larger than s as a function of n.

The proofs of both the theorems essentially repeat the above transformation as many times as possible. In the general case, we can repeat this any constantly many times and hence this would eventually yield Theorem 12.1. In the setting when the polynomial family has constant degree, we can repeat this more times and that yields Theorem 12.2.

We will see the details in the rest of the chapter.

12.2 The hardness-preserving variable reduction

Suppose $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a polynomial of degree d. Let $m = \lceil n^{1/3} \rceil$. To each of the variables x_i, we shall assign a unique non-commutative monomial $y_{i_1} y_{i_2} y_{i_3}$. Define the polynomial g obtained from f via by substituting the associated monomial for x_i:

$$\operatorname{Amp}_3(f) := g(y_1, \ldots, y_m) := f(y_{1_1} y_{1_2} y_{1_3}, \ldots, y_{n_1} y_{n_2} y_{n_3})$$

Clearly, g is an m-variate non-commutative polynomial with $\deg g \leq 3d$.

Lemma 12.3 (Hardness-preserving reduction)**.** *Let f be an n-variate non-commutative polynomial. Suppose there is a non-commutative circuit of size at most s that computes* $\mathrm{Amp}_3(f)$*. Then, there is a non-commutative circuit of size at most $s' = C \cdot s \cdot n^{\omega/3}$ for a universal constant C.*

We shall assume the above lemma for now and finish the proofs of Theorem 12.1 and Theorem 12.2. We shall define the following operator which applies this amplification multiple times.

$$\mathrm{Amp}_3^{(k)}(f) := \underbrace{\mathrm{Amp}_3 \circ \mathrm{Amp}_3 \circ \cdots \circ \mathrm{Amp}_3}_{k \text{ times}}(f).$$

Corollary 12.4 (Iterative hardness-preserving reduction)**.** *Let f be an n-variate non-commutative polynomial, of degree d, with $n = m^{3^k}$ for some positive integers m and k. Let $g(y_1, \ldots, y_m) =$ $\mathrm{Amp}_3^{(k)}(f)$, an m-variate non-commutative polynomial of degree $3^k d$. If g can be computed by a circuit of size s, then f can be computed by a circuit of size at most $s \cdot n^{\omega/2} \cdot C^k$ where C is the universal constant in Lemma 12.3*

Proof. By repeated applications of Lemma 12.3 yields that f can be computed by a circuit of size at most

$$\begin{aligned} s' &= s \cdot C^k \cdot \left(m^{\omega} \cdot m^{3\omega} \cdots m^{3^{k-1}\omega}\right) \\ &= s \cdot C^k \cdot m^{\omega \cdot (3^k - 1)/(3-1)} \\ &\leq s \cdot C^k \cdot m^{(\omega/2)\cdot 3^k} = s \cdot C^k \cdot n^{\omega/2}. \end{aligned}$$ □

Proof of Theorem 12.1 and Theorem 12.2. Let $g = \mathrm{Amp}_3^{(k)}(f)$ for a k that shall be fixed shortly. If $n = m^{3^k}$ of degree d, then g is an m-variate polynomial of degree $3^k \cdot d$. Corollary 12.4 states that if f requires circuits of size $\Omega(n^{(\omega/2)+\varepsilon})$ then $g(y_1, \ldots, y_m)$ requires circuits of size $C^k \cdot n^{\varepsilon}$.

- (Proof of Theorem 12.1) Set k large enough constant so that $3^k \cdot \varepsilon > c$. Then $\deg(g) = 3^k d = \mathrm{poly}(n) = \mathrm{poly}(m)$ as k is a constant. Also, g requires circuits of size $\Omega(n^{\varepsilon}) = \Omega(m^c)$ by the choice of k.
- (Proof of Theorem 12.2) Pick the smallest k such that $3^k \cdot k > \log n$; let $m = n^{1/3^k}$ and $g = \mathrm{Amp}_3^{(k)}(f)$. Note that for this choice of k, we have $k > \log m$. With this choice of

k and since $d = \deg(f)$ is a constant, we have

$$\deg(g) = 3^k \cdot d \approx 3^{(\log n)/3^k} \cdot d = 3^{\log m} \cdot d = \text{poly}(m).$$

Also, g requires circuits of size at least

$$C^k n^\varepsilon = m^{3^k \cdot \varepsilon} C^k \approx m^{3^k \cdot \varepsilon + \log C} = \Omega(\exp(m^\delta))$$

for some $\delta > 0$. □

In the rest of the chapter we shall see how to prove Lemma 12.3.

12.3 Proof of the main lemma

We are provided a circuit of size s that computes $\text{Amp}_3(f)$ and we want to use that to find a circuit for f. Basically, what we need to do is undo this monomial transformation.

Intuitively, we want to say that there shouldn't be a circuit much better than actually computing f and doing this transformation. Of course, given *such* a circuit for $\text{Amp}_3(f)$, we can just pull out a circuit for f. The main idea is that given *any* circuit for g, we can always structure the circuit in such a way that it really does look like a circuit computing f and then doing the monomial substitution.

We will first perform a *partial homogenisation* operation, very similar to Lemma 5.2 but much weaker. This would be the first step towards at least ensuring that each gate computes a polynomial of degree divisible by 3.

Definition 12.5 (A $(\bmod 3)$-homogeneous circuit)**.** *A circuit C is said to be $(\bmod 3)$-homogeneous if every gate is labelled by a pair $(i, j) \in \{1, 2, 3\}^2$ satisfying the following conditions:*

- *All leaves will be labelled by some (i, j) such that $j = i + 1 \bmod 3$ if it is a variable, and by some (i, i) if the leaf is a constant.*
- *If g is a $+$ gate with label (i, j), then all its children also have label (i, j).*
- *If $g = g_1 \times g_2$ with label (i, j) then there must be some $k \in \{0, 1, 2\}$ such that g_1 has label (i, k) and g_2 has label (k, j).* ◊

Intuitively, each gate's label of (i, j) says that its contribution will always be from a position that is $i \bmod 3$ and until $j \bmod 3$. We leave the proof of the following observation as an easy exercise.

Observation 12.6. *Any non-commutative polynomial that can be computed by a non-commutative circuit of size s can be equivalently computed by a* (mod3)*-homogeneous non-commutative circuit of size at most* $9s$*.*

Let us see what would happen if we started with the trivial circuit for $\mathrm{Amp}_3(f)$ obtained from a circuit for f and applying the monomial substitution. The resulting circuit is already (mod3)-homogeneous, and *every* gate has label $(1, 1)$ (and we are thinking of leaves as now being monomials; they also have label $(1, 1)$ then). This is the state we want to get to from an arbitrary circuit. The partial homogenisation is one step towards it but there are all kinds of labels for the gates and we want to fix that.

The beautiful idea of Carmosino et al [CILM18] is to replace each gate of the circuit by a $m \times m$ matrix of gates, such that each entry of this matrix would have label $(1, 1)$. We then want to simulate all additions and multiplications as matrix multiplications. We will need to build some notation.

Definition 12.7 (Division operators)**.** *For a polynomial $f \in \mathbb{F}\langle y_1, \ldots, y_m\rangle$ and a variable $y \in \{y_1, \ldots, y_m\}$, we shall define the left and right division operators as*

$$[y^{-1}]f = \sum_{\substack{\mathbf{w}\in\{y_1,\ldots,y_m\}^* \\ \mathbf{w}=y\mathbf{w}'}} \mathrm{coeff}_{\mathbf{w}}(f) \cdot \mathbf{w}'$$

$$f[y^{-1}] = \sum_{\substack{\mathbf{w}\in\{y_1,\ldots,y_m\}^* \\ \mathbf{w}=\mathbf{w}'y}} \mathrm{coeff}_{\mathbf{w}}(f) \cdot \mathbf{w}'$$

In words, $[y^{-1}]f$ divides from the left by y (and monomials that do not begin with y are zeroed out) and $f[y^{-1}]$ divides from the right. ♢

We now define an operator that maps every labelled gate of the circuit to a matrix of polynomials each of whose entries are (mod3)-homogeneous. We'll call it the *glacial* operator after the following sentence in [CILM18]:

> "This process is like a glacial movement during the ice age. An operator slides over the circuit and then disappears, drastically changing the landscape behind it."

Definition 12.8 (The *glacial* operator)**.** *Let g be a gate in a* (mod3)*-homogeneous non-commutative circuit C with label (a, b). Let g also denote the polynomial computed by the gate.*

The operator $\Phi(g)$ returns an $m \times m$ matrix of polynomials whose (i,j)-th entry is given by the (a,b)-th entry of the following matrix:

$$\begin{bmatrix} g & g\left[y_j^{-1}\right] & gy_j \\ y_i g & y_i g\left[y_j^{-1}\right] & y_i g y_j \\ \left[y_i^{-1}\right] g & \left[y_i^{-1}\right] g \left[y_j^{-1}\right] & \left[y_i^{-1}\right] g \left[y_j^{-1}\right] \end{bmatrix}.$$

$\Diamond$

What the above definition does is best described in plain words and a few examples. If g is labelled $(1,1)$, then basically g is both *start-aligned* and *end-aligned* and the matrix $\Phi(g)$ would just have g in every entry. If g is labelled $(2,2)$, then g "needs one symbol on the left to be added" to become *start-aligned* and "needs one symbol on the right to be removed" to become *end-aligned*. The (i,j)-th entry of $\Phi(g)$ then corresponds to $y_i g[y_j^{-1}]$, which *aligns* g by adding y_i on the left and removing y_j on the right.

Observation 12.9. *Let C be a* (mod3)*-homogenised circuit of size s. Suppose g is some internal gate in g.*

- *If $g = g_1 + g_2$, then $\Phi(g) = \Phi(g_1) + \Phi(g_2)$ (matrix addition).*
- *If $g = g_1 \times g_2$, then $\Phi(g) = \Phi(g_1) \times \Phi(g_2)$ (matrix multiplication).*

The proof of the above observation is a simple exercise and is left to the reader. Thus, all we need to do for now is replace every gate g of C by the entries of $\Phi(g)$, all addition and multiplication gates by matrix addition and matrix multiplication gates, and we would have obtained our (mod3)-homogeneous circuit with each gate labelled with $(1,1)$!

We start with the leaves of C which were computing variables. For any leaf ℓ, observe that $\Phi(\ell)$ would be an $m \times m$ matrix, each of whose entries is a monomial of degree exactly 3 or a constant. Thus, we can first compute $\Phi(\ell)$ for all the leaves ℓ. If $g = g_1 \circledast g_2$ is some internal gate for which we need to compute $\Phi(g)$, and we have computed $\Phi(g_1)$ and $\Phi(g_2)$, then we can compute the entries of $\Phi(g)$ by a suitable matrix addition/multiplication via Observation 12.9.

And finally, each matrix addition and matrix multiplication of $m \times m$ matrices can be simulated[1] by $O(m^\omega)$ usual additions and multiplications. Let us summarize this discussion as a lemma.

[1] technically, it should be $O(m^{\omega+\varepsilon})$ for every $\varepsilon > 0$... but meh.

Lemma 12.10 (Alignment lemma). *Suppose $f \in \mathbb{F}\langle y_1, \ldots, y_m\rangle$ is a polynomial such that every monomial of f has degree divisible by 3 and computable by a non-commutative circuit C of size s. Then, there is a* (mod3)*-homogeneous circuit C', each of whose gates/leaves are labelled by $(1,1)$, of size at most $O(s \cdot m^{\omega})$* □

Lemma 12.3 is a direct corollary of the above lemma as $m = n^{1/3}$ and this completes the proof of the hardness amplification for non-commtuative circuits.

Exercise 12.1 *The main monomial transformation was replacing the n variables by degree 3 monomials in $n^{1/3}$ variables. Naturally, one can choose any r and replace each of the n variables by degree r monomials in $n^{1/r}$ variables. How would these bounds change for an arbitrary r?*

Chapter 13

Lower bounds for multilinear models

Most of the polynomials that are studied usually, like those described in Chapter 2, are multilinear. A natural question is whether or not multilinear polynomials can be computed in a "multilinear fashion". This is formalized by what the model of multilinear circuits, in a way analogous to homogeneous circuits.

Definition 13.1 (Multilinear circuits)**.** *A circuit C is said to be* multilinear *if every gate of the circuit computes a multilinear polynomial. A circuit is said to be* syntactically multilinear *if for any $g = g_1 \times g_2$, there is no variable that has a path to both g_1 and g_2.* $\diamondsuit$

Note that syntactic multilinearity of course implies multilinearity as the definition forces all gates to compute multilinear polynomials. However, we could have a setting where there is a gate $g = g_1 \times g_2$ where some variable x has a path to both g_1 and g_2 but it so turns out that g_1 is independent of x due to other cancellations. However, for arithmetic formulas, the two notions are equivalent.

Exercise 13.1 *Given any arithmetic formula Φ that is multilinear, show that it can be converted to a formula Φ' of size* $\mathrm{poly}(\Phi)$ *that is syntactically multilinear.*
(Imp.) *Why does the same not work for multilinear circuits?*

This section shall deal mainly with multilinear formulas so we may assume without loss of generality that they are syntactically multilinear. Raz [Raz09] showed that multilinear formulas computing the Det_n or Perm_n must be of size $n^{\Omega(\log n)}$. The complexity measure used by Raz also led to exponential lower bounds for constant depth multilinear circuits [RY09] and super-linear lower bounds for syntactic multilinear circuits [RSY08]. Hrubeš and Yehudayoff [HY11] then showed a super-polynomial lower bound *homogeneous* multilinear formulas computing the elementary symmetric polynomial.

Although the lower bound of Hrubeš and Yehudayoff was subsequent to the results of Raz and Yehudayoff [Raz09, RY09], we shall see their lower bound first which uses a suprisingly simple complexity measure. For the non-homogeneous setting, the complexity measure used here would also be the *rank of the partial derivative matrix* from Chapter 11. We shall start off with a structural result whenever we are dealing with formulas, which builds on the depth reduction for formulas Lemma 5.5.

13.1 Log-product representations for formulas

The following structural lemma shows that any multilinear formula can be converted in to a small sum of *log-product* polynomials. The techniques of the following lemma can also be used in other settings with minor modifications, and we shall encounter a different version of this lemma later as well. These normal forms is from the survey of Shpilka and Yehudayoff [SY10], and also from the result of Hrubeš and Yehudayoff [HY11].

Definition 13.2. *A multilinear polynomial $f \in \mathbb{F}[X]$ is called a* multilinear log-product *polynomial if $f = g_1 \dots g_t$ and there exists a partition of variables $X = X_1 \sqcup \cdots \sqcup X_t$ such that*

- $g_i \in \mathbb{F}[X_i]$ *for all* $i \in [t]$.
- $\left(\frac{1}{3}\right)^i |X| \leq |X_i| \leq \left(\frac{2}{3}\right)^i |X|$ *for all* i, *and* $|X_t| = 1$.

◊

Lemma 13.3. *Let Φ be a multilinear formula of size s computing a polynomial p. Then p can be written as a sum of $(s+1)$ log-product multivariate polynomials.*

Proof. Similar to Lemma 5.5, let v be a node in Φ such that set of variables X_v that it depends on satisfies $\frac{|X|}{3} \leq |X_v| \leq \frac{2|X|}{3}$. If Φ_v is the polynomial computed at this node, then f can be written as

$$f \quad = \quad \Phi_v \cdot g_1 + \Phi_{v=0} \quad \text{for some } g_1 \in \mathbb{F}[X \setminus X_v].$$

where $\Phi_{v=0}$ is the formula obtained by replacing the node v by zero. Note that the subtree at the node v is completely disjoint from $\Phi_{v=0}$. Hence the sum of the sizes of Φ_v and $\Phi_{v=0}$ is at most s. Hence, $g_1 \in \mathbb{F}[X \setminus X_v]$ and $\frac{|X|}{3} \leq |X \setminus X_v| \leq \frac{2|X|}{3}$. Inducting on the formulas Φ_v and $\Phi_{v=0}$ gives the lemma. □

There are many variants of this formula depending on how you pick the node v in the proof above. Here is another variant that find a node v based on *degree* rather than the number of variables that it depends on. We state it here without proof but it follows exactly as in the lines of Lemma 13.3.

Lemma 13.4. *Let Φ be a homogeneous formula of size s computing a polynomial p of degree d. Then p can be written as a sum of $(s+1)$* log-product *polynomials, that is,*

$$p \quad = \quad f_1 + \cdots + f_r \qquad \text{with } r \leq s+1$$

where for each $i \in [r]$, we have $f_i = f_{i1} \cdots f_{i\ell}$ satisfying

- *each f_{ij} is homogeneous,*
- $(1/3)^j \cdot d \leq \deg(f_{ij}) \leq (2/3)^j \cdot d$,
- $f_{i\ell} = 1$.

In particular, each f_i factors into $\Omega(\log d)$ non-trivial factors of geometrically decreasing degrees.

Furthermore, if Φ was a multilinear formula to begin with, then so is the expression on the RHS.

We shall see yet another variant of this trick later in this chapter but we have enough for now to prove the lower bound of Hrubeš and Yehudayoff [HY11].

13.2 Lower bounds for homogeneous multilinear formulas

The main theorem of this section would be the following.

Theorem 13.5 ([HY11]). *Any homogeneous multilinear formula that computes the polynomial Sym_d, for $d \leq n/2$, must have size $d^{\Omega(\log d)}$.*

The complexity measure used here would be ridiculously simple — just the number of monomials! Surprisingly, this is enough to prove the lower bound for homogeneous multilinear formulas. Before we get into the proof, we would need the following approximation for the binomial coefficient. This follows from Stirling's approximation of $n!$.

Lemma 13.6 (Stirling's approximation of $\binom{n}{k}$)**.** *For $n \geq 3k/2$, the following*

$$\left(\frac{1}{3\sqrt{k}}\right) \cdot \left(\frac{n^n}{k^k \cdot (n-k)^{n-k}}\right) \quad \leq \quad \binom{n}{k} \quad \leq \quad \left(\frac{1}{\sqrt{k}}\right) \cdot \left(\frac{n^n}{k^k \cdot (n-k)^{n-k}}\right)$$

Proof of Theorem 13.5. If there is a size s homogeneous multilinear formula computing Sym_d, then by Lemma 13.4 we have an expression of the form

$$\mathsf{Sym}_d \quad = \quad \sum_{i=1}^{s+1} f_{i1} \cdots f_{i\ell}$$

with $\ell = \Omega(\log d)$. We shall show that each multilinear term of the form $f_{i1} \cdots f_{i\ell}$, with geometrically decreasing degrees, can contribute at most $d^{-\Omega(\log d)} \cdot \binom{n}{d}$ monomials. This would immediately imply that $s = d^{\Omega(\log d)}$.

Consider a term of the form $f_1 \cdots f_\ell$. Since the degrees drop geometrically, we may assume without loss of generality that $\ell = \Omega(\log d)$ and $\deg(f_i) \geq \sqrt{d}$ (by just multiplying all polynomials of smaller degree together). This is a homogeneous multilinear expression so let $\deg(f_i) = d_i$ and let f_i depend on the variables X_i. If $n_i = |X_i|$, then each f_i has at most $\binom{n_i}{d_i}$ monomials. Hence, the total number of monomials from this term is at most

$$\binom{n_1}{d_1} \cdots \binom{n_\ell}{d_\ell}.$$

All that's left to do is show that this is significantly smaller than $\binom{n}{d}$.

Lemma 13.7 ([HY11])**.** *Let $n \geq 2d$. For any set of non-negative integers satisfying $n_1 + \cdots + n_\ell = n$ and $d_1 + \cdots + d_\ell = d$, we have*

$$\binom{n_1}{d_1} \cdots \binom{n_\ell}{d_\ell} \quad \leq \quad 3\sqrt{\frac{d}{d_1 \cdots d_\ell}} \cdot \binom{n}{d}$$

The theorem immediately follows from this lemma as each $d_i \geq \sqrt{d}$ and there are $\Omega(\log d)$ of them.

Before we prove this theorem, let us quickly look at the simpler case when $d = n/2$. In that case, $\binom{n}{d} \approx 2^n/\sqrt{n}$ and the LHS is clearly upper bounded as

$$\binom{n_1}{d_1} \cdots \binom{n_\ell}{d_\ell} \quad \leq \quad \frac{2^{n_1+\cdots+n_\ell}}{\sqrt{n_1 \cdots n_\ell}}$$

giving us a similar bound. The case of general d requires a bit more work but is quite natural (once you know it is true).

Proof. [of Lemma 13.7] Without loss of generality, we may assume that $n_i \geq 2d_i$ for all i. We are trying to understand the following distribution:

> If we pick d elements out of n elements, what's the probability that we pick d_1 elements from the first n_1, and d_2 from the next n_2 ... etc.

The first step is to ask, for fixed values of $n_1, \ldots, n_\ell$, what values of $d_1, \ldots, d_\ell$ maximize this probability. Intuitively, each d_i ought to be proportional to n_i as that is what we would get in expectation. In the perfect regime where $n = \alpha \cdot d$ and $n_i = \alpha \cdot d_i$ for each $i \in [\ell]$, we can now use the approximation from Lemma 13.6 to get

$$\begin{aligned}
\binom{n}{d} &\geq \left(\frac{1}{3\sqrt{d}}\right) \cdot \left(\frac{n^n}{d^d \cdot (n-d)^{n-d}}\right) \\
&= \left(\frac{1}{3\sqrt{d}}\right) \cdot \left(\frac{d^n \cdot \alpha^n}{d^d \cdot d^{n-d} \cdot (\alpha-1)^{n-d}}\right) \\
&= \left(\frac{1}{3\sqrt{d}}\right) \cdot \left(\frac{\alpha^n}{(\alpha-1)^{n-d}}\right) \\
\text{and} \quad \binom{n_i}{d_i} &\leq \left(\frac{1}{\sqrt{d_i}}\right) \left(\frac{\alpha^{n_i}}{(\alpha-1)^{n_i-d_i}}\right) \\
\implies \binom{n_1}{d_1} \cdots \binom{n_\ell}{d_\ell} &\leq 3\sqrt{\frac{d}{d_1 \cdots d_\ell}} \cdot \binom{n}{d}
\end{aligned}$$

To complete the proof, we just have to show that $\binom{n_1}{d_1} \cdots \binom{n_\ell}{d_\ell}$ is maximized when $n_i/d_i \approx n/d$. To do this, let us consider a term $\binom{n_1}{d_1} \cdot \binom{n_2}{d_2}$ with $d_1/n_1 \gg d_2/n_2$ and show that $\binom{n_1}{d_1-1} \cdot \binom{n_2}{d_2+1}$ is larger. This is easy to see because

$$\begin{aligned}
\frac{\binom{n_1}{d_1-1} \cdot \binom{n_2}{d_2+1}}{\binom{n_1}{d_1} \cdot \binom{n_2}{d_2}} &= \frac{d_1 \cdot (n_2 - d_2)}{(n_1 - d_1 + 1) \cdot (d_2 + 1)} \\
&= \frac{\left(\frac{n_2+1}{d_2+1}\right) - 1}{\left(\frac{n_1+1}{d_1}\right) - 1}
\end{aligned}$$

$$> \ 1.$$

This is still a little incomplete as this ensures that the product is maximized when d_i is as close as possible to $d \cdot (n_i/n)$ but perhaps not quite exactly equal. This can be handled with some minor[1] changes in the calculation above. □ (Lemma 13.7)

The theorem follows immediately from this lemma. □

13.3 Lower bounds for (non-homogeneous) multilinear formulas

Homogeneity is crucially used in the proof of Theorem 13.5. A simple example is just the term $(x_1 + 1) \cdots (x_n + 1)$ which generates every possible multilinear monomial rendering a sparsity based complexity measure completely useless.

However, once again, we can use the *partial derivative matrix* that we studied in Chapter 11 here. In this section, we shall see the lower bounds of Raz [Raz09], and the lower bound of Raz and Yehudayoff [RY09].

Intuition

A natural first step is to try the simpler task of proving lower bounds for depth-3 multilinear circuits.

$$f \quad = \ell_{11} \dots \ell_{1d} + \cdots + \ell_{s1} \dots \ell_{sd}$$

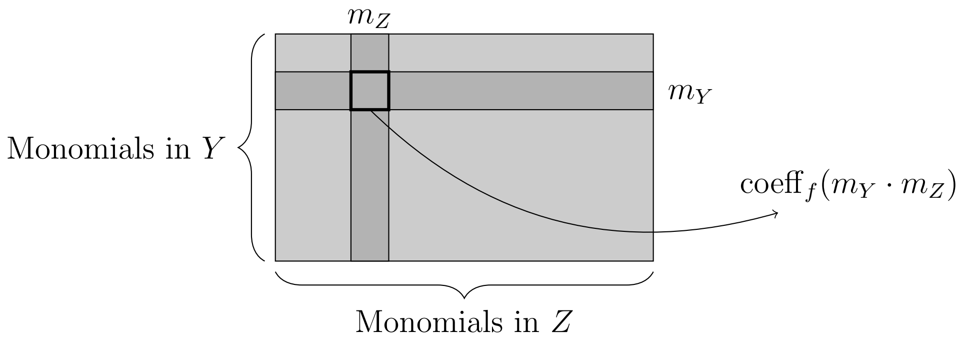
The task is now to construct a measure Γ such that $\Gamma(\ell_1 \dots \ell_d)$ is small whenever each ℓ_i is a linear polynomial and different ℓ_i's are over disjoint sets of variables. Consider the simplest case of $f = (a_1 + b_1 x)(a_2 + b_2 y)$. An observation is that the coefficients of f are given by the 2×2 matrix obtained as $[a_1 \ b_1]^T [a_2 \ b_2] = \begin{bmatrix} a_1 a_2 & a_1 b_2 \\ a_2 b_1 & b_1 b_2 \end{bmatrix}$. In other words, a polynomial $f = a_0 + a_1 x + a_2 y + a_3 xy$ factorizes into two variable disjoint factors if and only if the matrix $\begin{bmatrix} a_0 & a_1 \\ a_2 & a_3 \end{bmatrix}$ has rank 1. This is another way of reading f as a *non-commutative polynomial*, and this is precisely the partial derivative matrix of f with respect to $\{x, y\} = \{x\} \sqcup \{y\}$ as

[1] 'minor' if one is willing to afford some loss in parameters. But it is also possible to prove the statement claimed with some more work.

seen in Chapter 11.

We recall the definition of the partial derivative matrix with respect to $X = Y \sqcup Z$ from the picture:

$M_{Y,Z}(f) \quad =$



We shall use $\Gamma^{[\text{Raz}]}_{Y,Z}(f)$ to denote the rank of $M_{Y,Z}(f)$.

Here are some basic properties of the partial derivative matrix which would be extremely useful in later calculations.

Observation 13.8 (Sub-additivity)**.** *For any partition $X = Y \sqcup Z$ and any pair of multilinear polynomials f and g in $\mathbb{F}[X]$ we have $\Gamma^{[\text{Raz}]}_{Y,Z}(f+g) \quad \leq \quad \Gamma^{[\text{Raz}]}_{Y,Z}(f) \quad + \quad \Gamma^{[\text{Raz}]}_{Y,Z}(g)$.*

Proof. Follows from the linearity of the matrix. □

Observation 13.9 (Multiplicativity)**.** *If $f_1 \in \mathbb{F}[Y_1, Z_1]$ and $f_2 \in \mathbb{F}[Y_2, Z_2]$ with $Y = Y_1 \sqcup Y_2$ and $Z = Z_1 \sqcup Z_2$, then*

$$\Gamma^{[\text{Raz}]}_{Y,Z}(f_1 \cdot f_2) \quad = \quad \Gamma^{[\text{Raz}]}_{Y_1,Z_1}(f_1) \quad \cdot \quad \Gamma^{[\text{Raz}]}_{Y_2,Z_2}(f_2).$$

Proof. It is not hard to see that $M_{Y,Z}(f_1 \cdot f_2)$ is the tensor product $M_{Y_1,Z_1}(f_1) \otimes M_{Y_2,Z_2}(f_2)$, and the rank of a tensor product of two matrices is the product of the ranks. □

Observation 13.10 (Multiplication by univariates)**.** *For any $f \in \mathbb{F}[Y, Z]$ and any $g \in \mathbb{F}[Y]$, if $\mathbb{F}$ is large enough then we have that $\Gamma^{[\text{Raz}]}_{Y,Z}(g \cdot f) \leq \Gamma^{[\text{Raz}]}_{Y,Z}(f)$.*

By symmetry, the same is true when $g \in \mathbb{F}[Z]$.

Proof. The evaluation dimension of $g \cdot f$ is at most the evaluation dimension of g as any partial evaluation (by evaluating Y) of $g \cdot f$ is a scalar times a partial evaluation of g. The observation follows from Lemma 11.9. □

Observation 13.11 (A trivial bound)**.** *For any multilinear polynomial f, we have $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad \leq 2^{\min(|Y|,|Z|)}$.*

Proof. The number of rows is $2^{|Y|}$ and number of columns is $2^{|Z|}$, and hence the rank is upper-bounded by the minimum. □

Let us get back to lower bounds for multilinear models, and attempt to use $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$ defined above. Unfortunately, there are examples of simple polynomials like $f = (y_1 + z_1)\ldots(y_n + z_n)$ with $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) = 2^n$. Raz's idea here was to look at $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$ for a *random partition*, and show that with high probability the rank of the partial derivative matrix is far from full. As a toy example, we shall see why this has the potential to give lower bounds for depth-3 multilinear circuits.

Lemma 13.12. *Let $f(X) = \ell_1 \ldots \ell_d$ be an n-variate multilinear polynomial. If $X = Y \sqcup Z$ is a random partition with $|Y| = |Z| = |X|/2$, then with high probability we have*

$$\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad \leq \quad 2^{|X|/2} \cdot 2^{-|X|/16}.$$

It is to be noted that we should expect a random polynomial to be full-rank with respect to any partition, so the measure $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$ is expected to be $2^{|X|/2}$ which should yield a lower bound of $2^{\Omega(|X|)}$.

Sketch of Proof. Without loss of generality we can assume that each ℓ_i depends on at least two variables as removing the ℓ_i's that depend on just one variable does not alter $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$ with respect to any partition. Let $|X| = n$.

Using Observation 13.9, $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \leq 2^d$ and hence if $d < n/3$ then we are done. Hence assume that $d \geq n/3$. By a simple averaging argument, there must hence be at least $d/4$ of the ℓ_i's that depend on at most 3 variables; we shall refer to these as the *small* ℓ_i's.

Since the partition is chosen at random, on expectation a quarter of the small ℓ_i's would have all its variables mapped to either Y or Z, hence not contributing to $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$. Therefore, with high probability,

$$\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad \leq \quad 2^d \cdot 2^{-d/16} \quad \leq \quad 2^{n/2} \cdot 2^{-n/16}.$$

□

More generally, if $f = g_1(X_1)\ldots g_t(X_t)$ where the X_i's are mutually disjoint, then a random partition is very unlikely to partition all the X_i's into almost equal parts. This naturally calls for using the multilinear log-product representation from Lemma 13.3 for the case of multilinear formulas. The rest of this section would be proof of Raz's wonderful result [Raz09].

Theorem 13.13 ([Raz09]). *Any multilinear formula computing* Det_n *or* Perm_n *must be of size* $n^{\Omega(\log n)}$.

The proof would be as outlined. If f is computable by a size s multilinear formula, then by Lemma 13.3 we know that f admits a log-product representation:

$$f \quad = \quad \sum_{i=1}^{s+1} f_{i1} \cdots f_{i\ell}$$

Using this, we shall first obtain an upper-bound on $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f)$ for a random partition $X = Y \sqcup Z$ by showing that any single log-product is far from full-rank. Finally, for Det_n or Perm_n, we shall prove a lower bound on $\Gamma_{Y,Z}^{[\mathrm{Raz}]}$ for most partitions and that would complete the proof.

In the original paper of [Raz09], Raz considered *balanced* partitions $X = Y \sqcup Z$ with $|Y| = |Z| = |X|/2$ but this complicates issues because each variables are not independently assigned to Y or Z. However, there is a slightly simpler analysis when we just independently assign each variable to Y or Z and deal with the possible imbalance later on. This trick was communicated to us by Srikanth Srinivasan.

13.3.1 Log-products are far from full-rank on a random partition

The main technical part of the proof is to show that log-product multivariate polynomials are far from full-rank under a random partition of variables. This would let us show that a sum of log-product multivariate polynomials cannot be full rank unless it is a very large sum.

Main idea: Suppose $f = g_1 \ldots g_t$ where each $g_i \in \mathbb{F}[X_i]$. Let $X = Y \sqcup Z$ be a random partition, obtained by assigning each variable $x \in X$ independently to Y or Z with probability 1/2 each. Let $Y_i = Y \cap X_i$ and $Z_i = Z \cap X_i$ and say $d_i = \left|\frac{|Y_i| - |Z_i|}{2}\right|$ measure the imbalance between the sizes of Y_i and Z_i. We shall say X_i is k-imbalanced if $d_i \geq k$. Let $b_i = \frac{|Y_i| + |Z_i|}{2} = \frac{|X_i|}{2}$.

Suppose the partition was such that $|Y| = |Z|$ (as considered originally by Raz), then by Observation 13.9 we know that

$$\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad = \quad \Gamma_{Y_i,Z_i}^{[\mathrm{Raz}]}(g_1) \ldots \Gamma_{Y_i,Z_i}^{[\mathrm{Raz}]}(g_t)$$

$$\begin{aligned} &\leq \quad 2^{\min(|Y_1|,|Z_1|)} \cdot \dots \cdot 2^{\min(|Y_t|,|Z_t|)} \\ &= \quad 2^{b_1-d_1} \cdots 2^{b_t-d_t} = \frac{2^{|X|/2}}{2^{d_1+\cdots+d_t}}. \end{aligned}$$

Hence, even if one of the X_i's is a little imbalanced, then the product is far from full-rank.

Lemma 13.3 shows that the size of X_i decreases slowly with i, and it is not hard to show that $|X_i| \geq \sqrt{|X|}$ for $i \leq t' \stackrel{\text{def}}{=} \frac{\log|X|}{100}$. We wish to show that the probability that none of g_i (for $i \leq t'$) is k-unbalanced for $k = |X|^{1/20}$ is very small (inverse polynomially small). Since we have $t = \Theta(\log n)$ such independent events, the probabilty that *none* of the g_i's are k-unbalanced will be much smaller.

In the original setting of balanced partitions, these events are not exactly independent but one can make the calculations work out (using basic properties of hypergeometric distributions). But we'll work with partitions obtained by assigning each variable independently to Y or Z.

Let $\mathcal{E}_i$ be the event that g_i is *not* k-unbalanced. Since the variables of the g_i's are variable disjoint, and each variable is independently assigned to Y or Z, these events $\mathcal{E}_i$'s are independent. What is the probability that $\mathcal{E}_i$ holds? The event $\mathcal{E}_i$ is just the probability that $|Y_i| \in \left[\frac{|X_i|}{2} - k, \frac{|X_i|}{2} + k\right]$. Even the probability of $|Y_i| = |X_i|/2$ is at most $O(1/\sqrt{X_i})$. Hence,

$$\Pr[\mathcal{E}_i] \quad \leq \quad \frac{2k}{\sqrt{X_i}} \quad \leq \quad O\left(\frac{2k}{\sqrt[4]{|X|}}\right).$$

Since $t = \Theta(\log n)$, and these are independent events,

$$\Pr[\mathcal{E}_1 \wedge \cdots \wedge \mathcal{E}_{t'}] \quad \leq \quad |X|^{-\varepsilon \log|X|} \quad \text{for some } \varepsilon > 0.$$

Now suppose we further condition this on the event that $|Y| = |Z|$, how different can the above probability become?

$$\begin{aligned} &\Pr[\mathcal{E}_1 \wedge \cdots \wedge \mathcal{E}_{t'}] \geq \Pr[\mathcal{E}_1 \wedge \cdots \wedge \mathcal{E}_{t'} \mid |Y| = |Z|] \cdot \Pr[|Y| = |Z|] \\ \implies &\Pr[\mathcal{E}_1 \wedge \cdots \wedge \mathcal{E}_{t'} \mid |Y| = |Z|] \leq |X|^{-\varepsilon \log|X|} \cdot \sqrt{|X|} \end{aligned}$$

and the $\sqrt{|X|}$ is inconsequential when we have an inverse-quasipolynomial error probability.

Therefore,

$$\Pr_{\substack{X=Y\sqcup Z\\|Y|=|Z|}}\left[\Gamma_{Y,Z}^{[\mathrm{Raz}]}(g_1\dots g_t) > 2^{(|X|/2)-|X|^{1/20}}\right] \leq |X|^{-\varepsilon\log|X|}.$$

Hence, if $g_1\dots g_t$ is a log-product multilinear polynomial, then with probability at least $\left(1-|X|^{-\varepsilon\log|X|}\right)$ we have that $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(g_1\dots g_t) \leq 2^{(|X|/2)-|X|^{1/20}}$. Further, if f is computable by a multilinear formula of size s then, by Lemma 13.3, f can be written as a sum of $(s+1)$ log-product multilinear polynomials. Hence, with probability at least $\left(1-(s+1)|X|^{-\varepsilon\log|X|}\right)$ we have that

$$\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad\leq\quad (s+1)\cdot 2^{(|X|/2)-|X|^{1/20}}.$$

Hence, if $(s+1) < |X|^{(\varepsilon/2)\log|X|}$, then with high probability a random partition would ensure $\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \ll 2^{|X|/2}$. Let us record this as a lemma.

Lemma 13.14. *Let $f\in\mathbb{F}[X]$ be computed by a multilinear formula of size $s < |X|^{(\varepsilon/2)\log|X|}$ for a small enough constant $\varepsilon > 0$. Then with probability at least $(1-|X|^{-(\varepsilon/2)\log|X|})$ we have*

$$\Gamma_{Y,Z}^{[\mathrm{Raz}]}(f) \quad\leq\quad (s+1)\cdot 2^{|X|/2}\cdot 2^{-|X|^{1/20}}$$

for a random partition $X = Y\sqcup Z$ with $|Y|=|Z|=|X|/2$.

13.3.2 Det_n and Perm_n have large rank

The last step of the proof would be to find an explicit polynomial whose partial derivative matrix under a random partition has large rank. As earlier, our candidate polynomial would be Det_n or Perm_n. Unfortunately, both these polynomials are over n^2 variables and degree n. It is not hard to verify that the rank of the partial derivative matrix of Det_n or Perm_n can never be greater than 2^{2n}. Hence directly using Lemma 13.14, we would have $2^{O(n)}$ competing with $2^{n^2/2-n^{O(1)}}$ which is simply futile. A simple fix is to first randomly restrict ourselves to fewer variables and then apply Lemma 13.14.

Let $\tilde{X} = \{x_{11},\dots,x_{nn}\}$, and let us assume $n = 2m$ is even. We shall interpret all other variables as "scalars" by making our field $\mathbb{K} = \mathbb{F}\left(\{x_{ij} \;:\; i\neq j\}\right)$ instead. If we had a multilinear formula of size s, over $\mathbb{F}$, computing Det_n then we also have a multilinear formula, over $\mathbb{K}$, of size at most s. We shall apply Lemma 13.15 for this circuit.

However, we show that for every balanced partition $\tilde{X} = Y\sqcup Z$ the partial derivative matrix (over $\mathbb{K}$) has full-rank. We will prove that the matrix has full-rank by showing that it has full-rank even after a substitution of the variables $\{x_{ij} \;:\; i \neq j\}$. Consider the following restriction:

$$\begin{aligned} f &= \mathsf{Det}\begin{bmatrix} y_1 & 1 & & & \\ 1 & z_1 & & & \\ & & \ddots & & \\ & & & y_m & 1 \\ & & & 1 & z_m \end{bmatrix} \\ &= (y_1 z_1 - 1)\ldots(y_m z_m - 1). \end{aligned}$$

It is easy to check that $\Gamma^{[\text{Raz}]}_{Y,Z}(f) = 2^m$.

Lemma 13.15. *For every balanced partition $\tilde{X} = Y \sqcup Z$ with $|Y| = |Z| = m = n/2$, we have $\Gamma^{[\text{Raz}]}_{Y,Z}(\sigma(\mathsf{Det}_n)) = 2^m$.*

Combining Lemma 13.15 with Lemma 13.14, we have the following theorem.

Theorem 13.16 ([Raz09])**.** *Any multilinear formula computing Det_n or Perm_n must be of size $n^{\Omega(\log n)}$.* □

13.3.3 Constructing a full-rank polynomial

As alluded to earlier, if f is a random polynomial, then for any partition $X = Y \sqcup Z$ with $|Y| = |Z| = |X|/2$ we expect $\Gamma^{[\text{Raz}]}_{Y,Z}(f) = 2^{|X|/2}$. Can we construct such a polynomial explicitly? This was first answered by Raz [Raz06], and the construction was later simplified by Raz and Yehudayoff [RY08] and we shall describe the latter here.

Main idea: Suppose you know the partition as $Y = \{y_1, \ldots, y_n\}$ and $Z = \{z_1, \ldots, z_n\}$, then building a polynomial such that $M_{Y,Z}(f)$ is full-rank is easy — just take $f = (y_1 z_1 + 1)\ldots(y_n z_n + 1)$ which makes $M_{Y,Z}(f)$ the identity matrix. What we would like is to have one such copy for every partition, and we would also like to build it in a way so that f can be computed in VP.

The first attempt is to build the polynomial inductively for every partition into two equal parts, but "remembering" partial partitions take too much memory (similar to the situation we had in Theorem 3.10). Raz and Yehudayoff instead use a different construct that, rather than being based on partitions, is instead based on Dyck-strings or well-matched parentheses.

The language Dyck(n) refers to strings of length $2n$ over symbols '(' and ')' that is well-matched in the natural way. That is "()()" and "(())" belong to Dyck(2) but not "())(". Raz and Yehudayoff come up with a natural map to convert every such string to a full-rank polynomial, and we shall just state this by example.

$$\begin{aligned}\Omega(\text{``(())''}) &= (x_1x_4+1)\cdot(x_2x_3+1)\\ \Omega(\text{``()()''}) &= (x_1x_2+1)\cdot(x_3x_4+1)\end{aligned}$$

That is, if the opening bracket at position i is closed at position j, then there is a factor (x_ix_j+1). Let us define the polynomial as follows:

$$f(x_1,\ldots,x_{2n}) \quad = \quad \sum_{s\in\text{Dyck}(n)} \Omega(s)$$

Let us attempt to compute the polynomial using a small circuit in the following natural inductive manner.

$$f_{i,j}(\mathbf{x}) \quad = \quad \begin{cases} (x_ix_j+1) & \text{if } j=i+1\\ 0 & \text{if } j-i \text{ is even}\\ (x_ix_j+1)\cdot f_{i+1,j-1} & \\ \quad + \sum_{r=i+1}^{j-1} f_{i,r}\cdot f_{r+1,j} & \text{otherwise}\end{cases}$$

However, $f_{1,2n}$ is *not equal* to the polynomial f as the term corresponding to "()()()...()" has coefficient 1 in f but is C_n, the $(n-1)$-th Catalan number. Nevertheless, this inductively defined polynomial $f_{1,2n}$ has most of the properties we need. Raz and Yehudayoff take a slight variant of it by adding a few auxiliary variables $\{\omega_{i,j,k},\omega_{i,j}\}$ to build the following polynomial over the field $\mathbb{F}(\{\omega_{i,j,k},\omega_{i,j} \ : \ i,j,k\in[2n]\})$.

$$\tilde{f}_{i,j}(\mathbf{x}) \quad = \quad \begin{cases} (x_ix_j+1) & \text{if } j=i+1\\ 0 & \text{if } j-i \text{ is even}\\ (x_ix_j+1)\cdot \tilde{f}_{i+1,j-1}\omega_{i,j} & \\ \quad + \sum_{r=i+1}^{j-1} \tilde{f}_{i,r}\cdot \tilde{f}_{r+1,j}\cdot\omega_{i,r,j} & \text{otherwise}\end{cases}$$

Lemma 13.17 ([RY08]). *The polynomial $\tilde{f} = \tilde{f}_{1,2n}$ has the property that for every $X = Y \sqcup Z$*

with $|Y| = |Z| = |X|/2$ *we have* $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(\tilde{f}) = 2^{|X|/2}$.

Proof. The proof shall be over induction on n. For the base case, statement is obviously true for $n = 1$ as $\tilde{f} = (x_1x_2 + 1)$ in this case.

Suppose $n > 1$, and let $X = Y \sqcup Z$ be any partition with $|Y| = |Z| = |X|/2$. Then, either both x_1 and x_{2n} belong to the same side of the partition, or they belong to different sides. Let us handle each case separately.

Case 1: x_1 and x_{2n} are in different parts.

Then, polynomial $\tilde{f}$ where we set $\omega_{1,i,2n} = 0$ for all i. Under this substitution, $\tilde{f}$ is equal to $(x_1x_{2n} + 1) \cdot \tilde{f}_{2,2n-1}$. By induction, we know that $\tilde{f}_{2,2n-1}$ is full-rank under equal sized partition and hence so is $(x_1x_{2n} + 1) \cdot \tilde{f}_{2,2n-1}$. As setting variables to zero can only reduce the rank, it follows that $M_{Y,Z}(\tilde{f})$ must be full-rank as well.

Case 2: x_1 and x_{2n} are in the same part.

Since $|Y| = |Z| = |X|/2$, there must be some intermediate point r such that the two intervals $[1, r]$ and $[r + 1, 2n]$ are both split evenly by Y and Z. Using induction on each of these smaller intervals again, we get that $M_{Y,Z}(\tilde{f})$ is full-rank again.

□

This is one of the very few polynomials that we are aware of that can be computed by polynomial sized arithmetic circuits, but is not known to be computable by polynomial size ABPs.

13.4 Stronger lower bounds for constant depth multilinear formulas

Looking back at Lemma 13.14, we see that whenever $f(X)$ is computable by a size s multilinear formula $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f)$ is exponentially smaller than $2^{|X|/2}$ with probability $\left(1 - s \cdot |X|^{-\varepsilon \log |X|}\right)$. Hence we had to settle for a $n^{\Omega(\log n)}$ lower bound not because of the rank deficit but rather because of the bounds in the probability estimate. Unfortunately, this lower bound technique cannot yield a better lower bound for multilinear formulas as we have already seen (in Subsection 13.3.3) that there are explicit examples of polynomials computable by poly-sized multilinear circuits with $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f) = 2^{|X|/2}$ under *every* partition. However, the probability bound can be improved in the case of constant depth multilinear circuits to give stronger lower bounds.

Note that Lemma 13.14 was proved by considering *multilinear log-products* (Definition 13.2) as the building blocks. To show that a multilinear log product $g_1(X_1)\dots g_\ell(X_\ell)$ has small rank under a random partition, we argued that the probability that all the X_i's are partitioned in a roughly balanced fashion is quite small. This was essentially done by thinking of this as $\ell = O(\log n)$ close-to-independent events, each with probability $1/\text{poly}(n)$.

If ℓ was much larger than $\log n$ (with other parameters being roughly the same), it should be intuitively natural to expect a much lower probability of all the X_i's being partitioned in a roughly balanced manner. This indeed is the case for constant depth multilinear circuits, and we briefly sketch the key points where they differ from the earlier proof. The first is an analogue of Definition 13.2 in this setting.

Definition 13.18. *A multilinear polynomial f is said to be a* multilinear t-product *if f can be written as $f = g_1 \dots g_t$ with the following properties:*

- *The variable sets of the g_i are mutually disjoint*
- *Each g_i non-trivially depends on at least t variables*

◊

Lemma 13.19. *Let f be a multilinear polynomial of degree d over n variables that is computed by a depth-Δ multilinear formula Φ of size s. Then, f can be written as a sum of at most s multilinear t-products for $t = (n/100)^{1/2\Delta}$, and a multilinear polynomial of degree at most $n/100$.*

Proof. If $d < n/100$, then the lemma is vacuously true. Since Φ is a formula of depth Δ and computes a polynomial of degree $d > n/100$, there must be at least one product gate v of fan-in at least $\left(\frac{n}{100}\right)^{1/\Delta} = t^2$. Then similar to Lemma 13.3,

$$f \quad = \quad \Phi_v \cdot f' + \Phi_{v=0}$$

As Φ_v is a product of t^2 polynomials, by grouping the factors together we have that $\Phi_v \cdot f'$ is a multilinear t-product. Further, $\Phi_{v=0}$ is a multilinear polynomial that is computable by a depth-Δ formula of smaller size and we can induct on $\Phi_{v=0}$. □

Lemma 13.20. *Let $f(X)$ be an n-variate polynomial computed by a depth-Δ multilinear formula of size s. If $X = Y \sqcup Z$ is a randomly chosen partition with $|Y| = |Z| = n/2$, then* *with probability at least* $(1 - s \cdot \exp(-n^{\Omega(1/\Delta)}))$ *we have*

$$\Gamma_{Y,Z}^{[\text{Raz}]}(f) \quad \leq \quad (s+1) \cdot 2^{n/2} \cdot \exp(-n^{\Omega(1/\Delta)}).$$

Sketch of Proof. By Lemma 13.19, we have that f can be written as $g_0 + g_1 + \cdots + g_s$ where $\deg(g_0) \leq n/100$ and $g_1, \ldots, g_s$ are multilinear t-products. Note that since g_0 is a multilinear polynomial of degree at most $(n/100)$, the number of monomials in g_0 is at most $\binom{n}{n/100} \leq 2^{n/10}$. Hence, $\Gamma_{Y,Z}^{[\text{Raz}]}(g_0) \leq 2^{n/10}$.

For the other g_i's, we can bound the probability that $\Gamma_{Y,Z}^{[\text{Raz}]}(g_i)$ is large in a very similar fashion as in Lemma 13.14, as the probability that all the factors of g_i are partitioned in a balanced manner is the intersection of t independent events. By very similar estimates, this probability can be bounded by $(1/\text{poly}(n))^t$. Hence, with high probability

$$\Gamma_{Y,Z}^{[\text{Raz}]}(f) \quad \leq \quad \Gamma_{Y,Z}^{[\text{Raz}]}(g_0) + \cdots + \Gamma_{Y,Z}^{[\text{Raz}]}(g_s) \quad \leq \quad (s+1) \cdot 2^{n/2} \cdot \exp(-n^{\Omega(1/\Delta)}).$$

□

Combining Lemma 13.20 with Lemma 13.15, we have the following theorem of Raz and Yehudayoff.

Theorem 13.21 ([RY09]). *Any multilinear formula of depth* Δ *computing* Det_n *or* Perm_n *must be of size* $\exp(n^{\Omega(1/\Delta)})$. □

Chapter 14

Lower bounds for multi-k-ic models

Recall that a multilinear depth three circuit is a sum of *multilinear terms* that are polynomials of the form $T = \ell_1 \cdots \ell_d$ where every variable $x \in X$ is present in at most one ℓ_i. Kayal and Saha [KS15c] study more general depth three circuits that are sums of terms where very variable occurs in "few" linear factors.

Definition 14.1 (Multi-k-ic depth three circuits). *A product of linear polynomials $T = \ell_1 \cdots \ell_d$ is said to be a* multi-k-ic term *if every variable $x \in X$ occurs in at most k of the linear polynomials. A depth three circuit is said to be a* multi-k-ic depth three circuit *if it is a sum of multi-k-ic terms.* ♢

For example, the circuit computing $(x_1 + x_2)(x_2 + x_3)(x_1 + x_3) + (x_1 + x_3)(x_2 + 3x_3)$ multi-2-ic circuit.

Kayal and Saha [KS15c] studied the question of proving lower bound for such multi-k-ic depth-three circuits for small k and they showed that the techniques of [Raz09] can be generalized to give exponential lower bounds for multi-k-ic depth-three circuits for small k.

Theorem 14.2 ([KS15c]). *There is an explicit n-variate polynomial $f \in \mathsf{VNP}$ such that any multi-k-ic depth-three circuit computing it must have size $2^{\Omega(n/2^{100k})}$.*

14.1 Revisiting the measure

In all the multilinear lower bounds we saw, the measure used was the rank of the partial derivative matrix under a random partition. We shall use the same measure, but keeping in mind that monomials could be non-multilinear. Once again, for a partition $X = Y \sqcup Z$, we shall define the matrix $M_{Y,Z}(f)$ to be the matrix whose rows are indexed by all (possibly non-multilinear) monomials m_i in the Y variables, and columns are indexed by all (possibly non-multilinear) monomials m_j in the Z variables with the entry being the coefficient of $m_i m_j$ in f. We shall abuse notation and use $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f)$ to refer to the rank of $M_{Y,Z}(f)$.

As in the earlier lower bounds, we shall take a random partition $X = Y \sqcup Z$ and study $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f)$ for a small multi-k-ic circuit.

Remark. We shall take a slight deviation from the way we chose partitions earlier. Here, we shall take every variable $x \in X$ and map it to Y or Z with equal probability. Thereby, it is possible that $|Y|$ is not exactly $|X|/2$ but it would nevertheless be close to it with high probability. This modification gives us the useful feature that the random partition is made of independent events. This would turn out to be useful in the following calculations. $\Diamond$

14.2 Proof of Theorem 14.2

The proof strategy will be the same as earlier. For a random polynomial with degree in every variable bounded by k, we expect $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f)$ to be $\Omega(k^{\min(|Y|,|Z|)})$. We shall show that for a multi-k-ic term $T = \ell_1 \cdots \ell_d$, under a random partition $X = Y \sqcup Z$, the measure $\Gamma^{[\mathrm{Raz}]}(T)$ will be far from $k^{n/2}$ with high probability, if $n = |X|$.

The proof that we shall describe here is a simplification of the original proof of [KS15c]. This proof would be very similar to the proof of Lemma 13.12.

Lemma 14.3. *Let $X = Y \sqcup Z$ be a random partition. If T is a multi-k-ic term, then with probability $\left(1 - \exp(-\frac{|X|}{2^{6k+1}})\right)$ that*

$$\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f) \quad \leq \quad (1+k)^{\min(|Y|,|Z|)} \cdot \exp\left(-\frac{|Y|}{2^{3k+1}}\right).$$

Let us quickly recall how the proof of Lemma 13.12 proceeded. We essentially showed that for any linear polynomial involving "few" variables, with some non-trivial probability the random partition would map all the variables to one side of the partition. Thus, if there are many such linear polynomials, the is a large fraction of the linear polynomials that do not contribute to $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T)$. If there are very few such "small" linear polynomials, then the degree would have cannot be too large and we can use a trivial bound. The proof here shall proceed in a very similar fashion.

Proof of Lemma 14.3. Let $T = \ell_1 \cdots \ell_d$. We shall call a factor ℓ to be "large" if the number of variables it depends on is at least $3k$, and let T_L be the product of all ℓ_i's that are "large", and let T_S be the product of the remaining ℓ_i's.

Observation 13.9, generalized to this setting, gives that $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T) \leq \Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_L) \cdot \Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_S)$. Thus, it suffices to bound each term separately. The easy case is, as one would expect, handling T_L. Suppose Y is the smaller of the sets Y and Z. Let us list down every variable in Y that occurs in T_L in order as $y_1, \ldots, y_{r_L}$ (with repetition). Let $r_L = (1-\delta)|Y|k$ for some $0 \leq \delta \leq 1$. A trivial bound for $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_L)$ is

$$\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_L) \quad \leq \quad 2^{\deg(T_L)} \quad \leq \quad 2^{r_L/3k} \quad \leq \quad 2^{(1-\delta)|Y|/3}. \tag{14.4}$$

The trickier case is with T_S but intuitively the setting is very similar to what we encountered in the proof of Lemma 13.12. Since any factor ℓ of T_S depends on at most $3k$ variables, with probability at least $\left(\frac{1}{2^{3k-1}}\right)$, all the variables of ℓ would be on the same side of the partition. Thus, on expectation, there would be at least $\left(\frac{\deg(T_S)}{2^{3k-1}}\right)$ factors that would not contribute to $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_S)$ at all. In Lemma 13.12, we could show that the number of such factors is concentrated around the expectation since the linear polynomials were disjoint and the events were independent. However, in this setting a variable can occur in multiple factors. Nevertheless, one can still use the fact that every variable occurs in at most k factors to establish a concentration very similar to Chernoff's Bounds. The following beautiful theorem of Gavinsky, Lovett, Saks and Srinivasan [GLSS12] is exactly what we need.

Theorem 14.5 ([GLSS12]). *Let $X_1, \ldots, X_n$ be independent random variables. Let $E_1, \ldots, E_r$ be boolean random variables that are functions of $\{X_i\}$ such that each X_i influences at most k of the E_j's. If $\Pr[E_i = 1] \geq p$, then for any $\varepsilon > 0$, we have*

$$\Pr\left[E_1 + \cdots + E_r \quad \leq \quad (p-\varepsilon)r\right] \quad \leq \quad e^{-2\varepsilon^2 r/k}.$$

Again, let us list down every variable in Y that occurs in T_S in order as $y_1, \ldots, y_{r_S}$ (listed with repetition). Note that $r_S + r_L \leq |Y|k$, and since $r_L = (1-\delta)k|Y|$, we have that $r_S \leq \delta|Y|$. Let E_i be the indicator random variable that is 1 if all the variables of the factor ℓ that contains y_i are mapped to the same side of the partition; we shall call such an instance variable y_i as *ineffective*. In other words, the set of ineffective instances y_i (with $E_i = 1$) are those that do not contribute to $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_S)$.

Since each ℓ depends on at most $3k$ variables, we have that $\Pr[Y_i = 1] \geq p = \frac{1}{2^{3k-1}}$. Let us set $\varepsilon = \frac{p}{2}$ and use the above theorem. Hence, the probability at least $\left(1 - \exp(-\frac{r_S}{k2^{6k}})\right)$, there are at least $\left(\frac{r_S}{2^{3k}}\right)$ ineffective instances.

For every variable $x \in Y$, let d_x be the number of occurrences of x that is not ineffective. We know that $\sum_{x \in Y} d_x \leq r_S \cdot \left(1 - \frac{1}{2^{3k}}\right)$. On the other hand, $\prod_{x \in Y}(1 + d_x)$ is an upper-bound on the number of non-zero rows of $M_{Y,Z}(T_S)$. Hence,

$$\begin{aligned}
\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T_S) \quad \leq \quad \prod_{x \in Y}(1 + d_i) \quad & \leq \quad \left(\frac{\sum_{x \in Y}(1 + d_x)}{|Y|}\right)^{|Y|} \\
& \leq \quad \left(\frac{|Y| + r_S\left(1 - \frac{1}{2^{3k}}\right)}{|Y|}\right)^{|Y|} \\
& \leq \quad \left(1 + \delta k\left(1 - \frac{1}{2^{3k}}\right)\right)^{|Y|} \\
& \leq \quad (1 + \delta k)^{|Y|} \cdot \left(1 - \frac{1}{2^{3k+1}}\right)^{|Y|} \\
& \leq \quad (1 + k)^{\delta |Y|} \exp\left(-\frac{|Y|}{2^{3k+1}}\right).
\end{aligned}$$

Combining this with (14.4), we get with probability at least $\left(1 - \exp(-\frac{|X|}{2^{6k+1}})\right)$ that

$$\Gamma^{[\mathrm{Raz}]}_{Y,Z}(T) \quad \leq \quad (1 + k)^{\min(|Y|,|Z|)} \cdot \exp\left(-\frac{|Y|}{2^{3k+1}}\right).$$

□

Using an union bound over all multi-k-ic terms, we obtain the simple corollary.

Corollary 14.6. *Let $C = T_1 + \cdots + T_s$ be a multi-k-ic circuit over variables X. Then for a random partition $X = Y \sqcup Z$, with probability at least $\left(1 - s \cdot \exp(-\frac{|X|}{2^{6k+1}})\right)$, we have*

$$\Gamma^{[\mathrm{Raz}]}_{Y,Z}(C) \quad \leq \quad s \cdot (1 + k)^{\min(|Y|,|Z|)} \cdot \exp\left(-\frac{|Y|}{2^{3k+1}}\right).$$

In particular, if $s < \exp(\frac{|X|}{2^{6k+2}})$, then this happens with probability at least $1/2$.

All that is left to do is find an explicit f such that $\Gamma^{[\mathrm{Raz}]}_{Y,Z}(f) = (k + 1)^{\min(|Y|,|Z|)}$ with high probability and we would have our lower bound. Here is one example that is a slight generalization of the construction in Subsection 13.3.3 of Raz and Yehudayoff [RY08] that is defined as follows. We shall again work over the field $\mathbb{F}(\{\omega_{a,b,c}, \omega_{a,b} \ : \ a, b, c \leq 2n\})$.

$$f_{i,j}^{(k)} = \begin{cases} \sum_{r=0}^{k} x_i^r x_{i+1}^r & \text{if } j = i = 1 \\ 0 & \text{if } j - i \text{ is even} \\ \left(\sum_{r=0}^{k} x_i^r x_j^r\right) \cdot f_{i+1,j-1}^{(k)} \cdot \omega i, j, & \\ \quad + \sum_{\ell=i+1}^{j-1} f_{i,\ell}^{(k)} \cdot f_{\ell+1,j}^{(k)} \cdot \omega_{i,\ell,j} & \text{otherwise.} \end{cases}$$

The following lemma generalizes follows almost directly from Lemma 13.17.

Lemma 14.7. *The polynomial $f = f_{1,2n}^{(k)}$ defined above has the property that for every partition $X = \{x_1, \ldots, x_{2n}\} = Y \sqcup Z$, we have*

$$\Gamma_{Y,Z}^{[\text{Raz}]}(f) \quad = \quad (k+1)^{\min(|Y|,|Z|)}.$$

Further, this polynomial can be computed by a linear sized arithmetic circuit and hence is in VP.

Combining this with Corollary 14.6 directly gives the lower bound of Theorem 14.2 for a polynomial in VP.

Remark 14.8. *There has been a subsequent improvement by Kayal, Saha and Tavenas which I hope to add to this chapter soon.* ◊

Chapter 15

Separating multilinear ABPs and formulas

So far we have seen that the partial derivative matrix can be used to exhibit a weakness for multilinear formulas. Furthermore, this also yielded a separation between circuits and ABPs as we also saw a construction of a *full-rank* polynomial by a linear sized multilinear circuit.

A natural question now is where do multilinear ABPs sit? We do know that multilinear ABPs are sandwiched between multilinear formulas and multilinear circuits and hence at least one of the two containments has to be strict.

Definition 15.1 (Multilinear ABPs)**.** *An algebraic branching program is said to be multilinear if every path from s to t has variable-disjoint edge weights.* $\Diamond$

Dvir, Malod, Perifel and Yehudayoff [DMPY12] proved that multilinear formulas are strictly contained in multilinear ABPs. Formally, they prove the following result.

Theorem 15.2 ([DMPY12])**.** *For every $m \in \mathbb{N}$, there is an explicit polynomial F_n on $n = \operatorname{poly}(m)$ variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ such that*

- *F is computable by a multilinear algebraic branching program of size $O(n^2)$,*
- *any multilinear formula computing F requires size $n^{\Omega(\log n)}$.*

It is worth noting that one can not get an asymptotically better separation than this, as any $\operatorname{poly}(n)$ sized multilinear circuit can be converted into a multilinear formula of size $n^{O(\log n)}$. Therefore, among other things this result tells us that a better reduction from circuits to formulas is not possible at least in the multilinear regime.

Proof idea

The outline of the proof follows that of Raz's proof [Raz09] of a quasipolynomial lower bound for the *full-rank-polynomial* against multilinear formulas that we saw in Chapter 13. That is it used the *log-product decomposition* for multilinear formulas, and showed that the decomposition is "rank deficient" under a uniformly random equipartition of the variable set, with high probability.

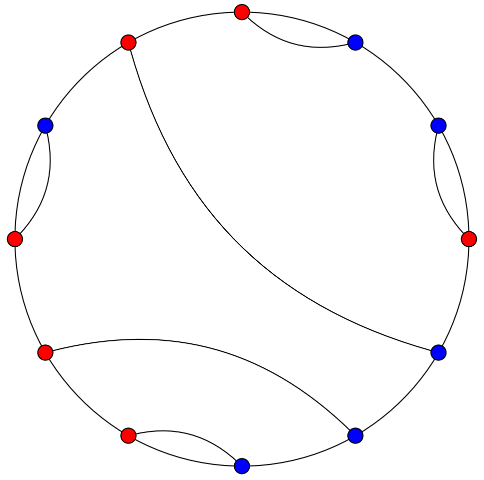
At this point, a possible line of argument is to use Raz's full-rank-polynomial and show that it is computable by multilinear ABPs. However, it is not known that this polynomial is computable by a small multilinear ABP. The idea of Dvir et al. is to construct a different polynomial that is full-rank for a certain *subset of restricted partitions*, and choose this subset of partitions carefully so that the polynomial is computable by ABPs. This carefully chosen set of partitions is referred to as *arc partitions* in the proof. The name becomes fairly clear from the process of "sampling" an arc partition that is used to describe the distribution.

The harder part is to show that Raz's lower bound proof continues to hold even for just these restricted partitions. That is, we now need to show that the log-product decomposition is rank deficient under a random *arc partition* with high probability. This proves to be quite tricky and a large chunk of the proof is just that, as we will soon sketch in the rest of this chapter.

15.1 Arc partitions

As mentioned above, the full-rank-polynomial as used in [Raz09] need not be computable by multilinear ABPs. We will therefore have to work with a polynomial that has full rank under a smaller set of partitions. We will call these partitions as *arc partitions*, which will be characterised by the support of the distribution $\mathcal{D}$ of the random process given below.

The random process will first pick $n/2$ pairs from $\mathbf{x}$ and then "bicolour" each pair with $\mathbf{y}$ and $\mathbf{z}$ uniformly at random, to obtain an equipartition. Imagine the indices of $\mathbf{x}$ from $\{0, \ldots, n-1\}$ arranged in order on an n-cycle, in a clockwise manner. The following example of such a partition from this distribution should help understand the formal sampling process.



The random process begins with the singleton set of pairs $P_1 = \{\{0, 1\}\}$. The process then picks one pair of unpicked elements in each iteration, till there are no more elements to pick. Throughout the process of picking these $n/2$ pairs, it maintains the invariant that all the pairs picked till some iteration t, together cover exactly a contiguous arc of length $2t$ on the cycle. Let the set of pairs picked ater iteration t be P_t, and let the corresponding arc be $[L_t, R_t]$, with $R_t - L_t \equiv 2t (\bmod n)$. Under the given restrictions there are three choices for the next arc to be picked, and the process picks one of these uniformly at random. That is:

$$P_{t+1} = \begin{cases} P_t \cup \{\{L_t - 2, L_t - 1\}\} & \text{w.p. } \frac{1}{3} \\ P_t \cup \{\{L_t - 1, R_t + 1\}\} & \text{w.p. } \frac{1}{3} \\ P_t \cup \{\{R_t + 1, R_t + 2\}\} & \text{w.p. } \frac{1}{3} \end{cases}$$

Here the additions and subtractions on the indices are *modulo n*. Let the final set of $n/2$ pairs be $P = P_{\frac{n}{2}}$. The process now goes over all pairs inside P and colours one variable by $\mathbf{y}$ and other by $\mathbf{z}$, uniformly at random to obtain the equipartition.

For ease of notation, we will say that both the set of $n/2$ pairs P and the actual partition $\pi(P) = \mathbf{y} \sqcup \mathbf{z}$ are generated using $\mathcal{D}$. That is, we will use both $\pi(P) \sim \mathcal{D}$ and $P \sim \mathcal{D}$. As we mentioned earlier, the set of arc partitions is exactly the support of this distribution $\mathcal{D}$. Similarly, an *arc-full-rank* polynomial will be a polynomial f for which $M_{\mathbf{y},\mathbf{z}}(f)$ has rank $2^{n/2}$ for every arc partition $\mathbf{y}, \mathbf{z}$.

The task is now in two parts. The first is to show that we can find a multilinear ABP that computes an *arc-full-rank* polynomial f. Then, we have to show that any small multilinear formula is rank-deficient under a random arc-partition.

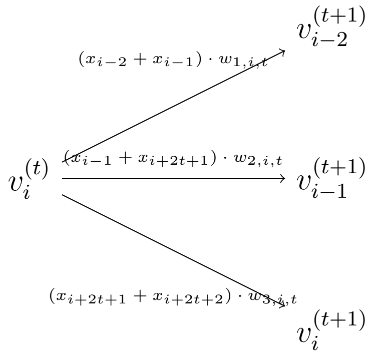
15.2 Upper bound with ABPs

We want the hard polynomial to have full rank with respect to every arc partition, so let us see how we can come up with a full rank polynomial for a fixed partition $\mathbf{y} \sqcup \mathbf{z}$. Say $\mathbf{y} = \{y_1, \ldots, y_m\}$ and $\mathbf{z} = \{z_1, \ldots, z_m\}$. We know that the polynomial $P(\mathbf{y}, \mathbf{z}) = \Pi_{i \in [m]}(y_i + z_i)$ has full rank with respect to the given partition.

Now we want to extend this idea to having full rank over any arc partition. One way to do this is to add a few (at most polynomially many) extra variables $\mathbf{w}$ so that for any arc partition π there will be an assignment ϕ to $\mathbf{w}$ that will ensure that $F_n(\phi(\mathbf{w}), \pi(\mathbf{x}))$ looks like $P(\mathbf{y}, \mathbf{z})$. We are going to do just that and additionally ensure that F_n is still computable by a small multilinear ABP.

Recall that for sampling of pairs from $\mathcal{D}$ we maintained the invariant that the pairs in P_t cover a contiguous arc $[L_t, R_t]$ of length exactly $2t$ on the n-cycle. This gives us that for any t, after t steps the number of distinct P_ts possible, is at most n. Moreover for the $(t+1)^{th}$ sampling step it is enough just to know L_t and R_t. This helps us construct the following ABP of width $\leq n$ and $n/2$ layers, that we describe by describing the underlying graph.

Let the nodes in each layer t be labelled as $v_1^{(t)}, \cdots, v_n^{(t)}$. The node $v_i^{(t)}$ would correspond to the fact that the current arc is $[i, i+2t] \bmod n$. We now describe the edges of the ABP. There would be three edges out of $v_i^{(t)}$ and they are as follows:


$v_{i-2}^{(t+1)}$
$(x_{i-2} + x_{i-1}) \cdot w_{1,i,t}$
$(x_{i-1} + x_{i+2t+1}) \cdot w_{2,i,t}$
$v_i^{(t)}$
$v_{i-1}^{(t+1)}$
$(x_{i+2t+1} + x_{i+2t+2}) \cdot w_{3,i,t}$
$v_i^{(t+1)}$

The ABP therefore has $O(n^2)$ vertices and since there are three edges out of each vertex, there are $O(n^2)$ edges. The polynomial computed by the ABP, F_n, would be our hard polynomial. From the construction, the following lemma is easy to verify and is left as an easy exercise.

Exercise 15.1 *Show that the polynomial F_n constructed above is full-rank with respect to every arc-partition.*

15.2.1 Lower bound against formulas

As seen earlier in Chapter 13, we will prove the lower bound by proving that the probability of a *log-product term* having rank that is n^{δ} close to full is inverse quasipolynomially small (for some small enough $\delta > 0$). The difference here is the distribution is over the arc-partitions and not the uniform distribution.

Suppose $|X| = n$. As in the previous settings, we have a partition $X = X_1 \sqcup \cdots \sqcup X_t$ from our log-product decomposition with $k = \Theta(\log n)$ and $|X_i| \geq |X|^{7/8}$ for all i. We'll refer to each X_i as a bucket. If we pick an arc-partition at random, we would like to estimate the probability that none of the X_i's are n^{δ} unbalanced. We would like to show that this is inverse quasi-polynomially small.

Arc partitions are chosen by first choosing a pairing P and then bi-colouring each pair $p \in P$. Firstly, note that if X_i picks up only "whole pairs" from the set of pairs P, then no matter how we bi-colour each pair we would have X_i being completely balanced. It is therefore sensible to look at buckets that pick up exactly one element from a lot of pairs.

$$V_i(P) = \{p \in P \ : \ |p \cap X_i| = 1\}$$
$$G(P) = \left\{i \in [t] \ : \ |V_i(P)| \geq n^{1/1000}\right\}$$

The set $G(P)$ refers to the "good buckets" that *cut* many pairs; the colours of these end-points are completely independent. Therefore, at the very least, for each $i \in G(P)$, we can hope to say that the probability that X_i is not n^{δ}-unbalanced is small (inverse polynomial).

Lemma 15.3. *Let P be a partition of X into pairs and let $G(P)$ be as defined above. Then, for any $i \in G(P)$,*

$$\Pr[X_i \textit{ is } n^{\delta}\textit{-balanced}] \leq O\left(\frac{2n^{\delta}}{n^{1/2000}}\right)$$

where the probability is over a random bi-colouring of the pairs in P.

This lemma is easy to prove. If the set $G(P)$ is large, we can then hope to say that we have a good number of independent events and hope for an inverse quasi-polynomial probability.

Arranging for many independent events

The following lemma states that for most pairings $P \sim \mathcal{D}$ we must have many buckets in $G(P)$. This is the most technical part of the paper and we'll see a very rough sketch in the next section.

Lemma 15.4. *Given a partition $X = X_1 \sqcup \cdots \sqcup X_t$ with $t = \Theta(\log n)$ and $|X_i| \geq n^{7/8}$ for all $i \in [t]$. Then,*

$$\Pr_{P \in \mathcal{D}} \left[|G(P)| \leq \frac{t}{1000} \right] \leq n^{-\Omega(t)}$$

where $\mathcal{D}$ is the distribution for arc partitions.

Hence, with high probability we have $|G(P)| \geq t/1000 = \Theta(\log n)$. Now we would like to say that there are many (almost) independent events among these good buckets.

Assume the good buckets are $X_1, \ldots, X_r$ without loss of generality. From Lemma 15.3, we know that the probability that X_1 is too balanced is small. However we now want to argue the same for X_2, even after we condition on X_1 being too balanced. This need not always be true. Consider the setting that both $X_1 \cup X_2$ is a just a union of pairs, but each pair has one end-point in X_1 and the other in X_2. Now if X_1 is very balanced, then X_2 would also be very balanced.

Thus, in order to say X_2 would likely be unbalanced even after conditioning on X_1 being balanced, we need to say that X_2 has many *other* pairs that are cut (that is, these aren't accounted for in X_1). More generally, want to ensure that for each $i \in [r]$, the bucket X_i cuts many pairs $p \in P$ whose other end-point is *not* in $X_1 \sqcup \cdots \sqcup X_{i-1}$.

Let us consider a graph $H(P)$ with the buckets $[t]$ as vertices and let us mark the good buckets $[r]$ as red. Add an edge (i, j) if $|V_i(P) \cap V_j(P)| \geq n^{1/1500}$; that is there are many pairs with one end-point in i and one in j. Note that each good bucket i has degree at least 1 in $H(P)$ as each good bucket cuts $n^{1/1000}$ pairs and we only have $O(\log n)$ buckets.

Lemma 15.5. *There exists a sequence $B = (b_1, \ldots, b_\ell)$ of good buckets from $H(P)$ with $\ell \geq \frac{r}{2}$ such that for all $i \in [\ell]$, the vertex b_i in the graph $H(P) \setminus \{b_1, \ldots, b_{i-1}\}$ has degree ≥ 1.*

Proof. Consider a spanning forest of $H(P)$. Since every good bucket has degree at least 1, the spanning forest has at least $r/2$ edges emanating from good buckets. The required sequence of vertices can be obtained by repeatedly removing leaves of this forest. □

Let $(b_1, \ldots, b_\ell)$ be such a sequence for $H(P)$, with $\ell \geq r/2 = \Omega(t)$. For brevity, let pairs that are not cut by any of the b_is be $\tilde{P}$. For every $i \in [r]$, define by P_i the pairs from $(P \setminus (P_1 \cup \cdots \cup P_{i-1}))$ that are cut by b_i. Now view the random colouring $\pi(P)$ of pairs from P to be happening in the order $P_1, \ldots, P_r, \tilde{P}$. From Lemma 15.5 we have that $|P_i| \geq n^{1/1500}$ for all $i \in [r]$. As mentioned above, this will help us yield an inverse polynomial factor for each of the r buckets, even after conditioning on previous colourings. Also let $n_i = |b_i|$ and for $\pi(P) = \mathbf{y}, \mathbf{z} \sim \mathcal{D}$, let $y_i = |b_i \cap \mathbf{y}|$. Let $\delta > 0$ be a suitably small constant. For $i \in [r]$, let the E_i refer to the event that $|y_i - n_i/2| \leq n^\delta$. In the case when $|G(P)| \geq K/1000$, we get that

$$\begin{aligned}
\Pr_{\pi(P)=\mathbf{y},\mathbf{z}\sim\mathcal{D}}\left[\operatorname{rank}\left(M_{Y,Z}(g)\right) \geq 2^{n/2-n^\delta}\right] &\leq \Pr_{\pi(P)=\mathbf{y},\mathbf{z}\sim\mathcal{D}}\left[\bigwedge_{i\in[r]} E_i\right] \\
&= \prod_{i\in[r]} \Pr_{\pi(P)=\mathbf{y},\mathbf{z}\sim\mathcal{D}}\left[E_i \middle| \bigwedge_{j<i} E_j\right] \\
&\leq \prod_{i\in[r]} O\left(\frac{2n^\delta}{\sqrt{|P_i|}}\right) \\
&\leq \prod_{i\in[r]} n^{-\Omega(1)} = n^{-\Omega(K)}.
\end{aligned}$$

Combining this with the fact that $G(P)$ must be large with high probability (Lemma 15.4), we get

$$\begin{aligned}
&\Pr_{\pi(P)=\mathbf{y},\mathbf{z}\sim\mathcal{D}}\left[\operatorname{rank}\left(M_{Y,Z}(g)\right) \geq 2^{n/2-n^\delta}\right] \\
&\leq \Pr_{\pi(P)=\mathbf{y},\mathbf{z}\sim\mathcal{D}}\left[\operatorname{rank}\left(M_{Y,Z}(g)\right) \geq 2^{n/2-n^\delta} \middle| |G(P)| \geq K/1000\right] \Pr_{P\sim\mathcal{D}}\left[|G(P)| \geq K/1000\right] \\
&\qquad + \Pr_{P\sim\mathcal{D}}\left[|G(P)| \leq K/1000\right] \\
&\leq n^{-\Omega(K)} + n^{-\Omega(K)} = n^{-\Omega(\log n)}.
\end{aligned}$$

This would finally complete the proof of Theorem 15.2.

15.3 Proof sketch for Lemma 15.4

We will a brief sketch of the proof of Lemma 15.4. The full proof is quite invovled and a full description can of course be found in their paper [DMPY12]. We will try to give a rough sketch of the main ideas in the proof.

We will call an index j a *jump* with respect to the bucket X_k, if $x_j \in X_k$ and $x_{j+1} \notin X_k$. The following observation is easy to see.

Observation 15.6. *For any $j \neq 0, 1$, the probability that $(j, j+1)$ is chosen as a pair by an arc partition is at least $1/9$.*

Therefore, if we have many buckets with lots of jumps, then a lot of them would contribute pairs that are cut. More formally, suppose we have more then $K/2$ buckets with at least $N = n^{1/100}$ jumps. We then collect at most N jumps from each bucket to get a total of $\geq KN/2$ distinct indices. Observation 15.6 says the probability that a particular jump results in a pair that is cut bounded below by some constant. We therefore conclude that $\geq KN/100$ of the total jumps get converted into cuts with high probability. Since we chose at most N jumps from each bucket, this means that at least $K/1000$ buckets contribute at least $N^{1/10} = n^{1/1000}$ cuts, which proves Lemma 15.4 for this case.

The trickier case is when there aren't many buckets with a lot of jumps. Let us consider an extreme case to understand this better — buckets form large contiguous segments of the circle.

Suppose we have a partial arc $[L_t, R_t]$, and suppose we have two different buckets X_i and X_j on the two ends that are to be processed. If these segments are sufficiently large, then a constant fraction of these would be matched between X_i and X_j (via the $(L_t - 1, R_t + 1)$ sort of pairs) with high probability, making both i and j good buckets.

The other case could be that the two segments at the ends of L_t and R_t belong to the same bucket say X_i. Let us call these segments A_L and A_R and let their lengths be a_L and a_R. The next few pairs would not yield any cuts as they will below to the same bucket. But the key insight is this — consider the arc partition when one of the two sides is completely processed (that is, A_L or A_R is completely contained in $[L_{t'}, R_{t'}]$ at that time). When this happens, the probability that there are very few elements left in the other part is inverse polynomially small. This is once again very similar to the fact that a random partition would not cut a set in too balanced a fashion. In similar spirit, the probability that there are too few elements in A_L left after A_R has been completely processed (or vice-versa) is inverse polynomially small. Hence, we would essentially reduce to the previous case when we have two sufficiently large segments from different buckets on the ends of L_t and R_t. Therefore, the probability that i fails to be a part of $G(P)$ is inverse polynomially small, and these events are almost independent for various i's. Therefore, the probability that $G(P) < t/1000$ is at most $n^{-\Omega(t)}$ as claimed by Lemma 15.4.

The formal proof requires some careful analysis and Dvir et al. [DMPY12] do this by considering the process as a "random walk on a distorted chess board".

15.4 What about IMM?

The result of Dvir, Malod, Perifel and Yehudayoff [DMPY12] that we dicussed separate the classes of syntactic multilinear ABPs and multilinear formulas. A natural question is whether this also works for the iterated matrix multiplication polynomial (IMM). Note that the polynomial used for this lower bound was a syntactic multilinear ABP but is *not* a multilinear projection of IMM! The same variable may occur in many edges even though each path may use each variable at most once. Therefore, such a reduction does not follow from this proof. In fact, this is an important open problem.

Open Problem 15.1 [Multilinear formula lower bounds for IMM]

Show that $\mathrm{IMM}_{n,d}$ *requires multilinear formulas of* $(nd)^{\Omega(\log nd)}$ *to compute it.*

Chapter 16

Tensor rank and formula lower bounds

In this chapter, we will establish an apriori surprising connection between lower bounds for homogeneous arithmetic formula and constructing explicit tensors of full rank. The chapter is based on a result of Raz [Raz10c].

16.1 Tensors

Tensors are natural *higher dimensional* analogues of matrices. A matrices is nothing but a two dimensional array filled in with numbers from some underlying field. A tensor is a higher dimensional version of this, where we have an d-dimensional cuboid filled with numbers. A tensor T is a map of the form

$$T : [m_1] \times \cdots \times [m_d] \longrightarrow \mathbb{F}$$

the same way an $m \times n$ matrix a map from $[m] \times [n] \to \mathbb{F}$. However, if one wants to understand a tensor more functionally (similarly to how it is useful to think of matrices as linear transformations on vector spaces), it is more natural to extend this definition linearly as follows.

Definition 16.1 (Tensor)**.** *A tensor T is a map of the form*

$$T : V_1 \times \cdots \times V_d \longrightarrow \mathbb{F}$$

where each V_i is a vector space over $\mathbb{F}$, of say dimension m_i, which is linear in every coor-

dinate i.e.

$$T(\mathbf{v}_1, \dots, \alpha\mathbf{v}_i + \beta\mathbf{v}'_i, \dots, \mathbf{v}_d) = \alpha T(\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_d) + \beta T(\mathbf{v}_1, \dots, \mathbf{v}'_i, \dots, \mathbf{v}_d).$$

The parameter d is called the order *of the tensor, and we say that the* shape of T *is* $m_1 \times \cdots \times m_d$. ♢

Since a tensor is linear in every coordinate, it suffices to specify the image of T at the basis of $V_1 \times \cdots \times V_d$ and extend it linearly. So a tensor can indeed be thought of as filling up a d-dimensional array of shape $[m_1] \times \cdots \times [m_d]$ by field elements, the same way an $m \times n$ matrix is specified by an $m \times n$ array filled up with field elements. Indeed, a matrix is nothing but an order-2 tensor.

It would sometimes be useful to switch between the two notions of thinking of a tensor as a multilinear map from $V_1 \times \cdots \times V_d$ to $\mathbb{F}$ and thinking a tensor as just a map from $[m_1] \times \cdots \times [m_d]$. So throughout this chapter, we shall fix a basis $\{\mathbf{e}_{i1}, \dots, \mathbf{e}_{im_i}\}$ for V_i and when we shall use $T[j_1, \dots, j_d]$ to really denote $T(\mathbf{e}_{1j_1}, \dots, \mathbf{e}_{dj_d})$.

16.1.1 Tensors as polynomials

In our setting, it would be useful to think of tensors as a restricted form of multilinear polynomials that are called *set-multilinear polynomials.*

Definition 16.2 (Set-multilinear polynomials)**.** *Let* $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ *be a partition of variables and let* $|\mathbf{x}_i| = m_i$. *A polynomial* $f(\mathbf{x})$ *is said to be* set-multilinear *with respect to the above partition if every monomial m in f satisfies* $|m \cap X_i| = 1$ *for all* $i \in [d]$. ♢

In other words, each monomial in f picks up at most one variable from each part in the partition. It is easy to see that many natural polynomials such as Det or Perm or NW are all set-multilinear for an appropriate partition of variables.

Observation 16.3. *For any tensor T of shape* $[m_1] \times \cdots \times [m_d]$, *we can associate a set-multilinear polynomial* $f(\mathbf{x})$ *with* $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ *and* $\mathbf{x}_i = \{x_{i1}, \dots, x_{im_i}\}$ *as*

$$f(\mathbf{x}) \quad = \quad \sum_{\substack{1 \leq i_j \leq m_j \\ \forall j \in [d]}} T(i_1, \dots, i_d) \cdot x_{1i_1} \cdots x_{di_d}. \tag{16.4}$$

Another representation is to just use a single variable x_j for a part $\mathbf{x}_j$ and use higher powers. That way, we can associate the following polynomial with a tensor T:

$$f(x_1, \ldots, x_d) = \sum_{\substack{1 \le i_j \le m_j \\ \forall j \in [d]}} T(i_1, \ldots, i_d) \cdot x_1^{i_1} \cdots x_d^{i_d}. \tag{16.5}$$

The same also holds in the other direction where we can interpret any set-multilinear polynomial as an appropriate tensor.

16.1.2 Rank of a tensor

Just like any matrix has a natural definition of *rank*, there is an analogue for tensors as well. The rank of a matrix M can be defined as the smallest r for which M can be written as a sum of r matrices of rank 1. A rank-1 matrix is just a matrix of the form $\mathbf{u}\mathbf{v}^T$ where the (i, j)-th entry is $u_i v_j$. We shall abuse[1] notation and use $\mathbf{u} \otimes \mathbf{v}$ to denote the order-2 tensor T where $T[i, j] = u_i v_j$. This naturally generalizes to higher order as well.

Definition 16.6 (Elementary tensors, and tensor rank)**.** *For* $\mathbf{v}_1 \in V_1, \ldots, \mathbf{v}_d \in V_d$*, define the tensor* $\mathbf{v}_1 \otimes \cdots \otimes \mathbf{v}_d$ *to be the tensor* E *given by*

$$E[j_1, \ldots, j_d] = (\mathbf{v}_1)_{j_1} \cdots (\mathbf{v}_d)_{j_d}.$$

We shall call such tensors as elementary tensors *or* rank-1 tensors*.*

For an arbitrary tensor T*, the* tensor rank of T*, denoted by* $\operatorname{rank}(T)$*, is the smallest* r *such that* T *can be expressed as a sum of* r *elementary tensors.* ◊

What do elementary tensors look like as polynomials? Let us consider the *set-multilinear* polynomial setting as in (16.4). It is easy to see that a rank-1 tensor is precisely a *set-multilinear* product of linear forms such as

$$f(\mathbf{x}) = \ell_1(\mathbf{x}_1) \cdots \ell_d(\mathbf{x}_d)$$

where each $\ell_i(\mathbf{x}_i)$ is a linear form in the variables in $\mathbf{x}_i$.

In the setting of (16.5), it is easy to see that a rank-1 tensor is precisely a *product of*

[1] it is abuse because it is really a tensor product of $\mathbf{u}$ and $\mathbf{v}^T$.

univariates such as

$$f(\mathbf{x}) \quad = \quad f_1(x_1) \cdots f_d(x_d).$$

Hence the following three questions are equivalent:

- Given a tensor T, find its rank.
- Given a set-multilinear polynomial f, find the smallest set-multilinear $\Sigma\Pi\Sigma$ circuit computing it.
- Given a polynomial f, find the smallest expression as a sum of product of univariates.

However, unlike matrices, computing the rank of even an order-3 tensor is NP-hard [Hås90]. But one could still ask if we can prove good upper or lower bounds for some specific tensors, or try to find a tensor with large rank. But before that, let us look at some basic properties that tensor rank satisfies.

Properties of tensor rank

The following are a couple of basic properties that follows almost immediately from the definitions.

Lemma 16.7 (Sub-additivity of tensor rank)**.** *Let T_1 and T_2 be two tensors of the same shape and order. Then, if $T = T_1 + T_2$, then* $\operatorname{rank}(T) \leq \operatorname{rank}(T_1) + \operatorname{rank}(T_2)$.

Lemma 16.8 (Sub-multiplicativity of tensor rank)**.** *Let $T_1 : V_1 \times \cdots \times V_{d_1} \to \mathbb{F}$ and $T_2 : W_1 \times \cdots \times W_{d_2} \to \mathbb{F}$ be two tensors. Then if $T = T_1 \otimes T_2$ given by*

$$T[i_1, \ldots, i_{d_1}, j_1, \ldots, j_{d_2}] = T_1[i_1, \ldots, i_{d_1}] \cdot T_2[j_1, \ldots, j_{d_2}],$$

then $\operatorname{rank}(T) \leq \operatorname{rank}(T_1) \cdot \operatorname{rank}(T_2)$.

16.1.3 Upper bounds on tensor rank

Let us consider an order-d tensor T of shape $n \times \cdots \times n$. How large can $\operatorname{rank}(T)$ be? One possible upper bound we could say is n^d. Surely, the tensor is an d-dimensional array with just n^d entries. We can certainly write it as a sum of n^d elementary tensors of the form $\mathbf{e}_{j_1} \otimes \cdots \otimes \mathbf{e}_{j_d}$. So clearly $\operatorname{rank}(T) \leq n^d$. But we can do a little better. Consider the case when $d = 2$, and we have an $n \times n$ matrix. The bound on the rank is not n^2 but rather n. This indicates that one should be able to do a bit better than n^d for the general case. Indeed we can.

Lemma 16.9. *Let T be an order-d tensor of shape $n \times \cdots \times n$. Then,* $\operatorname{rank}(T) \leq n^{d-1}$.

Proof. Let us revisit the case when $d = 2$, where we know an $n \times n$ matrix has rank at most n. Interpreting this statement via (16.5), this implies that of $q(x_1, x_2)$ is a bi-variate with degree in each variable bounded by n, then q can be written as

$$q(x_1, x_2) = \sum_{i=1}^{n} g_i(x_1) h_i(x_2).$$

Therefore, if $f(x_1, \ldots, x_d)$ is a polynomial with degree in each variable bounded by n, then we can write f as

$$\begin{aligned} f(x_1, \ldots, x_d) &= \sum_{m \in \operatorname{Mon}\{x_3, \ldots, x_d\}} m \cdot q_m(x_1, x_2) \\ &= \sum_{m \in \operatorname{Mon}\{x_3, \ldots, x_d\}} m \cdot \left(\sum_{i=1}^{n} g_{m,i}(x_1) \cdot h_{m,i}(x_2) \right) \end{aligned}$$

which is a sum of product of univariates of top fan-in $n \cdot n^{d-2} = n^{d-1}$. □

A counting argument would say that there do exist tensors of rank at least n^{d-1}/d as each elementary tensor has *nd degrees of freedom* and an arbitrary tensor has n^d *degrees of freedom*. One might think that the above upper bound of n^{d-1} should be tight. Bizarrely, it is not! For example (cf. [Pam85]), the maximum rank of any tensor of shape $2 \times 2 \times 2$ is 3 and not 4 as one might expect! Tensor rank also behaves in some strange ways under *limits* unlike the usual matrix rank. But a big open question is to find explicit tensors of such large rank.

Open Problem 16.1

Can we find an explicit tensor $T : [n]^d \to \mathbb{F}$ of rank $n^{d(1-o(1))}$?

Raz's [Raz10c] showed that in certain regimes, an answer to the above question would yield arithmetic formula lower bounds.

16.2 Tensor rank of small formulas

From this section onwards, we shall assume the *set-multilinear polynomial* interpretation of a tensor $T : [n]^d \to \mathbb{F}$ as described in (16.4). Hence our variables $\mathbf{x}$ is partitioned as $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ with $|\mathbf{x}_i| = n$ for all $i \in [d]$. The main motivating question would be the following:

> If f is a set-multilinear polynomial that is computed by a small formula, what can one say about its tensor rank?

To begin with, let us restrict ourselves to certain structured formulas that in a sense *respects* the partition defined.

Definition 16.10 (Set-multilinear formulas)**.** *A formula Φ is said to be a* set-multilinear formula *if for every gate in the formula computes a set-multilinear polynomial.* ◊

From the above definition, note that the set-multilinear formulas are a subclass of homogeneous formulas. As in the multilinear setting, it is easy to see that set-multilinearity for formulas can be made a *syntactic* restriction where each gate computes a tensor, with addition gates only adding “alike” tensors and multiplication gates multiplying disjoint tensors.

Exercise 16.1 *Show that set-multilinear formulas can, without loss of generality, be assumed to be syntactically set-multilinear formulas.*

An easier question to the one above would be the following:

> If f is a set-multilinear polynomial that is computed by a small *set-multilinear* formula, what can one say about its tensor rank?

In the rest of this chapter, we shall prove the following result of Raz [Raz10c].

Theorem 16.11 ([Raz10c])**.** *Let Φ be a set-multilinear formula of size $s \leq n^c$ computing a polynomial $f(\mathbf{x}_1, \ldots, \mathbf{x}_d)$. Then,*

$$\operatorname{rank}(f) \quad \leq \quad \frac{n^d}{n^{d/\exp(c)}}.$$

In the setting when d is small, Raz [Raz10c] also showed that formulas can be converted to set-multilinear formulas with a modest cost.

Theorem 16.12 ([Raz10c]). *Suppose $d = O\left(\frac{\log n}{\log\log n}\right)$. If Φ is a formula of size $s = \text{poly}(n)$ that computes a set-multilinear polynomial $f(\mathbf{x}_1, \ldots, \mathbf{x}_d)$, then there is a* set-multilinear *formula of* poly(s) *size that computes f as well.*

As a corollary, finding explicit tensors of almost full rank would imply super-polynomial formula lower bounds in the low-degree regime.

Corollary 16.13 ([Raz10c]). *If $f(\mathbf{x}_1, \ldots, \mathbf{x}_d)$ is an explicit tensor of rank $n^{d(1-o(1))}$ with $\omega(1) = d = O\left(\frac{\log n}{\log\log n}\right)$, then any formula computing f must be of super-polynomial size.*

The above two theorems are of very different flavours and should really be thought of as two independent surprising results. Theorem 16.11 is a tensor-rank upper bound and Theorem 16.12 is a structural result. We shall first address Theorem 16.11 in the next section and address Theorem 16.12 after that.

16.2.1 The tensor-rank upper-bound

We shall now prove Theorem 16.11. The proof described here is not the original proof in [Raz10c] but is an alternate proof by Suryajith Chillara, Mrinal Kumar and Ramprasad Saptharishi.

Proof of Theorem 16.11. For this we would need the slightly better depth reduction for homogeneous formulas (Theorem 5.23) of Saptharishi and Vinay [SV14]. We recall the statement here.

Theorem (Theorem 5.23). *Let f be a homogeneous n-variate degree d polynomial computed by a size s homogeneous formula. Then for any $0 < t \leq d$, f can be equivalently computed by a homogeneous $\Sigma\Pi^{[a]}\Sigma\Pi^{[t]}$ formula of top fan-in $s^{10(d/t)}$ where*

$$a \geq \frac{1}{10}\left(\frac{d}{t}\right)\log t.$$

It is a fairly straightforward observation to see that the above theorem preserves multilinearity and set-multilinearity as well. We shall start with the set-multilinear formula Φ of size $s = n^c$ that computes the polynomial $f(\mathbf{x}_1, \ldots, \mathbf{x}_d)$ and apply Theorem 5.23 for a suitable choice of t (that shall be set shortly). Therefore we now have a set-multilinear expression of the form

$$f \quad = \quad T_1 + \cdots + T_{s'}$$

where $s' \leq s^{10(d/t)} = n^{10c(d/t)}$ and each $T_i = Q_{i1} \cdots Q_{ia_i}$ is a set-multilinear product. Let us fix one such term $T = Q_1 \cdots Q_a$ and we know that this is a set-multilinear product with $a \geq \left(\frac{d \log t}{10t}\right)$. Let $d_i = \deg(Q_i)$. By the sub-multiplicativity of tensor rank (Lemma 16.8) and the trivial upper bound (Lemma 16.9) we have

$$\begin{aligned}
\operatorname{rank}(T) &\leq n^{d_1 - 1} \cdots n^{d_a - 1} \\
&= n^{d-a} \\
\implies \operatorname{rank}(f) &\leq s' \cdot n^{d-a} && \text{(Lemma 16.7)} \\
&= \left(\frac{n^d}{n^{a-10c(d/t)}}\right)
\end{aligned}$$

Let us focus on the exponent of n in the denominator. Using the lower bound on a from Theorem 5.23, we get

$$\begin{aligned}
a - 10c(d/t) &\geq \frac{d \log t}{10t} - 10c\left(\frac{d}{t}\right) \\
&= \left(\frac{d}{t}\right)\left(\frac{\log t}{10} - 10c\right) \\
&= \left(\frac{d}{2^{O(c)}}\right) && \text{if we set } \frac{\log t}{10} = 11c \\
\implies \operatorname{rank}(f) &\leq \frac{n^d}{n^{d/\exp(c)}}
\end{aligned}$$

which is what we set out to prove. □

16.2.2 Making formulas set-multilinear

In this section we shall prove Theorem 16.12. The proof would proceed in two steps, both of which are quite interested in their own right.

- **Homogenization :** In the first step, we shall convert a formula that compute a homogeneous polynomial of degree d in n variables to a n homogeneous formula computing the same polynomial. It would turn out that if $d = O(\log n)$, then this transformation only incurs a poly(n) blow-up in the size.
- **Set multilinearization :** In the second step, we show that for any homogeneous arithmetic formula that computes a set multilinear polynomial of degree d in n variables can be converted to a set-multilinear formula computing the same polynomial. In this

setting, if $d \leq O(\log n / \log\log n)$, the blow-up in the size of the formula will only be $\operatorname{poly}(n)$.

Homogenization of low-degree formulas

Lemma 16.14 ([Raz10c]). *Let Φ be a formula of size s computing an n-variate homogeneous polynomial f of degree d. Then, there is a* homogeneous formula Φ' *computing f of size at most* $\operatorname{poly}\left(s, \binom{d+\log s}{d}\right)$.

In particular, if $d = O(\log n)$ and $n = \operatorname{poly}(n)$ then we have $\operatorname{size}(\Phi') = \operatorname{poly}(n)$ *as well.*

Proof. Recall that due to the depth reduction result of Brent and Spira (Lemma 5.5), we can assume without loss of generality that the fan-in of every gate in Φ at most 2 and the depth of Φ is $O(\log s)$. The construction of the new formula Φ' would have no surprises – homogenize the formula Φ to obtain a circuit C using the standard homogenization (Lemma 5.2), and unravel the circuit to make it a formula in the most natural way. It is the analysis of the size that would give the lemma.

For every gate v in Φ, we have $d+1$ gates $(v,0),(v,2),\ldots,(v,d)$ in C. Semantically, the polynomial computed at such a gate (v,i) is the degree-i homogeneous component of the polynomial computed at v in Φ. As in Lemma 5.2, we shall connect these edges as follows:

$$\begin{array}{llcll} v = u + w & \Longrightarrow & (v,i) & = & (u,i) \quad + \quad (w,i) \quad \text{for all } i \\ v = u \times w & \Longrightarrow & (v,i) & = & \displaystyle\sum_{j=0}^{i} (u,j)\cdot(w,i-j) \quad \text{for all } i \end{array}$$

Hence, we now have a homogeneous circuit C that computes f and has size at most $s' = O(sd^2)$. Furthermore, the depth of this circuit is at most twice the depth of the formula Φ which was $O(\log s)$. Hence C is a homogeneous circuit of depth $O(\log s)$ and size $O(sd^2)$ computing f.

To convert C into a formula, we will do the natural operation of *recomputing* a node whenever we need to reuse the computation. This is equivalent to duplicating every gate (v,i) of C as many times are there are paths from this gate to the root fo C. Thus, in order to upper bound the size of the resulting formula, we will require an upper bound on the number of distinct paths from every gate (v,i) of C to its root.

Currently, C is a circuit because if v is a product gate of Φ with children u and w, then the out degree of (u,j) and (w,j) in C could be more than 1 as it *contributes* to (v,j') for all

$j \leq j' \leq d$. Hence, the resulting structure is a circuit and not a formula. However, at sum gates, the out degree of the children continue to be 1. But this gives us a good understanding of the many paths from (v, i) to the root in C. Let $v \to v_1 \to \cdots \to v_r$ be the path from v to the root ($= v_r$) in Φ. Then, the paths from (v, i) to (v_r, d) will be of the form

$$(v, i) \to (v_1, i_1) \to \cdots \to (v_r, i_r)$$

where $i \leq i_1 \leq \cdots \leq i_r = d$. But the number of such choices for $(i_1, \ldots, i_r)$ the same as the number of non-negative integer solutions to $b_1 + \cdots + b_r = d - i$ which is at most $\binom{r+d}{d}$. We know that the circuit has depth at most $O(\log s)$ and hence $r \leq O(\log s)$. Therefore, the number of paths from any (v, i) to the root is at most $\binom{d+O(\log s)}{d}$. Hence, if Φ' is the formula obtained by unravelling C, we have

$$\mathrm{size}(\Phi') = \mathrm{poly}\left(s, \binom{d + \log s}{d}\right).$$

□

Set-multilinearization of low-degree formulas

We shall now show that a homogeneous formula can be converted to a set-multilinear formula with a cost that is affordable if d is small.

Lemma 16.15 (Formula Set-multilinearization)**.** *Let $f(\mathbf{x})$ be an n-variate degree set-multilinear polynomial with respect to the partition $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ that is computed by a homogeneous formula Φ of size s. Then, there exists a set-multilinear formula Φ' of size at most $2^{O(d \log\log s)}$ which computes f.*

In particular, if $d = O\left(\frac{\log n}{\log\log n}\right)$ and $s = \mathrm{poly}(n)$ then the $\mathrm{size}(\Phi') = \mathrm{poly}(n)$.

Proof. To start with, without loss of generality, let us assume that the formula Φ is fan-in 2, homogeneous and has depth $O(\log s)$. In the first step, we set multilinearize Φ in the obvious way to obtain a circuit C. To this end, for every gate v in Φ, and vector $\mathbf{a} = (a_1, \ldots, a_d) \in \{0, 1\}^d$, there is a gate $(v, \mathbf{a})$ in C. Semantically, the the polynomial at $(v, \mathbf{a})$ consists of the monomials in the polynomial computed at v (in Φ) which contain exactly one variable from the set $\mathbf{x}_i$ for every i such that $a_i = 1$. The edges in C are connected in a natural way, namely for a gate v with children u and w, we have the following edges:

$$v = u + w \quad \Longrightarrow \quad (v, \mathbf{a}) \;=\; (u, \mathbf{a}) \;+\; (w, \mathbf{a}) \quad \text{for all } \mathbf{a}$$

$$v = u \times w \quad \Longrightarrow \quad (v, \mathbf{a}) \;=\; \sum_{\mathbf{b}+\mathbf{c}=\mathbf{a}} (u, \mathbf{b}) \cdot (w, \mathbf{c}) \quad \text{for all } \mathbf{a}.$$

Clearly, the size of C is at most $2^d \cdot s$. Moreover, the gates in C which have out degree more than one are of the form $(u, \mathbf{a})$ where u is a child of some multiplication gates at Φ. The root of C would now be $(\text{root}, \mathbf{1})$.

Like in the proof of Lemma 16.14, we now convert the circuit C to a formula by replicating nodes whenever we need to reuse computations. Hence, we would require as many copies of a gate $(v, \mathbf{a})$ as there are paths from $(v, \mathbf{a})$ to the root of C. In order to bound the blow up in size in the process, we will prove an upper bound on the number of such paths. Once again, let $v \to v_1 \to \cdots \to v_r$ be the path from v to the root ($= v_r$). Then any path from $(v, \mathbf{a})$ to $(v_r, \mathbf{1})$ is of the form

$$(v, \mathbf{a}) \to (v_1, \mathbf{a}_1) \to \cdots \to (v_r, \mathbf{1})$$

with $\mathbf{a} \leq \mathbf{a}_1 \leq \cdots \leq \mathbf{1}$ in the point-wise sense. Therefore, the number of paths from $(v, \mathbf{a})$ to the root is at most the number of sequences of "increasing" vectors $\mathbf{a}_1, \cdots, \mathbf{a}_r \in \{0, 1\}^d$. Note that the number of such sequences is at most r^d (why?) and hence the number of paths from $(v, \mathbf{a})$ to the root is at most $2^{d \log r} = 2^{O(d \log \log s)}$. Hence the size of the resulting formula Φ' is at most $s \cdot 2^{O(d \log \log s)}$. □

Lemma 16.14 and Lemma 16.15 complete the proof of Theorem 16.12.

Part V

Separations in the monotone world

Chapter 17

Separating monotone circuits and monotone ABPs

As mentioned earlier, we know that Formulas $\subseteq$ ABP $\subseteq$ Circuits and it is a major open problem to show if any of these are strict containments. Currently we do not even know how to prove super-polynomial lower bounds for any of these classes. However, we have just seen that we have techniques to prove lower bounds in the monotone setting. A natural question is therefore if any of these containments are strict in the monotone setting. In this chapter, we shall see a beautiful lower bound of Hrubeš and Yehudayoff [HY16].

Theorem 17.1 (Hrubeš and Yehudayoff [HY16])**.** *There is an explicit n-variate degree d polynomial P that is computable by a* $\operatorname{poly}(n,d)$ *sized monotone algebraic circuit such that any monotone algebraic ABP computing it must have size* $(nd)^{\Omega(\log nd)}$.

Monotone ABPs are defined analogously to Definition 8.5, where no path computes a monomial that is not present in the output polynomial. The above theorem show that

$$\text{Monotone ABPs} \subsetneq \text{Monotone Circuits.}$$

Weakness for ABPs?

What sort of weakness can we exploit for ABPs? There are many similarities between ABPs and circuits. Both can be homogenised without much blow-up in size. Even the frontier decomposition for ABPs and circuits looks similar:

If $f(\mathbf{x})$ is an n-variate degree d polynomial that is computable by a homogeneous size s circuit, then

$$f = \sum_{i=1}^{s} g_i h_i$$

where $\frac{d}{3} \leq \deg(g_i), \deg(h_i) \leq \frac{2d}{3}$ for all i.

If $f(\mathbf{x})$ is an n-variate degree d polynomial that is computable by a homogeneous ABP of size s, then for any $0 \leq k \leq d$ we have

$$f = \sum_{i=1}^{s} g_i h_i$$

where $\leq \deg(g_i) = k$ and $\deg(h_i) = d - k$ for all i.

Although the above frontier decompositions look very similar, observe that for ABPs we have a much finer control of what the degree of g_i and h_i are unlike for algebraic circuits. Hrubeš and Yehudayoff [HY16] show that this "weakness" can be exploited when working with the monotone setting.

17.1 The polynomial

Let d, m be parameters that will be set eventually (we will choose it such a way that $d = O(\log m)$). We shall identify $[m]$ with the additive group $\frac{\mathbb{Z}}{m\mathbb{Z}}$. Let T_d denote the complete binary tree of depth d (with 2^d leaves). A colouring of T_d will be a function $\chi : V(T_d) \to [m]$, which just assigns colours from $[m]$ to the vertices of the tree T_d. We shall say that a colouring χ is *valid* if for every internal node u with children v and w in T_d the colouring satisfies $\chi(u) = \chi(v) + \chi(w)$ (over $\frac{\mathbb{Z}}{m\mathbb{Z}}$).

Clearly, there are m^{2^d} possible valid colourings as the leaves can be assigned colours arbitrarily and the colours of the other vertices are forced. We will now build a polynomial from this.

Let $D = 2^{d+1} - 1 = |V(T_d)|$, the number of vertices of T_d, and let $\mathbf{x} = \{x_{ij} \ : \ i \in [D] \ , \ j \in [m]\}$ where $[D]$ is identified with the vertex set of T_d. For a valid colouring $\chi : V(T_d) \to [m]$ we assign a monomial $\text{Mon}(\chi) := \prod_{v \in V(T_d)} x_{v\chi(v)}$. We now define the polynomial $P_{m,d}$ as follows:

$$P_{m,d} = \sum_{\text{valid colourings } \chi} \text{Mon}(\chi)$$

This is a polynomial in $m \cdot (2^{d+1} - 1)$ of degree $2^{d+1} - 1$. Eventually, we shall choose $d = O(\log m)$ so this is a polynomial in $\operatorname{poly}(m)$ variables and degree $\operatorname{poly}(m)$.

Upper bound

Lemma 17.2. *The polynomial $P_{m,d}$ can be computed by a monotone algebraic circuit of size* $\operatorname{poly}(m, d)$.

Exercise 17.1 *Prove this lemma.*

17.1.1 Some intuition for the lower bound

What is left to show is that this polynomial cannot be computed by polynomial sized monotone ABPs. If you solve the above exercise, you would see that the polynomial $P_{m,d}$ can be computed via a *monotone* expression of the form

$$P_{m,d} = \sum_{i=1}^{\operatorname{poly}(m,d)} g_i h_i$$

where $\deg g_i = 2^d$ and $\deg h_i = 2^d - 1$ for all i. The property that we shall exploit is that if $P_{m,d}$ is computable by a monotone ABP, then for every $0 \leq k \leq 2^{d+1} - 1$, there must be a monotone expression computing $P_{m,d}$ of the form

$$P_{m,d} = \sum_{i=1}^{s} g_i h_i$$

where $\deg(g_i) = k$ and $\deg(h_i) = 2^{d+1} - 1 - k$. We need to find the right value of k that allows us to force s to be large (and clearly $k = 2^d$ or $2^d - 1$ are bad choices!). Hrubeš and Yehudayoff [HY16]'s cool idea was to look at the *isoperimetric profile* of the tree and choose k accordingly.

17.2 Isoperimetric profiles

The following definitions can be made for general graphs but we will restrict ourselves to the tree T_d as only that would be relevant here. For a parameter $k \leq |V(T_d)|$, we shall define

EIP(k) as

$$\mathrm{EIP}(k) := \min_{\substack{S \subseteq V(T_d)\\ |S|=k}} \left|E(S,\overline{S})\right|.$$

That is, EIP(k) is the size of the smallest cut induced by a subset of exactly k vertices.

Note that if $k = 2^\ell - 1$, then we can choose the subset S to be a subtree of depth ℓ and the size of the cut would just be 1. So there are several values of k for which EIP(k) is really small. But the point is that there are many other values of k for which EIP(k) is reasonably large.

Lemma 17.3. *Let k be the d-bit integer with binary representation* $1010\cdots 10$*. For this k, we have* $\mathrm{EIP}(k) \geq d/4$*.*

This is not a hard proof but is certainly believable. Let us just assume this and proceed with their proof (the interested reader can see a proof of this in the paper of Hrubeš and Yehudayoff [HY16]). From this point onwards, whenever we use the variable k, we mean this d-bit integer with binary representation $1010\cdots 10$.

17.3 Lower bound

Suppose that $P_{m,d}$ can be computed by a monotone ABP of size s. Then $P_{m,d}$ must have a monotone computation of the form

$$P_{m,d} = \sum_{i=1}^{s} g_i h_i$$

that satisfies the following constraints:

Degree constraints For each i we have $\deg g_i = k$ and $\deg h_i = D - k$,

Monotonicity If m_1 and m_2 are monomials in g_i and h_i respectively, then the product $m_1 m_2$ must be a monomial present in $P_{m,d}$.

Let's just focus on one summand $g \cdot h$. All the monomials in $P_{m,d}$ are of the form $\prod_{v \in V(T_d)} x_{v,c_v}$. This in particular means that each vertex v is "represented" exactly once in any monomial. Therefore, each vertex $v \in V(T_d)$ must appear in exactly one of g or h. This naturally partitions $V(T_d) = V_0 \sqcup V_1$ where g only involves variables corresponding to V_0 and h only involves varaibles corresponding to V_1.

The polynomial g (and similarly h) can now be thought of as a collection of *partial colourings* of just the vertices in V_0. Indeed, if $\prod_{v \in V_0} x_{v,c_v}$ is a monomial in g, then this corresponds to the partial colouring $\chi_0 : V_0 \to [m]$ that assigns colour c_v to v. Furthermore, if χ_0 is any partial colouring in g and χ_1 is any partial colouring in h, it must also be the case that $\chi = \chi_0 \cup \chi_1 : V(T_d) \to [m]$ is a valid colouring. The following lemma states that this more or less forces g and h to be quite sparse.

Lemma 17.4. *Let $V(T_d) = V_0 \sqcup V_1$ with $|V_0| = k$. Let $\mathcal{C}_g$ be a collection of colourings from V_0 to $[m]$ and $\mathcal{C}_h$ be a collection of colourings from V_1 to $[m]$. Then,*

$$|\mathcal{C}_g| \cdot |\mathcal{C}_h| \leq m^{-|E(V_0,V_1)|/4} \cdot (\#\text{ valid colourings of } V(T_d)).$$

Proof. Given a partition $V(T_d) = V_0 \sqcup V_1$, we shall call a non-leaf vertex $u \in V_i$ (where $i \in \{0,1\}$) a *pure node* if there is a leaf ℓ in the subtree rooted at u such that the entire path from u to ℓ besides u consists of vertices in V_{1-i}. That is, if $u \to u_1 \to \cdots \to u_r = \ell$ is the path, then $\{u, u_1, u_2, \ldots, u_r\} \cap V_i = \{u\}$. For a pure node u, there may be many leaves in the subtree rooted at u that is a witness to u being pure; we'll assign one such leaf ℓ_u arbitrarily and call them *pure leaves.* Let $\tilde{P}$ be the set of pure leaves in T_d with respect to the partition $V_0 \sqcup V_1$.

Subclaim 17.5. $\left|\tilde{P}\right| \geq E(V_0, V_1)/4.$

The proof of this is not too hard (it follows from basic graph theory, contraction of edges etc.) and we leave it as an exercise. We proceed with completing the proof of the lemma assuming the claim. We may assume that $\mathcal{C}_g, \mathcal{C}_h$ are both non-empty (for otherwise the lemma is vacuously true). We claim that any $\chi_0 : V_0 \to [m]$ in $\mathcal{C}_g$ is completely determined by its values on $(\text{Leaves}(T_d) \cap V_0) \setminus \tilde{P}$ and similarly any $\chi_1 : V_1 \to [m]$ in $\mathcal{C}_h$ is completely determined by its values on $(\text{Leaves}(T_d) \cap V_1) \setminus \tilde{P}$.

To see this fix some $\chi_1 \in \mathcal{C}_h$ and suppose χ_0 and χ_0' are two different colourings of V_0 that agree on $(\text{Leaves}(T_d) \cap V_0) \setminus \tilde{P}$. If $\chi_0 \neq \chi_0'$, then they must disagree on a leaf in V_0, and this leaf has to be a *pure leaf* as we already know that χ_0 and χ_0' agree on other leaves of V_0. Let this be $\ell_u \in \tilde{P}$ where u is a pure node in V_1. We will assume that this u is minimal in the sense that for any other pure node $u' \in V_1$ in the subtree rooted at u, we have $\chi_0(\ell_{u'}) = \chi_0'(\ell_{u'})$. Therefore, among the leaves in the subtree rooted at u, the colourings $\chi_0 \cup \chi_1$ and $\chi_0' \cup \chi_1$ agree on all leaves except ℓ_u. But then $\chi_0 \cup \chi_1$ and $\chi_0' \cup \chi_1$ cannot both be valid colourings. Therefore any colour in $\mathcal{C}_g$ is completely determined by the colours of

$(\text{Leaves}(T_d) \cap V_0) \setminus \tilde{P}$. Hence,

$$\begin{aligned} |\mathcal{C}_g| \cdot |\mathcal{C}_h| &\leq m^{\left|(\text{Leaves}(T_d) \cap V_0) \setminus \tilde{P}\right|} \cdot m^{\left|(\text{Leaves}(T_d) \cap V_1) \setminus \tilde{P}\right|} \\ &= m^{|\text{Leaves}(T_d)| - \left|\tilde{P}\right|} \\ &\leq m^{2^d} / m^{|E(V_0, V_1)|/4}. \end{aligned}$$

□

Exercise 17.2 *Prove Subclaim 17.5.*

Putting it all together

We will chose parameters now. Let $d = O(\log m)$ so that $P_{m,d}$ is a polynomial on $m \cdot D = \text{poly}(m)$ variables and degree $D = \text{poly}(m)$. From Lemma 17.2, we know that this can be computed by a poly(m)-sized monotone algebraic circuit.

Recall that for our chosen k, Lemma 17.3 tell us that $E(V_0, V_1) \geq d/4 = \Omega(\log m)$. If $P_{m,d}$ can be computed by a size s monotone ABP, then $P_{m,d} = \sum_{i=1}^{s} g_i h_i$ where this expression is *monotone* and $\deg(g_i) = k$ and $\deg(h_i) = D - k$. Lemma 17.4 states that the degree and monotonicity constraints force the sparsity of $g_i \cdot h_i$ to be at most $m^{2^d} / m^{d/16}$. Since $P_{m,d}$ is a polynomial with m^{2^d} monomials, we are forced to have $s \geq m^{d/16} = m^{\Omega(\log m)}$ which then completes the proof of Theorem 17.1. □

Chapter 18

Separation between monotone VP and VNP

We have seen that monotone computational models are rather restrictive and we have seen lower bounds that use just sparsity as the complexity measure (with a clever monomial counting making use of the restrictive nature of computatation). We have seen exponential lower bounds against monotone circuits (Theorem 8.6) and also a quasipolynomial separation between monotone ABPs and monotone circuits (Theorem 17.1). Here, we shall see a beautiful result of Yehudayoff [Yeh18] that separates VP and VNP in the monotone world.

The definition of the class "monotone VP" is clear. The following is an analogous definition of the class monotone VNP (or mon-VP). Throughout this chapter, we will restrict ourselves to polynomials over $\mathbb{R}$. Over a characteristic zero field such as $\mathbb{R}$, we will work with the *standard* definition of monotone models which is just that there are no negative constants.

Definition 18.1 (Monotone VNP). *A family of polynomial $\{f_n(\mathbf{x})\} \in \mathbb{R}[\mathbf{x}]$ is said to be in monotone* VNP *(or* mon-VNP*) if there is a family of polynomials $\{g_n(\mathbf{x},\mathbf{y})\} \in$ mon-VP with $|\mathbf{y}| = \operatorname{poly}|\mathbf{x}|$ such that*

$$\Diamond \qquad f(\mathbf{x}) = \sum_{\mathbf{b}\in\{0,1\}^{|\mathbf{y}|}} g(\mathbf{x},\mathbf{b}).$$

We are now ready to state the main theorem.

Theorem 18.2 ([Yeh18]). *There is an explicit polynomial family $\{f_n\} \in$ mon-VNP such that any mon-VP circuit computing it must have size $2^{\Omega(n/\log n)}$.*

The importance of coefficients

We already know lower bounds against monotone circuits. Can we not find a small modification such that the lower bound is using a polynomial that is in mon-VNP? The issue with this is that all the techniques for monotone lower bounds we have seen so far *only uses the structure of the set of monomials* and *not* the coefficients. Any technique that uses just the underlying set of monomials to prove a lower bound against mon-VP also necessarily yields a lower bound against mon-VNP.

Exercise 18.1 *Let $\Gamma : \mathbb{F}[\mathbf{x}] \to \mathbb{R}$ be a complexity measure such that*

- *For two polynomials f, g with the same set of monomials, $\Gamma(f) = \Gamma(g)$,*
- *For any $f \in \mathsf{VP}$, we have $\Gamma(f)$ is "small".*

Show that $\Gamma(h)$ is small for every $h \in$ mon-VNP as well.

Therefore, it is imperative that any separation between mon-VP and mon-VNP must necessarily use the coefficients of the underlying polynomials.

The polynomial

$$\begin{aligned} P(\mathbf{x}) &:= \frac{1}{2^n} \sum_{\mathbf{b} \in \{0,1\}^n} \prod_{i=1}^{n} \sum_{j=1}^{n} b_j x_{ij} \\ &= \sum_{\sigma : [n] \to [n]} \frac{1}{2^{|\mathrm{Range}(\sigma)|}} \cdot x_{1\sigma(1)} \cdots x_{n\sigma(n)} \end{aligned}$$

Lemma 18.3. *The polynomial $P \in$ mon-VNP.*

Proof. Follows just from the definition. □

Before we proceed towards describing the complexity measure, a bit of notation will help.

Some notation

The polynomial $P(\mathbf{x})$ is set-multilinear with respect to rows, and any monotone model computing this must also be set-multilinear. Therefore, all intermediate computations must involve monomials that picks at most 1 variable from each row.

For a set-multilinear monomial α, let $\mathrm{Row}(\alpha)$ denote the set of rows that it picks up variables from and similarly define $\mathrm{Col}(\alpha)$ analogously.

We shall say a polynomial $f(\mathbf{x})$ is *row-aligned* if $\mathrm{Row}(\alpha)$ is the same for every $\alpha \in f$. In such a case, define $\mathrm{Row}(f) = \mathrm{Row}(\alpha)$ for an arbitrary $\alpha \in f$.

18.1 Structural weakness of monotone circuits

We have already seen the standard structural weakness for monotone circuits that was used in both Section 8.2 and Chapter 17. We will use this specifically for a circuit computing a set-multilinear polynomial such as P.

Lemma 18.4. *Let C be a monotone circuit of size s computing the polynomial $P(\mathbf{x})$ defined above. Then, $P(\mathbf{x})$ can be written as*

$$P(\mathbf{x}) = \sum_{i=1}^{s} f_i(\mathbf{x}) \cdot g_i(\mathbf{x}) \tag{18.5}$$

such that

- *For each i, we have that f_i and g_i are row-aligned with $\mathrm{Row}(f_i) \sqcup \mathrm{Row}(g_i) = [n]$. Also, $\frac{n}{3} \leq \mathrm{Row}(f_i), \mathrm{Row}(g_i) \leq \frac{2n}{3}$.*
- *f_i's and g_i's contain only non-negative coefficients,*
- *for any i, and any $\alpha \in f_i$ and $\beta \in g_i$, we have*

  $$f_i[\alpha] \cdot g_i[\beta] \leq P[\alpha \cdot \beta]$$

  where by $h[\alpha]$ we denote the coefficient of the monomial α in h.

The proof of this is exactly along the same lines as we have seen earlier and we skip it. We can now describe the complexity measure.

We shall choose a parameter $\delta = O\left(\frac{1}{\log n}\right)$. Let $\Pi : \mathbb{R}[\mathbf{x}] \to \mathbb{R}[\mathbf{x}]$ be the map that projects down to only monomials that touch exactly δn columns. That is,

$$\Pi(h) = \sum_{\substack{\alpha \in h \\ |\mathrm{Col}(\alpha)| = \delta n}} h[\alpha] \cdot \alpha.$$

The complexity measure: Define $\Gamma : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ as

$$\begin{aligned}\Gamma(h(\mathbf{x})) &:= |\Pi(h)|_1 \\ &= \sum_{\substack{\alpha \in h \\ |\mathrm{Col}(\alpha)| = \delta n}} h[\alpha],\end{aligned}$$

the sum of the coefficients of monomials that touch exactly δn columns.

Given the above complexity measure, it will be useful to also have one more notation. Let $\mathcal{F}_{n,k}$ denote the set of onto (surjective) functions from $[n]$ to $[k]$, and let $F_{n,k}$ be the size of this set. We will be mostly interested in $k = \delta n$.

Proposition 18.6. *For $\delta = O\left(\frac{1}{\log n}\right)$, we have that*

$$\frac{1}{2} \cdot (\delta n)^n \leq F_{n,\delta n} \leq (\delta n)^n.$$

Sketch of Proof. The upper bound is clear. As for the lower bound, notice that for δ so small, a random function from $[n]$ to $[\delta n]$ would be onto with probability at least 1/2. □

Lemma 18.7. *(Upper bound for the polynomial) For the polynomial defined above,*

$$\Gamma(P) = \binom{n}{\delta n} \cdot 2^{-\delta n} \cdot F_{n,k}.$$

□

18.2 Upper bounding measure on building blocks

The strategy is the usual plan of bounding the measure for each building block in (18.5). The following is the main technical lemma.

Lemma 18.8 (Main technical lemma). *Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be row-aligned polynomials such that*

- $\mathrm{Row}(f) \sqcup \mathrm{Row}(g) = [n]$,
- $\frac{n}{3} \leq \mathrm{Row}(f), \mathrm{Row}(g) \leq \frac{2n}{3}$,
- *For each $\alpha \in f$ and $\beta \in g$, we have $f[\alpha] \cdot g[\beta] \leq P(\alpha \cdot \beta)$.*

Then,

$$\Gamma(f \cdot g) \leq 2^{-\Omega(\delta n)} \cdot \Gamma(P).$$

Clearly Theorem 18.2 follows readily from the above lemma. The rest of this chapter will be spent on the proof of this lemma. To make it a little easier to follow, we denote certain terms in red to indicate that the primary gain is coming from that term. Let $\varepsilon = 1/20$. We will call a monomial α to be "thin" if $|\mathrm{Col}(\alpha)| \leq (1-\varepsilon)\delta n$, and "wide" otherwise. We shall split $\Gamma(f \cdot g)$ into three sums.

$$\begin{aligned}
\Gamma(f \cdot g) := &\sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S}} f[\alpha] \cdot g[\beta] \\
\leq &\sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha \text{ thin}}} f[\alpha] \cdot g[\beta] && (T_1, \text{ involving thin } \alpha\text{'s}) \\
&+ \sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \beta \text{ thin}}} f[\alpha] \cdot g[\beta] && (T_2, \text{ involving thin } \beta\text{'s}) \\
&+ \sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ wide}}} f[\alpha] \cdot g[\beta]. && (T_3, \text{ the rest})
\end{aligned}$$

We will address the three terms seperately and show that each of them are substantially smaller than $\Gamma(P)$. It is easier to handle the thin terms first, as this follows just from the fact that the number of such terms is small.

Claim 18.9. $T_1 + T_2 \leq 2^{-\Omega(n)} \cdot \Gamma(P)$.

Proof. The main point is that there are not-too-many ways of splitting a monomial touching δn columns into two parts where one is thin. Fix a set S and let us estimate T_1 (the term T_2 is identical). The number of different ways of choosing α and β that cover columns S with α being thin is at most

$$\begin{aligned}
&\binom{\delta n}{< (1-\varepsilon)\delta n} \cdot ((1-\varepsilon)\delta n)^{|\mathrm{Row}(f)|} \cdot (\delta n)^{|\mathrm{Row}(g)|} \\
\leq\ &\binom{\delta n}{< (1-\varepsilon)\delta n} \cdot (1-\varepsilon)^{n/3} \cdot (\delta n)^n \\
\leq\ &2^{-\Omega(n)} \cdot (\delta n)^n
\end{aligned}$$

$$\leq\ 2^{-\Omega(n)} \cdot F_{n,\delta n}. \qquad \text{(Proposition 18.6)}$$

Summing over all such S, and using the fact that $f[\alpha] \cdot g[\beta] < P[\alpha \cdot \beta] = 2^{-\delta n}$, we have

$$\begin{aligned} T_1 + T_2 &\leq \binom{n}{\delta n} \cdot 2^{-\Omega(n)} \cdot F_{n,\delta n} \cdot 2^{-\delta n} \\ &\leq 2^{-\Omega(n)} \cdot \Gamma(P) \qquad \text{(Lemma 18.7)}. \end{aligned}$$

□

We are now left with the tricky part of estimating T_3. The rough intuition is the following. If we pick two "typical" monomials α and β, neither of which are thin, then $\alpha \cdot \beta$ would touch about $2(1-\varepsilon)\delta n$ columns as they would likely be disjoint. The coefficients of such monomials in P is just $2^{-2(1-\varepsilon)\delta n}$. Since we must ensure that $f[\alpha] \cdot g[\beta] = P[\alpha \cdot \beta] = 2^{-2(1-\varepsilon)\delta n}$ for all such α, β, both f and g must *typically* assign no more than $2^{-(1-\varepsilon)\delta n}$ for $f[\alpha], g[\beta]$. But in that case, when we pick *typical* α, β that touch the same set of δn columns, then $f[\alpha] \cdot g[\beta]$ would be about $2^{-2(1-\varepsilon)\delta n}$ though $P[\alpha \cdot \beta] = 2^{-(1-\varepsilon)\delta n}$ which is much larger.

On the other hand, if $f[\alpha]$ is much large for a specific α, then we are forced to have $g[\beta]$ quite small for any β that touches columns disjoint from α. Eventually we wish to "cover" the coefficient of monomials of P that touch δn columns as much as possible. Hence f and g have conflicting requirements on what should be assigned to its coefficients, and we want to exploit this.

Without loss of generality, we may assume (by scaling f and g by scalars if required) that

$$\max_{\substack{\alpha \in f \\ \delta n \geq |\mathrm{Col}(\alpha)| \geq (1-\varepsilon)\delta n}} f[\alpha] \quad = \quad \max_{\substack{\beta \in g \\ \delta n \geq |\mathrm{Col}(\beta)| \geq (1-\varepsilon)\delta n}} g[\beta] \quad =: M.$$

Case 1: $M \leq 2^{-(1-\varepsilon)\delta n}$. In this case, we can plug this in our bounds directly.

$$\begin{aligned} T_3 = \sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha, \beta \\ \mathrm{Col}(\alpha\cdot\beta) = S \\ \alpha, \beta \text{ wide}}} f[\alpha] \cdot g[\beta] \quad &\leq \quad \binom{n}{\delta n} \cdot (\delta n)^n \cdot 2^{-2(1-\varepsilon)\delta n} \\ \leq \binom{n}{\delta n} \cdot \frac{1}{2} \cdot F_{n,\delta n} \cdot 2^{-2(1-\varepsilon)\delta n} & \qquad \text{(Proposition 18.6)} \\ \leq \binom{n}{\delta n} \cdot 2^{-\delta n} \cdot F_{n,\delta n} \cdot 2^{-\Omega(\delta n)} & \end{aligned}$$

$$\leq 2^{-\Omega(\delta n)} \cdot \Gamma(P_1) \qquad \text{(Lemma 18.7)}.$$

Case 2: $M > 2^{-(1-\varepsilon)\delta n}$**.** Let $\alpha_0 \in f$ and $\beta_0 \in g$ be such that

$$f[\alpha_0], g[\beta_0] \geq 2^{-(1-\varepsilon)\delta n}.$$

Note that $\mathrm{Col}(\alpha_0 \cdot \beta_0) \leq 2\delta n$ and hence there are many columns that are untouched by α_0 or β_0. Therefore, if we pick a random δn sized set S at random, we should expect to have an intersection of at most $(2\delta)|S|$ with $\mathrm{Col}(\alpha_0\beta_0)$. The following is a straightforward application of the Chernoff's bound.

Claim 18.10.

$$\Pr_{S \in \binom{[n]}{\delta n}} \left[|S \cap \mathrm{Col}(\alpha_0\beta_0)| > (2\delta + \varepsilon)|S| \right] \leq 2^{-\Omega(\delta n)}.$$

We will call a set S to be *typical* if $|S \cap \mathrm{Col}(\alpha_0\beta_0)| \leq (2\delta+\varepsilon)\delta n$, and *atypical* otherwise. The above claim quantifies the fact that there aren't too many *atypical* sets. Then,

$$\begin{aligned}
T_3 &= \sum_{S \in \binom{[n]}{\delta n}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} f[\alpha]\cdot g[\beta] \\
&= \sum_{\substack{S \in \binom{[n]}{\delta n} \\ \text{typical}}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} f[\alpha]\cdot g[\beta] \quad + \sum_{\substack{S \in \binom{[n]}{\delta n} \\ \text{atypical}}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} f[\alpha]\cdot g[\beta] \\
&\leq \sum_{\substack{S \in \binom{[n]}{\delta n} \\ \text{typical}}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} f[\alpha]\cdot g[\beta] \quad + \sum_{\substack{S \in \binom{[n]}{\delta n} \\ \text{atypical}}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} P[\alpha\cdot\beta] \\
&\leq \sum_{\substack{S \in \binom{[n]}{\delta n} \\ \text{typical}}} \sum_{\substack{\alpha,\beta \\ \mathrm{Col}(\alpha\cdot\beta)=S \\ \alpha,\beta \text{ not thin}}} f[\alpha]\cdot g[\beta] \quad + \quad 2^{-\Omega(\delta n)}\cdot\Gamma(P).
\end{aligned}$$

Note that, for any typical S, we have that $\mathrm{Col}(\alpha_0)$ intersects S at at most $(2\delta+\varepsilon)\delta n$. Therefore, $\mathrm{Col}(\alpha_0)$ has at least $(1-\varepsilon-(2\delta-\varepsilon))\delta n$ columns that are not in $\mathrm{Col}(\beta)$.

$$g[\beta] \leq \frac{P[\alpha_0\beta]}{f[\alpha_0]} \leq 2^{-(1-2\varepsilon-2\delta)\delta n} \leq 2^{-\frac{2\delta n}{3}},$$

and a similar bound applies for $f[\alpha]$. Therefore,

$$\begin{aligned} T_3 &\le \sum_{\substack{S\in\binom{[n]}{\delta n}\\ \text{typical}}} \sum_{\substack{\alpha,\beta\\ \mathrm{Col}(\alpha\cdot\beta)=S\\ \alpha,\beta \text{ not thin}}} 2^{-\frac{\delta n}{3}}\cdot 2^{-\delta n} \quad + \quad 2^{-\Omega(\delta n)}\Gamma(P) \\ &\le 2^{-\frac{\delta n}{3}}\cdot\Gamma(P) \quad + \quad 2^{-\Omega(\delta n)}\cdot\Gamma(P) = 2^{-\Omega(\delta n)}\Gamma(P). \end{aligned}$$

This completes the bound for **Case 2**, and thus concludes the proof of Lemma 18.7 and hence Theorem 18.2.

Part VI

Lower bounds for depth four circuits: Shifted partial derivatives

Chapter 19

Lower bounds for depth-4 circuits with bounded bottom fan-in

This chapter shall address a recent technique for proving lower bounds for some depth-4 circuits.

Definition 19.1. *A* depth-4 circuit*, also referred to as a* $\Sigma\Pi\Sigma\Pi$ *circuit, computes a polynomial of the form*

$$f \quad = \quad Q_{11}\dots Q_{1d} \quad + \cdots + \quad Q_{s1}\dots Q_{sd}.$$

The number of summands s is called the top fan-in *of the circuit.*

Further, a $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit is a depth-4 circuit computing a polynomial of the form

$$f \quad = \quad Q_{11}\dots Q_{1a} \quad + \cdots + \quad Q_{s1}\dots Q_{sa} \quad \textit{where } \deg Q_{ij} \leq b \textit{ for all } i,j.$$

◊

19.1 Significance of the model

In a surprising series of results on depth reduction, Agrawal and Vinay [AV08] and subsequent strengthenings of Koiran [Koi12] and Tavenas [Tav15] showed that depth-4 circuits more or less capture the complexity of general circuits.

Theorem 19.2 ([AV08, Koi12, Tav15]). *If f is an n variate degree-d polynomial computed by a size s arithmetic circuit, then f can also be computed by a $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ circuit of* *size* $\exp\left(O(\sqrt{d}\log s)\right)$.

Conversely, if an n-variate degree-d polynomial requires $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ *circuits of size* $\exp\left(\Omega(\sqrt{d}\log s)\right)$, *then it requires arbitrary depth arithmetic circuits of size* $n^{\Omega(\log s/\log n)}$ *to compute it.*

Thus proving strong enough lower bounds for this special case of depth-4 circuits imply lower bounds for general circuits. The main results of the section is some recent lower bound [GKKS14, KSS14, FLMS15] that comes very close to the required threshold.

19.2 Building the complexity measure

As a simpler task, let us first attempt to prove lower bounds for expressions of the form

$$f \quad = \quad Q_1^d \quad + \cdots + \quad Q_s^d$$

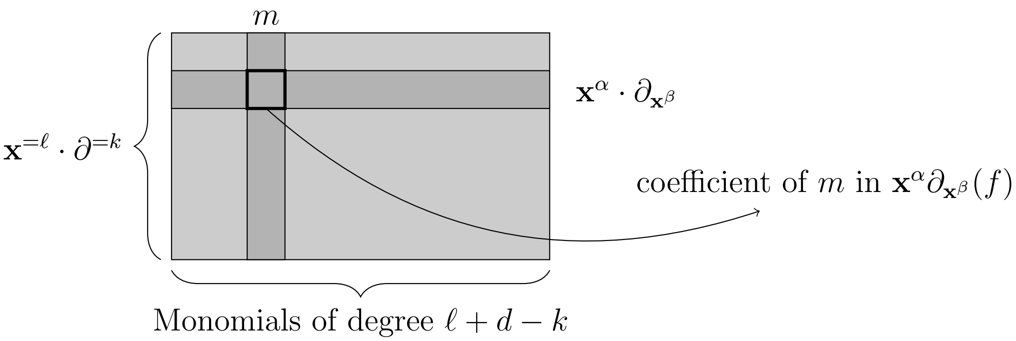
where each of the Q_i's are quadratics. This is exactly the problem studied by Kayal [Kay12], which led to the complexity measure for proving depth-4 lower bounds.

The goal is to construct a measure Γ such that $\Gamma(f)$ is small whenever f is a power of a quadratic. As a first attempt, let us look at the space of k-th order partial derivatives of Q^d (for a suitable choice of k). Unlike the case of $\Sigma\wedge\Sigma$-circuits where the the space of k-th order partial derivatives of ℓ^d had dimension 1, the space of partial derivatives of Q^d could be as large as it can be expected. Nevertheless, the following simple observation would provide the key intuition.

Observation 19.3. *Any k-th order partial derivative of Q^d is of the form $Q^{d-k}p$ where p is a polynomial of degree at most k. Hence, if $k \ll d$, then all k-th order partial derivatives of Q^d share large common factors.*

This suggests that instead of looking at linear combinations of the partial derivatives of Q^d, we should instead be analysing *low-degree polynomial combinations* of them.

Definition 19.4. *Let $\partial^{=k}(f)$ refer to the set of all k-th order partial derivatives of f, and $\mathbf{x}^{\leq \ell}$ refer to the set of all monomials of degree at most ℓ. The* shifted partials of f, *denoted by $\left\langle \partial^{=k}(f) \right\rangle_{\leq \ell}$, is the vector space spanned by $\left\{\mathbf{x}^{\leq \ell}\cdot\partial^{=k}(f)\right\}$. The dimension of this space shall be denoted by $\Gamma_{k,\ell}^{[\text{Kay}]}(f)$.* ◊


m
$\mathbf{x}^\alpha \cdot \partial_{\mathbf{x}^\beta}$
$\mathbf{x}^{=\ell} \cdot \partial^{=k}$
coefficient of m in $\mathbf{x}^\alpha \partial_{\mathbf{x}^\beta}(f)$
Monomials of degree $\ell + d - k$

The above observation shows that any element of $\left\langle \partial^{=k}\left(Q^d\right)\right\rangle_{\leq \ell}$ is divisible by Q^{d-k} and we thereby have the following lemma.

Lemma 19.5. *If $f = Q^d$ where Q is a quadratic, then $\Gamma_{k,\ell}^{[\mathrm{Kay}]}(f) \leq \binom{n+k+\ell}{n}$, the number of monomials of degree $(k+\ell)$.*

Note that if f was instead a random polynomial, we would expect the measure $\dim\left(\left\langle \partial^{=k}(f)\right\rangle_{\leq \ell}\right)$ to be about $\binom{n+k}{n} \cdot \binom{n+\ell}{n}$, which is *much* larger than $\binom{n+k+\ell}{n}$ for suitable choice of k, ℓ. Hence this measure $\Gamma_{k,\ell}^{[\mathrm{Kay}]}$ is certainly potentially useful for this model. Very similar to the above lemma, one can also show the following upper bound for the *building blocks* of $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuits.

Lemma 19.6. *Let $f = Q_1 \ldots Q_a$ with $\deg Q_i \leq b$ for all i. Then,*

$$\Gamma_{k,\ell}^{[\mathrm{Kay}]}(f) \quad = \quad \dim\left(\left\langle \partial^{=k}(f)\right\rangle_{\leq \ell}\right) \quad \leq \quad \binom{a}{k}\binom{n+(b-1)k+\ell}{n}.$$

It is easy to check that $\Gamma_{k,\ell}^{[\mathrm{Kay}]}$ is a sub-additive measure, and we immediately have this corollary.

Corollary 19.7. *Let f be an n-variate polynomial computed by a $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit of top fan-in s. Then,*

$$\Gamma_{k,\ell}^{[\mathrm{Kay}]}(f) \quad \leq \quad s \cdot \binom{a}{k}\binom{n+(b-1)k+\ell}{n}.$$

Or in other words for any choice of k, ℓ, we have that any $\Sigma\Pi^{[a]}\Sigma\Pi^{[b]}$ circuit computing a polynomial f must have top fan-in s at least

$$\frac{\Gamma_{k,\ell}^{[\mathrm{Kay}]}(f)}{\binom{a}{k}\binom{n+(b-1)k+\ell}{n}}.$$

Intuition from algebraic geometry

Another perspective for the shifted partial derivatives comes from algebraic geometry. Any zero $a \in \mathbb{F}^n$ of Q is a zero of *multiplicity* d of Q^d. This implies that the set of common zeroes of all k-th order partial derivatives of Q^d (for $k \approx \sqrt{d}$) is *large*. On the other hand if f is a random polynomial, then with high probability there are no roots of large multiplicity.

In algebraic geometry terminology, the common zeroes of a set of polynomials is called the *variety* of the ideal generated by them. Further there is also a well-defined notion of a *dimension of a variety* which measures how large a variety is. Let $\mathbb{F}[\mathbf{x}]_{\leq r}$ refer to the set of polynomials of degree at most r, and let $\gamma_I(r) = \dim\left(I \cap \mathbb{F}[\mathbf{x}]_{\leq r}\right)$. Intuitively, if $\gamma_I(r)$ is large, then there are *many constraints* and hence the variety is *small*. In other words the growth of $\gamma_I(r)$ is inversely related to the dimension of the variety of I, and this is precisely captured by what is known as the *Affine Hilbert function of I*. More about the precise definitions of the Affine Hilbert function etc. can be found in any standard text in algebraic geometry such as [CLO07].

In our setting, the ideal we are interested in is $I = \left\langle \partial^{=k} f \right\rangle$. If f is a homogeneous polynomial, then $I \cap \mathbb{F}[\mathbf{x}]_{\leq r} = \left\langle \partial^{=k}(f) \right\rangle_{\leq \ell}$ where $\ell = r - (\deg(f) - k)$. Hence studying the dimension of shifted partial derivatives is exactly studying $\gamma_I(r)$ which holds all information about the dimension of the variety.

19.3 Lower bounding shifted partials of explicit polynomials

For a random polynomial $R(\mathbf{x})$, we would expect that

$$\Gamma_{k,\ell}^{[\mathrm{Kay}]}(R) \quad \approx \quad \min\left\{ \binom{n+\ell+d-k}{n}, \binom{n+k}{n}\binom{n+\ell}{n} \right\}.$$

The terms on the RHS correspond to trivial upper bounds, where first term is the total number of monomials of degree $(\ell + d - k)$ and the second term is the total number shifted partials.

Claim 19.8. *For $k = \varepsilon\sqrt{d}$ for a small enough $\varepsilon > 0$, and $\ell = \frac{cn\sqrt{d}}{\log n}$ for a large enough* *constant c, we have*

$$\frac{\min\left\{\binom{n+\ell+d-k}{n}, \binom{n+k}{n}\binom{n+\ell}{n}\right\}}{\binom{O(\sqrt{d})}{k}\binom{n+(\sqrt{d}-1)k+\ell}{n}} \quad = \quad 2^{\Omega(\sqrt{d}\log n)}.$$

The proof of this claim is easily obtained by using standard asymptotic estimates of binomial coefficients. Note that using Corollary 19.7, the above claim shows that if we can find an explicit polynomial whose dimension of shifted partials are as large as above, then we would have an $\exp(\Omega(\sqrt{d}\log n))$ lower bound for the top fan-in of $\Sigma\Pi^{[\sqrt{d}]}\Sigma\Pi^{[\sqrt{d}]}$ circuits computing this polynomial.

If we have a set of polynomials with distinct leading monomials, then they are clearly linearly independent. Hence one way of lower bounding the dimension of a space of polynomials is to find a sufficiently large set of polynomials with distinct monomials in the space. The vector space of polynomials we are interested is $\left\langle \partial^{=k}(f)\right\rangle_{\leq \ell}$, and if we choose a structured polynomial f we can hope to be able to estimate the number of distinct leading monomials in this vector space.

19.3.1 Shifted partials of the determinant and permanent

The first lower bound for $\Sigma\Pi^{[\sqrt{d}]}\Sigma\Pi^{[\sqrt{d}]}$ circuits was by Gupta, Kamath, Kayal and Saptharishi [GKKS14] for the determinant and the permanent polynomial. We shall describe the lower bound for Det_n, although it would carry over immediately to Perm_n as well. As mentioned earlier, we wish to estimate the number of distinct leading monomials in $\left\langle \partial^{=k}(\mathsf{Det}_n)\right\rangle_{\leq \ell} = \operatorname{span}\left\{\mathbf{x}^{\leq \ell}\partial^{=k}\mathsf{Det}_n\right\}$. [GKKS14] made a relaxation to merely count the number of distinct leading monomials among the generators $\left\{\mathbf{x}^{\leq \ell}\partial^{=k}\mathsf{Det}_n\right\}$ instead of their span.

The first observation is that any k-th order partial derivative of Det_n is just an $(n-k)\times(n-k)$ minor. Let us fix a monomial ordering induced by the lexicographic ordering on the variables:

$$x_{11} \succ x_{12} \cdots \succ x_{1n} \succ x_{21} \succ \cdots \succ x_{nn}.$$

Under this ordering, the leading monomial of any minor is just the product of variables on the main diagonal of the sub-matrix corresponding to the minor, and hence is a term of the form $x_{i_1 j_1} \ldots x_{i_{(n-k)}, j_{(n-k)}}$ where $i_1 < \cdots < i_{n-k}$ and $j_1 < \cdots < j_{n-k}$; let us call such a sequence of indices as an $(n-k)$-increasing sequences in $[n] \times [n]$. Further, for any $(n-k)$-increasing sequence, there is a unique minor M whose leading monomial is precisely the product of the variables indexed by the increasing sequence. Therefore, the task of lower bounding distinct leading monomials in $\left\{\mathbf{x}^{\leq \ell} \partial^{=k} \mathsf{Det}_n\right\}$ reduces to the following combinatorial problem.

Claim 19.9. *For any $k, \ell > 0$, we have*

$$\Gamma_{k,\ell}^{[\mathrm{Kay}]}(\mathsf{Det}_n) \quad \geq \quad \# \left\{ \begin{array}{c} \textit{monomials of degree } (\ell + n - k) \textit{ that} \\ \textit{contain an } (n-k)\textit{-increasing sequence} \end{array} \right\}.$$

We could start with an $(n-k)$-increasing sequence, and multiply by a monomial of degree ℓ to obtain a monomial containing an increasing sequence. Of course, the issue is that this process is not invertible and hence we might overcount. To fix this issue, [GKKS14] assign a *canonical increasing sequence* to every monomial that contains an increasing sequence and multiply by monomials of degree ℓ that do not change the canonical increasing sequence.

Definition 19.10. *Let $D_2 = \{x_{1,1}, \ldots, x_{n,n}, x_{1,2}, x_{2,3}, \ldots, x_{n-1,n}\}$, the main diagonal and the diagonal just above it. For any monomial m define the* canonical increasing sequence *of m, denoted by $\chi(m)$, as $(n-k)$-increasing sequence of m that is entirely contained in D_2 and is ordered highest according to the ordering '$\succ$'. If m contains no $(n-k)$-increasing sequence entirely in D_2, then we shall say the canonical increasing sequence is empty.* ◊

The reason we restrict ourselves to D_2 is because it is easier to understand which monomials change the canonical increasing sequence and which monomials do not.

Lemma 19.11. *Let S be an $(n-k)$-increasing sequence completely contained in D_2, and let m_S be the monomial obtained by multiplying the variables indexed by S. There are at least $(2(n-k)-1)$ variables in D_2 such that if m is any monomial over these variables, then $\chi(m_S) = \chi(m \cdot m_S)$.*

Proof. Note that for any $x_{i,j} \in D_2$ other than $x_{n,n}$, exactly one of $x_{i+1,j}$ or $x_{i,j+1}$ is in D_2 as well; let us refer to this element in D_2 as the *companion* of $x_{i,j}$. It is straightforward to check that for any $(n-k)$-increasing sequence S, the elements of S and their companions do not alter the canonical increasing sequence. □

It is a simple exercise to check that the number of $(n-k)$-increasing sequences contained in D_2 is $\binom{n+k}{2k}$. Further, as we are free to use the $n^2 - 2n + 1$ variables outside D_2, and the $2(n-k) - 1$ variables that don't alter the canonical increasing sequence, we have the following lemma.

Lemma 19.12. *For any $k, \ell \geq 0$,*

$$\dim\left(\left\langle \partial^{=k}\left(\mathsf{Det}_n\right)\right\rangle_{\leq \ell}\right) \quad \geq \quad \binom{n+k}{2k}\binom{(n^2-2n+1)+2(n-k)-1+\ell}{\ell}.$$

Although this lower bound is not as large as expected for a random polynomial, this is still sufficient to give strong lower bounds for depth-4 circuits. By choosing $k = \varepsilon\sqrt{n}$ for a small enough $\varepsilon > 0$, and $\ell = n^2\sqrt{n}$, Lemma 19.12 with Corollary 19.7 yields the lower bound of Gupta, Kamath, Kayal and Saptharishi [GKKS14]

Theorem 19.13. *Any $\Sigma\Pi^{[O(\sqrt{n})]}\Sigma\Pi^{[\sqrt{n}]}$ circuit computing Det_n or Perm_n has top fanin $2^{\Omega(\sqrt{n})}$.* □

It is worth noting that although Claim 19.8 suggests that we should be able to obtain a lower bound of $\exp(\Omega(\sqrt{n}\log n))$ for Det_n, [GKKS14] also showed that the above estimate for the dimension of shifted partial derivatives for the determinant is fairly tight. Hence the dimension of shifted partials cannot give a stronger lower bound for the determinant polynomial. However, it is possible that the estimate is *not* tight for the permanent and the dimension of shifted partial derivatives of the permanent is provably strictly larger than that of the determinant! It is conceivable that one should be able to prove an $\exp(\Omega(\sqrt{n}\log n))$ lower bound for the permanent using this measure.

Indeed, subsequently an $\exp(\Omega(\sqrt{d}\log n))$ was proved [KSS14, FLMS15] for other explicit polynomials which we now outline.

19.3.2 Shifted partials of the Nisan-Wigderson polynomial

Very shortly after [GKKS14]'s $2^{\Omega(\sqrt{n})}$ lower bound, Kayal, Saha and Saptharishi [KSS14] gave a stronger lower bound for a different polynomial. Their approach was to engineer an explicit polynomial F for which the dimension of shifted partial derivatives is easier to estimate. The main idea was that, if any k-th order partial derivative of the engineered polynomial is a monomial, then once again estimating $\dim\left(\left\langle \partial^{=k}(F)\right\rangle_{\leq \ell}\right)$ reduces to a monomial counting problem. If we could ensure that no two monomials of F have a gcd of degree k or more, then we would immediately get that all k-th order partial derivatives of F are just monomials (albeit possibly zero). If we were to interpret the set of non-zero monomials of F as just subsets over the variables, then the above constraint can be rephrased as a set system with *small pairwise intersection*. Such systems are well studied and are known as Nisan-Wigderson designs [NW94]. With this in mind, [KSS14] studied the following polynomial family inspired by an explicit construction of a Nisan-Wigderson design. The definition below is a specialization of Definition 2.4.

Definition 19.14 (Nisan-Wigderson Polynomial)**.** *. Let n be a power of 2 and let $\mathbb{F}_n$ be the finite field with n elements that are identified with the set $\{1, \ldots, n\}$. For any $0 \leq k \leq n$, the polynomial NW_k is a n^2-variate polynomial of degree n defined as follows:*

$$\mathrm{NW}_k(x_{1,1}, \ldots, x_{n,n}) \quad = \quad \sum_{\substack{p(t) \in \mathbb{F}_n[t] \\ \deg(p) < k}} x_{1,p(1)} \ldots x_{n,p(n)}.$$

$\Diamond$

We recall the main observation about the set of monomials of the Nisan-Wigderson polynomial which is the *low pairwise-intersection* property.

Observation 19.15. *Any two monomials of NW_k intersect in less than k variables. Hence, any k-th order partial derivative of $\mathrm{NW}_k(\mathbf{x})$ is a monomial (which could possibly be zero).* □

Hence, the problem of lower bounding the shifted partials of NW_k reduces to the problem of counting distinct monomials of degree $\ell + d - k$ that are divisible by one of these k-th order derivatives. [KSS14] additionally used the observation that two random k-th order partial derivatives of NW_k are monomials that are *far* from each other. Using this, they estimate the number of distinct shifts of these monomials and showed that the dimension of shifted partial derivatives of NW_k is very close to the trivial upper bound as in Claim 19.8. We sketch the argument by Chillara and Mukhopadhyay [CM14]. Formally, for any two multilinear monomials m_1 and m_2, let the $\Delta(m_1, m_2)$ denote $\min\{|m_1| - |m_1 \cap m_2|, m_2 - |m_1 \cap m_2|\}$ (abusing notation by identifying the multilinear monomials with the set of variables that divide it).

Lemma 19.16 ([CM14])**.** *Let $m_1, \ldots, m_s$ be monomials over N variables such that $\Delta(m_i, m_j) \geq d$ for all $i \neq j$. Then the number of distinct monomials that may be obtained by multiplying some m_i by arbitrary monomials of degree ℓ is at least $s\binom{N+\ell}{N} - \binom{s}{2}\binom{N+\ell-d}{N}$.*

Proof. For $i = 1, \ldots, s$, let A_i be the set of monomials that can be obtained by multiplying m_i with a degree ℓ monomial. By inclusion-exclusion,

$$\left|\bigcup_{i=1}^{s} A_i\right| \quad \geq \quad \sum_{i=1}^{s} |A_i| \quad - \quad \sum_{i<j} |A_i \cap A_j|\,.$$

Note that each A_i is of size exactly $\binom{N+\ell}{N}$. Further, since $\Delta(m_i, m_j) \geq d$, any monomial that is divisible by m_i and m_j must necessarily be divisible by m_i and the variables in m_j not in m_i. Hence, $|A_i \cap A_j| \leq \binom{N+\ell-d}{N}$. The lemma follows by substituting these above. □

Note that any two distinct monomials of NW_k intersect in at most k places. For each monomial m_i of NW_k, let m_i' be any non-zero k-th order partial derivative of m_i. Therefore, $\Delta(m_i', m_j') \geq n - 2k \geq \frac{n}{2}$ for $k = \varepsilon\sqrt{n}$. Since we have n^k monomials of pairwise distance at least $n/2$, the above lemma immediately yields a lower bound for the shifted partials of NW_k.

Theorem 19.17 ([KSS14]). *Let $k = \varepsilon\sqrt{d}$ for some constant $\varepsilon > 0$. Then for any $\ell = \Theta\left(\frac{n^2\sqrt{n}}{\log n}\right)$,*

$$\dim\left(\left\langle \partial^{=k}\left(\mathrm{NW}_k\right)\right\rangle_{\leq \ell}\right) \quad \geq \quad \frac{n^k}{2} \cdot \binom{n^2+\ell}{n^2}$$

Sketch of Proof. As mentioned earlier, we have n^k monomials $\{m_i'\}$ with pairwise distance at least $\frac{n}{2}$. Using Lemma 19.16, it suffices to show that

$$n^k \cdot \binom{n^2+\ell}{n^2} \quad \geq \quad 2 \cdot \binom{n^k}{2} \cdot \binom{n^2+\ell-\frac{n}{2}}{n^2}$$

and this follows easily from standard binomial coefficient estimates. □

Combining with Corollary 19.7, we have the lower bound of [KSS14] using standard estimates.

Theorem 19.18 ([KSS14]). *Any $\Sigma\Pi^{[O(\sqrt{n})]}\Sigma\Pi^{[\sqrt{n}]}$ computing the* NW_k *polynomial, where $k = \varepsilon\sqrt{n}$ for a sufficiently small $\varepsilon > 0$, must have top fan-in* $\exp(\Omega(\sqrt{n}\log n))$. □

[KSS14] used the above lower bound to give an $n^{\Omega(\log n)}$ lower bound for a subclass of formulas called *regular formulas*. The interested reader can refer to [KSS14] for more details.

19.3.3 Shifted partials of the Iterated-matrix-multiplication polynomial

Fourier, Limaye, Malod and Srinivasan [FLMS15] showed the same lower bound as [KSS14] but for the *iterated matrix multiplication* polynomial which is known to have polynomial sized circuits computing it. Let's recall the definition from Chapter 2

Definition (Iterated matrix multiplication polynomial)**.** *Let $M_1, \ldots, M_d$ be $n \times n$ matrices with distinct variables as entries, i.e. $M_k = \left(\!\left(x_{ij}^{(k)}\right)\!\right)_{i,j \leq n}$ for $k = 1, \ldots, d$. The polynomial $\mathrm{IMM}_{n,d}$ is a $(n^2 d)$-variate degree-d polynomial defined as the $(1,1)$-th entry of the matrix product $M_1 \ldots M_d$:*

$$\mathrm{IMM}_{n,d}(\mathbf{x}) \quad = \quad (M_1 \ldots M_d)_{1,1}\,.$$

$\Diamond$

A more useful perspective is to interpret this as a *canonical algebraic branching program*, which we recall again.

Definition (Algebraic branching program)**.** *An algebraic branching program (ABP) comprises of a layered directed graph G with $(d+1)$ layers of vertices, where the first and last layer consists of a single node (called source and sink respectively), all other layers consist of n vertices, and edges are only between successive layers and have linear polynomials as edge-weights. The ABP is set to compute the polynomial f defined as*

$$f(\mathbf{x}) \quad = \quad \sum_{\textit{source-sink path } \rho} \mathrm{weight}(\rho)$$

where the weight *of any path is just the product of the edge weights on the path.* $\Diamond$

The canonical ABP comprises of the graph where the i-th vertex of layer $(\ell - 1)$ is connected to the j-th vertex of layer ℓ with edge-weight $x_{ij}^{(\ell)}$ for every choice of i, j and ℓ. It is easy to see that the polynomial computed by the canonical ABP is in fact $\mathrm{IMM}_{n,d}$.

To lower bound the dimension of shifted partial derivatives of $\mathrm{IMM}_{n,d}$, firstly note that a derivative with respect to any variable (or edge) simply results in the sum of all source-sink paths that *pass* through this edge. [FLMS15] use the following simple but crucial observation to assist in bounding the dimension of shifted partials.

Observation 19.19. *Assume that d is even. Let $e_1, e_3, \ldots, e_{d-1}$ be an arbitrary set of edges such that e_i is between layer i and $i+1$. Then, there is a unique path from source to sink that passes through all these edges.*

Proof. Since these are edges in alternate layers, their starting and ending points uniquely determine the edges that are picked up from the even-numbered layers to complete the source-sink path. □

Since we are interested in k-th order derivatives for $k \approx \varepsilon\sqrt{d}$, [FLMS15] consider the following restriction by removing some edges from the underlying graph:

- Select $(2k-1)$ layers $\ell_1, \ldots, \ell_{2k-1}$ that are roughly equally spaced between the first and the last layer. These layers, and the first and the last layers, shall be untouched and shall be called *pristine layers*.

- In all the other layers, retain only those edges connecting vertex i of this layer to vertex i of the next.

This restriction effectively makes the graph similar to an ABP with $2k+1$ layers. Let the polynomial computed by the restricted ABP be $\mathrm{IMM}'_{n,d}(\mathbf{x})$. Since $\mathrm{IMM}'_{n,d}$ was obtained by just setting some variables of $\mathrm{IMM}_{n,d}$ to zero, the dimension of shifted partial derivatives of $\mathrm{IMM}'_{n,d}$ can only be smaller than that of $\mathrm{IMM}_{n,d}$. Similar to Observation 19.19, we have the following observation.

Observation 19.20. *For every choice of k edges from odd-numbered pristine layers, there is a unique source-sink path that passes through them.*

In other words, for any choice of k variables chosen by picking one from each odd-numbered pristine layer, then the k-th order partial derivative of $\mathrm{IMM}'_{n,d}$ with respect to these k variables is a non-zero monomial.

Once again, we can lower bound the dimension of shifted partial derivatives of $\mathrm{IMM}'_{n,d}$ by a monomial counting problem. Similar to the earlier case, [FLMS15] show that the monomials thus obtained are *far* from one another. We state their main lemma below without proof.

Lemma 19.21 ([FLMS15])**.** *There are at least $n^{k/2}$ monomials of $\mathrm{IMM}'_{n,d}$ of pairwise distance at least $\frac{n}{4}$.*

Again, using Lemma 19.16 and standard binomial coefficient estimates, this implies that the shifted partial derivatives of $\mathrm{IMM}'_{n,d}$ is almost as large as the trivial upper bound.

Theorem 19.22 ([FLMS15])**.** *Let $k = \varepsilon\sqrt{d}$ for a sufficiently small $\varepsilon > 0$ and ℓ be an integer such that $n^{1/16} \leq \frac{N+\ell}{\ell} \leq n^{1/4}$ where N is the number of variables $\mathrm{IMM}'_{n,d}$ depends on. Then,*

$$\begin{aligned}\dim\left(\left\langle \partial^{=k}\left(\mathrm{IMM}_{n,d}\right)\right\rangle_{\leq \ell}\right) &\geq \dim\left(\left\langle \partial^{=k}\left(\mathrm{IMM}'_{n,d}\right)\right\rangle_{\leq \ell}\right) \\ &= \Omega\left(n^{k/2} \cdot \binom{N+\ell}{\ell}\right).\end{aligned}$$

□

Combining with Corollary 19.7, we get the lower bound of [FLMS15].

Theorem 19.23 ([FLMS15])**.** *Any* $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ *circuit computing* $\mathrm{IMM}_{n,d}$*, with* $d \leq n^{\delta}$ *for a sufficiently small* $\delta > 0$*, has top fan-in* $\exp(\Omega(\sqrt{d}\log n))$. □

Similar to [KSS14], the above result also implies $n^{\Omega(\log n)}$ lower bounds for regular formulas computing $\mathrm{IMM}_{n,d}$.

19.4 A bottom fan-in hierarchy theorem

Subsequent to these results, Kumar and Saraf [KS14b] showed finer separations even within the class of depth-4 circuits with small bottom fan-in. We state the result below without proof.

Theorem 19.24 ([KS14b])**.** *For any* t *satisfying* $\log n \ll t \leq \frac{d}{40}$*, we can construct* n*-variate degree* d *polynomials that are computable by homogeneous* $\Sigma\Pi\Sigma\Pi^{[t]}$ *such that any homogeneous* $\Sigma\Pi\Sigma\Pi^{[t/20]}$ *circuit computing it requires size* $n^{\Omega(d/t)}$.

Chapter 20

Lower bounds for homogeneous depth four circuits

The model for which we shall be interested in proving lower bounds are homogenous depth four circuits. These circuits compute polynomials of the form

$$f \quad = \quad \sum_i Q_{i1} \dots Q_{ia_i}$$

where each Q_{ij} is a homogeneous polynomial. This immediately forces that $\sum_{j=1}^{a_i} \deg(Q_{ij}) = \deg(f)$ for all i.

Goal. Find an explicit polynomial f (of degree d, and over n variables) such that any homogeneous depth four circuit requires size $n^{\Omega(\sqrt{d})}$. That is, if

$$f \quad = \quad \sum_i Q_{i1} \dots Q_{ia_i}$$

for homogeneous polynomials Q_{ij}'s, then the total number of monomials present among the Q_{ij}'s must be $n^{\Omega(\sqrt{d})}$.

Intuition towards the measure - (1)

Consider an expression of the form

$$C \quad = \quad \sum_{i=1}^{s} Q_{i1} \dots Q_{ia_i}$$

We shall call a summand $Q_{i1} \dots Q_{ia_i}$ *good* if the degree of each $Q_{ij} \leq \sqrt{d}$. Let us split the above sum into *good* terms and the rest.

$$\begin{aligned} C_1 &= \sum_{i=1}^{s_1} Q_{i1} \dots Q_{ia_i} \quad \text{where} \quad \deg(Q_{ij}) \leq \sqrt{d} \quad \text{for all } i, j & (20.1) \\ C_2 &= \sum_{i=s_1+1}^{s} Q_{i1} \dots Q_{ia_i} \quad \text{where} \quad \deg(Q_{i1}) > \sqrt{d} \quad \text{for all } i > s_1 & (20.2) \end{aligned}$$

If one were to just prove a lower bound for (20.1), then using the dimension of shifted partial derivatives we can obtain a lower bound of $n^{\Omega(\sqrt{d})}$. Hence let us focus on an expression of the form (20.2) and see if we can come up with a measure that gives a $n^{\Omega(\sqrt{d})}$ lower bound there as well.

Starting with (2), let us expand each Q_{i1} as a sum of monomials to obtain an expression of the form

$$C_2 \quad = \quad \sum_{i=1}^{s'} m_i \cdot Q'_i$$

where each m_i is a monomial of degree greater than $\sqrt{d}$, and Q'_i some polynomial of degree $d - \deg(m_i)$. The number of summands s' would be at most the size of the circuit we started out with.

Key Idea: Suppose the polynomial C_2 was multilinear, i.e. the degree in each variable is bounded by 1. Further, say $s' \leq n^{\sqrt{d}/10}$. Apply a random restriction ρ on the variables by setting each variable independently to zero with probability $p < \frac{1}{n^{1/20}}$.

If m was any monomial that was divisible by $\sqrt{d}$ disjoint variables, then $\rho(m) \neq 0$ with probability at most $\frac{1}{n^{\sqrt{d}/20}}$. Hence, the probability that $\rho(m_i) \neq 0$ for some $i \leq s'$ that is divisible by $\sqrt{d}$ variables is at most $\frac{1}{n^{\sqrt{d}/10}}$. Hence, the only terms that would survive on the RHS are terms of the form $\rho(m_i \cdot Q'_i)$ where m_i is divisible by at most $\sqrt{d}$ distinct variables. But recall that $\deg(m_i) > \sqrt{d}$ and this implies that m_i is non-multilinear. If that is the case, then every monomial on the RHS is non-multilinear! Thus as long as $\rho(C_2) \neq 0$, there would be at least one multilinear monomial that survives. This would contradict our original assumption that $s' \leq n^{\sqrt{d}/10}$, giving us the lower bound we were after.

Thus, the measure for the sum of *good* terms is the dimension of shifted partial derivatives. The measure for the sum of non-*good* terms was *the number of non-zero multilinear monomials after a random restriction.* Hopefully some combination of these measures would give us a measure for their sum.[1]

Intuition towards the measure - (2)

The idea of using random restrictions as defined above essentially kills all monomials that are divisible by 'too many' variables. Let us consider an extreme case where every monomials in each Q_{ij} is just a power of a single variable. We shall first try to prove a lower bound for expression of the form

$$C \quad = \quad \sum_i Q_{i1} \cdots Q_{ia_i}$$

where every monomial in any Q_{ij} is a power of a single variable, i.e. each Q_{ij} is a sum of univariate polynomials.

Define the operator MultiQuad that acts on any polynomial Q such that MultiQuad(Q) is just the sum of monomials of Q of degree at most 2 in every variable. Then,

$$\begin{aligned} C &= \sum_i \text{MultiQuad}(Q_{i1}) \cdots \text{MultiQuad}(Q_{ia_i}) \quad + \quad \text{other terms} \\ &= \quad C_1 \quad + \quad C_2 \end{aligned}$$

Notice that C_1 corresponds to a $\Sigma\Pi^{[d/2]}\Sigma\Pi^{[2]}$ circuit since we assume that each Q_{ij} is a sum of univariates. The dimension of shifted partial derivatives would yield a lower bound for such $\Sigma\Pi^{[d/2]}\Sigma\Pi^{[2]}$ circuits. But what really happens to C_2 as we take some partial derivative?

Key Observation. For any multilinear monomial m, the partial derivative $\partial_m(C_2)$ only consists of non-multilinear monomials.

Thus, this points towards the following modification of the traditional dimension of shifted partial derivatives:

> For any polynomial P, look at the set of polynomials of obtained as $m_1 \cdot \partial_{m_2}(P)$ where m_1 and m_2 are *multilinear monomials* of a certain degree, and compute the dimension of the *multilinear component* of these polynomials i.e. erase all mono-

[1] There are some instances when this strategy can fail spectacularly. See [KS14b]

mials that are non-multilinear and then compute the dimension of the residual polynomials.

This basically allows us to completely ignore the contribution of C_2 as we have that multilinear component of $m_1 \partial_{m_2}(C_2))$ is zero for every m_1 and m_2 that are multilinear.

Both these point us to a modification of the shifted partials, which [KLSS17, KS14c] refer to as *projected shifted partial derivatives.*

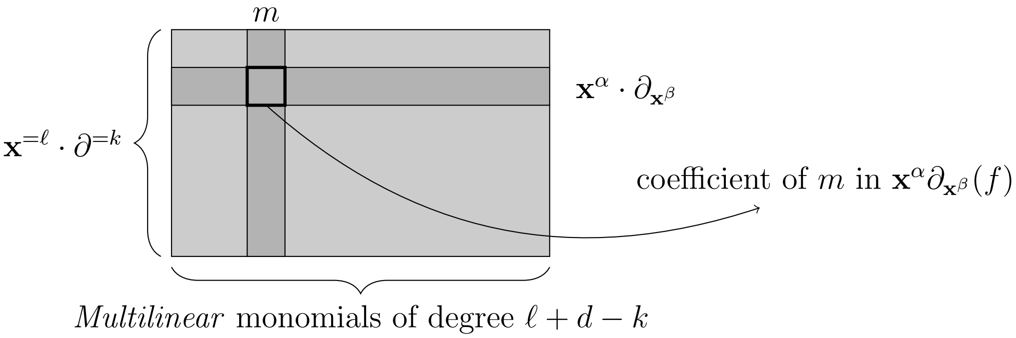
Definition 20.3 (Projected Shifted Partial Derivatives)**.** *Fix parameters $k, \ell > 0$. For any polynomial P, the set of projected shifted partials of P, denoted by $\mathrm{PSD}_{k,\ell}(P)$ is defined as follows*

$$\mathrm{PSD}_{k,\ell}(P) \quad = \quad \left\{ \mathrm{mult}(m_1 \cdot \partial_{m_2}(P)) \ : \ \begin{array}{c} \deg(m_1) = \ell \ , \ \deg(m_2) = k, \\ m_1 \text{ and } m_2 \text{ are multilinear} \end{array} \right\}$$

where $\mathrm{mult}(f)$ *refers to the polynomial f projected to only the multilinear monomials of f.*

The measure $\Gamma^{\mathrm{PSD}}_{k,\ell}(P)$ is defined as the dimension of the above set of polynomials, i.e.

$$\Diamond \qquad \Gamma^{\mathrm{PSD}}_{k,\ell}(P) \quad = \quad \dim\left(\mathrm{span}(\mathrm{PSD}_{k,\ell})\right)$$


m
$\mathbf{x}^{\alpha} \cdot \partial_{\mathbf{x}^{\beta}}$
$\mathbf{x}^{=\ell} \cdot \partial^{=k}$
coefficient of m in $\mathbf{x}^{\alpha}\partial_{\mathbf{x}^{\beta}}(f)$
Multilinear monomials of degree $\ell + d - k$

The works of [KLSS17, KS14c] use this measure to prove a lower bound for "*low-support* depth 4 circuits". As sketched earlier, the task of proving lower bounds for general homogeneous depth 4 circuits can be reduced to the *low-support* depth 4 circuits via random restrictions.

20.1 Reducing to 'low-support' depth 4 circuits

We have already seen a sketch of how this can be done via a random restriction but let us formalize this as a lemma.

Lemma 20.4. *Let P be an n-variate degree d polynomial computed by a homogeneous depth 4 circuit C of size $s \leq n^{c\sqrt{d}}$, for some $c > 0$. Let ρ be a random restriction that sets each variable to zero independently with probability $1 - 1/n^{2c}$. Then with probability at least $(1 - 1/s)$, the polynomial $\rho(P)$ is computed by a homogeneous depth 4 circuit C' with bottom support at most $\sqrt{d}$ and size at most s.*

Proof. Let $\{m_1, \ldots, m_r\}$ be the set of all monomials computed at the lowest layer of the depth 4 circuit C that are divisible by more than $\sqrt{d}$ distinct variables. Since the size of C is at most s, we also have that $r \leq s$. Then,

$$\begin{aligned} \forall i \in [r] \qquad \Pr[\rho(m_i) \neq 0] &\leq \frac{1}{n^{2c\sqrt{d}}} \\ \implies \qquad \Pr[\exists i \ : \ \rho(m_i) \neq 0] &\leq \frac{r}{n^{2c\sqrt{d}}} \leq \frac{1}{n^{c\sqrt{d}}} \leq \frac{1}{s} \end{aligned}$$

Thus, with probability at least $(1 - 1/s)$, all the large support monomials are killed and C reduces to a homogeneous depth 4 circuit of bottom support at most $\sqrt{d}$. □

20.1.1 Handling random restrictions

The previous section outlined how in essence, it would suffice to try and find an explicit polynomial for which we can prove a good enough lower bound for bounded bottom-support depth 4 circuits. Let us say that we have found an explicit polynomial g that requires depth 4 circuits of size at least $n^{\sqrt{d}/100}$. Are we done? Let us write things down formally to see exactly what we need.

Say the polynomial we wish to show requires large homogeneous depth 4 circuits is f. Let us assume on the contrary that f can be computed by homogeneous depth 4 circuits of size $s < n^{\sqrt{d}/10000}$. Then, by Lemma 20.4, $\rho(f)$ can be computed by a homogeneous depth 4 circuits of bottom support bounded by $\sqrt{d}/1000$ of size s. We want to be able to say that this is a contradiction. We might be able to say that if $\rho(f)$ has g as *a projection*, that is, but setting more variables to zero in $\rho(f)$ we obtain g.

Both the results of [KLSS17] and [KS14c] proceed by showing that the polynomial g, for which they show a lower bound for bounded bottom support circuits, is robust enough to yield the lower bound even after random restriction. The calculations become trickier because the calculations of $\Gamma_{k,\ell}^{[\mathrm{PSD}]}(\rho(f))$. However, in this survey we shall use an easier approach to generically lift any g to a different polynomial f such that $\rho(f)$ has g as a projection. This trick came up during discussions with Mrinal Kumar.

Lemma 20.5. *Let ρ be a random restriction that sets each variable to zero independently with probability $1-p$. For any polynomial $f(y_1, \ldots y_n)$, define $f \circ \mathrm{Lin}_p$ as*

$$f \circ \mathrm{Lin}_p \quad = \quad f\left(\sum_{i=1}^{t} y_{1i}, \ldots, \sum_{i=1}^{t} y_{nt}\right) \quad \text{where } t = \left(\frac{1}{p}\right) n \log n$$

Then, $\rho(f \circ \mathrm{Lin}_p)$ has f as a projection with probability $1 - 1/2^n$.

Proof. For any $i = 1, \ldots, n$

$$\begin{aligned} \Pr[\rho(y_{i1}) = \ldots \rho(y_{it}) = 0] &= (1-p)^t \\ &= \frac{1}{n \cdot 2^n} \\ \implies \Pr[\exists i \ : \ \rho(y_{i1}) = \ldots \rho(y_{it}) = 0] &\leq \frac{1}{2^n} \end{aligned}$$

Hence, with probability at least $1 - 1/2^n$, for each i there is some j such that $\rho(y_{ij}) \neq 0$. Therefore, with probability at least $1 - 1/2^n$, the polynomial f is a projection of $\rho(f \circ \mathrm{Lin}_p)$. □

In all the applications, as in Lemma 20.4, we would have $p = 1/n^{O(1)}$. Thus, we would only incur a polynomial blow-up in the number of variables from f to $f \circ \mathrm{Lin}_p$. Hence, we can focus on proving a lower bound a homogeneous depth 4 circuit of bottom support at most r (which would eventually be something like $\sqrt{d}/100$).

Lemma 20.6 ([KLSS17])**.** *Let P be an n-variate degree d polynomial computed by a homogeneous depth 4 circuit of size s and bottom-support at most r. Then for any k, ℓ such that $\ell + rk \leq n/2$,*

$$\Gamma_{k,\ell}^{\mathrm{PSD}}(P) \quad \leq \quad s \cdot \binom{\frac{2d}{r} + k}{k} \cdot \binom{n}{\ell + rk}.$$

The proof of this lemma is exactly along the description in of Intuition - (2): split the circuit into multiquadratic and non-multiquadratic part, and show that the non-multiquadratic part contributes no multilinear monomials. But to just put things in perspective, we shall be dealing with parameters $r = \sqrt{d}/100$, $k = \sqrt{d}$ and $\ell = \frac{n}{2}(1-\varepsilon)$ for $\varepsilon = O\left(\frac{\log d}{\sqrt{d}}\right)$. The above bound, by Lemma 4.7, can be seen to reduce to

$$\Gamma_{k,\ell}^{\mathrm{PSD}}(P) \quad \leq \quad s \cdot \binom{n}{\ell} \cdot (1+\varepsilon)^{2rk} \cdot 2^{O(\sqrt{d})}$$

Sanity checks

Let us first check if this measure can at least in principle yield a lower bound for us. The best way to do this is to get some heuristic estimate of what we expect the measure to be for a random n-variate degree d polynomial R.

Heuristic Estimate. For a random n-variate degree d polynomial R, we expect the $\Gamma_{k,\ell}^{\mathrm{PSD}}(R)$ to be as large as it can be, i.e.

$$\Gamma_{k,\ell}^{\mathrm{PSD}}(R) \quad \approx \quad \min\left(\binom{n}{k} \cdot \binom{n}{\ell}, \binom{n}{\ell+d-k}\right)$$

As a first step, one should first check that if we could indeed find a polynomial P for which the bound is as large as stated above, do we get a useful lower bound from Lemma 20.6? Turns out that if we were to choose our parameters carefully, we do indeed get the lower bound. Just to give a sense of how *careful* we need to be, here is some of the parameters that are chosen in [KLSS17, KS14c].

- The number of variables n is at least the cube of the degree d.
- The model we shall be working with is bottom-support r where $r = \sqrt{d}/1000$.
- The order of derivatives $k = \sqrt{d}$.
- The degree of the shift ℓ shall be chosen as $\ell = \frac{n}{2}(1-\varepsilon)$ where $\varepsilon = \frac{\log d}{c\sqrt{d}}$ for a suitable constant c.

The above choice of parameters might already seem pretty fragile but these are not the most delicate choices! While proving the lower bound on $\Gamma_{k,\ell}^{\mathrm{PSD}}$ for an explicit polynomial, the number of monomials etc. need to be tailored to perfection to make the proof work.

20.2 The surrogate rank approach of [KLSS17]

The goal is now to find an explicit polynomial P such that $\mathrm{PSD}_{k,\ell}(P)$ has large rank. One way to prove that a set of polynomials are linearly independent is to show that they have distinct leading monomials (as used [GKKS14] etc.) Another method is to show that these polynomials are *almost orthogonal*. An example of this phenomenon can be seen in the following fact.

Fact 20.7. *Let M be a square matrix such that the absolute value of the diagonal entry is larger than sum of the absolute values of the non-diagonal entries in that row or column, i.e. $|M_{ii}| \geq \sum_{j \neq i} |M_{ij}|$ for all i. Then the matrix M is full rank.*

Such matrices are also called *diagonally dominant matrices*, and captures the notion of *almost orthogonal* vectors alluded to earlier. For symmetric matrices M, the following bound of Alon [Alo09].

Lemma 20.8 ([Alo09])**.** *For any real symmetric matrix M,*

$$\mathrm{rank}(M) \quad \geq \quad \frac{(\mathrm{Tr}(M))^2}{\mathrm{Tr}(M^2)}$$

We'll see the proof of this shortly but it would shed some more intuition to see what the above lemma yields for a diagonally dominant matrix. Let M be a matrix of the form

$$M \quad = \quad \begin{bmatrix} D & d & \dots & d \\ d & D & \dots & d \\ \vdots & \vdots & \ddots & \vdots \\ d & d & \dots & D \end{bmatrix}_{r \times r}$$

Then, $\mathrm{Tr}(M) = D \cdot r$, and $\mathrm{Tr}(M^2) = (D^2 + (r-1)d^2)r = O(D^2 r + r^2 d^2)$. If $D > (r-1)d^2$, then $\mathrm{Tr}(M^2) = O(D^2 r)$. Thus, the above lemma gives that $\mathrm{rank}(M) = \Omega(r)$.

Proof. By the spectral theorem, any real symmetric matrix has a basis of eigen vectors with eigenvalues $\lambda_1, \dots, \lambda_n$ where n is the dimension of the matrix. If $\lambda_1, \dots, \lambda_r$ are the non-zero eigenvalues, then

$$\mathrm{Tr}(M) \quad = \quad \sum_{i=1}^{r} \lambda_i$$

$$
\begin{aligned}
& \leq \sqrt{r} \cdot \left(\sum_{i=1}^{r} \lambda_i^2\right) = \sqrt{r} \cdot \mathrm{Tr}(M^2) \\
\implies r & \geq \frac{(\mathrm{Tr}(M))^2}{\mathrm{Tr}(M^2)}
\end{aligned}
$$

□

The bound of [KLSS17] for an explicit polynomial P proceeds by considering the matrix B where each row is indexed by a pair of multilinear monomials (m_1, m_2) of degree k and ℓ respectively, and the row is just the coefficients of the monomials of $\mathrm{mult}(m_2 \partial_{m_1}(P))$ in a fixed order. Note that B is not even a square matrix, and certainly not symmetric. However, the matrix $M = BB^T$ is a symmetric square matrix such that $\mathrm{rank}(M) \leq \mathrm{rank}(B)$.

Let us spend some time understand the entries of M. The (i, j)-th entry of M is precisely the inner-product of row i and row j of B. If P is a polynomial with just zero-one coefficients, then the i-th diagonal entry is precisely the number of non-zero entries in row i of B. Thus,

$$
\begin{aligned}
\mathrm{Tr}(M) &= \text{number of non-zero entries in } B \\
&= (\#\text{ cols of } B) \cdot \mathop{\mathbb{E}}_{i}[\#\text{ non-zero entries in } i\text{-th col of } B]
\end{aligned}
$$

The calculation for $\mathrm{Tr}(M^2)$ requires a little more care. Let M_i refer to the i-th row of M and B_i refer to the i-th row of B. Then,

$$
\begin{aligned}
\mathrm{Tr}(M^2) &= \sum_i \langle M_i, M_i \rangle \\
&= \sum_i \sum_j \langle B_i, B_j \rangle^2 \quad = \quad \sum_i \sum_j \left(\sum_m B_{im} B_{jm}\right)^2 \\
&= \sum_i \sum_j \sum_m B_{im}^2 B_{jm}^2 \quad + \quad \sum_i \sum_j \sum_{m \neq m'} B_{im} B_{im'} B_{jm} B_{jm'} \\
&= \sum_m \left(\sum_i \sum_j B_{im} B_{jm}\right) \quad + \quad \sum_i \sum_j \sum_{m \neq m'} B_{im} B_{im'} B_{jm} B_{jm'} \\
&= \qquad T_1 \qquad\qquad + \qquad\qquad T_2
\end{aligned}
$$

The first term T_1 is easy to calculate:

$$
T_1 \quad = \quad (\#\text{ cols of } B) \cdot \mathop{\mathbb{E}}_{i}[(\#\text{ non-zero entries in } i\text{-th col of } B)^2]
$$

$$\overset{\text{(hopefully)}}{\approx} \quad (\#\text{ cols of } B) \cdot \mathop{\mathbb{E}}_{i}[(\#\text{ non-zero entries in } i\text{-th col of } B)]^2$$

The term T_2 roughly corresponds to the number of 2×2 submatrices of B that is $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. If we could somehow show that there are not too many such submatrices, then $\mathrm{Tr}(M^2)$ is essentially dominated by T_1. That would then yield that $\mathrm{rank}(M) \gtrapprox (\#\text{ cols of } B)$.

Obtaining a bound on T_2:

$$T_2 \quad = \quad \sum_i \sum_j \sum_{m \neq m'} B_{im} B_{im'} B_{jm} B_{jm'}$$

Each term $B_{im} B_{im'} B_{jm} B_{jm'}$ that is non-zero corresponds to a 2×2 submatrix of B (indexed by rows i, j and columns m, m') that is $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

The columns of B are indexed by multilinear monomials of degree $\ell + d - k$, and the rows of B are indexed by a derivative and a shift. Let row i correspond to $\mathrm{mult}(\gamma_1 \cdot \partial_{\alpha_1}(P))$ and row j to $\mathrm{mult}(\gamma_1 \cdot \partial_{\alpha_1}(P))$. Thus, if the 2×2 minor indexed by rows i, j and columns m, m' equals $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, then there exists $\beta_1, \beta_2, \beta_3, \beta_4 \in P$ such that

$$\begin{aligned} m \quad &= \quad \frac{\beta_1}{\alpha_1} \cdot \gamma_1 \quad = \quad \frac{\beta_3}{\alpha_2} \cdot \gamma_2 \\ m' \quad &= \quad \frac{\beta_2}{\alpha_1} \cdot \gamma_1 \quad = \quad \frac{\beta_4}{\alpha_2} \cdot \gamma_2 \\ &\implies \frac{\beta_1}{\beta_3} \quad = \quad \frac{\beta_2}{\beta_4} \end{aligned}$$

Following notation used in [KLSS17], we shall call $\beta_1, \beta_2, \beta_3, \beta_4$ as the *label* of the 2×2 minor. Since $m \neq m'$, we also have that $\beta_1 \neq \beta_2$. What we'd like to say that the only way $\beta_1/\beta_3 = \beta_2/\beta_4$ is if $\beta_3 = \beta_1$ and $\beta_2 = \beta_4$. This need not be true in general of course, but this is where the choice of the polynomial comes in.

Claim 20.9. *If P is the $\mathrm{NW}_{d,d^3,e}$ polynomial for $e = \frac{d}{3}$ then any 2×2 minor of B (with the order of derivatives $k = o(d)$) that is $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$ has label $\beta_1, \beta_2, \beta_3, \beta_4$ where $\beta_1 = \beta_3$ and $\beta_2 = \beta_4$, or $\beta_1 = \beta_2$ and $\beta_3 = \beta_4$.*

Proof. Assume that $\beta_1 \neq \beta_3$. Then by Lemma 2.5 we know that they differ in at least $2d/3$ places. But then, $\beta_1/\beta_3 = \beta_2/\beta_4$ forces that β_1 and β_3 must agree at least $2d/3$ places forcing $\beta_1 = \beta_2$. □

Thus, for the NW-polynomial the number of such boxes is quite small. Using this, albeit with a reasonable amount of sweat, one can estimate T_2 to show that $T_2 = O(T_1)$. Thus, [KLSS17] obtain the following bound.

Lemma 20.10 ([KLSS17])**.** *For the polynomial* $\mathrm{NW}_{d,d^3,e}$*, for* $e = \frac{d}{3}$*, and* $k = \sqrt{d}$ *and* $\ell = \frac{n}{2}\left(1 - \frac{\log d}{\sqrt{d}}\right)$ *we have the bound*

$$\Gamma^{\mathrm{PSD}}_{k,\ell}(\mathrm{NW}_{d,d^3,e}) \quad \geq \quad \frac{1}{\mathrm{poly}(n,d)} \cdot \min\left(\binom{n}{\ell + d - k}, \binom{d}{k}^2 \cdot d^k \cdot k! \cdot \binom{n}{\ell}\right)$$

Note that the first term of the min in the RHS is the number of columns of B, as we had heuristically estimated. Simplifying the RHS using Lemma 4.7, we get

$$\Gamma^{\mathrm{PSD}}_{k,\ell}(\mathrm{NW}_{d,d^3,e}) \quad \geq \quad \frac{1}{\mathrm{poly}(n,d)} \cdot \binom{n}{\ell} \cdot \exp\left(c \cdot \varepsilon(d-k)\right)$$

for some constant $c > 0$. Since $\varepsilon = \frac{\log d}{\sqrt{d}}$, we get

$$\Gamma^{\mathrm{PSD}}_{k,\ell}(\mathrm{NW}_{d,d^3,e}) \quad \geq \quad \frac{1}{\mathrm{poly}(n,d)} \cdot \binom{n}{\ell} \cdot \exp\left(c \cdot \sqrt{d} \cdot \log d\right)$$

With the above bound and Lemma 20.6, we get the lower bound of [KLSS17].

Theorem 20.11 ([KLSS17])**.** *Any depth* 4 *homogeneous circuit of bottom support* $r = \sqrt{d}/1000$ *computing the polynomial* $\mathrm{NW}_{d,d^3,d/3}$ *over a characteristic zero field must have top fan-in* $s = d^{\Omega(\sqrt{d})}$.

In fact, more generally, any homogeneous depth 4 *circuit of bottom support bounded by* r *computing* $\mathrm{NW}_{d,m,e}$ *for suitably chosen parameters must have top fanin* $s = d^{\Omega(d/r)}$.

Coupling with Lemma 20.5, we obtain (a slight reformulation of) their main theorem.

Theorem 20.12 ([KLSS17])**.** *Any depth* 4 *homogeneous computing the polynomial* $\mathrm{NW}_{d,d^3,d/3} \circ \mathrm{Lin}$ *over a characteristic zero field must have size* $s = d^{\Omega(\sqrt{d})}$.

20.3 The leading monomial approach of [KS14c]

Shortly after [KLSS17], a purely combinatorial proof of the result was presented by Kumar and Saraf [KS14c]. More over, they were able to prove the lower bound of $n^{\Omega(\sqrt{d})}$ for the size of any homogeneous depth 4 circuit computing $\mathrm{IMM}_{n,d}$ (for some suitable choices of n and d). This was a strengthening of [KLSS17] in two ways – (1) it worked over any field, and (2) the lower bound was for a polynomial that we know can be computed small arithmetic circuit.

The calculations of [KS14c] are much more trickier than [KLSS17] but there are quite a few interesting ideas that would even have application in other areas.

The earlier lower bounds of [GKKS14, KSS14, FLMS15] required a lower bound on the dimension of shifted partial derivatives of a polynomial P, and this was obtained by finding a *large* set of *distinct leading monomials*. In [KS14c], they take this approach but require a very careful analysis. The key difference in this setting is the following:

> If β is the leading monomial of a polynomial P, then for any monomial γ, we also have that $\beta \cdot \gamma$ is the leading monomial of γP.
>
> However, the leading monomial of $\mathrm{mult}(\gamma P)$ could be $\beta' \cdot \gamma$ for some $\beta' \neq \beta$ (as higher monomials could be made non-multilinear during the shift by γ).

The multilinear projection makes the task of counting leading monomials much harder and [KS14c] come up with a clever method to estimate this.

Leading monomials after multilinear projections

Let P the polynomial for which we are trying to lower bound $\Gamma^{\mathrm{PSD}}_{k,\ell}(P)$. For every monomial multilinear monomial α of degree k, and a monomial $\beta \in \partial_\alpha(P)$, define the set $A(\alpha,\beta)$ as

$$A(\alpha,\beta) \quad = \quad \left\{ \gamma \ : \ \begin{array}{c} \deg(\gamma) = \ell + d - k \text{ and there is a } \gamma' \text{ of degree } \ell \\ \text{such that } \gamma = \mathrm{LM}(\mathrm{mult}(\gamma' \cdot \partial_\alpha(P))) = \gamma' \cdot \beta \end{array} \right\}$$

In other words, we want the number of distinct monomials that are contributed by β, which are also distinct leading monomials obtained from $\partial_\alpha(P)$ that are divisible by β. We then have

$$\Gamma^{\mathrm{PSD}}_{k,\ell}(P) \quad \geq \quad \left|\bigcup_{\alpha,\beta} A(\alpha,\beta)\right|$$

Choice of derivatives: Instead of looking at all derivatives in $\partial^{=k}$, we shall restrict ourselves to just a subset of derivatives. Restricting the above union to a subset $\Delta \subset \mathbf{x}^{=k}$ still continues to remain a lower bound for $\Gamma^{\mathrm{PSD}}_{k,\ell}(P)$. Keeping in mind that we are dealing with $P = \mathrm{NW}_{d,m,e}$ we shall choose Δ to be a set of monomials of the form $x_{1a_1} \cdots x_{ka_k}$ with each $a_i \leq m$ so as to have m^k derivatives in total. This shall become relevant later.

$$\Gamma^{\mathrm{PSD}}_{k,\ell}(P) \quad \geq \quad \left|\bigcup_{\substack{\alpha\in\Delta \\ \beta\in\mathbf{x}^{=\ell}}} A(\alpha,\beta)\right| \tag{20.13}$$

The standard technique to obtain a lower bound on the union of sets is via the *Inclusion-Exclusion* principle.

Lemma 20.14 (Inclusion-Exclusion Principle)**.** *For any collection of sets* $A_1, \ldots, A_r$*,*

$$\left|\bigcup_i A_i\right| \quad \geq \quad \sum_i |A_i| \quad - \quad \sum_{i\neq j} |A_i \cap A_j|$$

If we were to somehow show that $\sum_{i\neq j} |A_i \cap A_j| \leq \frac{1}{2}\sum_i |A_i|$, then we obtain that $|\cup_i A_i| \geq \frac{1}{2}\cdot\sum_i |A_i|$. This is what shall be employed for the sets $A(\alpha,\beta)$, except that we quickly run into two immediate problems.

1. How do we even estimate $A(\alpha,\beta)$? The set of γ' such that $\gamma'\beta = \mathrm{LM}(\partial_\alpha(P))$ do not seem to have any nice combinatorial structure.

2. What if it so happens that $\sum |A(\alpha_1,\beta_1) \cap A(\alpha_2,\beta_2)| = 100 \sum |A(\alpha,\beta)|$? Inclusion-Exclusion does not yield anything in that case.

It so turns out that the second point actually is the case. In fact for $\mathrm{IMM}_{n,d}$, the second term turns out to be greater than the first term by a factor of $n^{\sqrt{d}/1000}$ or so! In [KS14c], they prove a wonderful strengthened version of the Inclusion-Exclusion principle which allows them to handle the second hurdle.

Lemma 20.15 (Stronger Inclusion-Exclusion [KS14c])**.** *Let $A_1, \ldots, A_r$ be sets such that there is some $\lambda > 1$ such that*

$$\sum_{i \neq j} |A_i \cap A_j| \quad \leq \quad \sum_i \lambda \cdot |A_i|$$

Then,

$$\left|\bigcup_i A_i\right| \quad \geq \quad \left(\frac{1}{4\lambda}\right) \cdot \left(\sum_i |A_i|\right)$$

In other words, as long as the second term of the Inclusion-Exclusion principle is *not too much larger* than the first term, we still can get non-trivial bounds on the union.

Proof. Let $p = \frac{1}{2\lambda} < 1$. Define sets $A'_1, \ldots, A'_r$ such that $A'_i \subseteq A_i$ obtained by adding each element of A_i to A'_i independently with probability p. Since $A'_i \subseteq A_i$, we also have that $|\cup A_i| \geq |\cup A'_i|$. By linearity of expectation,

$$\mathbb{E}\left[\sum_i |A'_i|\right] \quad = \quad p \sum_i |A_i|$$

More importantly, by the sampling process,

$$\mathbb{E}\left[\left|A'_i \cap A'_j\right|\right] \quad = \quad p^2 \cdot |A_i \cap A_j|$$

as any common element must be added to both A'_i *and* A'_j, and either of these events happen independently with probability p each. Since $\sum_{i,j} \left|A'_i \cap A'_j\right|$ drops by a factor of p^2, we are now in a position to apply the Lemma 20.14 to the A'_is.

$$\begin{aligned}
\left|\bigcup A_i\right| \quad &\geq \quad \mathbb{E}\left[\left|\bigcup A'_i\right|\right] \\
&\geq \quad \mathbb{E}\left[\sum_i |A'_i|\right] \quad - \quad \mathbb{E}\left[\left|A'_i \cap A'_j\right|\right] \\
&= \quad p\left(\sum_i |A_i|\right) \quad - \quad p^2\left(\sum_{i \neq j} |A_i \cap A_j|\right) \\
&\geq \quad p\left(\sum_i |A_i|\right) \quad - \quad p^2 \lambda\left(\sum_i |A_i|\right)
\end{aligned}$$

$$\geq \frac{p}{2}\left(\sum_i |A_i|\right) \quad = \quad \frac{1}{4\lambda}\left(\sum_i |A_i|\right)$$

□

Corollary 20.16. *Considers sets $A_1, \ldots, A_r$ and let $S_1 = \sum_i |A_i|$ and $S_2 = \sum_{i \neq j} |A_i \cap A_j|$. Then,*

$$\left|\bigcup A_i\right| \quad \geq \quad \frac{S_1}{4} \cdot \min\left(1, \frac{S_1}{S_2}\right)$$

We can now proceed to lower bound $|\bigcup A(\alpha, \beta)|$ via inclusion exclusion.

Estimating $|\bigcup A(\alpha, \beta)|$ via Inclusion-Exclusion

$$\left|\bigcup_{\alpha,\beta} A(\alpha,\beta)\right| \quad \geq \quad \sum_{\alpha,\beta} |A(\alpha,\beta)| \quad - \quad \sum_{(\alpha,\beta)\neq(\alpha',\beta')} |A(\alpha,\beta) \cap A(\alpha',\beta')|$$

Let us first address the term $\sum |A(\alpha, \beta)|$. As mentioned earlier, it is not an easy task to get a good handle on the set $A(\alpha, \beta)$ for polynomial such as NW or IMM, for any reasonable monomial ordering. However, [KS14c] circumvent this difficult by using an indirect approach to estimate this term.

For any derivative α and $\beta \in \partial_\alpha(P)$, define the set $S(\alpha, \beta)$ as the following set of multilinear monomials of degree ℓ that is disjoint from β.

$$S(\alpha,\beta) \quad = \quad \left\{\gamma \ : \begin{array}{c} \gamma \text{ is multilinear, has} \\ \text{degree } \ell \text{ and } \gcd(\beta,\gamma) = 1 \end{array}\right\}$$

This on the other hand is independent of any monomial ordering, and is also easy to calculate:

$$\text{For every } \alpha, \beta \qquad |S(\alpha,\beta)| \quad = \quad \binom{n-d+k}{\ell}.$$

Lemma 20.17 ([KS14c])**.** *For any α,*

$$\sum_\beta |A(\alpha,\beta)| \quad \geq \quad \left|\bigcup_\beta S(\alpha,\beta)\right|$$

Proof. Consider any $\gamma \in \bigcup_\beta S(\alpha,\beta)$. By definition, there is at least one non-multilinear monomial in $\gamma \cdot \partial_\alpha(P)$. Thus, in particular $\mathrm{LM}(\mathrm{mult}(\gamma \cdot \partial_\alpha(P))$ is non-zero and equal to some $\gamma \cdot \beta$ for some monomial $\beta \in \partial_\alpha(P)$. This also implies that $\gamma' = \gamma \cdot \beta \in A(\alpha,\beta)$. This yields an injective map ϕ

$$\phi : \bigcup_\beta S(\alpha,\beta) \quad \rightarrowtail \quad \{(\beta,\gamma') \ : \ \beta \in \partial_\alpha(P) \ , \ \gamma' \in A(\alpha,\beta)\}$$

Since the size of the RHS is precisely $\sum_\beta |A(\alpha,\beta)|$, the lemma follows. □

Thus, by another use of Inclusion-Exclusion on the $S(\alpha,\beta)$'s, we get

$$\begin{aligned}\left|\bigcup_{\alpha,\beta} A(\alpha,\beta)\right| \ &\geq \ \sum_{\alpha,\beta} |A(\alpha,\beta)| \quad - \sum_{(\alpha,\beta)\neq(\alpha',\beta')} |A(\alpha,\beta)\cap A(\alpha',\beta')| \\ &\geq \ \sum_\alpha\left(\sum_\beta |S(\alpha,\beta)|\right) \ - \ \sum_\alpha\left(\sum_{\beta\neq\beta'} |S(\alpha,\beta)\cap S(\alpha,\beta')|\right) \\ &\qquad - \sum_{(\alpha,\beta)\neq(\alpha',\beta')} |A(\alpha,\beta)\cap A(\alpha',\beta')|\end{aligned}$$

Let us call the three terms in the RHS of the last equation as T_1, T_2 and T_3 respectively. Since we know the size of each $S(\alpha,\beta)$ exactly, the value of T_1 is easily obtained.

Lemma 20.18 ([KS14c]).

$$T_1(\alpha) \quad := \quad \sum_\beta |S(\alpha,\beta)| \ = \ \textit{(\# mons in a deriv)} \cdot \binom{n-d+k}{\ell}$$

We shall be simplifying such binomial coefficients very often so let us recall the Lemma 4.7.

Lemma 4.7. *Let n and ℓ be parameters such that $\ell = \frac{n}{2}(1-\varepsilon)$ for some $\varepsilon = o(1)$. For any a, b such that $a, b = O(\sqrt{n})$,*

$$\binom{n-a}{\ell-b} \quad = \quad \binom{n}{\ell}\cdot 2^{-a}\cdot(1+\varepsilon)^{a-2b}\cdot\exp(O(b\cdot\varepsilon^2)).$$

Since our of parameters would be $\varepsilon = \Theta\left(\frac{\log d}{\sqrt{d}}\right)$, the bound on T_1 can be simplified as

$$T_1(\alpha) \ = \ (\#\text{ mons in a deriv})\cdot\binom{n}{\ell}\cdot\left(\frac{1+\varepsilon}{2}\right)^{d-k}\cdot\exp(O(\log^2 d))$$

$$= \quad m^{e-k} \cdot \binom{n}{\ell} \cdot \left(\frac{1+\varepsilon}{2}\right)^{d-k} \cdot \exp(O(\log^2 d))$$

Remark. *To avoid writing this factor of* $\exp(O(\log^2 d))$*, we shall use* $\approx$ *of* $\gtrsim$ *or* $\lesssim$ *to indicate that a factor* $\exp(O(\log^2 d))$ *is omitted.* $\Diamond$

So far we have not used any property of the polynomial P. But this becomes crucial in the calculation of T_2 and T_3. To get a sense of how these calculations proceed in [KS14c], we present the full calculation for the case of $P = \mathrm{NW}_{d,m,e}$ for suitable choices of the parameters m, d, e.

Lemma 20.19 ([KS14c])**.** *For the polynomial* $\mathrm{NW}_{d,m,e}$*, if* $n = md$ *and* $\ell = \frac{n}{2}(1-\varepsilon)$ *for* $\varepsilon = \Theta\left(\frac{\log d}{\sqrt{d}}\right)$*, for every* $\alpha \in \Delta$*,*

$$T_2(\alpha) \quad := \quad \sum_{\beta \neq \beta'} |S(\alpha,\beta) \cap S(\alpha,\beta')| \quad \lesssim \quad m^{2(e-k)} \cdot \binom{n}{\ell} \cdot \left(\frac{1+\varepsilon}{2}\right)^{2d-2k}$$

Proof. Recall that $S(\alpha,\beta) \cap S(\alpha,\beta')$ is just set of all multilinear monomials γ of degree ℓ that are disjoint from both β and β'. Hence, for any pair of multilinear degree $(d-k)$ monomials $\beta \neq \beta' \in \partial_\alpha(P)$ such that $\deg(\gcd(\beta,\beta')) = t$, we know that

$$|S(\alpha,\beta) \cap S(\alpha,\beta')| \quad = \quad \binom{n-2d+2k+t}{\ell}$$

Thus, if we can count the number of pairs (β,β') that agree on exactly t places, we can obtain $T_2(\alpha)$. Note that for $\mathrm{NW}_{d,m,e}$, any two $\beta,\beta' \in \partial_\alpha(\mathrm{NW}_{d,m,e})$ can agree on at most $e-k$ places. Further, the number of pairs that agree in exactly $0 \leq t \leq e-k$ places is at most

$$m^{e-k} \cdot \binom{d-k}{t} \cdot (m-1)^{e-k-t}$$

as there are m^{e-k} choices for β, and $\binom{d-k}{t}$ choices for places where they may agree, and $(m-1)^{e-k-t}$ choices for β' that agree with β on those t places. Thus,

$$T_2(\alpha) \;\leq\; \sum_{t=0}^{e-k} m^{e-k} \cdot \binom{d-k}{t} \cdot (m-1)^{e-k-t} \cdot \binom{n-2d+2k+t}{\ell}$$

$$\begin{aligned}
&\approx \quad \sum_{t=0}^{e-k} m^{e-k} \cdot \binom{d-k}{t} \cdot (m-1)^{e-k-t} \cdot \binom{n}{\ell} \frac{1}{2^{2d-2k-t}} \cdot (1+\varepsilon)^{2d-2k-t} \\
&\leq \quad m^{2(e-k)} \binom{n}{\ell} \left(\frac{1+\varepsilon}{2}\right)^{2d-2k} \cdot \sum_{t=0}^{e-k} \binom{d-k}{t} \left(\frac{2}{(1+\varepsilon)m}\right)^{t} \\
&\leq \quad m^{2(e-k)} \binom{n}{\ell} \left(\frac{1+\varepsilon}{2}\right)^{2d-2k} \cdot \left(1 + \frac{2}{(1+\varepsilon)m}\right)^{d-k} \\
&= \quad m^{2(e-k)} \cdot \binom{n}{\ell} \cdot \left(\frac{1+\varepsilon}{2}\right)^{2d-2k} \cdot O(1) \qquad \text{if } m = \Omega(d) \qquad \square
\end{aligned}$$

Combining this with Lemma 20.18 and using Inclusion-Exclusion (Corollary 20.16),

$$\begin{aligned}
\left|\bigcup_{\beta} S(\alpha, \beta)\right| \quad &\gtrsim \quad T_1(\alpha) \cdot \min\left(1, \frac{T_1(\alpha)}{T_2(\alpha)}\right) \\
&\approx \quad T_1(\alpha) \cdot \min\left(1, \frac{\left(\frac{2}{1+\varepsilon}\right)^{d-k}}{m^{e-k}}\right)
\end{aligned}$$

To maximize this, if we choose the parameters m, d, e such that $T_1(\alpha) \approx T_2(\alpha)$, we obtain the following corollary.

Corollary 20.20. *Consider the polynomial* $\mathrm{NW}_{d,m,e}$ *with* $n = md$ *and* $m = \Omega(d)$. *If* $\ell = \frac{n}{2}(1-\varepsilon)$ *for* $\varepsilon = \Theta\left(\frac{\log d}{\sqrt{d}}\right)$ *and* e *chosen so that*

$$m^{e-k} \quad \overset{\text{poly}}{=} \quad \left(\frac{2}{1+\varepsilon}\right)^{d-k}$$

then

$$\sum_{\substack{\alpha \in \Delta \\ \beta \in \partial_\alpha(\mathrm{NW})}} |A(\alpha, \beta)| \quad \gtrsim \quad |\Delta| \cdot \binom{n}{\ell}$$

Proof. By Lemma 20.17, we know that

$$\sum_{\substack{\alpha \in \Delta \\ \beta \in \partial_\alpha(P)}} |A(\alpha, \beta)| \quad \geq \quad |\Delta| \cdot \left|\bigcup_{\beta} S(\alpha, \beta)\right|.$$

Furthermore, from the discussion above, if $T_1(\alpha) \approx T_2(\alpha)$ then

$$\begin{aligned}\left|\bigcup_{\beta} S(\alpha,\beta)\right| &\gtrsim T_1(\alpha)\cdot \min\left(1, \frac{T_1(\alpha)}{T_2(\alpha)}\right)\\ &= T_1(\alpha)\\ &\approx \binom{n}{\ell}\end{aligned}$$

as $T_1(\alpha) \approx T_2(\alpha)$ forces $m^{e-k} \approx \left(\frac{2}{1+\varepsilon}\right)^{d-k}$. Therefore,

$$\square \qquad \sum_{\substack{\alpha\in\Delta\\ \beta\in\partial_\alpha(P)}} |A(\alpha,\beta)| \quad \gtrsim \quad |\Delta|\cdot\binom{n}{\ell}$$

Note that e needs to tailored very precisely to force the above condition! If e is chosen too large or small, we get nothing from this whole exercise!

In the case of IMM this calculations gets a lot messier. The calculation would similarly force that the number of monomials must be in a very narrow range. This is achieved by instead looking at a random subgraph of the generic ABP of suitable sparsity to ensure the following two properties:

- The number of monomials in any derivative is exactly as demanded.
- 'Most' pairs of monomials (β, β') agree on 'few' places.

Upper bounding $\sum |A(\alpha,\beta) \cap A(\alpha',\beta')|$

We are still left with the task of upper bounding

$$T_3 \quad = \quad \sum_{(\alpha,\beta)\neq(\alpha',\beta')} |A(\alpha,\beta)\cap A(\alpha',\beta')|$$

As mentioned earlier, we really do not have a good handle on the set $A(\alpha,\beta)$, and certainly not on the intersection of two such sets. Once again, we shall use a proxy that is easier to estimate to upper bound T_3.

The set $A(\alpha,\beta)\cap A(\alpha',\beta')$ consists of multilinear monomials γ of degree $\ell + d - k$ such that there exists multilinear monomials γ', γ'' of degree ℓ satisfying

$$\gamma \;=\; \gamma'\beta \;=\; \gamma''\beta',$$

$$\begin{aligned} \gamma'\beta &= \mathrm{LM}(\mathrm{mult}(\gamma'\partial_\alpha(P))) \\ \text{and} \quad \gamma''\beta' &= \mathrm{LM}(\mathrm{mult}(\gamma''\partial_{\alpha'}(P))) \end{aligned}$$

This in particular implies that γ must be divisible by both β and β'.

Observation 20.21. *If* $\deg(\gcd(\beta, \beta')) = t$*, then*

$$|A(\alpha, \beta) \cap A(\alpha', \beta')| \quad \leq \quad \binom{n - 2d + 2k + t}{\ell - d + k + t}$$

Proof. Every monomial $\gamma \in A(\alpha, \beta) \cap A(\alpha', \beta')$ must be divisible by β and β'. Since $|\beta \cup \beta'| = 2d - 2k - t$, the number of choices of γ is precisely

$$\binom{n - (2d - 2k - t)}{(\ell + d - k) - (2d - 2k - t)} \quad = \quad \binom{n - 2d + 2k + t}{\ell - d + k + t}$$

□

One needs a similar argument as in the case of T_2 to figure out how many pairs $(\alpha, \beta) \neq (\alpha', \beta')$ are there with $\deg(\gcd(\beta, \beta')) = t$ and sum them up accordingly.

Lemma 20.22 ([KS14c]). *For the polynomial* $\mathrm{NW}_{d,m,e}$*, and* $n = md$ *and* $\ell = \frac{n}{2}(1 - \varepsilon)$ *for* $\varepsilon = \Theta\left(\frac{\log d}{\sqrt{d}}\right)$*,*

$$T_3 \quad \lesssim \quad |\Delta|^2 \cdot \left(\frac{m^{e-k}}{2^{d-k}}\right)^2 \cdot \binom{n}{\ell}$$

Proof. Fix a pair of derivatives α, α'. As before, we shall first count the number of pairs of monomials $\beta \in \partial_\alpha P$ and $\beta' \in \partial_{\alpha'} P$ such that $\gcd(\beta, \beta') = t$. Note that since α may differ from α', we could potentially have $\gcd(\beta_1, \beta_2) = e$. Once again, this is easily seen to be at most

$$m^{e-k} \cdot \binom{d - k}{t} \cdot (m - 1)^{e-k-t}.$$

Therefore, using Observation 20.21,

$$\begin{aligned} T_3(\alpha, \alpha') \quad &\leq \quad \sum_{t=0}^{e} m^{e-k} \cdot (m-1)^{e-k-t} \binom{d-k}{t} \binom{n - 2d + 2k + t}{\ell - d + k + t} \\ &\approx \quad \sum_{t=0}^{e} m^{e-k} \cdot (m-1)^{e-k-t} \binom{d-k}{t} \cdot \binom{n}{\ell} \left(\frac{1}{2}\right)^{2d-2k-t} (1 + \varepsilon)^t \end{aligned}$$

$$\begin{aligned} &\leq \frac{m^{2(e-k)}}{2^{2(d-k)}} \cdot \binom{n}{\ell} \cdot \left(1 + \frac{2(1+\varepsilon)}{m}\right)^{d-k} \\ &\approx \frac{m^{2(e-k)}}{2^{2(d-k)}} \cdot \binom{n}{\ell} \qquad (\text{as } m = \Omega(d)) \\ \implies T_3 &\lesssim |\Delta|^2 \cdot \left(\frac{m^{e-k}}{2^{d-k}}\right)^2 \cdot \binom{n}{\ell} \end{aligned}$$ □

Recalling that we have chosen our parameters so that

$$(\#\text{ mons per deriv}) \approx \left(\frac{2}{1+\varepsilon}\right)^{d-k}$$

the above equation reduces to

$$T_3 \quad \lesssim \quad |\Delta|^2 \left(\frac{1}{1+\varepsilon}\right)^{2(d-k)} \cdot \binom{n}{\ell}.$$

We shall choose our set of derivatives so that $|\Delta| \approx (1+\varepsilon)^{2(d-k)}$. With that setting, we can readily see that $T_3 \lesssim T_1$.

Combining with Corollary 20.20, we obtain the required bound for $|\bigcup A(\alpha, \beta)|$ via Inclusion-Exclusion (Corollary 20.16).

Lemma 20.23. *Let $m = d^2$ (so that $n = md = d^3$). Let $k = O(\sqrt{d})$ and $\ell = \frac{n}{2}(1-\varepsilon)$ for $\varepsilon = \frac{\log d}{c\sqrt{d}}$ where c is a constant. If c and e are tailored so that*

$$\begin{aligned} |\Delta| \quad = \quad m^k &\gtrsim (1+\varepsilon)^{2d-2k} \\ m^{e-k} &\approx \left(\frac{2}{1+\varepsilon}\right)^{d-k} \end{aligned}$$

Then, for the polynomial $\mathrm{NW}_{d,m,e}$, if we consider a subset of non-zero derivatives order k of size $\lfloor (1+\varepsilon)^{2d-2k} \rfloor$, then

$$\Gamma_{k,\ell}^{\mathrm{PSD}}(\mathrm{NW}_{d,m,e}) \quad \geq \quad \left|\bigcup_{\alpha,\beta} A(\alpha,\beta)\right| \quad \gtrsim \quad \binom{n}{\ell} \cdot (1+\varepsilon)^{2d-2k}.$$

By Lemma 20.6, we know that any homogeneous depth-4 circuit C of size s and bottom fan-in r satisfies

$$\Gamma_{k,\ell}^{\mathrm{PSD}}(C) \quad \leq \quad s \cdot \binom{n}{\ell} \cdot (1+\varepsilon)^{rk} \cdot 2^{O(\sqrt{d})}.$$

Hence, if r was small enough (say $r = \sqrt{d}/1000$) so that $rk \leq (d-k)$, then we have a lower bound of $s \geq (1+\varepsilon)^{d-k} \cdot 2^{O(\sqrt{d})}$ which is $d^{\Omega(\sqrt{d})}$ by the choice of ε.

Theorem 20.24 ([KS14c])**.** *Any homogeneous depth 4 circuit with bottom support bounded by $r = \sqrt{d}/1000$ computing, over any field $\mathbb{F}$, the polynomial $\mathrm{NW}_{d,m,e}$ with parameters as defined above must have top fan-in $s = d^{\Omega(\sqrt{d})}$.*

In fact, more generally, any homogeneous depth 4 circuit of bottom support bounded by r computing $\mathrm{NW}_{d,m,e}$ for suitably chosen parameters must have top fanin $s = d^{\Omega(d/r)}$.

Again, coupling with Lemma 20.5, we obtain (a slight reformulation of) their theorem.

Theorem 20.25 ([KLSS17, KS14c])**.** *Any homogeneous depth 4 circuit computing, over any field $\mathbb{F}$, the polynomial $\mathrm{NW}_{d,m,e} \circ \mathrm{Lin}$ with parameters as defined above must have top fan-in $s = d^{\Omega(\sqrt{d})}$.*

A similar lower bound $d^{\Omega(\sqrt{d})}$ holds also for the polynomial $\mathrm{IMM}_{n,d} \circ \mathrm{Lin}$ for suitable choices of n and d.

Exercise 20.1 *Show that there indeed does exist settings of c and e so as to satisfy the constraints in Lemma 20.23.*

Part VII

Further applications of shifted partial derivatives

Chapter 21

Quick summary of key points

The chapters that follow would delve deeper to show more general models where variants of shifted or projected shifted partial derivatives can be used to prove lower bounds. A lot of the lower bounds would use specific observation or tricks used in the proofs in Chapter 19 and Chapter 20.

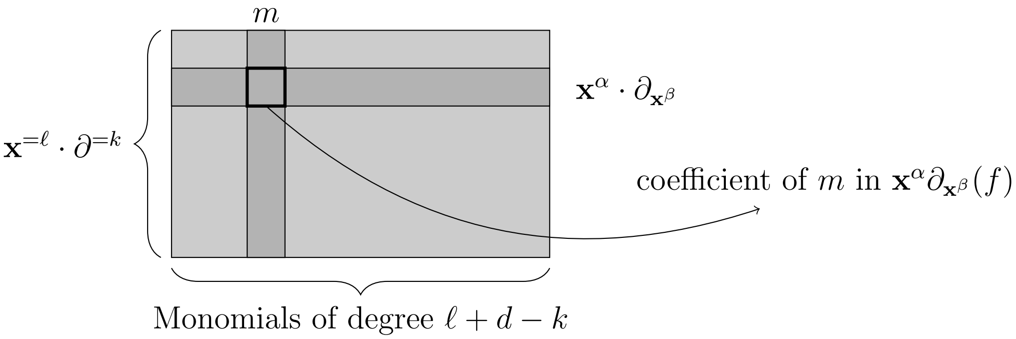
> **Note to reader:** All these lower bounds would involve a delicate play between the various parameters involved. In order to completely understand, it is *imperative* that you work out the calculations in Chapter 19 and Chapter 20 (specifically those in Section 20.3) at least once completely. Doing that would give some sense of how the different parameters interact and this is crucial to a lot of the lower bounds that follow.

Having said that, this chapter shall be devoted to restating the most important points to remember from the previous two chapters. These would be enough to get a high-level view of the lower bounds that follow but one has to get their hands dirty somewhere to completely understand these.

21.1 Shifted Partial Derivatives

Definition (Shifted Partial Derivatives, Definition 19.4)**.** *Let $\partial^{=k}(f)$ refer to the set of all k-th order partial derivatives of f, and $\mathbf{x}^{\leq \ell}$ refer to the set of all monomials of degree at most ℓ. The* shifted partials *of f, denoted by $\left\langle \partial^{=k}(f)\right\rangle_{\leq \ell}$, is the vector space spanned by $\left\{\mathbf{x}^{\leq \ell} \cdot \partial^{=k}(f)\right\}$. The dimension of this space shall be denoted by $\Gamma_{k,\ell}^{[\mathrm{SPD}]}(f)$.* ◊

To express this pictorially, $\Gamma^{[\mathrm{SPD}]}_{k,\ell}(f)$ is the rank of the following matrix.


m
$\mathbf{x}^{\alpha} \cdot \partial_{\mathbf{x}^{\beta}}$
$\mathbf{x}^{=\ell} \cdot \partial^{=k}$
coefficient of m in $\mathbf{x}^{\alpha}\partial_{\mathbf{x}^{\beta}}(f)$
Monomials of degree $\ell + d - k$

This measure is used to prove lower bounds for the top fan-in of depth four circuits *with bounded bottom fan-in.*

Lemma (Upper bound for hom. $\Sigma\Pi\Sigma\Pi^{[t]}$ circuits, restating Corollary 19.7)**.** *Let f be an n-variate degree d polynomial of the form $f = Q_1 \cdots Q_a$ with $\deg(Q_i) \leq t$. Then for any k, ℓ, we have*

$$\Gamma^{[\mathrm{SPD}]}_{k,\ell}(f) \quad \leq \quad \binom{a}{k} \cdot \binom{n + \ell + k(t-1)}{n}$$

By grouping factors of degree much smaller than t, one can assume without loss of generality that $a = O(d/t)$. One should note that the first binomial coefficient $\binom{a}{k}$ is at most 2^a. Thus if the goal is to prove a lower bound of $n^{\Omega(d/t)} = n^{\Omega(a)}$, then the first term is not too relevant.

Exercise 21.1 *Let $H(Q_1, \ldots, Q_a)$ be an arbitrary polynomial function applied to $Q_1, \ldots, Q_a$. Suppose $\deg(Q_i) \leq t$ for all i. Show that*

$$\Gamma^{[\mathrm{SPD}]}_{k,\ell}(H(Q_1, \ldots, Q_a)) \quad \leq \quad \binom{a+k}{a} \cdot \binom{n + \ell + k(t-1)}{n}.$$

The above lemma is a special case of the above more general exercise.

The second part of the lower bound is to show that the measure is large for explicit polynomials. The Nisan-Wigderson polynomial, $\mathrm{NW}_{d,m,e}$, is designed so that the measure is almost as large as possible. The iterated matrix multiplication polynomial, IMM, also has a large value of $\Gamma^{[\mathrm{SPD}]}_{k,\ell}$, though not as large as the value for NW. For the right range of parameters,

$$\Gamma^{[\mathrm{SPD}]}_{k,\ell}(Q_1 \cdots Q_a) \ll \Gamma^{[\mathrm{SPD}]}_{k,\ell}(\mathrm{IMM}) \ll \Gamma^{[\mathrm{SPD}]}_{k,\ell}(\mathrm{NW}) \approx \text{Maximum possible.}$$

21.2 Projected Shifted Partial Derivatives

An important variant that shall be heavily used in the following chapters is the measure of *projected shifted partial derivatives* defined in Chapter 20.

Definition (Projected shifted partial derivatives, Definition 20.3)**.** *Fix parameters $k, \ell > 0$. For any polynomial P, the set of projected shifted partials of f, denoted by $\mathrm{PSD}_{k,\ell}(f)$ is defined as follows*

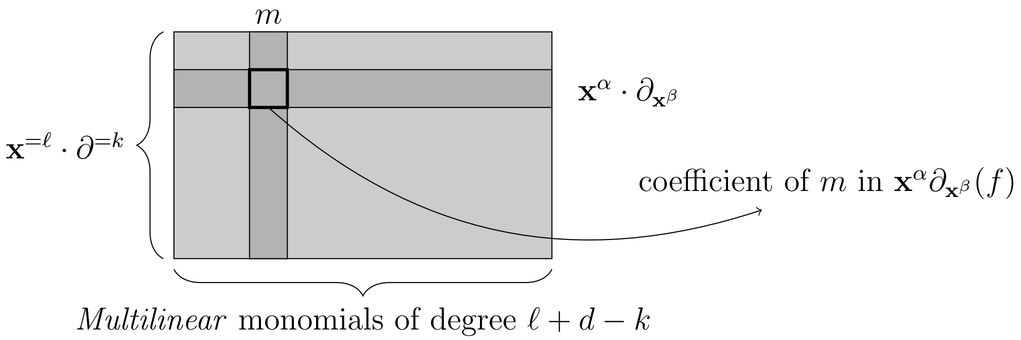
$$\mathrm{PSD}_{k,\ell}(f) \quad = \quad \left\{ \mathrm{mult}(m_1 \cdot \partial_{m_2}(f)) \;:\; \begin{array}{c} \deg(m_1) = \ell\ ,\ \deg(m_2) = k, \\ m_1 \textit{ and } m_2 \textit{ are multilinear} \end{array} \right\}$$

where $\mathrm{mult}(f)$ *refers to the polynomial f projected to only the multilinear monomials of f.*

The measure $\Gamma^{\mathrm{PSD}}_{k,\ell}(f)$ is defined as the dimension of the above set of polynomials, i.e.

$$\Diamond \qquad \Gamma^{\mathrm{PSD}}_{k,\ell}(f) \quad = \quad \dim\left(\mathrm{span}(\mathrm{PSD}_{k,\ell})\right)$$

Pictorially, the measure is the rank of the following matrix.


m
$\mathbf{x}^{\alpha} \cdot \partial_{\mathbf{x}^{\beta}}$
$\mathbf{x}^{=\ell} \cdot \partial^{=k}$
coefficient of m in $\mathbf{x}^{\alpha}\partial_{\mathbf{x}^{\beta}}(f)$
Multilinear monomials of degree $\ell + d - k$

One can think of this as a sub-matrix for the shifted partial derivatives obtained by throwing out the columns corresponding to non-multilinear monomials.

This measure was used to prove lower bounds on the *total size* of homogeneous depth four circuits (with no bottom fan-in restrictions), unlike the previous setting which was a top fan-in lower bound. But the following is an important note to bear in mind:

Projected shifted partial derivatives is designed to prove lower bounds on the top fan-in of homogeneous depth four circuits with *low bottom support size.*

It so happens that any homogeneous depth four circuit of small total size can be reduced to a depth four circuit of small bottom support size. But this distinction is important and should be stressed.

21.2.1 Depth four circuits of low bottom support size

Let $f = Q_1 \cdots Q_a$ with $d = \deg(f)$ and suppose all the monomials of any Q_i depend on just r variables. There are two important observations made to prove the upper bound of the complexity measure on such circuits.

Observation (Non-multiquadratic terms do not contribute)**.** *Let g be a polynomial such that every monomial of g is divisible by some x_i^3, or in other words each monomial of g is* non-multiquadratic*. Then for an k, ℓ, we have $\Gamma^{[\text{PSD}]}_{k,\ell}(g) = 0$.*

Observation (Decomposition of low support size products)**.** *Let $f = Q_1 \cdots Q_a$ be a polynomial of degree d such that all monomials of any Q_i depends on at most t variables. Then f can be expressed as*

$$f \quad = \quad Q'_1 \cdots Q'_a \quad + \quad g$$

where $\deg(Q'_i) \leq 2r$ for all i, and every monomial of g is non-multiquadratic.

Therefore, we have

$$\Gamma^{[\text{PSD}]}_{k,\ell}(Q_1 \cdots Q_a) = \Gamma^{[\text{PSD}]}_{k,\ell}(Q'_1 \cdots Q'_a)$$

and the RHS is a low bottom degree product. Thus similar to the upper bound for shifted partial derivatives, bearing in mind that we only care about multilinear monomials, one can easily show the following.

Lemma (Upper bound for low bottom support size circuits, Lemma 20.6)**.** *Let $f = Q_1 \cdots Q_a$ be an n-variate degree d polynomial with each Q_i a sum of monomials depending on at most r variables. Then for any k, ℓ with $\ell + kr \leq \frac{n}{2}$,*

$$\Gamma^{[\text{PSD}]}_{k,\ell}(f) \quad \leq \quad \binom{(2d/r) + k}{k} \cdot \binom{n}{\ell + kr}$$

For a very delicate range of parameters, we have a very similar behaviour of the measure on the standard polynomials.

$$\Gamma_{k,\ell}^{[\text{PSD}]}(Q_1 \cdots Q_a) \ll \Gamma_{k,\ell}^{[\text{PSD}]}(\text{IMM}) \ll \Gamma_{k,\ell}^{[\text{PSD}]}(\text{NW}) \approx \text{Maximum possible.}$$

In the right range of parameters, this gives an $n^{\Omega(d/t)}$ lower bound on the top fan-in of any homogeneous depth four circuit of bottom support size bounded by t that computes NW or IMM.

The calculations involved are quite delicate but it would be useful to just keep the case of NW in mind as the full calculations were described in Section 20.3. But a couple of thing to keep in mind is that the calculations for $\text{NW}_{d,m,e}$ work over any field but as long as m^e roughly equal to 2^d (the precise constraints are explicit described in Section 20.3).

So far, this only addresses depth four circuits of small bottom support size. In order to reduce the general setting of homogeneous depth four circuits to this case, there is one additional trick employed.

21.2.2 Reducing to depth four circuits of low bottom support size

The key observation here is that if we have a depth four circuit of small size, then the number of distinct monomials computed at the layer closest to the leaves is bounded by the size of the circuit. As a concrete instance, say we have a depth four circuit of size $s = n^{(0.1)\sqrt{d}}$. We shall now pick a *lot* of variables at random and set them to zero or to be more precise we shall set each variable to 0 independently with probability $1 - \frac{1}{n^{0.2}}$. With very high probability, we would now be left with about $n^{0.8}$ variables but the resulting circuit remains homogeneous as setting variables to zero maintains homogeneity.

However, if m is a monomial that depends on $\sqrt{d}$ or more variables, the probability that this monomials survives this random restriction is at most $\frac{1}{n^{(0.2)\sqrt{d}}}$. Thus, even if we union bound over all monomials present in the depth four circuit of size $n^{(0.1)\sqrt{d}}$ we get that the probability that some monomial of support $\sqrt{d}$ or more survives this process is at most $\frac{1}{n^{(0.1)\sqrt{d}}} = o(1)$. Thus, almost surely, the resulting circuit is now a homogeneous depth four circuit with bottom support size at most $\sqrt{d}$.

This however complicates the other side of the argument where we now need to find a polynomial P such that *even after a random restriction ρ is applied on P*, we must have $\Gamma_{k,\ell}^{[\text{PSD}]}(\rho(P))$ to be large. This is extremely non-trivial to see if NW or IMM are so robust. Fortunately, there is a hack that allows us to circumvent this at the cost of making the polynomial uglier.

Lemma (Linear blow-up trick to handle random restrictions, Lemma 20.5)**.** *Let ρ be a random restriction that sets each variable to zero independently with probability $1-\alpha$. For any polynomial $P(x_1, \ldots x_n)$, define $P \circ \mathrm{Lin}_\alpha$ as*

$$P \circ \mathrm{Lin}_\alpha \quad = \quad P\left(\sum_{i=1}^{t} y_{1i}, \ldots, \sum_{i=1}^{t} y_{nt}\right) \quad \textit{where } t = \left(\frac{1}{\alpha}\right) n \log n$$

Then, $\rho(f \circ \mathrm{Lin}_\alpha)$ has f as a projection with probability $1 - 1/2^n$.

Basically, we replace each variable in NW by a sum of t new distinct variables where $t = (1/\alpha)\log n$. The point is that, if a variable is kept alive with probability α, then with very high probability, one of the t variables $\{y_{ij}\}_{j \in [t]}$ will be kept alive for each x_i. Hence, there is a copy of $\mathrm{NW}_{d,m,e}$ sitting inside $\rho(\mathrm{NW}_{d,m,e} \circ \mathrm{Lin}_\alpha)$ with very high probability.

Therefore, if we assume on the contrary that C is a homogeneous depth four circuit of size $n^{(0.1)\sqrt{d}}$ computing $\mathrm{NW}_{d,m,e} \circ \mathrm{Lin}_\alpha$, then there is a homogeneous circuit C' with size at most $n^{(0.1)\sqrt{d}}$ and bottom support size at most $\sqrt{d}$ that computes $\mathrm{NW}_{d,m,e}$. But since we already have a lower bound for homogeneous depth four circuits of low bottom support size computing $\mathrm{NW}_{d,m,e}$, we get a contradiction.

Hopefully this would give the readers a rough description of the main observations. But to really understand them, one *has* to work out the calculations in Section 20.3 at least once to get a better grip of how these parameters interact. We now move on to some other models for which projected shifted partials, or variants of it, can again be employed.

Chapter 22

Evaluation perspective on projected shifted partial derivatives

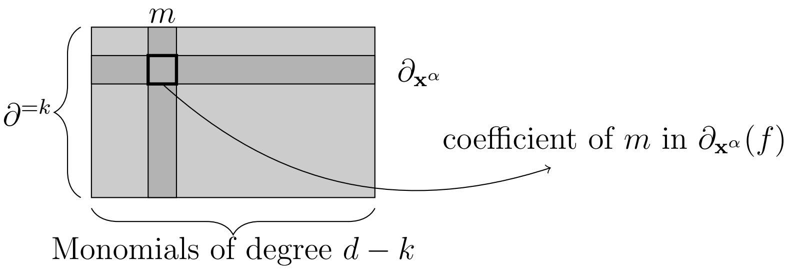
The measure used in the lower bounds of Kayal, Limaye, Saha, Srinivasan [KLSS17] and Kumar, Saraf [KS14c] was the dimension of projected shifted partials. As seen in that chapter, the calculations are extremely delicate. In this chapter, we shall see some slight modifications of this measure that is in a sense more *algebraic* and hence useful in other lower bounds.

22.1 Coefficients vs evaluations

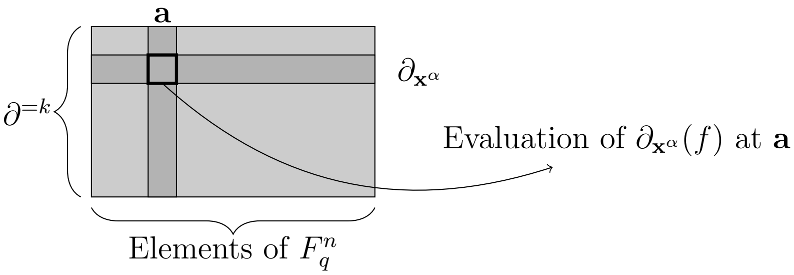
For a moment, let us revisit the lower bounds of Nisan and Wigderson [NW97]. The measure studied for the class of homogeneous depth-3 circuits in Theorem 8.3 was the dimension of partial derivatives.

$$\Gamma_k^{[\mathrm{NW}]}(f) \quad := \quad \dim\left\{\partial^{=k}(f)\right\}$$

More precisely, we interpret any element of $\partial^{=k}(f)$ as a long vector of coefficients and look at the dimension of this collection of vectors. That is, $\Gamma_k^{[\mathrm{NW}]}(f)$ is the rank of the following matrix.


m
$\partial^{=k}$
$\partial_{\mathbf{x}^\alpha}$
coefficient of m in $\partial_{\mathbf{x}^\alpha}(f)$
Monomials of degree $d-k$

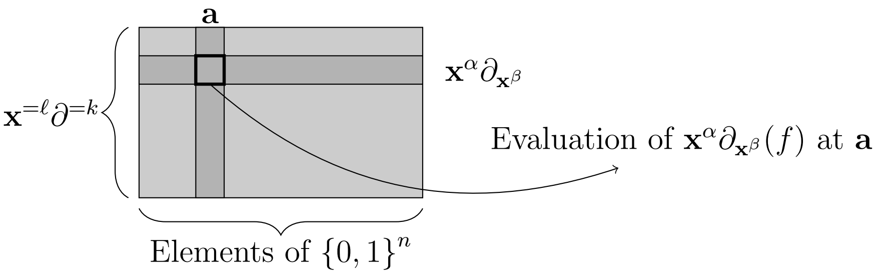
Grigoriev and Karpinski [GK98], for their lower bound for $\Sigma\Pi\Sigma$ circuits over finite fields instead looked at the dual *evaluation perspective* by studying a matrix of the form


$\mathbf{a}$
$\partial^{=k}$
$\partial_{\mathbf{x}^\alpha}$
Evaluation of $\partial_{\mathbf{x}^\alpha}(f)$ at $\mathbf{a}$
Elements of F_q^n

As seen in Chapter 10, the measure $\Gamma_k^{[\mathrm{GK}]}$ was the rank of the above matrix (with a few columns removed). Intuitively, we expect that if the rank of the *coefficient* matrix is large, then the rank of the *evaluation* matrix should also be large. Sometimes, the evaluation perspective allows us to handle the circuit model better. In a way, the proof of Grigoriev and Karpinski [GK98] can be thought of as a formalization of this intuition for $\Sigma\Pi\Sigma$ circuits.

A similar perspective can also be adopted for the dimension of shifted partial derivatives. For the dimension of projected shifted partials however, this connection is not that clean. Roughly speaking, throwing away non-multilinear monomials changes the evaluations of the shifted partials at points. Formally, the multilinear projection of a polynomial f can be interpreted as looking at the residue of f mod $\{x_i^2 \ : \ i \in [n]\}$, that is just replacing any x_i^2 by zero. However this reduction does not work well with evaluations as $f(\mathbf{a})$ could be different from $(f \bmod \{x_i^2 \ : \ i \in [n]\})(\mathbf{a})$ for each $\mathbf{a} \in \mathbb{F}^n$.

Let us turn the question around and ask if we wish to understand the rank of the following matrix



a
$\mathbf{x}^\alpha \partial_{\mathbf{x}^\beta}$
$\mathbf{x}^{=\ell}\partial^{=k}$
Evaluation of $\mathbf{x}^\alpha \partial_{\mathbf{x}^\beta}(f)$ at $\mathbf{a}$
Elements of $\{0,1\}^n$

what is the *coefficient* analogue of this measure? Turns out, the answer is a different notion of projected shifted partial derivatives where the projection is not modulo $\{x_i^2 \,:\, i \in [n]\}$ but instead modulo $\{x_i^2 - x_i \,:\, i \in [n]\}$. It should be intuitively clear that as long as we are only interested in evaluations on $\{0,1\}^n$, the evaluation of f mod $\{x_i^2 - x_i \,:\, i \in [n]\}$ at $\mathbf{a} \in \{0,1\}^n$ yields the same as $f(\mathbf{a})$.

What can we say about this modification of projected shifted partial derivatives? Is this also a measure that can give the homogeneous depth-4 lower bounds studied in Chapter 20? Turns out the answer is indeed yes, and this perspective also allows one to generalize the lower bounds to homogeneous depth-5 circuits over finite fields by Kumar and Saptharishi [KS17]. There would be like the evaluation perspective of Grigoriev and Karpinski [GK98] (that we saw in Chapter 10) of the lower bound of Nisan and Wigderson [NW97] (that we saw in Subsection 8.1.3).

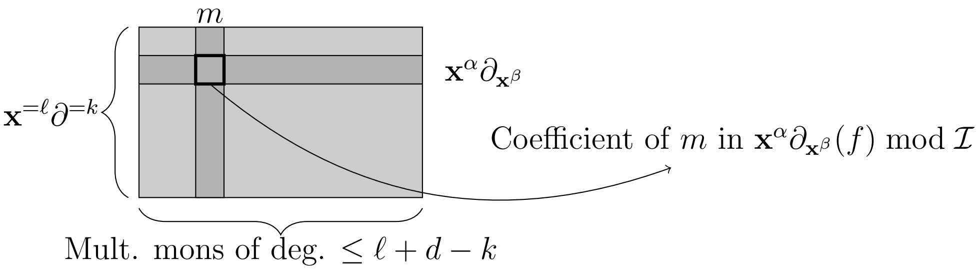
22.2 Projected shifted partials via $\left\{x_i^2 - x_i \,:\, i \in [n]\right\}$

The following definition is a little more general but would be useful later in this chapter. But for now, it would be useful to just consider $\mathcal{I} = \langle x_i^2 - x_i \,:\, i \in [n]\rangle$.

Definition 22.1 (PSDs with respect to $\mathcal{I}$)**.** *The* projected shifted partial derivatives with respect to $\mathcal{I}$ *for a polynomial f, denoted by $\Gamma_{k,\ell,\mathcal{I}}(f)$, is defined as*

$$\Diamond \qquad \Gamma^{\text{PSD}}_{k,\ell,\mathcal{I}}(f) \quad := \quad \dim\left\{\left(\mathbf{x}^{=\ell} \cdot \partial^{=k}(f)\right) \bmod \mathcal{I}\right\}.$$

In the setting where $\mathcal{I} = \langle x_i^2 - x_i \,:\, i \in [n]\rangle$, every polynomial f, there is a unique multilinear polynomial g for which $f = g$ mod $\mathcal{I}$ and we shall refer to this g by f mod $\mathcal{I}$. Thus, $\Gamma_{k,\ell,\mathcal{I}}(f)$ is the rank of the following matrix:


m
$\mathbf{x}^{=\ell}\partial^{=k}$
$\mathbf{x}^{\alpha}\partial_{\mathbf{x}^{\beta}}$
Coefficient of m in $\mathbf{x}^{\alpha}\partial_{\mathbf{x}^{\beta}}(f) \bmod \mathcal{I}$
Mult. mons of deg. $\leq \ell + d - k$

In Definition 20.3, we are essentially working with $\mathcal{I} = \langle x_i^2 \ : \ i \in [n] \rangle$ but working modulo other ideals is at times more useful. In fact, there is a fairly large class of ideals $\mathcal{I}$ for which $f \bmod \mathcal{I}$ has a unique multilinear representative. But to make it applicable for lower bounds for homogeneous depth-4 circuits, we will need a mechanism to transform a "low support polynomial" to a "low degree polynomial".

Definition 22.2 (Support-to-degree ideals)**.** *An ideal $\mathcal{I}$ is said to be a* support-to-degree *ideal if there exist linear polynomials $\ell_1, \ldots, \ell_n$ such that*

$$\Diamond \qquad \mathcal{I} \quad = \quad \left\langle x_i^2 - \ell_i(x_i) \ : \ i \in [n] \right\rangle .$$

Observation 22.3. *For any polynomial f and a support-to-degree ideal $\mathcal{I}$, there is a unique multilinear polynomial g such that $f = g \bmod \mathcal{I}$.*

Furthermore, if f is a polynomial such that every monomial in f depends on at most r variables, then the unique multilinear polynomial $g = f \bmod \mathcal{I}$ in fact has degree at most r.

It is important to note that the support-to-degree ideal has generators that replace x_i^2 by a *linear polynomial in just x_i*. Relaxing this to $x_i^2 - \ell$ for an arbitrary linear polynomial ℓ could lead to monomials of large support.

Proof. The proof follows immediately by repeatedly replacing x_i^2 by $\ell_i(x_i)$. □

As mentioned before, we would need the above more general definition in a later lower bound but for now it would help to just keep ideals such as $\langle x_i^2 \ : \ i \in [n] \rangle$ or $\langle x_i^2 - x_i \ : \ i \in [n] \rangle$ in mind.

In order to check if Definition 22.1 is a measure useful for homogeneous depth-4 circuits, we need to show two things — (1) the measure is small for a small homogeneous depth-4 circuit (of low bottom support), and (2) the measure is large for an explicit polynomial. These together would imply the practicability of dimension of projected shifted partials with respect to an arbitrary support-to-degree ideal.

The second claim would be easier to prove so let us address that first.

Lemma 22.4 (PSD wrt $\mathcal{I}$ for homogeneous polynomials)**.** *Let f be any homogeneous polynomial of degree d. For any choice of k, ℓ and any support-to-degree ideal $\mathcal{I}$, we have*

$$\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}(f) \quad \geq \quad \Gamma^{\mathrm{PSD}}_{k,\ell}(f).$$

Proof. Let $g \in \mathbf{x}^{=\ell} \cdot \partial^{=k}(f)$. The main difference between $g \bmod \langle x_i^2 \ : \ i \in [n]\rangle$ and $g \bmod \mathcal{I}$ is just that the first projection just zeros out any non-multilinear monomial of degree $\ell + d - k$ whereas $g \bmod \mathcal{I}$ reduces non-multilinear monomials to lower degree monomials. Hence, $g \bmod \langle x_i^2 \ : \ i \in [n]\rangle$ is just $g \bmod \mathcal{I}$ but just restricted to the multilinear monomials of degree $\ell + d - k$. Thus it follows that the rank of $\left(\mathbf{x}^{=\ell}\partial^{=k} g\right) \bmod \mathcal{I}$ is lower bounded by the rank of $\left(\mathbf{x}^{=\ell}\partial^{=k} g\right) \bmod \langle x_i^2 \ : \ i \in [n]\rangle$. □

We now want to show that if C is a homogeneous depth-4 circuit with bottom support at most $r = O(\sqrt{d})$, then we can give a good upper bound for $\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}(C)$.

Lemma 22.5 (PSD wrt $\mathcal{I}$ for a hom. $\Sigma\Pi\Sigma\Pi$ circuit)**.** *Let C be a homogeneous $\Sigma\Pi\Sigma\Pi$ computing an n-variate degree d polynomial of top fan-in s and bottom support bounded by r. Then for any choice of k, ℓ and any support-to-degree ideal $\mathcal{I}$ we have*

$$\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}} \quad \leq \quad s \cdot \binom{3(d/r)+1}{k} \cdot \binom{n}{\ell + kr}$$

Proof. Let us consider a single summand $T = Q_1 \cdots Q_m$ of C. We shall partition this set into those polynomials $Q_1, \ldots, Q_a$ of degree at most r, and polynomials $Q'_1, \ldots, Q'_b$ of degree more than r. By homogeneity of C, we know that $b \leq d/r$. Since some of the Q_is could have very small degree, a could potentially be as large as d. To handle this, if we find any Q_i and Q_j both of degree at most $r/2$, we shall replace them by their product. This ensures that all Q_is have degree more than $r/2$ except perhaps one. Hence, (reusing some symbols) we can write T as

$$T \quad = \quad \tilde{Q}_1 \cdots \tilde{Q}_a \ \cdot \ Q'_1 \cdots Q'_b$$

where each $a \leq 2(d/r) + 1$, $b \leq (d/r)$, each Q_i has degree at most r and every monomial in a Q'_i depends on at most r variables. For brevity, we shall use the notation $\tilde{Q}_A$ to denote $\prod_{i \in A} \tilde{Q}_i$, and similarly Q'_B to denote $\prod_{i \in B} Q'_i$.

Let $\partial_{\mathbf{x}^\alpha}(T)$ be a k-th order partial derivative of T. By the chain rule of differentiation,

$$\begin{aligned}
\partial_{\mathbf{x}^\alpha}(T) &\in \operatorname{span}\left\{\partial_{\mathbf{x}^\beta}(\tilde{Q}_A)\cdot\partial_{\mathbf{x}^\gamma}(Q'_B)\cdot\tilde{Q}_{\overline{A}}\cdot Q'_{\overline{B}} \;:\; \begin{array}{c}\mathbf{x}^\alpha=\mathbf{x}^\beta\cdot\mathbf{x}^\gamma\\ |A|+|B|\le k\end{array}\right\}\\
&\subseteq \operatorname{span}\left\{\mathbf{x}^{\le k(r-1)}\cdot\partial_{\mathbf{x}^\gamma}(Q'_B)\cdot\tilde{Q}_{\overline{A}}\cdot Q'_{\overline{B}} \;:\; \begin{array}{c}\mathbf{x}^\alpha=\mathbf{x}^\beta\cdot\mathbf{x}^\gamma\\ |A|+|B|\le k\end{array}\right\}\\
\implies \mathbf{x}^{=\ell}\cdot\partial_{\mathbf{x}^\alpha}(T) &\subseteq \operatorname{span}\left\{\mathbf{x}^{\le \ell+k(r-1)}\cdot\partial_{\mathbf{x}^\gamma}(Q'_B)\cdot\tilde{Q}_{\overline{A}}\cdot Q'_{\overline{B}} \;:\; \begin{array}{c}\mathbf{x}^\alpha=\mathbf{x}^\beta\cdot\mathbf{x}^\gamma\\ |A|+|B|\le k\end{array}\right\}
\end{aligned}$$

We now have to look at $\mathbf{x}^{=\ell}\cdot\partial_{\mathbf{x}^\alpha}(T) \bmod \mathcal{I}$ and for that notice that Q'_B is a product of polynomials of low-support, that is each monomial Q'_i depends on at most $\sqrt{d}$ variables. Therefore, by applying the product rule on $\partial_{\mathbf{x}^\gamma}(Q'_B)$, we know that this can be written as a linear combination of products of low-support polynomials. By Observation 22.3, every polynomial f of support at most r there is a unique multilinear polynomial $g = f \bmod \mathcal{I}$ of degree at most r. Hence, we get that

$$\begin{aligned}
\mathbf{x}^{=\ell}\cdot\partial_{\mathbf{x}^\alpha}(T) \bmod \mathcal{I} &\subseteq \operatorname{span}\left\{\mathbf{x}^{\le \ell+k(r-1)}\cdot\mathbf{x}^{\le k(r)}\cdot\tilde{Q}_{\overline{A}}\cdot Q'_{\overline{B}} \;:\; \begin{array}{c}\mathbf{x}^\alpha=\mathbf{x}^\beta\cdot\mathbf{x}^\gamma\\ |A|+|B|\le k\end{array}\right\} \bmod \mathcal{I}\\
&= \operatorname{span}\left\{\mathbf{x}^{\le \ell+kr}\cdot\tilde{Q}_{\overline{A}}\cdot Q'_{\overline{B}} \;:\; \begin{array}{c}\mathbf{x}^\alpha=\mathbf{x}^\beta\cdot\mathbf{x}^\gamma\\ |A|+|B|\le k\end{array}\right\} \bmod \mathcal{I}.
\end{aligned}$$

Therefore, an upper bound on $\left\{\mathbf{x}^{=\ell}\partial^{=k}(T) \bmod \mathcal{I}\right\}$ is

$$\binom{2(d/r)+1+(d/r)}{k}\cdot\binom{n}{\ell+kr}\cdot n$$

Hence, if $C = T_1 + \cdots + T_s$, then by sub-additivity we get

$$\square \qquad \Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}(C) \quad = \quad s\cdot\binom{3(d/r)+1}{k}\cdot\binom{n}{\ell+kr}\cdot n.$$

The bound above is almost the same as in Lemma 20.6 and the difference is only $\exp(O(d/r))$ due to the first binomial coefficient above. Lemma 22.4 and Lemma 20.23 yields a lower bound for $\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}(\mathrm{NW})$ as well.

What did we gain by looking at $\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}$ at the end of all this? The key point is makes it easier to look at the evaluation perspective for such measures when the ideal $\mathcal{I}$ *cooperates* with the evaluation operation. A concrete instance of this was the lower bound by Kumar and Saptharishi [KS17] for homogeneous depth five circuits over finite fields.

22.3 Lower bounds for depth five circuits over finite fields

The plan would be to combine the ideas from the lower bound of Grigoriev and Karpinski [GK98] (discussed in Chapter 10) with the above perspective of projected shifted partial derivatives. The main theorem of this chapter would be the result of Kumar and Saptharishi [KS17].

Theorem 22.6 ([KS17]). *Consider* $\mathrm{NW}_{d,m,e}$ *for some suitable choice of parameter. For any finite field* $\mathbb{F}_q$*, any homogeneous depth-*5 *circuit computing* $\mathrm{NW}_{d,m,e}$ *must be of size* $\exp(\Omega_q(\sqrt{d}))$.

The plan that one could adopt is the following.

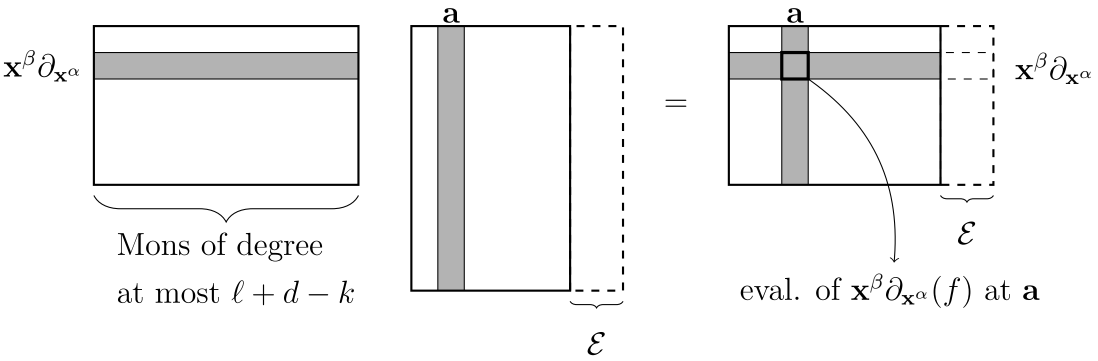
1. Take the measure to be $\Gamma^{\mathrm{PSD}}_{k,\ell,\mathcal{I}}$ where $\mathcal{I} = \langle x_i^q - x_i \;:\; i \in [n]\rangle$, which is the *evaluation perspective* of the shifted partial derivatives on $\mathbb{F}_q^n$.
2. As an analogue of “support” for the hom. $\Sigma\Pi\Sigma\Pi$ lower bounds, we shall look at the “rank” of products of linear polynomials computed by the bottom two layers of the circuit.
3. Intuitively, any term of rank at most r should essentially behave like a term of degree at most $(q-1)r$, as $\mathcal{I}$ reduces any variable of exponent q or above.

   Hence, the “low-rank” part of the circuit (only considering terms of rank at most $\sqrt{d}$) should behave like a homogeneous $\Sigma\Pi\Sigma\Pi^{[q\sqrt{d}]}$ circuit, for which we can upper bound the measure in a similar fashion.
4. To eliminate the “high-rank” products of linear polynomials, just as in the lower bound of Grigoriev and Karpinski [GK98] (as discussed in Chapter 10), these “high-rank” gates are zero on almost all evaluations. Therefore, we can modify our measure to not look at all evaluations on $\mathbb{F}_q^n$ but instead at $\mathbb{F}_q^n \setminus \mathcal{E}$ where $\mathcal{E}$ a tiny subset of $\mathbb{F}_q^n$ (of about $\exp(-\sqrt{d}) \cdot q^n$ size).
5. For the hard polynomial $\mathrm{NW}_{d,m,e}$, hopefully the measure continues to remain large even under the evaluation perspective on $\mathbb{F}_q^n \setminus \mathcal{E}$.

This is certainly a legitimate plan but the difficulty is in proving item 5. The calculations for the projected shifted partials were intricate enough, and it unclear how to convert those to the evaluation perspective with a few columns removed. Furthermore, it is important to keep the rough ballpark of parameters in mind. If we are hoping to have parameters for k and ℓ similar to those in Chapter 20, then we are dealing with matrices with roughly $2^n \cdot \exp(O(\sqrt{d} \log n))$ rows (indexed by derivatives of order $k = O(\sqrt{d})$ and shifts of order $\ell \approx n/2$) but close to q^n columns. It would have been fine if we actually had all q^n columns but we have to instead work with evaluations only on $\mathbb{F}_q^n \setminus \mathcal{E}$ for a set $\mathcal{E}$ of size $\exp(-\sqrt{d}) \cdot q^n \gg 2^n$.

In Chapter 10, we used the property that any linear combination of determinantal minors have many non-zero evaluations. Such a statement is simply false for this setting as we are dealing with polynomials that are multiplied by monomials of degree $\approx n/2$ and are hence going to be zero at many evaluations. Therefore, one needs a different strategy.

Idea 1: Since we are dealing with a matrix of evaluations, it can be naturally written as a product of a "coefficient matrix" and a "Vandermonde matrix".


$\mathbf{x}^\beta \partial_{\mathbf{x}^\alpha}$
Mons of degree
at most $\ell + d - k$
$\mathbf{a}$
$\mathcal{E}$
$=$
$\mathbf{a}$
$\mathbf{x}^\beta \partial_{\mathbf{x}^\alpha}$
$\mathcal{E}$
eval. of $\mathbf{x}^\beta \partial_{\mathbf{x}^\alpha}(f)$ at $\mathbf{a}$

Formally, let C be a matrix with rows indexed by shifted partials and columns indexed by monomials of the right degree, so that each row of C is just the coefficients of a shifted partial $\mathbf{x}^\beta \partial_{\mathbf{x}^\alpha}(f) \bmod \mathcal{I}$ listed down. Also, let V be the evaluation matrix of the monomials where rows are indexed by monomials and columns indexed by points in $\mathbb{F}_q^n$. Then the product $C \cdot V$ is precisely the evaluation matrix of shifted partials modulo $\mathcal{I}$. Furthermore, discarding a set $\mathcal{E}$ of evaluations is just discarding the corresponding columns from V.

In Chapter 20, we essentially showed that the matrix C for $\mathrm{NW}_{d,m,e}$ has pretty large row-rank. Furthermore, it is easy to show that V also has full-rank. Since $\mathcal{E}$ is a fairly small set, we should expect $\tilde{V}$, which is V with the columns indexed by $\mathcal{E}$ removed, to also have very large rank. Can we therefore claim that $C \cdot \tilde{V}$ also has large rank? In some sense, we certainly can.

Lemma 22.7 (Rank of products)**.** *For matrices $X \cdot Y = Z$, we always have that*

$$\operatorname{rank}(X) \quad \geq \quad \operatorname{rank}(Y) \; + \; \operatorname{rank}(Z) \; - \; (\text{common dimension})$$

where common dimension *denotes the number of columns of X or the number of rows of Y.*

Of course, if $\tilde{V}$ has full-rank then $\operatorname{rank}(M) = \operatorname{rank}(C)$. The problem is that if we are forced to look at a subset of evaluations and hence $\tilde{V}$ may not be full rank. When we are looking at all evaluations on $\mathbb{F}_q^n$, then the matrix V and $\tilde{V}$ has much more 2^n rows as row indices would also involve non-multilinear monomials. However C just has 2^n rows and hence they could potentially all be in the left-kernel of $\tilde{V}$ thus making $C \cdot \tilde{V} = 0$. Thus, we need a way of reducing the number of rows in V to about 2^n.

Idea 2: Do not consider evaluations on $\mathbb{F}_q^n$ but just evaluations on $\{0,1\}^n \subset \mathbb{F}_q^n$.

That way, it is equivalent to reducing all polynomials $\mathcal{I} = \langle x_i^2 - x_i \; : \; i \in [n]\rangle$ and hence the rows of V (or columns of C) would be indexed by only multilinear monomials of degree at most $\ell + d - k$, which is the same regime as in Chapter 20. Hence, we can hope to use Lemma 22.7 to lower bound the measure for $\mathrm{NW}_{d,m,e}$.

The problem now becomes item 4, the "high rank" gates. Consider the high-rank term $(x_1 + 2) \cdots (x_n + 2)$ over say $\mathbb{F}_7$. This term remains non-zero when we evaluate on any point in $\{0,1\}^n$.

Idea 3: Do not consider evaluations on $\{0,1\}^n$ but rather on a *shift* of $\{0,1\}^n$.

Picking a random point $\mathbb{F}_q^n$ can be thought of as first picking a point $\mathbf{c} \in \mathbb{F}_q^n$, and then picking a shift $\mathbf{x} \in \{0,1\}^n$ and returning $\mathbf{c} + \mathbf{x}$. Therefore, if we have an event that is *good* for a random point in $\mathbb{F}_q^n$, this also shows that there exists a $\mathbf{c} \in \mathbb{F}_q^n$ for which the event is *good* for a random point in $\mathbf{c} + \{0,1\}^n$.

Lemma 22.8. *Let A be an function on $\mathbb{F}_q^n$ and suppose $\mathbb{E}_{\mathbf{y} \in \mathbb{F}_q^n}[A(\mathbf{y})] \geq \delta$. Then, there exists a point $\mathbf{c} \in \mathbb{F}_q^n$ such that*

$$\mathop{\mathbb{E}}_{\mathbf{y} \in \mathbf{c} + \{0,1\}^n}[A(\mathbf{y})] \quad \geq \quad \delta$$

In this setting, if the event is the vanishing of all high-rank gates, we now have that there exists a *translate of a hypercube* $\mathbf{c} + \{0,1\}^n$ such all high rank gates vanish on most points in this translate of a hypercube.

We are now back to item 5, lower bounding the rank of the evaluation matrix for $\mathrm{NW}_{d,m,e}$. Since we are only dealing with a translate of a hypercube, we can finding a support-to-degree ideal $\mathcal{I}$ that respects evaluations on $\mathbf{c} + \{0,1\}^n$.

Lemma 22.9 ([KS17] Multilinearization for translate of hypercubes)**.** *Let* $\mathbf{c} + \{0,1\}^n$*. Then there exists a support-to-degree ideal* $\mathcal{I}$ *such that every polynomial* $f(\mathbf{x}) \in \mathbb{F}_q^n$ *has a unique multilinear representative* g *with* $\deg(g) \leq \deg(f)$ *that agrees with* f *on* $\mathbf{c} + \{0,1\}^n$*.*

Proof. Since each x_i can only take values c_i or $c_i + 1$, it suffices to replace each x_i^2 by either c_i^2 or $(c_i+1)^2$ depending on the value of x_i. It is easy to find a linear polynomial ℓ_i such that $\ell_i(c_i) = c_i^2$ and $\ell_i(c_i+1) = (c_i+1)^2$ — a simple calculation yields $\ell_i(x) = c_i^2 + (x - c_i)(2c_i+1)$. Hence, if we define $\mathcal{I}$ to be

$$\mathcal{I} \quad = \quad \left\{x_i^2 - c_i^2 - (x_i - c_i)(2c_i + 1) \ : \ i \in [n]\right\},$$

clearly this is a support-to-degree ideal (Definition 22.2) and by definition does not alter the evaluation on $\mathbf{c} + \{0,1\}^n$. □

With this multilinearization lemma, the coefficient matrix C has columns indexed by just multilinear monomials of degree at most $\ell + d - k$ which makes the dimensions C more balanced.

These are the main ideas that go into the proof of Kumar and Saptharishi [KS17] and from here on is just setting the right parameters etc. The rest of the exposition will leave a lot of the proofs as exercises to the readers as the main intuitions have been exposed.

The formal measure

For a set $\mathcal{A} \subseteq \mathbb{F}_q^n$, define the measure $\Gamma_{k,\ell,\mathcal{A}}^{[\mathrm{KS}]}$ as follows

$$\Gamma_{k,\ell,\mathcal{A}}^{[\mathrm{KS}]}(f) \quad = \quad \mathrm{rank}\left\{\left(\mathbf{x}^{=\ell} \cdot \partial^{=k}(f)\right)_{\mathbf{a} \in \mathcal{A}}\right\}$$

or in other words is the rank of the matrix where rows are indexed by shifts and partial derivatives, columns are indexed by elements $\mathcal{A}$ and the corresponding entry being the evaluation of the shifted partial derivative of f at the point $\mathbf{a}$.

22.3.1 Upper bound for a homogeneous depth-5 circuit

The upper bound would proceed in the following natural steps, similar to the discussion in 10.

Lemma 22.10 ([KS17] Eliminating high-rank gates)**.** *Let C be a homogeneous depth-5 circuit that computes an n-variate degree d polynomial over $\mathbb{F}_q$. Assume that* $\text{size}(C) \leq 2^{\sqrt{d}/100}$*. Let $\tau = q\sqrt{d}/6$ and $k = \tau/2q^3 = O_q(\sqrt{d})$. Then, there is a set $\mathcal{E}$ of size at most* $\exp -O(\sqrt{d}) \cdot q^n$ *such that all products of linear polynomials of rank greater than τ that are computed by the bottom to layers of C vanish on all of $\mathbb{F}_q^n \setminus \mathcal{E}$ with multiplicity at least k.*

Exercise 22.1 *Prove this.*

Lemma 22.11 ([KS17] Upper bound)**.** *Let C be a homogeneous depth-5 circuit of size at most $2^{\sqrt{d}/100}$ that computes an n-variate degree d polynomial over $\mathbb{F}_q$. Let $\tau, k, \mathcal{E}$ be as above. Then, for any $\mathcal{A} \subset \mathbb{F}_q^n \setminus \mathcal{E}$ that is contained in some $\mathbf{c} + \{0,1\}^n$, we have*

$$\Gamma^{[\mathrm{KS}]}_{k,\ell,\mathcal{A}}(C) \quad \leq \quad 2^{\sqrt{d}/100} \cdot \binom{\frac{4d}{\tau}+1}{k} \cdot \binom{n}{\ell + k\tau q} \cdot \text{poly}(n)$$

This proof would proceed in a few steps.

Claim 22.12. *Consider the polynomial C' which is obtained from C by* dropping all products of linear polynomials or rank at least τ. *Then, by the choice of $\mathcal{A}$,*

$$\Gamma^{[\mathrm{KS}]}_{k,\ell,\mathcal{A}}(C) \quad = \quad \Gamma^{[\mathrm{KS}]}_{k,\ell,\mathcal{A}}(C')$$

Claim 22.13. *If C' is a homogeneous depth-5 circuit with all products of linear forms having rank at most τ, and if $\mathcal{A}$ is contained in some translate of a hypercube $\mathbf{c} + \{0,1\}^n$, then*

$$\Gamma^{[\mathrm{KS}]}_{k,\ell,\mathcal{A}}(C') \quad = \quad \text{size}(C') \cdot \binom{\frac{4d}{\tau}+1}{k} \cdot \binom{n}{\ell + k\tau q} \cdot \text{poly}(n)$$

Exercise 22.2 *Complete the proofs using Observation 22.3* and *Lemma 22.9.*

22.3.2 Lower bound for $\mathrm{NW}_{d,m,e}$

As mentioned earlier, to lower bound the rank of the evaluation matrix we shall write this as $C \cdot V$ and use Lemma 22.7 to lower bound the rank. The matrices C and V shall be the following:

> **The matrix C:**
>
> Rows are indexed by shifted partials, columns are indexed by multilinear monomials of degree at most $\ell + d - k$, and the entry in $(\mathbf{x}^\alpha \cdot \partial_{\mathbf{x}^\beta}, m)$ is the coefficient of the monomial m in the unique multilinear representative of $\mathbf{x}^\alpha \cdot \partial_{\mathbf{x}^\beta}(f) \bmod \mathcal{I}$ (Lemma 22.9).
>
> **The matrix $V(\mathcal{A})$:**
>
> Rows are indexed by multilinear monomials of degree at most $\ell + d - k$, columns are indexed by points in $\mathcal{A}$, and the entry at $(m, \mathbf{a})$ is the evaluation of m at $\mathbf{a}$.

We shall set parameters in $\mathrm{NW}_{d,m,e}$ as we had in Lemma 20.23, then by Lemma 22.4, we have the following bound for the rank of C.

$$\begin{aligned} \operatorname{rank}(C) &\geq \binom{n}{\ell} \cdot (1+\varepsilon)^{2d-2k} \cdot \exp(-O(\log^2 n)) \\ &= \binom{n}{\ell + d - k} \cdot \exp(-O(\log^2 n)) \end{aligned} \tag{22.14}$$

Moving on to the rank of $V(\mathcal{A})$. Currently, $\mathcal{A}$ is contained in some translate $\mathbf{c} + \{0,1\}^n$. The following observation allows us to instead look at subsets of $\{0,1\}^n$, which is easier to study.

Observation 22.15. *For any set $\mathcal{A}$ and a point $\mathbf{c} \in \mathbb{F}_q^n$, we have*

$$\operatorname{rank}(V(\mathcal{A}) = \operatorname{rank}(V(\mathcal{A} - \mathbf{c})).$$

Exercise 22.3 *Prove this.*

Due to this observation, we might as well assume that $\mathcal{A} \subset \{0,1\}^n$. Suppose we consider the simpler matrix $\tilde{V}$ where we have a column for every $\mathbf{a} \in \{0,1\}^n$. How do we show that $\tilde{V}$ has full rank? This is because the matrix $\tilde{V}$ has an lower-triangular matrix sitting inside with ones on the diagonal. To elaborate a bit, if we were to order the rows by decreasing order of degrees, and match each row (monomial) by the characteristic vector (a point $\mathbf{a}$ of hamming weight at most $\ell + d - k$), then it is easy to see that this is a lower-triangular matrix with ones on the diagonal.

Due to this, if it so happens that $\mathcal{A}$ has many points from

$$\mathcal{H}_{\leq \ell+d-k} = \{\mathbf{a} \in \{0,1\}^n \ : \ \mathrm{wt}(\mathbf{a}) \leq \ell + d - k\},$$

then $\mathrm{rank}(V)$ would also be large. The point is that we only need to avoid the set $\mathcal{E}$ but we are free to choose any translate we want. The same averaging argument comes into play here to ensure that we have a large intersection with a translate of $\mathcal{H}_{\leq \ell+d-k}$.

Lemma 22.16. *Let $\mathcal{E} \subset \mathbb{F}_q^n$ be a set of size $\delta \cdot q^n$. Then, there exists $\mathbf{c} \in \mathbb{F}_q^n$ such that*

$$|(\mathbf{c} + \mathcal{H}_{\leq \ell+d-k}) \cap \mathcal{E}| \quad \leq \quad \delta \cdot |\mathcal{H}_{\ell+d-k}|$$

From this lemma, the rank bound follows.

Lemma 22.17. *For any set $\mathcal{E} \in \mathbb{F}_q^n$ of size at most $\delta \cdot q^n$, there exists a set $\mathcal{A}$ that is contained in some $\mathbf{c} + \{0,1\}^n$ for which*

$$\mathrm{rank}(V(\mathcal{A})) \quad \geq \quad \binom{n}{\ell + d - k} \cdot (1 - \delta)$$

Exercise 22.4 *Prove this.*

Putting all of this together yields the require lower bound for the rank of the evaluation matrix for $\mathrm{NW}_{d,m,e}$.

Lemma 22.18. *Let the parameters of $\mathrm{NW}_{d,m,e}$ be chosen appropriately. Then, for any set $\mathcal{E} \subset \mathbb{F}_q^n$ of size at most $\exp(-O(\sqrt{d})) \cdot q^n$, there exists a set $\mathcal{A}$ contained in $\mathbf{c} + \{0,1\}^n$ and is disjoint from $\mathcal{E}$ such that*

$$\Gamma^{[\mathrm{KS}]}_{k,\ell,\mathcal{A}}(\mathrm{NW}_{d,m,e}) \quad \geq \quad \binom{n}{\ell + d - k} \cdot \exp(-O(\log^2 n))$$

Combining this with 22.11, the main theorem (Theorem 22.6) of Kumar and Saptharishi [KS17] (almost[1]) follows.

[1] there is a slight technicality here that depending on F_q we choose k accordingly and hence we get that

"for each q there is a polynomial f for which the bound holds". This can be fixed to instead get the right order of quantifiers — "There is a polynomial f such that for every q ..."

Chapter 23

The power of non-homogeneous depth three circuits

A $\Sigma\Pi\Sigma$ circuit computes a polynomial of the form

$$f \quad = \quad \sum_{i=1}^{s} \ell_{i1} \dots \ell_{iD}.$$

If the circuit is non-homogeneous, the degree of the circuit D could potentially be much larger than $\deg(f)$.

The class of depth three arithmetic circuits can compute polynomials in non-trivial ways. To illustrate a couple of examples, there is a homogeneous $\Sigma\Pi\Sigma$ circuit for Perm_n of size $2^{O(n)}$ called Ryser's Formula [Rys63]

$$\mathsf{Perm}_n \quad = \quad \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \left(\sum_{j \in S} x_{ij} \right) \tag{23.1}$$

On the other hand, no $\Sigma\Pi\Sigma$ circuit for the Det_n significantly better than writing it as a sum of $n!$ monomials was known (until [GKKS16]). Further, the elementary symmetric polynomials $\mathsf{Sym}_k(x_1, \dots, x_n)$ of degree k defined as

$$\mathsf{Sym}_k(\mathbf{x}) \quad = \quad \sum_{\substack{S \subset \mathbf{x} \\ |S| = k}} \prod_{x_i \in S} x_i$$

can be computed by a non-homogeneous depth 3 circuit of size $O(n^2)$ over any characteristic zero field. In stark contrast, [NW97] showed that any homogeneous depth 3 circuit computing Sym_k requires size $n^{\Omega(k)}$. [NW97] also showed a $2^{\Omega(n)}$ lower bound for homogeneous depth 3 circuits computing Perm_n or Det_n.

Also, the results of [GR00, GK98] showed a $2^{\Omega(n)}$ lower bound for $\Sigma\Pi\Sigma$ circuits *over finite fields* that compute Det_n or Perm_n. All these results seemed to suggest that there perhaps is an $2^{\Omega(n)}$ lower bound for $\Sigma\Pi\Sigma$ circuits computing Det_n over characteristic zero fields as well. If it was true over finite fields, and for homogeneous $\Sigma\Pi\Sigma$ circuits, how much power can characteristic zero fields and non-homogeneity add? As it turns out, quite a lot!

Theorem 23.2 ([GKKS16])**.** *Let f be an n-variate degree d polynomial computed by an arithmetic circuit of size s over any characteristic zero field. Then there is a $\Sigma\Pi\Sigma$ circuit of size $s' \leq s^{O(\sqrt{d})}$ that computes f.*

Corollary 23.3 ([GKKS16])**.** *There is a $\Sigma\Pi\Sigma$ circuit over $\mathbb{Q}$, the field of rational numbers, of size $n^{O(\sqrt{n})}$.*

The proof is quite short and comprises of two steps using known reductions, and going through a bizarre intermediate model of *depth 5 powering circuits.* We present a longer route towards this result that perhaps sheds more light on the reduction.

23.1 Handling non-homogeneous depth-3 circuits

Non-homogeneous models are generally difficult to deal with in the context of lower bounds. A natural question to ask if any such non-homogeneous circuit can be converted to a suitable homogeneous model which may then be attacked. This was first studied by Shpilka and Wigderson [SW01].

Let f be a homogeneous degree d polynomial computed by a possibly non-homogeneous depth 3 circuit C of the form

$$f \quad = \quad \sum_{i=1}^{s} \ell_{i1} \dots \ell_{iD}$$

As a first step, let us extract the degree d homogeneous component of each summand on the RHS. Since f is a homogeneous degree d polynomial, f has to be sum of the degree d

homogeneous components of each summand on the RHS. Consider a single term of the form

$$T \quad = \quad (\ell_1 + \alpha_1) \cdots (\ell_D + \alpha_D)$$

where each ℓ_i is a homogeneous linear polynomial, and α are elements from the field. Assuming that the first r of the α_i's are zero, we can write T in the form (with some reuse of symbols)

$$\begin{aligned} T &= \alpha \cdot \ell_1 \ldots \ell_r \cdot (\ell_{r+1} + 1) \ldots (\ell_D + 1) \\ \implies \quad \mathrm{Hom}_d(T) &= \ell_1 \ldots \ell_r \cdot \mathsf{Sym}_{d-r}(\ell_{r+1}, \ldots, \ell_D) \end{aligned}$$

where $\mathsf{Sym}_k(\mathbf{x})$, the elementary symmetric polynomial of degree k defined as

$$\mathsf{Sym}_k(\mathbf{x}) \quad = \quad \sum_{\substack{S \subset \mathbf{x} \\ |S| = k}} \prod_{x_i \in S} x_i$$

Hence, if we can show that $\mathsf{Sym}_{d-r}(\mathbf{x})$ has a not-too-large homogeneous depth 4 circuit, then we can immediately infer that f can be computed by a not-too-large homogeneous depth 5 circuit. The following identities, attributed to Newton (cf. [Kal00]), is exactly what we need. Define the *power symmetric polynomials*, denoted by $\mathrm{Pow}_k(\mathbf{x})$ as

$$\mathrm{Pow}_k(\mathbf{x}) \quad = \quad \sum_{x_i \in \mathbf{x}} x_i^k$$

Lemma 23.4 (Newton Identities)**.** *Let* $\mathsf{Sym}_k(x_1, \ldots, x_m)$ *and* $\mathsf{Pow}_k(x_1, \ldots, x_m)$ *denote the* elementary symmetric *and* power symmetric *polynomials of degree k respectively, as defined above. Then,*

$$\mathsf{Sym}_k \quad = \quad \frac{1}{k!} \cdot \begin{vmatrix} \mathsf{Pow}_1 & 1 & 0 & \cdots & 0 & 0 \\ \mathsf{Pow}_2 & \mathsf{Pow}_1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ \mathsf{Pow}_{k-1} & \mathsf{Pow}_{k-2} & \mathsf{Pow}_{k-3} & \cdots & \mathsf{Pow}_1 & k-1 \\ \mathsf{Pow}_k & \mathsf{Pow}_{k-1} & \mathsf{Pow}_{k-2} & \cdots & \mathsf{Pow}_2 & \mathsf{Pow}_1 \end{vmatrix}.$$

Expanding the determinant on the RHS, we obtain a homogeneous expression

$$\mathsf{Sym}_k(\mathbf{x}) \quad = \quad \sum_{\mathbf{a}\,:\,\sum_i ia_i = k} \alpha_{\mathbf{a}} \cdot (\mathsf{Pow}_1)^{a_1} \dots (\mathsf{Pow}_k)^{a_k} \tag{23.5}$$

The number of summands bounded by the number of non-negative solutions to $\sum ia_i = k$, which is precisely the number of partitions of k. By the estimates of [HR18], we know that the number of partitions of k is bounded by $2^{\Theta(\sqrt{k})}$. Thus, (23.5) yields a homogeneous depth 4 circuit for $\mathsf{Sym}_k(x_1, \dots, x_m)$ of size $2^{\Theta(\sqrt{k})} \cdot m$. In fact, the circuit is a homogeneous $\Sigma\Pi\Sigma\wedge$ circuit, i.e. a $\Sigma\Pi\Sigma\Pi$ circuit where the bottom layer of multiplication in fact just raises a single variable to a higher power.

Let us summarizing this as a lemma.

Lemma 23.6 ([SW01])**.** *For every $d \leq n$, the elementary symmetric polynomial Sym_d can be computed by a homogeneous $\Sigma\Pi\Sigma\wedge$ circuit of size $2^{O(\sqrt{d})} \cdot \mathrm{poly}(n)$ over any field $\mathbb{F}$ of characteristic zero.*

Thus, if f is a homogeneous degree d polynomial computed by a (possibly non-homogeneous) depth-3 circuit C of size s over a field $\mathbb{F}$ of characteristic zero, then f can be equivalently computed by a $\Sigma\Pi\Sigma\wedge\Sigma$ of size at most $2^{O(\sqrt{d})} \cdot \mathrm{poly}(s)$.

23.2 Depth reduction to depth three circuits

In this section we shall see the proof of the depth reduction of [GKKS16]. As mentioned earlier, the proof is quite short but we shall give a slightly lengthier exposition that is close to how the result was discovered. This route via an attempt to prove better lower bounds for depth-4 circuits might be more insightful than the actual proof itself.

Towards proving better lower bounds for depth 4 circuits

From the depth reduction to depth-4 (Theorem 5.17), it suffices to prove a better lower bound for explicit polynomials computed as

$$f \quad = \quad \sum_{i=1}^{s} Q_{i1} \dots Q_{ir} \quad \text{where} \quad \deg(Q_{ij}) \leq \sqrt{d}\ ,\ r \leq O(\sqrt{d}) \tag{23.7}$$

A noble goal is to show a lower bound of $s = n^{\omega(\sqrt{d})}$. Perhaps a simpler question to ask is to prove a lower bound for expressions of the form

$$f \quad = \quad \sum_{i=1}^{s} Q_i^{\sqrt{d}} \quad \text{where } \deg(Q_i) \leq \sqrt{d} \tag{23.8}$$

Fortunately, if the goal is to prove lower bounds of $n^{\omega(\sqrt{d})}$, then without loss of generality we can focus on this equation instead.

Lemma 23.9. *Over any characteristic zero field, given an expression of the form*

$$f \quad = \quad \sum_{i=1}^{s} Q_{i1} \dots Q_{ir} \quad \textit{where} \quad \deg(Q_{ij}) \leq \sqrt{d}\ ,\ r \leq O(\sqrt{d})$$

there is an equivalent equation

$$f \quad = \quad \sum_{i=1}^{s'} Q_i^r \quad \textit{where} \quad \deg(Q_i) \leq \sqrt{d}$$

with $s' \leq s \cdot 2^{O(\sqrt{r})}$.

Proof. Consider Ryser's formula (23.1) applied for to the $r \times r$ matrix where each row is $[y_1, \dots, y_r]$.

$$\mathsf{Perm} \begin{bmatrix} y_1 & \dots & y_r \\ \vdots & \ddots & \vdots \\ y_1 & \dots & y_r \end{bmatrix} \quad = \quad r! \cdot y_1 \dots y_r \quad = \quad \sum_{S \subseteq [r]} (-1)^{r-|S|} \left(\sum_{j \in S} y_j \right)^r$$

The lemma follows by applying this identity on each term $Q_{i1} \dots Q_{ir}$. □

A very similar identity, to convert a product into sums of powers of linear polynomials, is often attributed to Fischer [Fis94]. We shall refer to this as the Ryser-Fischer trick.

Lemma 23.10 (Ryser-Fischer Trick)**.**

$$y_1 \dots y_r = \frac{1}{r!} \cdot \sum_{S \subseteq [r]} (-1)^{r-|S|} \left(\sum_{j \in S} y_j \right)^r$$

Note that since we need to divide by $r!$, the above lemma fails over low characteristic fields, in particular finite fields. Thus, proving an $n^{\omega(\sqrt{d})}$ lower bound for expressions such as

(23.8) implies an $n^{\omega(\sqrt{d})}$ lower bound for expressions such as (23.7). We shall call expressions such as (23.8) as $\Sigma\wedge\Sigma\Pi^{[\sqrt{d}]}$ circuits.

Just as we converted the top Π layer into powering layers using the Ryser-Fischer identity, the same can be done to the lower layer of Π gates as well.

Corollary 23.11. *If a homogeneous n-variate degree d polynomial f can be computed by a $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ of size $s = n^{O(\sqrt{d})}$, then f can also be computed by an $\Sigma\wedge^{[O(\sqrt{d})]}\Sigma\wedge^{[\sqrt{d}]}\Sigma$ circuit of size $s' = s \cdot 2^{O(\sqrt{d})}$.*

Conversely, if f requires $\Sigma\wedge^{[O(\sqrt{d})]}\Sigma\wedge^{[\sqrt{d}]}\Sigma$ circuits of size $s' = n^{\omega(\sqrt{d})}$ to compute it, then f cannot be computed by polynomial sized arithmetic circuits.

We shall take a small detour to apply this to the conversion of non-homogeneous depth 3 circuits to homogeneous shallow circuits.

Revisiting non-homogeneous depth 3 circuits

From Section 23.1, we know that any non-homogeneous $\Sigma\Pi\Sigma$ circuit can be converted to a homogeneous $\Sigma\Pi\Sigma\wedge\Sigma$ circuit, and this was essentially by writing elementary symmetric polynomial Sym_d has a homogeneous $\Sigma\Pi\Sigma\wedge$ circuit of size $2^{O(\sqrt{d})} \cdot \mathrm{poly}(n)$:

$$\mathsf{Sym}_d(\mathbf{x}) \quad = \quad \sum_{\mathbf{a}\,:\,\sum_i ia_i = d} \alpha_{\mathbf{a}} \cdot (\mathsf{Pow}_1)^{a_1} \dots (\mathsf{Pow}_d)^{a_d}$$

To convert this $\Sigma\Pi\Sigma\wedge$ circuit to a $\Sigma\wedge\Sigma\wedge$ circuit, we could use Ryser-Fischer's identity again. At first sight, it appears as though this would yield a blow up of 2^d as some of the product gates could have fan-in d. However, notice that the sum is over a_i's satisfying $\sum i \cdot a_i = d$. Hence, there can be at most $O(\sqrt{d})$ of the a_i's that are non-zero. By looking at Ryser-Fischer's identity applied on $y_1^{a_1} \dots y_d^{a_d}$ more carefully, we see that it uses at most $(1 + a_1)\dots(1 + a_d) \le d^{O(\sqrt{d})}$ distinct linear polynomials instead of the naïve bound of 2^d. This fact of expressing any degree d monomial over m variables as a $\Sigma\wedge\Sigma$ circuit of size $d^{O(m)}$ was also observed by Ellison [Ell69].

Thus, if f admits a poly-sized depth three circuit (possibly non-homogeneous), then f also admits a homogeneous $\Sigma\wedge\Sigma\wedge\Sigma$ circuit of size $d^{O(\sqrt{d})} \cdot \mathrm{poly}(n)$. The following lemma summarizes this discussion.

Lemma 23.12. *Let f be an n-variate degree d polynomial that is computable by depth three circuit of size s over $\mathbb{Q}$. Then, f is equivalently computable by a homogeneous $\Sigma\wedge\Sigma\wedge\Sigma$ circuit of size $d^{O(\sqrt{d})} \cdot \mathrm{poly}(s)$.*

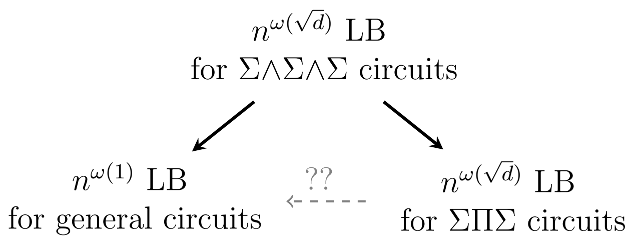

$n^{\omega(\sqrt{d})}$ LB
for $\Sigma\wedge\Sigma\wedge\Sigma$ circuits
$n^{\omega(1)}$ LB
for general circuits
??
$n^{\omega(\sqrt{d})}$ LB
for $\Sigma\Pi\Sigma$ circuits

Figure 23.1: Power of $\Sigma\wedge\Sigma\wedge\Sigma$ ckts.

Conversely, if f requires $\Sigma\wedge\Sigma\wedge\Sigma$ circuits of size $n^{\omega(\sqrt{d})}$ over $\mathbb{Q}$ to compute it, then f requires depth three circuits of size $n^{\omega(\sqrt{d})}$.

In fact, this bound can be improved further and we shall address this shortly.

Completing the picture

We now have an interesting situation (Figure 23.1). On one hand, Corollary 23.11 states that a lower bound of $n^{\omega(\sqrt{d})}$ for $\Sigma\wedge\Sigma\wedge\Sigma$ circuits would yield a super-polynomial lower bound for general arithmetic circuits. On the other, Lemma 23.12 states that an $n^{\omega(\sqrt{d})}$ lower bound for $\Sigma\wedge\Sigma\wedge\Sigma$ circuits would yield a lower bound of $n^{\omega(\sqrt{d})}$ for depth three circuits.

Could this just be a coincidence? Or, is it the case that any poly-sized arithmetic circuit can be equivalently expressed as a depth three circuit of size $n^{O(\sqrt{d})}$ over $\mathbb{Q}$? As it turns out, there is indeed a depth reduction to convert any arithmetic circuit to a not-too-large depth three circuit over $\mathbb{Q}$.

To complete the picture, it suffices to show that a $\wedge\Sigma\wedge$ circuit can be expressed as a $\Sigma\Pi\Sigma$ circuit. This would automatically imply a reduction from $\Sigma\wedge\Sigma\wedge\Sigma$ circuits to $\Sigma\Pi\Sigma$ circuits. The last step of the puzzle is the *duality trick* of [Sax08]. A similar version of this trick also appeared in the lower bound of Shpilka and Wigderson [SW01] but this statement is from the work of Saxena [Sax08].

Lemma 23.13 (The Duality Trick [Sax08]). *There exists univariate polynomials f_{ij}'s of degree at most b such that*

$$(z_1 + \cdots + z_s)^b \quad = \quad \sum_{i=1}^{sb+1} f_{i1}(z_1)\ldots f_{is}(z_s).$$

It is worth noting that the degree of each term on the RHS is sb, whereas the LHS just has degree b. This is the place where non-homogeneity is introduced. Applying the above lemma for a $\wedge\Sigma\wedge$ circuit such as $(y_1^a + \cdots + y_s^a)^b$ gives

$$\begin{aligned}(y_1^a + \cdots + y_s^a)^b &= \sum_{i=1}^{sb+1} \prod_{j=1}^{s} f_{ij}(y_j^a) \\ &= \sum_{i=1}^{sb+1} \prod_{j=1}^{s} \tilde{f}_{ij}(y_j)\end{aligned}$$

where $\tilde{f}_{ij}(y) = f_{ij}(y^a)$. Since each $\tilde{f}_{ij}(y)$ is a univariate polynomial, it can be factorized completely over the $\mathbb{C}$, the field of complex numbers. Hence, if $f_{ij}(y) = \prod_k (y - \zeta_{ijk})$, then we get

$$\begin{aligned}(y_1^a + \cdots + y_s^a)^b &= \sum_{i=1}^{sb+1} \prod_{j=1}^{s} \tilde{f}_{ij}(y_j) \\ &= \sum_{i=1}^{sb+1} \prod_{j=1}^{s} \prod_{k=1}^{b} (y_j - \zeta_{ijk})\end{aligned}$$

which is a depth three circuit! Thus, $(y_1^a + \cdots + y_s^a)$ can be expressed as a depth three circuit of size $\operatorname{poly}(s, a, b)$ over $\mathbb{C}$. With a little more effort, one can construct a depth three circuit over $\mathbb{Q}$ as well. Summarizing this is a lemma, we have the following.

Lemma 23.14. *Any n-variate degree d polynomial f computed by a homogeneous $\Sigma\wedge\Sigma\wedge\Sigma$ of size s over a characteristic zero field $\mathbb{F}$ can also be computed by a depth three circuit of size $\operatorname{poly}(s, n, d)$ over $\mathbb{F}$.*

Combining with Corollary 23.11 and Theorem 5.17, we obtain the main result of [GKKS16].

Theorem 23.2 (restated). *Let f be an n-variate degree d polynomial computed by an arithmetic circuit of size s over any characteristic zero field. Then there is a $\Sigma\Pi\Sigma$ circuit of size $s' \leq s^{O(\sqrt{d})}$ that computes f.*

Remark. Note that if we were to start with a degree d polynomial f and apply the above depth reduction, all the linear polynomials that we obtain at bottom are essentially from the application of Ryser-Fischer's identity on the bottom Π layer of fanin $\sqrt{d}$ of the $\Sigma\Pi^{[O(\sqrt{d})]}\Sigma\Pi^{[\sqrt{d}]}$ circuit. Hence, the each of the linear polynomials that appear in the final $\Sigma\Pi\Sigma$ circuit depend on at most $\sqrt{d}$ variables. In other words, the above Theorem yields a reduction to $\Sigma\Pi\Sigma^{[\sqrt{d}]}$ circuits.

23.3 Revisiting the depth-five powering circuit for Sym_d

As mentioned earlier, the conversion of a non-homogeneous depth-3 circuit to a homogeneous depth-5 circuit proceeds via the construction of Shpilka and Wigderson [SW01] for the elementary symmetric polynomial. We present a slightly more careful study of the resulting depth-5 circuit of this construction.

The first step in building the circuit for Sym_d was (23.5), to express it as

$$\mathsf{Sym}_d(\mathbf{x}) \quad = \quad \sum_{\mathbf{a}\,:\,\sum_i ia_i = d} \alpha_{\mathbf{a}} \cdot (\mathsf{Pow}_1)^{a_1} \dots (\mathsf{Pow}_d)^{a_d}.$$

In the next step, we used the Ryser-Fischer trick (Lemma 23.10) to replace the top ×-gate by a $\Sigma\wedge\Sigma$ circuit. We then observed that a monomial $y_1^{a_1}\cdots y_d^{a_d}$ can be expressed as a $\Sigma\wedge\Sigma$ circuit of top fan-in at most $(1+a_1)\cdots(1+a_d)$. Since we know that the above expression has $\sum i\cdot a_i = d$, at most $O(\sqrt{d})$ of the a_i's are non-zero and this gives a bound of $d^{O(\sqrt{d})}$.

However, one can very easily obtain a slightly better bound of $2^{O(\sqrt{d})}$ instead of $d^{O(\sqrt{d})}$.

Proposition 23.15. *If $a_1, \dots, a_d$ are non-negative integers with $\sum_{i=1}^d i\cdot a_i = d$, then*

$$(1+a_1)\cdots(1+a_d) \quad \leq \quad 2^{O(\sqrt{d})}$$

Proof. Without loss of generality, we may assume that the non-zero a_i's are the first r for some $r = O(\sqrt{d})$. Given $\sum i\cdot a_i = d$, we wish to infer a bound for $(1+a_1)\cdots(1+a_r)$ and a natural approach to do this is to use the AM-GM inequality somehow. Indeed,

$$\begin{aligned}
(1+a_1)+(2+2a_2)+\cdots+(r+ra_r) \quad &= \quad \frac{r(r+1)}{2}+d \;\leq\; 2d \\
\implies (1+a_1)\cdot(2+2a_2)\cdots(r+ra_r) \quad &\leq \quad \left(\frac{2d}{r}\right)^r \quad \text{(AM-GM inequality)} \\
\implies (1+a_1)\cdot(1+a_2)\cdots(1+a_r) \quad &\leq \quad \left(\frac{2d}{r}\right)^r\left(\frac{1}{r!}\right) \quad = \quad \left(\frac{d}{r^2}\right)^r\cdot\exp(r)
\end{aligned}$$

which is $2^{O(\sqrt{d})}$ for all $r = O(\sqrt{d})$. □

Corollary 23.16. *For every $d \leq n$, the elementary symmetric polynomial Sym_d can be computed by a homogeneous $\Sigma\wedge\Sigma\wedge$ circuit of size at most $2^{O(\sqrt{d})} \cdot \operatorname{poly}(n)$ over any field $\mathbb{F}$ of characteristic 0.*

Thus, specializing to non-homogeneous depth-3 circuits, we have the following immediate corollary.

Corollary 23.17. *Let f be computed by a (possibly non-homogeneous) depth-3 circuit C of size s. Then, for every $d \leq \deg(f)$, the d-th homogeneous part of f can be computed by a $\Sigma\wedge\Sigma\wedge\Sigma$ circuit of size $\exp(\sqrt{d}) \cdot poly(s)$.*

Further, any bound on the bottom fan-in of the circuit C translates to a similar bound on the bottom fan-in of the resulting $\Sigma\wedge\Sigma\wedge\Sigma$ circuit.

Corollary 23.16 in particular implies that the monomial $x_1 \cdots x_n$ has a $\Sigma\wedge\Sigma\wedge$ circuit of size $2^{O(\sqrt{n})}$. This is pretty surprising as we know of a $2^{\Omega(n)}$ lower bound for $\Sigma\wedge\Sigma$ circuits computing the monomial. Allowing higher powers at bottom layer might appear to be of no assistance in computing a multilinear monomial but surprisingly it does![1]

Open Problem 23.1

Show a lower bound on the size of an $\Sigma\wedge\Sigma\wedge$ circuit computing the monomial $x_1 \cdots x_n$.

[1] This observation is by Michael Forbes.

Chapter 24

Depth three circuits of low bottom fan-in

Kayal and Saha [KS15a] show that the technique of projected shifted partial derivatives can also be used to prove lower bounds for subclasses of non-homogeneous depth three circuits, namely depth three circuits with *bounded bottom fan-in*. We shall denote the class of depth three circuits of bottom fan-in bounded by r as $\Sigma\Pi\Sigma^{[r]}$ circuits. The ideas involved have also been useful in addressing depth-4 circuits with “low-arity” [KS16b, KS15b] that we shall see later.

24.1 $\Sigma\Pi\Sigma$ circuits with bottom fan-in $O(\sqrt{d})$

Now let us focus on $\Sigma\Pi\Sigma^{[r]}$ circuits, where all linear polynomials in the circuit depend on at most r variables. The following is the key observation of Kayal and Saha [KS15a] and can be verified easily from the proof of Lemma 23.6.

Observation 24.1 ([KS15a])**.** *Starting with a $\Sigma\Pi\Sigma^{[r]}$ circuit C of size s computing a homogeneous n-variate polynomial of degree d, the resulting $\Sigma\Pi\Sigma\wedge\Sigma$ circuit C' obtained from Lemma 23.6 is in fact a $\Sigma\Pi\Sigma\wedge\Sigma^{[r]}$ circuit of size $s' = \operatorname{poly}(s) \cdot 2^{O(\sqrt{d})}$.*

Thus, by expanding the all powers of linear polynomials computed in the bottom two layers of the $\Sigma\Pi\Sigma\wedge\Sigma^{[r]}$ circuit C', the circuit C' can be rewritten as a homogeneous depth 4 circuit of bottom support bounded by r and size $s'' \leq s' \cdot d^r$

We would like to combine this with Lemma 20.23 but the only catch is the additional $d^r = d^{\sqrt{d}}$ cost we incur. Had that been $2^{O(\sqrt{d})}$, we would have been fine as we are hoping to prove something like a $n^{\sqrt{d}/1000}$ lower bound. And so far, we were dealing with $\mathrm{NW}_{d,m,e}$ with $n \approx d^3$ and in this regime $d^{\sqrt{d}} = n^{(1/3)\sqrt{d}}$. But if we look back at the constraints for $\mathrm{NW}_{d,m,e}$, we only needed to ensure that $m^e \approx 2^d$, where the total number of variables $n = md$. Fortunately, there is enough freedom to make d much much smaller than m but still ensure that $m^e = 2^d$ by reducing e appropriately. Hence for any $\varepsilon > 0$, with a suitable trade-off between d and n, we can always make sure that $d^{\sqrt{d}}$ is much smaller than $n^{\varepsilon\sqrt{d}}$.

This immediately the main theorem of [KS15a].

Theorem 24.2 ([KS15a])**.** *Over any characteristic zero field $\mathbb{F}$, any $\Sigma\Pi\Sigma^{[r]}$ circuit C computing the polynomial $\mathrm{NW}_{d,m,e}$, for suitably chosen parameters n and d with $n = d^{O(1)}$, must have size $s = n^{\Omega(d/r)}$.*

24.2 $\Sigma\Pi\Sigma$ circuits with bottom fan-in $n^{1-\varepsilon}$

Kayal and Saha [KS15a] also prove lower bounds for depth three circuits where the bottom fan-in is bounded away from n by any polynomial factor.

Theorem 24.3 ([KS15a])**.** *Let $\varepsilon > 0$ be any constant. Then over any characteristic zero field, there exists an explicit polynomial P such that any $\Sigma\Pi\Sigma^{[n^{1-\varepsilon}]}$ circuit computing P must have size $n^{\Omega_\varepsilon(\sqrt{d})}$.*

We shall work with $\varepsilon = 0.1$ to save on some variables. All the ideas here can be made to work for any $\varepsilon > 0$.

As a first step, we shall start with a $\Sigma\Pi\Sigma^{[n^{0.9}]}$ circuit and use Observation 24.1 to convert it to a homogeneous $\Sigma\Pi\Sigma\wedge\Sigma^{[n^{0.9}]}$ circuit computing the same polynomial. As we have just seen, if the fan-in of all the linear polynomials at the bottom were instead $O(\sqrt{d})$, then we can directly apply Observation 24.1 and use Theorem 20.11 to prove the lower bound via projected shifted partial derivatives. What we would like to do is reduce to this case somehow, and a natural approach is to use random restrictions.

Attempt 1

We have linear forms of size $n' = n^{0.9}$ and we want to use a random restriction to reduce keep only $d' = \sqrt{d}$ of the the n' variables alive and set the rest to zero. The natural choice is to keep a variable alive with probability $p < \frac{d'}{n'}$ and use Chernoff's bound. Hopefully the error probability is low enough and we can union bound over all linear forms in the circuit.

Lemma 24.4 (Chernoff's Bound)**.** *Let $X_1, \ldots, X_m$ be independent $\{0,1\}$ random variables with $\Pr[X_i = 1] = p$ for all i. Then, if $X = \sum X_i$ and $\mu = \mathbb{E}[X]$, for any $\delta > 0$, we have the following bounds:*

$$\Pr[X > (1+\delta)\mu] \quad \leq \quad \begin{cases} e^{-\delta^2\mu/3} & \text{if } \delta < 1 \\ e^{-\delta\mu/3} & \text{if } \delta > 1 \end{cases}$$

Hence, in the regime we are interested in, we would like $(1+\delta)\mu = d'$ where $\mu = pn'$ if p is the probability with which we keep a variable alive. If $\delta < 1$, or in other words $p \approx \frac{d'}{n'}$, then this error term cannot be better than $\exp(-d') = \exp(-\sqrt{d})$ and this is not sufficient for the union bound over all gates as we have $\exp(\Omega(\sqrt{d}\log n))$ of them.

The other possibility is that we choose $p \ll \frac{d'}{n'}$ but choose δ large enough so that $(1+\delta)\mu = d'$. However in this regime, $\delta > 1$ and hence $\delta\mu = O(d')$. Once again Lemma 24.4 only gives an error of $\exp(-O(d'))$. Seems like we are unable to use Chernoff's bound to reduce to the case when the bottom fan-in is $\sqrt{d}$. Kayal and Saha [KS15a] do a two-step random restriction process to make handle this differently, but let us spend a moment thinking about this as this should be intuitively weird.

The plan was to set p small enough so that the $\Pr[X > \sqrt{d}]$ becomes $\exp(-\Omega(\sqrt{d}\log n))$. Certainly as we decrease p, this probability must go down but this is somehow not seen from Lemma 24.4 as the error seems stuck at $\exp(-O(d'))$. What this shows is that the bounds provided by Lemma 24.4 are not tight enough to work in the regime when μ is very small. Fortunately, there are better bounds known and we shall use that. (This is different from the original analysis of Kayal and Saha [KS15a].)

24.2.1 Stronger Chernoff Bounds

The following statement is the tightest bound we know for the Chernoff bounds. We are interested in understanding a sum of m independent, identically distributed $\{0,1\}$ random variables, and say $\Pr[X_i = 1] = p$. Suppose we are interested in the event that $X = \sum X_i > (p+\varepsilon)m$. Intuitively, such an event makes it seem like $\Pr[X_i = 1] = p + \varepsilon$ rather than p. Thus one should expect that the probability of this event happening should be related to the *distance between* the distributions $\Pr[X_i = 1] = p$ and $\Pr[X_i = 1] = p + \varepsilon$. Indeed, the following bound formalizes this by using the *relative entropy* or *KL-divergence* as the distance measure.

Lemma 24.5 (Chernoff's bound via relative entropy)**.** *Let $X_1, \ldots, X_m$ be independent, identically distributed $\{0,1\}$ random variables with $\Pr[X_i = 1] = p$ and let $X = \sum X_i$. For any $p' > p$, we have*

$$\Pr[X > p'm] \quad \leq \quad e^{-m \cdot \mathbb{D}(p' \| p)}$$

where $\mathbb{D}(\alpha\|\beta)$ is the relative entropy *or* KL-divergence *between the distributions $\Pr[X_i = 1] = \alpha$ and $\Pr[X_i = 1] = \beta$ defined as*

$$\mathbb{D}(\alpha\|\beta) \quad := \quad \alpha \cdot \log\left(\frac{\alpha}{\beta}\right) \; + \; (1-\alpha)\log\left(\frac{1-\alpha}{1-\beta}\right).$$

All the usual bounds for Chernoff are essentially obtained by approximating the relative entropy term in some way. We shall use this formulation to analyze the random restriction process in a single shot. A few simplifications are in order.

Claim 24.6. *For any $0 < \beta < \alpha < 1/2$,*

$$0 \geq (1-\alpha)\log\left(\frac{1-\alpha}{1-\beta}\right) \geq -2\alpha.$$

Proof. First note that since $\beta < \alpha$, we have $1 \geq \frac{(1-\alpha)}{(1-\beta)}$ and hence its logarithm is negative.

$$\begin{aligned}
(1-\alpha)\log\left(\frac{1-\alpha}{1-\beta}\right) &\geq \log\left(\frac{1-\alpha}{1-\beta}\right)\\
&= \log(1-\alpha) - \log(1-\beta)\\
&\geq \log(1-\alpha) \qquad (\because \beta < 1)\\
&= -\alpha - \frac{\alpha^2}{2} - \frac{\alpha^3}{3}\cdots\\
&\geq -\alpha - \alpha^2 - \alpha^3\cdots\\
&\geq -2\alpha \qquad\qquad\qquad \Box
\end{aligned}$$

Thus effectively, in the definition of $\mathbb{D}(\alpha\|\beta)$, the dominant term is the first term when $\beta \ll \alpha$.

Corollary 24.7. *For any $0 < \beta < \alpha < 1/2$, then*

$$\mathbb{D}(\alpha\|\beta) \quad \geq \quad \alpha\left(\log\left(\frac{\alpha}{\beta}\right) - 2\right).$$

In particular, if $\beta \ll \alpha$, the negative 2 above is not relevant as it is dominated by the growing function $\log(\alpha/\beta)$ so we shall drop that for simplicity.

Let's get back to the setting we were interested in. We have $n' = n^{0.9}$ variables, each kept alive with some probability p. The goal was to find a suitable p so that the probability more than $d' = \sqrt{d}$ among the n' are kept alive is at most $\exp(-\Omega(\sqrt{d}\log n))$. If $p'n' = d'$, then

$$\begin{aligned} \Pr[X > d'] &\geq \exp\left(-n'\left(p'\log\left(\frac{p'}{p}\right)\right)\right) \\ &= \exp\left(-d'\left(\log(p'/p)\right)\right) \end{aligned}$$

Therefore, all we need to do is to choose p so that $\log(p'/p) = \Omega(\log n)$ and we would are done! We summarize this as a lemma, since we'd use this in the next chapter as well.

Lemma 24.8. *Let $S_1, \ldots, S_r$ be subsets of $[n]$ of size at most $n^{0.9}$ each and suppose $r \leq n^{0.01\sqrt{d}}$. If we pick a set $R \subseteq [n]$ by choose every element independently with probability $\frac{\sqrt{d}}{n^{0.92}}$, then*

$$\Pr[\textit{For all } i,\ |S_i \cap R| \leq \sqrt{d}] \quad = \quad 1 - o(1).$$

Therefore, as long as the size of the $\Sigma\Pi\Sigma^{[n^{0.9}]}$ circuit is not too large, we can apply a random restriction with probability $p = \frac{\sqrt{d}}{n^{0.92}}$ and reduce to the case of $\Sigma\Pi\Sigma^{[\sqrt{d}]}$ circuits.

24.2.2 Building the hard polynomial

The task now is to build a hard polynomial P with a suitable trade-off between the number of variables and its degree so that *even after* a random restriction ρ_ε for say $\varepsilon = n^{-0.99}$, the polynomial $\rho_\varepsilon(P)$ has a large dimension of projected shifted partial derivatives. We can once again use the NW polynomial family and make it robust to such projections using the linear blow-up trick (Lemma 20.5).

Fix some parameter d and choose m, e as in Lemma 20.23 so that $\mathrm{NW}_{d,m,e}$ has nearly maximal dimension of projected shifted partial derivatives. Now consider the hard polynomial to be $\mathrm{NW}_{d,m,e} \circ \mathrm{Lin}$ obtained by replacing each variable of $\mathrm{NW}_{d,m,e}$ by a sum of d^{1000} fresh variables. Hence we are now in a setting where we have an n-variate polynomial of degree d with $n \geq d^{1000}$.

Now suppose this polynomial is computed by a $\Sigma\Pi\Sigma^{[n^{0.9}]}$ circuit. Applying ρ_ε, we get that $\rho_\varepsilon(\mathrm{NW}_{d,m,e} \circ \mathrm{Lin})$ is computed by a $\Sigma\Pi\Sigma^{[\sqrt{d}]}$ circuit.

$$\Sigma\Pi\Sigma\Pi^{[2\sqrt{d}]} + (\text{non-multiquadratic}).$$

Since $\rho_\varepsilon(\mathrm{NW}_{d,m,e})$ has a copy of $\mathrm{NW}_{d,m,e}$ sitting inside, by setting additional variables to zero, we now have a circuit of the same form as above computing $\mathrm{NW}_{d,m,e}$ and we have our lower bound from Lemma 20.6 and Lemma 20.23:

$$\begin{aligned}
\Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Sigma\Pi\Sigma^{[\sqrt{d}]}\right) &\leq \Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Sigma\Pi\Sigma\Pi^{[\sqrt{d}]}\right) + \Gamma_{k,\ell}^{[\mathrm{PSD}]}(\text{non-multiquadratic}) \\
&= \Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Sigma\Pi\Sigma\Pi^{[\sqrt{d}]}\right) \\
&\ll \Gamma_{k,\ell}^{[\mathrm{PSD}]}(\mathrm{NW}_{d,m,e}).
\end{aligned}$$

This completes the proof of Theorem 24.2. □

Exercise 24.1 *Use these techniques to prove a similar lower bound for hom.* $\Sigma\Pi\Sigma\Pi\Sigma^{[n^{0.9}]}$ *circuits.*

Remark. *In the paper of Kayal and Saha [KS15a], the focus was on the* NW *polynomial as their proof used the calculations seen in Theorem 20.11 which are tailored to* NW*. Bera and Chakrabarti [BC15] instead gave a lower bound for* IMM *but for hom.* $\Sigma\Pi\Sigma\Pi\Sigma^{[n^{0.5-\varepsilon}]}$ *circuits and showed an* $n^{\Omega(\sqrt{d})}$. ◊

Chapter 25

Depth four circuits of low arity

Subsequent to the Kayal and Saha [KS15a] lower bound for non-homogeneous depth $\Sigma\Pi\Sigma^{[n^{0.9}]}$ with small bottom fan-in, this was almost simultaneously generalized by two independent works of Kumar and Saraf [KS16b] and Kayal and Saha [KS15b].

25.1 The model of computation

Definition 25.1 (Low arity depth-4 circuits). *We say that a polynomial f can be computed by an* arity-r depth-4 circuit *of size S if it can be written as*

$$f \quad = \quad \sum_{i=1}^{s} \prod_{j=1}^{t} Q_{ij}$$

where $S \leq s \cdot t$ and each Q_{ij} is an arbitrary polynomial on just r variables.

We shall use $\Sigma\Pi\circledast^{[r]}$ to refer to such computations, where the $\circledast^{[r]}$ stands for an arbitrary polynomial on r variables. $\Diamond$

Clearly, $\Sigma\Pi\Sigma^{[r]}$ circuits are a special case of $\Sigma\Pi\circledast^{[r]}$ circuits. In the above definition, we stress that the Q_{ij}s are *any* polynomials and can even be of exponential degree. The only constraint the model imposes is that they depend on only r variables.

The main theorem of Kumar and Saraf [KS16b] and Kayal and Saha [KS15b] is the following.

Theorem 25.2 ([KS16b, KS15b]). *Assume that the characteristic of the base field is* 0 *(or large enough). For every $\varepsilon > 0$, there is an explicit n-variate degree d homogeneous* *polynomial $P \in \mathsf{VNP}$ such that any $\Sigma\Pi\circledast^{[n^{1-\varepsilon}]}$ circuit computing it must have size at least $n^{\Omega_\varepsilon(\sqrt{d})}$.*

The rest of the chapter, we shall see a proof of this. Once again, we shall work with $n^{0.9}$ instead of $n^{1-\varepsilon}$ as that would have all the ideas and save us some notation.

Reducing to $\sqrt{d}$-arity circuits

Let us assume that the circuit is at most $n^{0.01\sqrt{d}}$. As seen in the previous chapter, we can always use a random restriction by setting each variable independently to zero with probability $1 - \frac{\sqrt{d}}{n^{0.92}}$ to reduce the arity of such a circuit to $\sqrt{d}$ via Lemma 24.8. Hence, it suffices to work with just the case of $\sqrt{d}$-arity circuits.

25.2 Warm-up: Small product fan-in case

As a first step, let us consider a simpler setting where the fan-in of the Π layer is bounded by d, the degree of the polynomial. That is, we have an expression of the form

$$f = \sum_{i=1}^{s} Q_{i1} \cdots Q_{id}$$

where each Q_{ij} is an arbitrary polynomial on $\sqrt{d}$ variables. (If the Q_{ij}s were linear, then we are looking at the case of *homogeneous* $\Sigma\Pi\Sigma$ circuits as a warm-up). We would like to show that for an explicit polynomial f any expression such as the one above must have $s = n^{\Omega(\sqrt{d})}$. Let us expand each Q_{ij} as a sum of monomials, and let Q'_{ij} be the multiquadratic part of Q_{ij}. Hence,

$$f = \sum_{i=1}^{s} Q'_{i1} \cdots Q'_{id} + \text{(non-multiquadratic)}$$

However, since Q'_{ij} depends on just $\sqrt{d}$ variables and is multiquadratic, its degree can be at most $2\sqrt{d}$. Thus the first term above is a $\Sigma\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]}$ circuit. Therefore,

$$\Gamma_{k,\ell}^{[\mathrm{PSD}]}(f) \leq \Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Sigma\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]}\right)$$

If we can show that the RHS is not too large, we would be done. Repeating the standard calculations from Lemma 20.6, it can be seen that

$$\Gamma_{k,\ell}^{[\mathrm{PSD}]}(f) \quad \leq \quad s \cdot \binom{d}{k} \cdot \binom{n}{\ell + k\sqrt{d}}$$

The only difference from the expression in Lemma 20.6 is the first $\binom{d}{k}$ term which was $\binom{O(\sqrt{d})}{k}$ earlier. But nevertheless we would be choosing $k \leq \sqrt{d}$, we know that $\binom{d}{k} = d^{\sqrt{d}}$. Since we are hoping for a lower bound something like $n^{0.01\sqrt{d}}$, we can always make $n \gg d$ in order to ensure that $d^{\sqrt{d}} \ll n^{0.01\sqrt{d}}$. Therefore, this first term is a lower order term that won't really affect the calculations.

Since the second term is essentially the same as in Lemma 20.6, we can use Lemma 20.23 to get

$$\Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]}\right) \ll \Gamma_{k,\ell}^{[\mathrm{PSD}]}(\mathrm{NW}_{d,m,e})$$

giving us the lower bound on s.

25.3 Taming the product fan-in

It is useful to keep the $\Sigma\Pi\Sigma^{[r]}$ case at the back of our minds. How were we able to control the Π-fanin there? There, we use Sym to *extract* the d-th homogeneous part from the $\Sigma\Pi\Sigma$ circuit and expressed that as a homogeneous $\Sigma\Pi\Sigma\wedge\Sigma$ circuit. We shall do something similar here. For any polynomial P and positive integer i, recall that $\mathrm{Hom}_i(P)$ refers to the i-th homogeneous component of P and let $\mathrm{Hom}_{\leq i}(P)$ refer to the sum of the homogeneous parts of degree up to i of P.

Let's focus on one term $T = Q_1 \cdots Q_t$. Say the first a of the Q_is have a zero constant term. As for the rest, we may assume that the constant term is one by scaling T appropriately. Hence, we have an expression of the form

$$T \quad = \quad Q_1 \cdots Q_a \cdot (1 + Q_1') \cdots (1 + Q_t')$$

Firstly, note that if $a > d$, then $\mathrm{Hom}_d(T) = 0$ as all monomials in the RHS have degree more than d. Hence we can assume that $a \leq d$.

Claim 25.3. *If $T = Q_1 \cdots Q_a \cdot (1 + Q'_1) \cdots (1 + Q'_t)$, then*

$$\mathrm{Hom}_d(T) \quad = \quad \mathrm{Hom}_d\left(Q_1 \cdots Q_a \cdot \sum_{i=0}^{d} \mathsf{Sym}_i(Q'_1, \ldots, Q'_t)\right).$$

Proof. The only monomials present in T that are not in $Q_1 \cdots Q_a \cdot \sum_{i=0}^{d} \mathsf{Sym}_i(Q'_1, \ldots, Q'_t)$ are monomials that are obtained by multiplying more than d of the Q'_is. But such monomials must have degree more than d and hence cannot contribute to $\mathrm{Hom}_d(T)$. □

Therefore if T is a $\Pi\circledast^{[\sqrt{d}]}$ circuit, since each $\mathsf{Sym}_i(y_1, \ldots, y_t)$ can be expressed as a $\Sigma\Pi^{[d]}\Sigma\wedge$ circuit of size at most $2^{O(\sqrt{d})}\,\mathrm{poly}(t, d)$ (Lemma 23.6), we have that $\mathrm{Hom}_d(T)$ can be expressed as the d-th homogeneous part of a $\Sigma\Pi^{[d]}\Sigma\wedge\circledast^{[\sqrt{d}]}$ of size at most $2^{O(\sqrt{d})}\,\mathrm{poly}(t, d)$, which is of course also a $\Sigma\Pi^{[d]}\Sigma\circledast^{[\sqrt{d}]}$ circuit (by absorbing the powering gate). The factor of $2^{O(\sqrt{d})}$ is affordable as we are hoping to prove a $n^{\Omega(\sqrt{d})}$ lower bound.

$$\mathrm{Hom}_d(T) \quad = \quad \mathrm{Hom}_d\left(\Sigma\Pi^{[d]}\Sigma\circledast^{[\sqrt{d}]}\right).$$

As we are only interested in the degree d homogeneous part, we might as well assume that each of the $\circledast^{[\sqrt{d}]}$ computations are polynomials of degree at most d, as the higher degree monomials cannot contribute to $\mathrm{Hom}_d(T)$. Thus, we have that $\mathrm{Hom}_d(T)$ is the homogeneous degree d part of a $\Sigma\Pi^{[d]}\Sigma\circledast^{[\sqrt{d}]}$ circuit of formal degree at most d^2.

We have already seen earlier that the homogeneous parts of any low-degree circuit can be computed via interpolation (Lemma 5.4) Therefore, there is a $\Sigma\Pi^{[d]}\Sigma\circledast^{[\sqrt{d}]}$ circuit of size at most $2^{O(\sqrt{d})} \cdot \mathrm{poly}(d, t)$ computing $\mathrm{Hom}_d(T)$. Therefore, if f is a homogeneous degree d polynomial computed by a $\Sigma\Pi\circledast^{[\sqrt{d}]}$ circuit of size s, then by extracting the homogeneous parts of degree d from each summand we have

$$f \quad = \quad \sum_{i=1}^{s} \mathrm{Hom}_d(T_i) \quad \in \quad \Sigma\Pi^{[d]}\Sigma\circledast^{[\sqrt{d}]},\ \text{size } 2^{O(\sqrt{d})}\,\mathrm{poly}(d, t).$$

Just as we did in the warm-up case, we can now split each $\circledast^{[\sqrt{d}]}$ computation as its multi-quadratic part and the non-multiquadratic part to get an expression of the form

$$f \quad = \quad \Sigma\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]} \quad + \quad (\text{non-multiquadratic}).$$

Hence using Lemma 20.23 and Lemma 20.6,

$$\begin{aligned}\Gamma_{k,\ell}^{[\mathrm{PSD}]}\left(\Sigma\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]} + \text{non-multiquadratic}\right) &\leq s \cdot 2^{O(\sqrt{d})} \cdot \mathrm{poly}(d,t) \\ &\quad \cdot \ \Gamma_{k,\ell}^{[\mathrm{PSD}]}\Pi^{[d]}\Sigma\Pi^{[2\sqrt{d}]} \\ &\ll \Gamma_{k,\ell}^{[\mathrm{PSD}]}(\mathrm{NW}_{d,m,e})\end{aligned}$$

unless $s \cdot t = n^{\Omega(\sqrt{d})}$. Therefore any $\Sigma\Pi\circledast^{[\sqrt{d}]}$ circuit computing $\mathrm{NW}_{d,m,e}$ must have size at least $n^{\Omega(\sqrt{d})}$.

Furthermore, using Lemma 20.5, any $\Sigma\Pi\circledast^{[n^{0.9}]}$ circuit computing $\mathrm{NW}_{d,m,e} \circ \mathrm{Lin}$ must have size at least $n^{\Omega(\sqrt{d})}$, and that completes the proof of Theorem 25.2. □

Chapter 26

Arithmetic circuits with locally low algebraic rank

We have been studying depth four circuits over the last few chapters. One of the goals has been to handle some amount of non-homogeneity while assuming other structure on the circuit. The main reason for this quest is that proving a lower bound of $n^{\omega(d^{1/3})}$ for *non-homogeneous* depth four circuit would suffice to separate VP and VNP.

In the last chapter, we looked at $\Sigma\Pi\Sigma\Pi$ circuits that were non-homogeneous while we imposed the restriction that the $\Sigma\Pi$ layer closer to the leaves computed a polynomial on few variables. We use $\circledast^{[n^{0.9}]}$ to refer to such polynomials and proved lower bounds for such $\Sigma\Pi\circledast^{[n^{0.9}]}$ circuits.

In this chapter we shall study restrictions on the Π layer closer to the root. Until now, we were somehow reducing to the case of $\Sigma\Pi^{[d]}\Sigma\Pi$, where the top layer of Π gates multiply *at most d* polynomials together. As a warm-up to the more general models that we shall study in this chapter, let us spend a few moments studying $\Sigma\circledast^{[d]}\Sigma\Pi$ circuits.

26.1 Preliminaries

26.1.1 Warm-up: Compositions of sparse polynomials

Definition 26.1 ($\Sigma\circledast^{[d]}\Sigma\Pi$ circuits)**.** *We shall say a polynomial f is computed by a $\Sigma\circledast^{[d]}\Sigma\Pi$ circuit of size s it can be expressed as*

$$f \quad = \quad \sum_i H_i(Q_{i1}, \ldots, Q_{id}),$$

where each $H_i(y_1, \ldots, y_d)$ is an arbitrary polynomial, and the sum of the sparsity of the Q_{ij}s is at most s. ♢

Can projected shifted partial derivatives be used to prove lower bounds for this model? Yes, indeed. The first step, as before, would be to use a random restriction to ensure that each Q_{ij} is a sparse polynomial of *low support size.* But for simplicity, let us assume for now that $\deg(Q_{ij}) \leq \sqrt{d}$ and see how the upper bound calculations work.

Consider a single term $H(Q_1, \ldots, Q_d)$. What can we say about any k-th order partial derivative of this? By the chain-rule of differentiation,

$$\partial_x H(Q_1, \ldots, Q_d) \quad = \quad \sum_{i=1}^{d} \left(\partial_{y_i}(H)\right)(Q_1, \ldots, Q_d) \cdot \partial_x(Q_i).$$

Hence repeating this, it is easy to see that

$$\begin{aligned} \partial^{=k} H(\mathbf{Q}) \quad &\subseteq \quad \mathbb{F}\text{-span}\left\{(\partial^{=k}H)(\mathbf{Q}) \cdot \partial_{m_1}(Q_{i_1}) \cdots \partial_{m_k}(Q_{i_k}) \ : \ \deg(m) = k\right\} \\ &\subseteq \quad \mathbb{F}\text{-span}\left\{(\partial^{=k}H)(\mathbf{Q}) \cdot \mathbf{x}^{\leq k(\sqrt{d}-1)}\right\} \\ \implies \mathbf{x}^{=\ell}\partial^{=k} H(\mathbf{Q}) \quad &\subseteq \quad \mathbb{F}\text{-span}\left\{(\partial^{=k}H)(\mathbf{Q}) \cdot \mathbf{x}^{\leq \ell + k(\sqrt{d}-1)}\right\} \end{aligned}$$

The key point is that, irrespective of how high the degree of H is, there are at most $\binom{d+\sqrt{d}}{d}$ distinct partial derivatives of H. Hence, the upper bound is effectively the same value as seen earlier for homogeneous depth-4 circuits such as in Lemma 20.6. Thus it is pretty clear that projected shifted partials would give an $n^{\Omega(\sqrt{d})}$ lower bound here.

Observation 26.2. *Assume we are working over a characteristic zero field. There exists an explicit polynomial in* VP *(namely* IMM *with appropriate parameters) such that any $\Sigma\circledast^{[d]}\Sigma\Pi$ circuit computing it requires size $n^{\Omega(\sqrt{d})}$.*

Kumar and Saraf [KS16a] studied a generalization of this model by looking at what they called 'locally low algebraic rank' depth four circuits and showed an $n^{\Omega(\sqrt{d})}$ lower bound for such circuits over fields of characteristic zero. Subsequently, Pandey, Saxena and Sinhababu [PSS18] extended it to arbitrary fields. We will need a bit of background on algebraic rank to describe the model and we do that first.

26.1.2 Algebraic Rank

We are familiar with the notion of linear dependence between polynomials, which is to say that there is a linear combination of the polynomials that is zero. A natural extension of this notion is algebraic independence.

Definition 26.3 (Algebraic rank)**.** *A set of polynomials* $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_t\} \subseteq \mathbb{F}[\mathbf{x}]$ *is said to be algebraically independent over* $\mathbb{F}$ *if there is no nonzero polynomial* $R \in \mathbb{F}[y_1, \ldots, y_t]$ *such that* $R(Q_1, \ldots, Q_t)$ *is identically zero.*

A maximal subset of $\mathbf{Q}$ *which is algebraically independent is said to be a transcendence basis of* $\mathcal{Q}$ *and the size of such a set is said to be the algebraic rank of* $\mathbf{Q}$*, denoted by* $\mathrm{algRank}(\mathbf{Q})$. ♢

It is a non-trivial observation that all maximal algebraically independent sets have the same size and hence the notion of *algebraic rank* is indeed well-defined.

If a set of t polynomials is algebraically dependent, then the above definition says that there is a non-zero polynomial in t variables over the underlying field, which vanishes when composed with this set. Such a polynomial is called an annihilating polynomial of this set. The first basic property of algebraic rank is that it is upper bounded the number of variables. We leave this as an exercise with a hint.

Exercise 26.1 [Upper bound on algebraic rank] *Show that any set of* $n+1$ *polynomials over* n *variables have some algebraic dependency. In other words, the algebraic rank of any set of polynomials is upper bounded by the number of variables.*

Hint: $(D+1)^{n+1} > \binom{nD+n}{n}$ *if* D *is large enough.*

It is natural question to ask if one can show good upper bounds on the lowest degree of an annihilating polynomial of a given set of polynomials. The following lemma of Kayal shows such a bound which would be useful to us later on.

Lemma 26.4 (Kayal [Kay09])**.** *Let $\mathbb{F}$ be a field and let $\mathbf{Q} = (Q_1, Q_2, \ldots, Q_t)$ be a set of polynomials of degree d in n variables over the field $\mathbb{F}$ having algebraic rank k. Then there exists a $\mathbf{Q}$-annihilating polynomial of degree at most $(k+1) \cdot d^k$.*

Given a set of polynomials, can its algebraic rank be computed efficiently? A natural approach is to search for an annihilating polynomial but as seen in the lemma above, the degree could be very large making this infeasible. In fact, Kayal [Kay09] showed that computing even the constant term of the annihilator is #P-hard. However, there is a fantastic result of Jacobi from the 1800s that gives a criterion to check if a set of polynomials is algebraically dependent, over fields of characteristic zero.

Lemma 26.5 (Jacobian Criterion)**.** *Let $Q_1, \ldots, Q_t \in \mathbb{F}[x_1, \ldots, x_n]$ be polynomials over a field $\mathbb{F}$ of characteristic zero. Then, the algebraic rank of the set $\{Q_1, \ldots, Q_t\}$ is equal to the rank of the following matrix, called* the Jacobian of $\mathbf{Q}$*, interpreted over the function field $\mathbb{F}(x)$:*

$$\mathcal{J}(Q_1, \ldots, Q_t) \quad := \quad \begin{bmatrix} \partial_1(Q_1) & \partial_2(Q_1) & \cdots & \partial_n(Q_1) \\ \partial_1(Q_2) & \partial_2(Q_2) & \cdots & \partial_n(Q_2) \\ \vdots & \vdots & \ddots & \vdots \\ \partial_1(Q_t) & \partial_2(Q_t) & \cdots & \partial_n(Q_t) \end{bmatrix}$$

Hence, we have the following randomized algorithm to compute the algebraic rank of a given set of polynomials — compute the Jacobian of the given set of polynomials, evaluate it at a random point of $\mathbb{F}^n$ and find its rank. It is a simple exercise to see that by the Schwartz-Zippel lemma, the rank of the Jacobian evaluated at a random point on $\mathbb{F}^n$ is equal to the rank of the matrix over the function field.

Although the above lemma is only over characteristic zero fields, Pandey, Saxena and Sinhababu [PSS18] modified the criterion to work over any field. The statement is a little technical to explain here and for simplicity we shall work only over characteristic zero fields here.

We are now ready to describe the model of computation studied by Kumar and Saraf [KS16a].

26.1.3 Locally low algebraic rank circuits

Intuitively, if we have a set of polynomials $\{Q_1, \ldots, Q_t\}$ with algebraic rank at most r, then morally this set *behaves* like a set of just r polynomials. In the case of linear independence, any composition on a set of polynomials of rank at most r can be interpreted as a composition on just r polynomials. Motivated by this, Kumar and Saraf [KS16a] study the following class of circuits.

Definition 26.6. *Let $\mathbb{F}$ be any field. A $\Sigma\circledast^{\{\!\{r\}\!\}}\Sigma\Pi$ circuit C in n variables over $\mathbb{F}$ is a representation of an n variate polynomial as*

$$C = \sum_{i=1}^{T} H_i(Q_{i1}, Q_{i2}, \ldots, Q_{it})$$

where H_i is an arbitrary polynomial and for each $i \in [T]$, $\mathrm{algRank}\{Q_{ij} \;:\; j \in [t]\} \leq r$.

The size of such a circuit will denote the sum of the number of monomials of each Q_{ij} (the complexity of the composition H_i is irrelevant to the size). $\Diamond$

The symbol $\circledast^{\{\!\{r\}\!\}}$ is to denote that we have an arbitrary composition of polynomials with algebraic rank bounded by r. In the paper of Kumar and Saraf [KS16a], they use the notation $\Sigma\Gamma^{(r)}\Sigma\Pi$ to denote such circuits but we shall use the above notation just to maintain consistency with the previous chapters.

The first thing to note is that if we look at the class of $\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi$ circuits, then clearly includes the class of homogeneous $\Sigma\Pi\Sigma\Pi$ circuits is a subclass of them where each H_i is just a product of at most d polynomials. Thus the above model is a vast generalization of the class of homogeneous depth-4 circuits. Kumar and Saraf [KS16a] show that even for this more general model, projected shifted partial derivatives can prove an $n^{\Omega(\sqrt{d})}$ lower bound.

Theorem 26.7 ([KS16a, PSS18]). *Let $\mathbb{F}$ be any field of characteristic zero. There exists a family $\{P_d\}$ of polynomials in* VNP*, such that P_d is a polynomial of degree d in $n = d^{O(1)}$ variables, and for any $\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi$ circuit C that computes computes P_d over $\mathbb{F}$ must have size $n^{\Omega(\sqrt{d})}$.*

26.2 Lower bounds for locally low algebraic rank circuits

We begin with some intuition for why we can expect to prove lower bounds for this model via projected shifted partial derivatives. We have already seen in Subsection 26.1.1 that projected shifted partial derivatives can be used to give lower bounds for $\Sigma\circledast^{[d]}\Sigma\Pi$ circuits.

So the question is if we can somehow go from a $\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi$ circuit to a $\Sigma\circledast^{[d]}\Sigma\Pi$ circuit. Let us look at the case of linear rank to get some intuition.

Suppose we have a polynomial $H(Q_1, \ldots, Q_t)$ with $\dim\{Q_1, \ldots, Q_t\} \leq r$. Then, there exists some r of the Q_is such that every other Q_i can be written as a linear combination of these r. Therefore, $H(Q_1, \ldots, Q_t)$ can be re-written as some $H'(Q_{i_1}, \ldots, Q_{i_r}) \in \circledast^{[r]}\Sigma\Pi$.

The main point is that for linear dependence, any polynomial Q that is linearly dependent on $Q_1, \ldots, Q_t$ can be expressed as a linear combination of them. Can we do the same thing for algebraic independence? Unfortunately no, for a silly reason. Consider the set $\{x, x^2\}$. Clearly, the polynomial x is algebraically dependent on x^2. However, any $x \neq H(x^2)$ for any polynomial H.

Nevertheless, Kumar and Saraf [KS16a] that such a Q can infact be "expressed" as a polynomial combination of the Q_is under a looser sense. The following lemma is key to their lower bound.

Lemma 26.8 (Algebraic dependence to functional dependence)**.** *Let $\mathbb{F}$ be any field of characteristic zero or sufficiently large characteristic. Let $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ be a set of algebraically independent polynomials in n. Let Q be a polynomial of degree at most d such that Q is algebraically dependent on $\mathbf{Q}$. Then, for most random[1] $\mathbf{a} \in \mathbb{F}^n$, there exists a polynomial F on r variables such that*

$$Q(\mathbf{x}+\mathbf{a}) \quad = \quad \operatorname{Hom}_{\leq d}\left(F(Q_1(\mathbf{x}+\mathbf{a}), Q_2(\mathbf{x}+\mathbf{a}), \ldots, Q_r(\mathbf{x}+\mathbf{a}))\right).$$

Revisiting the earlier example of $\{x, x^2\}$, while it is true that $x \neq F(x^2)$ for any polynomial F, we nevertheless have

$$(x+a) \quad = \quad \operatorname{Hom}_{\leq 1}\left(\frac{(x+a)^2}{2a} + \frac{a}{2}\right)$$

as a valid equality for all $a \neq 0$.

We shall defer the proof of this theorem to the end of the chapter and see how this can be used. Recall that since we are working with a $\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi$ circuit, we would always have $r \leq d$ in the above lemma. Let's try to see if we can remove the $\operatorname{Hom}_{\leq d}$ operation at a small cost. Since $Q(\mathbf{x}+\mathbf{a})$ is a polynomial of degree at most d, we would like to collect all terms from the RHS of degree at most d. This seems difficult as written as each of $Q_i(\mathbf{x}+\mathbf{a})$ is

[1] Here random $\mathbf{a}$ means an $\mathbf{a}$ chosen from a large enough grid in $\mathbb{F}^n$. The size of this grid depends on the degrees of the polynomials

non-homogeneous and the even very high degree monomials of F when evaluated on these non-homogeneous polynomials could yield lower degree terms. But we shall keep in mind that we do not really care about the complexity of the composition F as long as we can show that it is a composition of *few* polynomials.

Lemma 26.9. *Let $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r\}$ be a set of algebraically independent polynomials in n. Let Q be a polynomial of degree at most d such that Q is algebraically dependent on $\mathbf{Q}$. Then, for a random $\mathbf{a} \in \mathbb{F}^n$, there exists a polynomial $\tilde{F}$ in $r(d+1)$ variables*

$$Q(\mathbf{x}+\mathbf{a}) \quad = \quad \tilde{F}\left(Q_1^{(0)}, \ldots, Q_1^{(d)}, \ldots, Q_r^{(0)}, \ldots, Q_r^{(d)}\right)$$

where each $Q_i^{(j)} = \mathrm{Hom}_j(Q_i(\mathbf{x}+\mathbf{a}))$.

Proof. From Lemma 26.8, we have a we have a polynomial F so that

$$Q(\mathbf{x}+\mathbf{a}) \quad = \quad \mathrm{Hom}_{\leq d}\left(F(Q_1(\mathbf{x}+\mathbf{a}), \ldots, Q_r(\mathbf{x}+\mathbf{a}))\right). \tag{26.10}$$

Define a polynomial $F'(y_1^{(0)}, \ldots, y_1^{(d)}, \ldots, y_r^{(0)}, \ldots, y_r^{(d)})$ by

$$F'(y_1^{(0)}, \ldots, y_1^{(d)}, \ldots, y_r^{(0)}, \ldots, y_r^{(d)}) \quad := \quad F(y_1^{(0)} + \cdots + y_1^{(d)}, \ldots, y_r^{(0)} + \cdots + y_r^{(d)})$$

It is trivial to observe that

$$Q(\mathbf{x}+\mathbf{a}) \quad = \quad \mathrm{Hom}_{\leq d}\left(F'(Q_1^{(0)}, \ldots, Q_1^{(d)}, \ldots, Q_r^{(0)}, \ldots, Q_r^{(d)})\right) \tag{26.11}$$

replacing $Q_i(\mathbf{x}+\mathbf{a})$ by $\mathrm{Hom}_{\leq d}(Q_i(\mathbf{x}+\mathbf{a}))$ does not affect any monomial of degree at most d in (26.10). Seems like we haven't done much but the advantage is that all the inputs to F' in the above equation are homogeneous polynomials. A monomial $(y_1^0)^{e_{1,0}} \cdots (y_r^{(d)})^{e_{r,d}}$ of F' can contribute a term of degree at most d in (26.11) if and only if

$$\sum_{\substack{1 \leq i \leq r \\ 0 \leq j \leq d}} (j \cdot e_{i,j}) \quad \leq \quad d.$$

Therefore, if $\tilde{F}$ is the sum of all monomials of F' satisfying $\sum_{i,j}(j \cdot e_{i,j}) \leq d$, then

$$\begin{aligned} \tilde{F}(Q_1^{(0)}, \ldots, Q_1^{(d)}, \ldots, Q_r^{(0)}, \ldots, Q_r^{(d)}) \quad &= \quad \mathrm{Hom}_{\leq d}\left(F'(Q_1^{(0)}, \ldots, Q_1^{(d)}, \ldots, Q_r^{(0)}, \ldots, Q_r^{(d)})\right) \\ &= \quad Q(\mathbf{x}+\mathbf{a}) \end{aligned}$$

□

Corollary 26.12. *Let* $\mathbf{Q} = \{Q_1, \ldots, Q_t\}$ *be a set of polynomials of degree at most* d *satisfying* $\operatorname{algRank}(\mathbf{Q}) = r$ *and let* $\{Q_1, \ldots, Q_r\}$ *be a transcendence basis. For any arbitrary composition* $H(Q_1, \ldots, Q_t)$*, for almost all* $\mathbf{a} \in \mathbb{F}^n$*, we have*

$$H(Q_1(\mathbf{x}+\mathbf{a}), \ldots, Q_t(\mathbf{x}+\mathbf{a})) \quad = \quad H'\left(Q_1^{(0)}, \ldots, Q_1^{(d)}, \ldots, Q_r^{(0)}, \ldots, Q_r^{(d)}\right)$$

for some polynomial $H' \in \mathbb{F}[y_1^{(0)}, \ldots, y_r^{(d)}]$ *with each* $Q_i^{(j)} = \operatorname{Hom}_j(Q_i(\mathbf{x}+\mathbf{a}))$*.* □

With this corollary we are almost done. The only catch is that we need to look at $Q_i(\mathbf{x}+\mathbf{a})$ and unfortunately translates of sparse polynomials are not sparse. But on the other hand, if we knew something more on the structure of the Q_is, say a bound on its degree or a bound on the support size of all monomials, then we get the same bound for $Q_i(\mathbf{x}+\mathbf{a})$ as well. Once again, using a random restriction to set each variable independently to zero, we can assume that all monomials computed at the lowest level of a $\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi$ circuit have support size at most $\sqrt{d}$. Hence the overall structure would be the following:

$$C \in \Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi \quad \overset{\text{random restr.}}{\Longrightarrow} \quad C \in \begin{array}{c}\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi,\text{ with}\\ \sqrt{d}\text{-bottom support size}\end{array}$$

$$\Longrightarrow \; C(\mathbf{x}+\mathbf{a}) \in \begin{array}{c}\Sigma\circledast^{\{\!\{d\}\!\}}\Sigma\Pi,\text{ with}\\ \sqrt{d}\text{-bottom support size}\end{array} \quad \overset{\text{Corollary 26.12}}{\subseteq} \quad \begin{array}{c}\Sigma\circledast^{[d(d+1)]}\Sigma\Pi,\text{ with}\\ \sqrt{d}\text{-bottom support size}\end{array}$$

Therefore by Observation 26.2,

$$\Gamma_{k,\ell}^{[\mathrm{PSD}]}(C(\mathbf{x}+\mathbf{a})) \quad \leq \quad \Gamma_{k,\ell}^{[\mathrm{PSD}]}(C(\mathbf{x})) \quad \ll \quad \Gamma_{k,\ell}^{[\mathrm{PSD}]}(\mathrm{NW}_{d,m,e})$$

unless of course the size of C is $n^{\Omega(\sqrt{d})}$ giving us the lower bound. (Once again, we would have to use Lemma 20.5 to make $\mathrm{NW}_{d,m,e}$ robust to random restrictions). This completes the proof of Theorem 26.7, assuming Lemma 26.8. □(Theorem 26.7)

We only need to finish the proof of Lemma 26.8 and we shall do that in the rest of this chapter.

26.2.1 Proof of Lemma 26.8

We have an algebraically independent set of polynomials $\mathbf{Q} = \{Q_1, \ldots, Q_r\}$ and a polynomial Q of degree at most d that is algebraically dependent on it. In other words, there is a non-zero *annihilator* $A(y_1, \dots, y_r, z)$ such that

$$A(Q_1, \dots, Q_r, Q) \quad = \quad 0.$$

Let us assume that A is the smallest degree annihilator. We can say a few things about the polynomial $A'(\mathbf{x}, z) := A(Q_1, \dots, Q_r, z)$.

$$A(Q_1, \dots, Q_r, z) \quad =: \quad A'(\mathbf{x}, z) \quad = \quad A_0(\mathbf{Q}) + A_1(\mathbf{Q})z + \cdots + A_D(\mathbf{Q})z^D$$

Firstly, it must depend on z since otherwise $A(Q_1, \dots, Q_r, 0) = 0$ contradicts the assumption that $\mathbf{Q}$ was algebraically independent. Therefore, $A'(\mathbf{x}, z) = A(Q_1, \dots, Q_r, z)$ is a non-zero polynomial with $A'(\mathbf{x}, Q) = 0$. Hence, $(z - Q)$ must divide the polynomial $A'(\mathbf{x}, z)$.

Given that Q is a *root* of $A'(\mathbf{x}, z)$, can we express Q as a polynomials in the coefficients of A' (which are in turn polynomials in $\mathbf{Q}$)? The following beautiful lemma of Dvir, Shpilka and Yehudayoff [DSY09] shows that we indeed can, under some mild non-degeneracy condition.

Lemma 26.13 (Dvir, Shpilka, Yehudayoff [DSY09])**.** *For a field $\mathbb{F}$, let $P \in \mathbb{F}[\mathbf{x}, z]$ be a non-zero polynomial of degree at most D in z. Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial such that $P(\mathbf{x}, f) = 0$ and $\partial_z P(\mathbf{0}, f(\mathbf{0})) \neq 0$. If*

$$P(\mathbf{x}, z) \quad = \quad \sum_{i=0}^{D} P_i(\mathbf{x}) \cdot z^i.$$

Then, for every $t \geq 0$, there exists a polynomial G such that

$$\mathrm{Hom}_{\leq t}[f(\mathbf{x})] \quad = \quad \mathrm{Hom}_{\leq t}[G_t(P_0, P_1, \dots, P_D)].$$

For now, let us assume this lemma and finish the proof of Lemma 26.8. We would like to use Lemma 26.13 on the polynomial $A'(\mathbf{x}, z)$. The only thing to be checked is that the non-degeneracy condition $\partial_z A'(\mathbf{0}, Q(\mathbf{0})) \neq 0$.

There are a couple of reasons why this may fail in general. One of them is if it so happens that $(z - Q)^2$ divides $A'(\mathbf{x}, z)$, that is Q is a *repeated root* of $A'(\mathbf{x}, z)$. In this particular instance, $\partial_z A'(\mathbf{x}, Q)$ is identically zero. Can this happen in our setting?

First, observe that $A''(\mathbf{x}, z) := \partial_z A'(\mathbf{x}, z) = \partial_z A(Q_1, \dots, Q_r, z)$ is not identically zero, as we knew that $A(Q_1, \dots, Q_r, z)$ must depend on the variable z. But then if $A''(\mathbf{x}, Q) \equiv 0$, then $\partial_z A(y_1, \dots, y_r, z)$ is a smaller degree polynomial such that exhibits an algebraic dependency among $\{Q_1, \dots, Q_r, Q\}$ contradicting our choice of A.

Thus we can assume that $A''(\mathbf{x}, Q) \not\equiv 0$. However, $A''(\mathbf{0}, Q(\mathbf{0}))$ could now become zero despite $A''(\mathbf{x}, Q)$ being a non-zero polynomial[2]. This is where the shifts come in. If for most random $\mathbf{a} \in \mathbb{F}^n$, we know that $A''(\mathbf{a}, Q(\mathbf{a})) \neq 0$. We can now shift everything by the point $\mathbf{a}$.

Let $\tilde{Q}_i(\mathbf{x}) := Q_i(\mathbf{x} + \mathbf{a})$ and $\tilde{Q} = Q(\mathbf{x} + \mathbf{a})$. Therefore, we also have

$$A(\tilde{Q}_1, \ldots, \tilde{Q}_r, \tilde{Q}) = 0.$$

Thus if $\tilde{A}'(\mathbf{x}, z) = A(\tilde{Q}_1, \ldots, \tilde{Q}_r, z)$, we know that $(z - \tilde{Q})$ divides $\tilde{A}'$. Furthermore, $\partial_z \tilde{A}'(\mathbf{0}, \tilde{Q}(\mathbf{0})) = \partial_z A'(\mathbf{a}, Q(\mathbf{a})) \neq 0$. Thus, all the conditions of Lemma 26.13 are met and we have the proof of Lemma 26.8. □(Lemma 26.8)

Proof of Lemma 26.13. Let $\partial_z P(\mathbf{0}, f(\mathbf{0})) = \varepsilon_0 \neq 0$. The proof would be an induction on t. For the case of $t = 0$, we can just set $G_0(P_0, \ldots, P_D)$ to be the constant $f(\mathbf{0})$ and hence the base case is done.

Let's assume that we have found a polynomial $g = G_t(P_0, \ldots, P_D)$ such that

$$\mathrm{Hom}_{\leq t}[f] \quad = \quad \mathrm{Hom}_{\leq t}[g].$$

If $\mathrm{Hom}_{\leq t+1}[f] = \mathrm{Hom}_{\leq t+1}[g]$, then we already have our inductive step and there is nothing to be done. Hence let's assume that $\mathrm{Hom}_{t+1}[f] \neq \mathrm{Hom}_{t+1}[g]$ and therefore every monomial in $f - g$ has degree at least $t + 1$.

$$\begin{aligned}
0 \quad = \quad P(\mathbf{x}, f) \quad &= \quad \sum_i P_i \cdot f^i \\
&= \sum_i P_i \cdot (g + (f - g))^i \\
&= \sum_i P_i \cdot g^i \quad + \quad \left(\sum_i P_i \cdot (i g^{i-1})\right) \cdot (f - g) \quad + \quad (\deg > (t+1) \text{ terms}) \\
&= P(\mathbf{x}, g) \quad + \quad (\partial_z P(\mathbf{x}, g)) \cdot (f - g) \quad + \quad (\deg > (t+1) \text{ terms}) \\
&= P(\mathbf{x}, g) \quad + \quad \varepsilon_0 \cdot (f - g) \quad + \quad (\deg > (t+1) \text{ terms})
\end{aligned}$$

where the last equation is because every monomial of $f - g$ has degree at least $t + 1$ and the constant term of $\partial_z P(\mathbf{x}, g) = \partial_z P(\mathbf{0}, g(\mathbf{0})) = \partial_z P(\mathbf{0}, f(\mathbf{0})) = \varepsilon_0$. Hence, by rearranging

[2] This would happen for example if $(z - Q)(z - Q')$ divides $A'(\mathbf{x}, z)$ with $Q \neq Q'$ but $Q(\mathbf{0}) = Q'(\mathbf{0})$

terms,

$$\mathrm{Hom}_{\leq t+1}[f] \quad = \quad \mathrm{Hom}_{\leq t+1}\left[g \quad - \quad \frac{P(\mathbf{x}, g)}{\varepsilon_0}\right].$$

And of course, if g can be expressed as a polynomial combination of $P_0, \ldots, P_D$, then so can $g - \frac{P(\mathbf{x},g)}{\varepsilon_0}$ and that completes the inductive step and hence the proof of Lemma 26.13. □

26.3 Functional dependence to algebraic dependence

In Lemma 26.8, we saw that we can convert an algebraic dependence between polynomials to some sort of a functional dependence. It was observed by Pandey, Saxena and Sinhababu [PSS18] that a converse of Lemma 26.8 is also true. Next we outline a simple proof of this over fields of characteristic zero using the Jacobian described in Lemma 26.5.

Lemma 26.14 (Functional dependence to algebraic dependence)**.** *Let $\mathbb{F}$ be any field of characteristic zero. Let $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_r, Q_{r+1}\}$ be a set of algebraically independent polynomials. Then for almost all $\mathbf{a}$,*

$$\nexists F \quad \text{such that} \quad \mathrm{Hom}_{\leq 1}\left[Q_{r+1}(\mathbf{x}+\mathbf{a})\right] \quad = \quad \mathrm{Hom}_{\leq 1}\left[F(Q_1(\mathbf{x}+\mathbf{a}), \ldots, Q_r(\mathbf{x}+\mathbf{a}))\right].$$

Proof. By the Jacobian criterion of Lemma 26.5, the Jacobian $\mathcal{J}(Q_1, \ldots, Q_{r+1})$ has rank $r+1$ over the function field $\mathbb{F}(\mathbf{x})$.

$$\mathcal{J}(Q_1, \ldots, Q_{r+1}) \quad := \quad \begin{bmatrix} \partial_1(Q_1) & \partial_2(Q_1) & \cdots & \partial_n(Q_1) \\ \partial_1(Q_2) & \partial_2(Q_2) & \cdots & \partial_n(Q_2) \\ \vdots & \vdots & \ddots & \vdots \\ \partial_1(Q_{r+1}) & \partial_2(Q_{r+1}) & \cdots & \partial_n(Q_{r+1}) \end{bmatrix}$$

By the Schwartz-Zippel lemma, for almost all $\mathbf{a} \in \mathbb{F}^n$, the above matrix evaluated at $\mathbf{a}$ continues to be full rank. Fix any such $\mathbf{a}$. Now consider the polynomial $Q_{r+1}(\mathbf{x}+\mathbf{a})$ and collect all the degree one terms. The coefficient of x_i in $Q_{r+1}(\mathbf{x}+\mathbf{a})$ is precisely $\partial_i Q_{r+1}(\mathbf{a})$.

Similarly, let us collect the degree one terms in any composition $F(Q_1(\mathbf{x}+\mathbf{a}), \ldots, Q_r(\mathbf{x}+\mathbf{a}))$. The coefficient of x_i is precisely

$$\sum_{j=1}^{r} (\partial_j F)(Q_1(\mathbf{a}), \ldots, Q_r(\mathbf{a})) \cdot \partial_i Q_j(\mathbf{a}).$$

If $\mathrm{Hom}_{\leq 1}\left[Q_{r+1}(\mathbf{x}+\mathbf{a})\right] = \mathrm{Hom}_{\leq 1}\left[Q_1(\mathbf{x}+\mathbf{a}), \ldots, Q_r(\mathbf{x}+\mathbf{a})\right]$, then we would have the following matrix identity

$$\begin{bmatrix} \partial_1 Q_{r+1}(\mathbf{a}) & \cdots & \partial_n Q_{r+1}(\mathbf{a}) \end{bmatrix} \quad = \quad \begin{bmatrix} F_1 & \cdots & F_r \end{bmatrix} \cdot \begin{bmatrix} \partial_1 Q_1(\mathbf{a}) & \cdots & \partial_n Q_1(\mathbf{a}) \\ \partial_1 Q_2(\mathbf{a}) & \cdots & \partial_n Q_2(\mathbf{a}) \\ \vdots & \ddots & \vdots \\ \partial_1 Q_r(\mathbf{a}) & \cdots & \partial_n Q_r(\mathbf{a}) \end{bmatrix}$$

where $F_i := (\partial_j F)(Q_1(\mathbf{a}), \ldots, Q_r(\mathbf{a}))$. But the above equation yields a contradiction as we know that the $r+1$ rows of the Jacobian of $\{Q_1, \ldots, Q_{r+1}\}$ are linearly independent and hence the LHS in the above matrix equation cannot be written as a linear combination of rows of $\mathcal{J}(Q_1, \ldots, Q_r)$. □

Over fields of low characteristic, Pandey, Saxena and Sinhababu [PSS18] use the modified Jacobian criterion to obtain a similar converse but the exact statement is a bit technical to describe here.

Part VIII

Limitations of lower bound techniques

Chapter 27

Limitations of sub-additive rank methods

So far, almost all lower bound proofs that we have seen follow this template:

1. Identify a set $\mathcal{B}$ of *building blocks* or *simple polynomials*.
2. Build a function $\Gamma : \mathbb{F}[\mathbf{x}] \to \mathbb{N}$, where $\Gamma(f)$ is the rank of an associated matrix $M(f)$ whose entries are linear functions in the coefficients of f.
3. Show that for each building block $g \in \mathcal{B}$ we have that $\operatorname{rank}(M(g)) \leq U$.
4. Find an explicit polynomial f such that $\operatorname{rank}(M(f)) \geq L$.
5. This shows that if $f = g_1 + \cdots + g_s$ where each $g_i \in \mathcal{B}$, then $s \geq L/U$.

For such a template, what is the largest we can make the ratio L/U? This would give us an indication of the limitations of techniques that use subadditivity via building blocks to prove the required lower bound. A beautiful result of Efremenko, Garg, Oliveira and Wigderson [EGOW18] prove *unconditional* limitations of these techniques for the instances of Waring rank[1] and tensor rank and we shall see their proof in this chapter. Throughout this chapter, the field $\mathbb{F}$ will be assumed to have characteristic zero.

Theorem 27.1 (Limitations for Waring Rank). *Let $M : \mathbb{F}[\mathbf{x}]_{=d} \to \operatorname{Matrix}(\mathbb{F})$ be a map that assigns a matrix to every homogeneous polynomial of degree d, that acts linearly, that is*

[1] Waring rank of a polynomial is the top fan-in of the smallest $\Sigma\wedge\Sigma$-circuit computing it.

$M(f+g) = M(f)+M(g)$. Suppose $\mathcal{B} = \left\{\ell^d \ : \ \ell = \sum a_i x_i\right\}$; define $\operatorname{rank}(\mathcal{B}) = \max\left\{\operatorname{rank}(g) \ : \ g \in \mathcal{B}\right\}$*. Then, for any $f \in \mathbb{F}[\mathbf{x}]_{=d}$ we have that*

$$\frac{\operatorname{rank}(M(f))}{\operatorname{rank}(M(\mathcal{B}))} \leq (d+1) \cdot \binom{n+(d/2)}{n}.$$

This shows that such sub-additive rank measures as outlined above cannot prove a lower bound for $\Sigma\wedge\Sigma$-circuits much better than $\binom{n+(d/2)}{n}$, though counting arguments show that there are n-variate degree-d polynomials that require $\Sigma\wedge\Sigma$-circuits of size $\frac{1}{\operatorname{poly}(n)} \cdot \binom{n+d}{n}$ to compute it.

Efremenko, Garg, Oliveira and Wigderson [EGOW18] also show a similar limitation for tensor rank.

Theorem 27.2 (Limitations for Tensor Rank)**.** *Let $X = X_1 \sqcup \cdots \sqcup X_d$ with each $|X_i| = n$, and let $\mathbb{F}[X]_{SML}$ refer to the class of set-multilinear polynomials with respect to the above partition (these are synonymous to tensors of order-d).*

Let $M : \mathbb{F}[\mathbf{x}]_{SML} \to$ Matrix($\mathbb{F}$) *be a map that assigns a matrix to every set-multilinear polynomial that acts linearly, that is $M(f+g) = M(f) + M(g)$. Suppose*

$$\mathcal{B} = \left\{\ell_1(X_1)\cdots\ell_d(X_d) \ : \ \ell_i = \sum a_{ij}x_{ij}\right\},$$

*which is synonymous to rank-*1 *tensors. Define* $\operatorname{rank}(\mathcal{B}) = \max\left\{\operatorname{rank}(g) \ : \ g \in \mathcal{B}\right\}$*. Then, for any $f \in \mathbb{F}[\mathbf{x}]_{SML}$ we have that*

$$\frac{\operatorname{rank}(M(f))}{\operatorname{rank}(M(\mathcal{B}))} \leq 2^d \cdot n^{\lfloor d/2 \rfloor}.$$

Once again, a counting argument would tell us that there are order-d tensors of rank $\frac{1}{d} \cdot n^{d-1}$. Furthermore, both in the setting of $\Sigma\wedge\Sigma$-circuits and tensor rank, current vanilla techniques achieve a lower bound of $n^{\lfloor d/2 \rfloor}$. The above theorems show that this template of proving lower bounds cannot do too much better.

We shall see the proof of Theorem 27.1 in complete detail; the proof of Theorem 27.2 is very similar.

What does this limitation mean?

Currently, the limitations hold for tensor rank and Waring rank but it is conceivable that there is a broader setting where such limitations can be proven. What does that imply for algebraic circuit lower bounds?

Almost all the lower bounds we have discussed so far follows the template of constructing a suitable linear matrix, upper bounding the rank of some building blocks, and lower bounding the rank for the target polynomial. There are, nevertheless, a few exceptions that we have seen in this survey. The first is the lower bound of Grigoriev and Karpinski [GK98] that we saw in Chapter 10. The complexity measure used by Grigoriev and Karpinski [GR00] is indeed via a linear matrix, but we study the rank of this matrix *after a certain set of columns have been dropped*. It is unclear if such non-deterministic choices can also be captured in the framework of Efremenko, Garg, Oliveira and Wigderson.

Another lower bound that does not go via sub-additive rank measures is the determinantal complexity lower bound of Mignon and Ressayre [MR04] (as seen in Chapter 7). This lower bound, though is again a rank of an associated matrix, is not proven using sub-additivity of building blocks.

Furthermore, it should also be noted that even for Waring rank and tensor rank, we *can* provide super-polynomial lower bounds via sub-additive rank measures. Of course, in the tensor-rank setting, a lower bound of $n^{\lfloor d/2 \rfloor}$ is trivial since this just involves flattening to a matrix. But for most other applications, we are aiming for much weaker lower bounds than $\binom{n+d}{d}$. Hence, even if this framework extends for models such as homogeneous $\Sigma\Pi\Sigma\Pi$ circuits, we cannot rule out the possibility that sub-additive rank methods can give lower bounds of $n^{\omega(\sqrt{d})}$, which is sufficient to separate VP and VNP.

Proof outline

We have a set $\mathcal{B}$ such that every polynomial we care about can be expressed as a linear combination of polynomials in $\mathcal{B}$. We want to think of $\mathcal{B}$ as a set of *simple polynomials* in the sense that $\operatorname{rank}(\mathcal{B}) = U$.

Now suppose that the set $\mathcal{B}$ of *simple polynomials* can be generated by one polynomial $B(\mathbf{y})$ with $|\mathbf{y}| = m$, that is for every $g \in \mathcal{B}$ there is some $\mathbf{a} \in \mathbb{F}^m$ such that $g = B(\mathbf{a})$. In the case of Waring rank, this polynomial would be

$$B(\mathbf{y}) = (x_1 y_1 + \cdots + x_n y_n)^d,$$

and in the case of tensor-rank this polynomial would be

$$B(\mathbf{y}) = \left(\sum_{j=1}^{n} y_{1j}x_{1j}\right) \times \cdots \times \left(\sum_{j=1}^{n} y_{dj}x_{dj}\right).$$

The fact that $\max_{\mathbf{a}} \operatorname{rank}(M(B(\mathbf{a}))) = U$ implies that the rank of the symbolic matrix $M(B(\mathbf{y}))$ *over the function field* $\mathbb{F}(\mathbf{y})$ is U. Now suppose f was an arbitrary polynomial, we know that $f(\mathbf{x}) = \sum_{\mathbf{a}\in A} B(\mathbf{a})$ for some finite set A since $\mathcal{B}$ forms a spanning set. Therefore, we have

$$M(f) = \sum_{\mathbf{a}\in A} M(B(\mathbf{a})).$$

What we know is that any fixed evaluation $M(B(\mathbf{a}))$ of $M(B(\mathbf{a}))$ has rank at most U and hence its rows/columns are spanned by some set of U row/column vectors. Suppose it turns out that there are row/column vectors $V_1, \ldots, V_t$ (with t not much larger than U) such that *every evaluation* $M(B(\mathbf{a}))$ has its rows/columns spanned by these t vectors. Then, by linearity, so must the rows/columns of $M(f)$! This would let us show that $\operatorname{rank}(M(f)) \leq t$ for any arbitrary f. Efremenko, Garg, Oliveira and Wigderson show that something almost similar is true for the setting of Waring rank or tensor rank.

Lemma 27.3 (Bounding rank of all evaluations)**.** *Consider the set-up as before for the Waring rank or tensor rank setting. Suppose* $\operatorname{rank}_{\mathbb{F}(\mathbf{y})}(M(B(\mathbf{y})) = U$. *Then there exists a set of row vectors* $R_1, \ldots, R_t$ *and column vectors* $C_1, \ldots, C_{t'}$ *such that every evaluation* $M(B(\mathbf{a})) = C + R$ *where the columns of* C *are spanned by* $\{C_1, \ldots, C_{t'}\}$ *and the rows of* R *are spanned by* $\{R_1, \ldots, R_t\}$.

- *For the setting of Waring rank, we have* $t + t' \leq U \cdot (d+1) \cdot \binom{n+(d/2)}{n}$.
- *For the setting of tensor rank, we have* $t + t' \leq U \cdot 2^d \cdot n^{\lfloor d/2 \rfloor}$.

This lemma immedietely yields that $\operatorname{rank}(M(f)) \leq U \cdot (d+1) \cdot \binom{n+(d/2)}{n}$ in the Waring rank setting, and $\operatorname{rank}(M(f)) \leq U \cdot 2^d \cdot n^{\lfloor d/2 \rfloor}$ in the tensor rank setting.

In fact, as it would soon be evident from the proof, the same set-up can be used to prove similar limitations in a more general setting. Suppose $B(\mathbf{y})$ is the polynomial whose evaluations generate the class $\mathcal{B}$ and suppose $\deg_{\mathbf{y}} B(\mathbf{y}) = \deg f$ and $|\mathbf{y}| = m$. Then for any

$f \in \operatorname{span}(\mathcal{B})$ we have

$$\frac{\operatorname{rank}(M(f))}{\operatorname{rank}(M(\mathcal{B}))} \leq (d+1) \cdot \binom{m + \lfloor d/2 \rfloor}{m}.$$

The Waring rank setting was the instantiation with $m = n$. Of course, if $m = \Omega(n^2)$, then we get a meaningless limitation that these techniques cannot prove lower bounds better than n^d (which is roughly the number of monomials in f anyway!) but we get a meaningful limitation for all $m = n^{2-\varepsilon}$. (In fact, the tensor rank setting can also be cast this way with $m = dn$, but Theorem 27.2 improves this above bound.)

In the rest of this chapter, we shall see a proof of the key lemma (Lemma 27.3) for the general instance when the building blocks are generated by a $B(\mathbf{y})$.

27.1 The main decomposition lemma

We have a polynomial $B(\mathbf{x}, \mathbf{y}) \in \mathbb{F}[\mathbf{x}, \mathbf{y}]$ that is homogeneous in $\mathbf{y}$ and $\deg_{\mathbf{y}} B = \deg f = d$; let $|\mathbf{y}| = m$ and $|\mathbf{x}| = n$. We shall sometime denote $B(\mathbf{x}, \mathbf{y})$ by just $B(\mathbf{y})$ as we will be more interested in B as a function of $\mathbf{y}$.

$$\mathcal{B} = \{B(\mathbf{a}) \ : \ \mathbf{a} \in \mathbb{F}^m\}$$

Let us assume that $M(f)$ is an $N \times N$ matrix, without loss of generality, for some $N \geq 0$. If $U = \operatorname{rank}(M(\mathcal{B}))$, then we have that $\tilde{M} = \operatorname{rank}_{\mathbb{F}(\mathbf{y})}(M(B(\mathbf{y}))) = U$ as a symbolic matrix. Therefore, $\tilde{M}$ can be written as

$$\tilde{M} = V_1 W_1^T + \cdots V_U W_U^T$$

for N-length vectors V_i, W_i with entries from $\mathbb{F}(\mathbf{y})$. This is a little annoying that even though $\tilde{M}$ only involves entries that are homogeneous degree d polynomials in $\mathbf{y}$, the entries of the vectors could involve rational functions in $\mathbf{y}$ of arbitrary numerators and denominators. A natural question is whether every *homogeneous matrix* of small rank has a *small homogeneous rank-*1 *decomposition*. Efremenko, Garg, Oliveira and Wigderson show that this is indeed true.

Lemma 27.4 (Homogeneous rank-1 decomposition)**.** *Suppose* $\tilde{M}$ *is a matrix with entries in* $\mathbb{F}[\mathbf{y}]_{=d}$ *and suppose* $\operatorname{rank}_{\mathbb{F}(\mathbf{y})} \tilde{M} = U$*. Then there are row vectors* $P_1, \ldots, P_t$ *and* $Q_1, \ldots, Q_t$

such that

$$\tilde{M} = P_1 Q_1^T + \cdots + P_t Q_t^T,$$

satisfying

- $t \leq (d+1)U$,
- *For each i there is some $0 \leq d_i \leq d$ such the vectors P_i and Q_i consists of homogeneous polynomials of degree d_i and $d - d_i$ respectively.*

In other words, if the rank of a matrix $\tilde{M}$ involving homogeneous polynomials of degree d as its entries is at most U, then its *homogeneous rank*[2] is at most $(d+1)U$.

An analogue of the above lemma holds for set-multilinear polynomials as well, where we would like to have each rank-1 term to also be set-multilinear. This is the analogue required in the proof of Theorem 27.2.

The proof of Lemma 27.4 is along the lines of Strassen's division elimination and we will defer this to later and see how to prove Lemma 27.3 using this.

Proof of Lemma 27.3. Consider the homogeneous rank-1 decomposition of $\tilde{M}$ given above by Lemma 27.4:

$$\tilde{M} = P_1 Q_1^T + \cdots + P_t Q_t^T.$$

Write $\tilde{M} = \tilde{R} + \tilde{C}$ where

$$\tilde{R} := \sum_{i:\deg P_i \leq \frac{d}{2}} P_i Q_i^T \quad , \quad \tilde{C} := \sum_{i:\deg P_i > \frac{d}{2}} P_i Q_i^T.$$

We will focus on $\tilde{R}$; the analysis for $\tilde{C}$ would be analogous.

$$\begin{aligned}
\tilde{R}(\mathbf{y}) &= \sum_{i:\deg P_i \leq \frac{d}{2}} P_i(\mathbf{y}) Q_i(\mathbf{y})^T \\
&= \sum_{i:\deg P_i \leq \frac{d}{2}} \sum_{\mathbf{e}:|\mathbf{e}| \leq \frac{d}{2}} \mathbf{y}^{\mathbf{e}} \cdot \operatorname{coeff}_{\mathbf{y}^{\mathbf{e}}}(P_i) Q_i(\mathbf{y})^T \\
\implies \tilde{R}(\mathbf{a}) &= \sum_{i:\deg P_i \leq \frac{d}{2}} \sum_{\mathbf{e}:|\mathbf{e}| \leq \frac{d}{2}} \mathbf{a}^{\mathbf{e}} \cdot \operatorname{coeff}_{\mathbf{y}^{\mathbf{e}}}(P_i) Q_i(\mathbf{a})^T
\end{aligned}$$

[2] Defined analogously as the smallest *homogeneous* rank-1 decomposition

Hence, every row of $\tilde{R}(\mathbf{a})$ is spanned by $\left\{\mathrm{coeff}_{\mathbf{y}^{\mathbf{e}}}(P_i) \;:\; \deg P_i \leq \frac{d}{2}\;,\; \deg \mathbf{y}^{\mathbf{e}} \leq \frac{d}{2}\right\}$. Similarly, every column of $\tilde{C}(\mathbf{a})$ is spanned by $\left\{\mathrm{coeff}_{\mathbf{y}^{\mathbf{e}}}(Q_i)^T \;:\; \deg P_i > \frac{d}{2}\;,\; \deg \mathbf{y}^{\mathbf{e}} \leq \frac{d}{2}\right\}$. Together, we have at most $t \cdot \binom{m+(d/2)}{m}$ vectors. The statement of Lemma 27.3 follows since $M(f)$ is a linear combination of the evaluations $\{R(\mathbf{a}) + C(\mathbf{a}) \;:\; \mathbf{a} \in \mathbb{F}^m\}$ □

Proof of the decomposition lemma

We begin with the standard rank-1 decomposition of the symbolic matrix $\tilde{M}$.

$$\tilde{M} = V_1 W_1^T + \cdots V_U W_U^T$$

As stated, the entries of V_i, W_i could involve rational functions in $\mathbf{y}$. Nevertheless, we can clear denominators and obtain

$$\tilde{M} = \frac{1}{g(\mathbf{y})} \cdot \left(V_1' W_1'^T + \cdots V_U' W_U'^T\right)$$

where now $g(\mathbf{y})$ and the entries of V_i' and W_i' are honest-to-god polynomials in $\mathbf{y}$. Assume without loss of generality that $g(\mathbf{0}) = 1$ (by suitable translation and scaling if necessary). Then, $g(\mathbf{y}) = 1 - g'(\mathbf{y})$ where $g'(\mathbf{0}) = 0$. Therefore,

$$\frac{1}{g(\mathbf{y})} = 1 + g'(\mathbf{y}) + (g'(\mathbf{y}))^2 + \cdots .$$

Therefore,

$$\tilde{M} = \left(V_1' W_1'^T + \cdots V_U' W_U'^T\right) \cdot \left(1 + g'(\mathbf{y}) + (g'(\mathbf{y}))^2 + \cdots\right).$$

Here comes the important point: even though the right hand side has infinitely many terms, every entry on the left-hand side has degree at most d. Hence, we may just collect homogeneous components from the RHS to obtain our homogeneous decomposition. Define $\tilde{g}(\mathbf{y}) = 1 + g'(\mathbf{y}) + \cdots + (g'(\mathbf{y}))^d$. Then,

$$\tilde{M} = \sum_{i=0}^{d} \sum_{j=1}^{U} \left(\mathrm{Hom}_i\left(\tilde{g} \cdot V_j'\right)\right)\left(\mathrm{Hom}_{d-i}\left(W_j'\right)\right)^T . \quad \square$$

Part IX

Breakthrough!

Chapter 28

Lower bounds for constant depth circuits

In 2021, Nutan Limaye, Srikanth Srinivasan and Sébastien Tavenas [LST21] proved super-polynomial lower bounds for constant depth circuits! Not just that, but they gave a ridiculously simple proof of this fact! Needless to say, many parts of the introductory parts of various chapters alluding to lack of progress etc. has been completely blown out of the waters. Admittedly, it is a little silly to discuss fairly complicated approaches to prove lower bounds for depth-4 circuits etc. for the last how-many-ever chapters only to give a completely simple proof of super-polynomial lower bound for arbitrary constant depth circuits. But, well, it is what it is.

In this chapter, we are going to see the full proof of their beautiful result and then worry about correcting the narrative in the survey later.

Theorem 28.1 (Limaye-Srinivasan-Tavenas [LST21])**.** *Let d, n such that $d = o(\log n)$. For any $\Delta > 0$, any product-depth Δ circuit computing $\mathrm{IMM}_{n,d}$, over any field of characteristic zero, requires size $n^{d^{-\exp(\Delta)}}$.*

For the particular case of $\Sigma\Pi\Sigma$ circuits, the lower bound is $n^{\Omega(\sqrt{d})}$.

As we have seen in Theorem 23.2, this lower bound for $\Sigma\Pi\Sigma$ circuits is tight, up to constants in the exponent.

Comparing with depth-reduction results. We have seen in Exercise 5.5 that any polynomial sized circuit can be equivalently computed by a homogeneous product-depth Δ circuit of size at most $n^{O(d^{1/\Delta})}$. Furthermore, if we are willing to forgo homogeneity, then Theorem 23.2 can be used to yield a non-homogeneous product-depth Δ circuit of size at most

$n^{O(d^{1/2\Delta})}$. Therefore, the above theorem is tight for the setting of $\Delta = 1$ but not so for larger Δ.

28.1 Proof overview

The proof is surprisingly extremely simple. Roughly speaking, the proof proceeds in three broad steps, many components of which were known!

1. Convert any arbitrary product-depth Δ circuit into a homogeneous product-depth 2Δ circuit computing the same polynomial.
2. Convert any homogeneous product-depth 2Δ circuit computing a set-multilinear polynomial into a set-multilinear product-depth 2Δ circuit computing the same polynomial.
3. Prove a lower bound for set-multilinear constant depth circuits.

Note that Lemma 23.6 already showed a version of item 1 to convert a non-homogeneous $\Sigma\Pi\Sigma$ circuit to a homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ circuit. Turns out, this can be generalised to larger depth as well and we'll discuss this later in the chapter.

As for item 2, we did see a set-multilinearisation in the context of Theorem 16.12. Turns out, the same can also be done for constant depth circuits.

As for item 3 of proving lower bounds for set-multilinear formulas of constant depth, we even have lower bounds for multilinear formulas of arbitrary depth! However, an important point is that both the above steps are efficient only when $d \ll n$ whereas many of the multilinear and set-multilinear lower bounds required d to be comparable to n. Nevertheless, Limaye, Srinivasan and Tavenas show how to use the same technique with a small but ingenious modification.

28.2 Reducing to the set-multilinear world

We will first see how we can transform any small-depth circuit computing a lower-degree set-multilinear polynomial into a syntactically set-multilinear small-depth circuit without too much blow-up in size. This involves two steps as mentioned above — homogenisation and set-multilinearisation.

28.2.1 Homogenisation within small depth

It is good to keep in mind that it is perhaps too much to expect to homogenise a non-homogeneous arbitrary depth-Δ circuit into a homogeneous depth-Δ circuit of comparable size. An excellent counter-example to this, with respect to $\Sigma\Pi\Sigma$ circuits is the elementary symmetric polynomial. We know there are non-homogeneous $\Sigma\Pi\Sigma$ circuits of polynomial size that compute it, but any homogeneous $\Sigma\Pi\Sigma$ circuit computing it must be of exponential size (Theorem 9.2).

However, if one is willing to increase the depth a little (by factor of 2), then one can indeed homogenise an arbitrary circuit with a not-too-large blow-up in size.

Lemma 28.2 (Homogenisation within constant depth)**.** *Let C be a product-depth Δ circuit of size s computing an n-variate degree d polynomial. Then, there is a* homogeneous *circuit C' of product-depth 2Δ and size $2^{O(\sqrt{d})} \cdot \operatorname{poly}(s)$ computing the same polynomial.*

We'll give the sketch of the proof here. It is instructive to go back to what we did in the case of $\Sigma\Pi\Sigma$ circuits. The crux is to try and extract the homogeneous part of a term of the form

$$T = (1 + \ell_1) \cdots (1 + \ell_m)$$

where each of the ℓ_i's are homogeneous linear polynomials. It is clear that

$$\mathrm{Hom}_d(T) = \mathsf{Sym}_d(\ell_1, \ldots, \ell_m).$$

Then, we can use the fact that $\mathsf{Sym}(z_1, \ldots, z_m)$ can be computed by a homogeneous depth-4 (or product-depth 2) circuit of size at most $2^{O(\sqrt{d})} \operatorname{poly}(m)$ (Lemma 23.6). Therefore, $\mathrm{Hom}_d(\Sigma\Pi\Sigma)$ can be computed by a homogeneous depth-5 circuit of size at most $2^{O(\sqrt{d})} \operatorname{poly}(s)$ size.

This proves the above lemma in the setting when $\Delta = 1$. In order to generalise this to higher depth, what exactly would we need? If there is a multiplication gate $g = g_1 \times \cdots \times g_m$, and lets assume that we have managed to break each g_i into its homogeneous parts $g_{i,0} + g_{i,1} + \cdots + g_{i,d}$. Therefore, we wish to extract the homogeneous part

$$\mathrm{Hom}_d(g) = \mathrm{Hom}_d\left(\prod_{i=1}^{m}\left(\sum_{j=0}^{d} g_{i,j}\right)\right),$$

which seems very similar to the elementary symmetric polynomial except that it is a "weighted" version of it:

$$\mathsf{WSym}_{m,d}(z_{1,0},\ldots,z_{1,d},\ldots,z_{m,0},\ldots,z_{m,d}) = \sum_{\substack{\sigma:[m]\to\{0,\ldots,d\} \\ \sum_i \sigma(i)=d}} \prod_{i=1}^{m} z_{i,\sigma(i)}.$$

In words, we want all monomials obtained by picking up one homogeneous part from each block such that the total degree is exactly d. Very similar to Newton identities that relate the Sym_i families to the Pow_i families, there are similar relations between the weighted versions of them. Mukherjee and Bera [MB18] gave a combinatorial proof of this fact.

Lemma 28.3 ([MB18])**.** *The polynomial* $\mathsf{WSym}_{m,d}$ *can be computed by a homogeneous* $\Sigma\Pi\Sigma\Pi$ *circuit of size at most* $2^{O(\sqrt{d})}\cdot \mathrm{poly}(m)$*.* □

Applying this at each gate of an arbitrary depth-Δ circuit yields Lemma 28.2.

28.2.2 Set-multilinearisation

Lemma 28.4 (Set-multilinearisation in constant depth)**.** *Let C be a homogeneous product-depth Δ circuit of size s computing a set-multilinear n-variate degree d polynomial f. Then, there is a* set-multilinear circuit *of product-depth Δ and size $d^{O(d)}\cdot \mathrm{poly}(s)$ computing the same polynomial.*

This is exactly the standard set-multilinearisation operation that we used in Theorem 16.12 which just needs to be inspected that the same can be done with constant depth, arbitrary fan-in circuits.

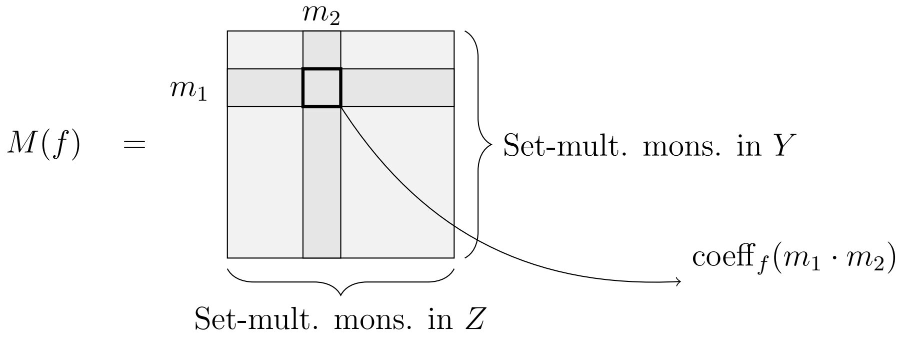
Exercise 28.1 [Set-multilinearisation in constant depth] *Prove Lemma 28.4.*

28.3 Lower bound for set-multilinear, small-depth circuits

Now that we have seen that any n-variate, degree d, set-multilinear polynomial that is computed by a small product-depth Δ circuit can also be computed by a set-multilinear product-depth 2Δ circuit of size $d^{O(d)}\cdot \mathrm{poly}(s)$, it suffices to prove strong enough lower bounds for small-depth set-multilinear circuits.

28.3.1 The complexity measure

Suppose $X = X_1 \sqcup \cdots \sqcup X_d$, we will be using the familiar measure of the rank of the partial derivative matrix. We will split d parts into two sets of buckets, that we'll refer to as Alice parts and Bob parts and denote them by $Y_1 \sqcup \cdots \sqcup Y_r$ and $Z_1 \sqcup \cdots \sqcup Z_{r'}$ respectively (with $r + r' = d$). For this partition of variables and this split, and for any set-multilinear polynomial f, we will defined the matrix $M(f)$ as follows:



That is, this is the *communication matrix* when Alice is given all variables of the Alice parts and Bob is given all variables in the Bob parts. The number of rows is $\prod |Y_i|$ and the number of columns is $\prod |Z_i|$. The measures used will be related to the rank of this matrix but it would be convenient to use a normalised version of this, called *relative-rank* (denoted by rel-rank).

$$\mu(f) := \text{rel-rank}(M(f)) := \frac{\text{rank}(M(f))}{\sqrt{\#\text{rows} \cdot \#\text{cols}}}$$

Roughly speaking, rel-rank is a way of capturing the rank of the matrix and also take into account the *aspect-ratio* of the matrix.

The following are a few simple observations of rel-rank that follows readily from the definition.

Observation 28.5 (Simple properties of μ)**.** *The following are some simple properties about the measure μ:*

1. ***Trivial upper bound.*** *For any set-multilinear polynomial f, we have*

$$\mu(f) \leq \min\left\{\sqrt{\frac{\#rows}{\#cols}}, \sqrt{\frac{\#cols}{\#rows}}\right\}.$$

In particular, $\mu(f) \leq 1$.

2. ***Sub-additivity.*** *For any set-multilinear sum*[1] $f + g$, *we have*

$$\mu(f + g) \leq \mu(f) + \mu(g).$$

3. ***Multiplicativity.*** *For any set-multilinear product*[2]

$$\mu(f \cdot g) = \mu(f) \cdot \mu(g).$$

To illustrate the key idea, we will illustrate how to get super-polynomial lower bounds for $\Sigma\Pi\Sigma$ circuits, which translates to proving good enough lower bound for set-multilinear $\Sigma\Pi\Sigma\Pi\Sigma$ circuits.

28.3.2 Lower bounds for set-multilinear depth-5 circuits

Until now, we didn't really bother much about the sizes of the parts with Alice and Bob but this will be crucial! Let k be a parameter that is set to $(\log_2 n)/2$ and we will assume that $k > \frac{\sqrt{d}}{2}$.

We will only consider set-multilinear polynomials of degree d, where each Alice part Y_i has size exactly $2^{k - \frac{k}{\sqrt{d}}}$ and each Bob part Z_j has size exactly 2^k. (To avoid the use of floors and ceilings, we will assume that k is divisible by $\sqrt{d}$). However, we will set up the number of Alice parts and Bob parts so that

$$\prod |Y_i| \approx \prod |Z_i|$$

so that the matrix $M(f)$ is almost a square matrix. The imbalance between the sizes of the Alice parts and the Bob parts will be crucial in the proof.

The hard polynomial, for now, will be whatever P makes the matrix $M(P) = I$ and hence $\mu(P) = 1$. Noticing that we can actually find an efficiently computable polynomial with $\mu(P) = 1$ is mostly an after-thought and we'll do that later. (TODO)

[1] That is, f and g set-multilinear with respect to the same parts.

[2] That is, f and g set-multilinear on disjoint sub-partitions.

Warm-up: Lower bounds for set-multilinear $\Sigma\Pi\Sigma$ circuits

If ℓ is any set-multilinear linear polynomial, then ℓ must only involve a single part X_i. Therefore,

$$\mu(\ell) \leq \frac{1}{\sqrt{|X_i|}}.$$

Hence, if C is a size s set-multilinear $\Sigma\Pi\Sigma$ circuit, then

$$\mu(C) \leq s \cdot \frac{1}{\sqrt{|X_1| \cdots |X_d|}}$$

The lower bound

Lemma 28.6 (Lower bound for set-multilinear depth-5 circuits)**.** *Let f be a set-multilinear polynomial with respect to the partition defined above, with $d = o(\log^2 n)$. If f is computable by a size s, set-multilinear $\Sigma\Pi\Sigma\Pi\Sigma$ circuit, then*

$$\mu(f) \leq s \cdot 2^{-k\sqrt{d}/8}.$$

Once we have this lemma, the lower bound is an immediate corollary.

Corollary 28.7. *Let f be a set-multilinear polynomial with respect to the above parts with $\mu(f) = 1$. Then, any set-multilinear $\Sigma\Pi\Sigma\Pi\Sigma$ circuit computing it must have size $n^{\Omega(\sqrt{d})}$.* □

Proof of Lemma 28.6. The proof of this Lemma is embarrassingly simple. Suppose, $C = T_1 + \cdots + T_r$ where each T_i is a set-multilinear $\Pi\Sigma\Pi\Sigma$ circuit of size s_i, then we know that $s \leq \sum s_i$. Thanks to sub-additivity, it suffices to show that $\mu(T_i) \leq s_i \cdot 2^{-k\sqrt{d}/8}$. Hence, let us focus on one such term

$$T = Q_1 \ldots Q_t$$

where (reusing some variables) T has size bounded by s and is a set-multilinear product of set-multilinear $\Sigma\Pi\Sigma$ circuits. Without loss of generality, let Q_1 have the largest degree among $Q_1, \ldots, Q_t$.

Case 1: $\deg(Q_1) > \sqrt{d}/2$**.** In this case, Q_1 is a *size-able* factor and hence "the loss" from Q_1 is going to be sufficient for us.

$$\begin{aligned}\mu(T) &= \mu(Q_1)\cdots\mu(Q_t) \leq \mu(Q_1)\\ &\leq s \cdot \frac{1}{\sqrt{|X_{i_1}|\cdots|X_{i_{d'}}|}} \leq s \cdot 2^{-\frac{\sqrt{d}}{4}\cdot\left(k-\frac{k}{\sqrt{d}}\right)} \leq s \cdot 2^{-k\sqrt{d}/8}.\end{aligned}$$

Case 2: $\deg(Q_i) \leq \sqrt{d}/2$ **for all** i**.** Since μ is multiplicative, we have

$$\mu(T) = \mu(Q_1)\cdots\mu(Q_t).$$

We will show that $\mu(Q_i) \ll 1$ purely from the fact that $M(Q_1)$ will be far from a square matrix!

Let us focus on one factor Q. Suppose this factor depends on a Alice parts and b Bob parts. Further, $a + b = \deg(Q) \leq \sqrt{d}/2$. Let us get a sense of the number of rows and columns in the matrix $M(Q)$.

$$\begin{aligned}\#\text{rows} &= 2^{a\cdot\left(k-\frac{k}{\sqrt{d}}\right)},\\ \#\text{cols} &= 2^{b\cdot k},\\ \implies \mu(Q) &\leq \min\left\{\sqrt{\frac{\#\text{rows}}{\#\text{cols}}}, \sqrt{\frac{\#\text{cols}}{\#\text{rows}}}\right\}\\ &= 2^{-k\cdot\left|a\left(k-\frac{k}{\sqrt{d}}\right)-bk\right|} =: 2^{-k\cdot|E|}\end{aligned}$$

where $E = a\left(1-\frac{1}{\sqrt{d}}\right) - b$. The key point is that E is not close to zero just for the simple reason that the fractional part of E is substantial.
If $a \leq b$, then E is negative. Hence,

$$\begin{aligned}E &= a\left(1-\frac{1}{\sqrt{d}}\right) - b\\ &\leq \frac{\deg(Q)}{2}\cdot\left(1-\frac{1}{\sqrt{d}}\right) - \frac{\deg(Q)}{2}\\ &= \frac{-\deg(Q)}{2\sqrt{d}}.\end{aligned}$$

If $a > b$, then E is positive. There, we have

$$\begin{aligned} E \geq \{E\} &= \left\{1 - \frac{1}{\sqrt{d}}\right\} \\ &> \frac{1}{2} > \frac{\deg(Q)}{4\sqrt{d}}. \end{aligned}$$

Therefore,

$$\begin{aligned} & \mu(Q) \leq 2^{-\frac{k \cdot \deg(Q)}{4\sqrt{d}}} \\ \implies & \mu(T) = \mu(Q_1) \cdots \mu(Q_t) \leq 2^{-\frac{k \cdot \deg(T)}{4\sqrt{d}}} = 2^{-k\sqrt{d}/4} \end{aligned}$$

And that completes the proof! □

28.3.3 Lower bounds for larger depth

The lower bound for larger depth proceeds exactly along the same lines (with a few minor technical issues that need to get sorted out). Suppose we are dealing with a depth-7 set-multilinear circuit instead. The idea is to proceed exactly along the same lines. Say this circuit $C = T_1 + \cdots + T_r$, it suffices to focus on each summand T_i. Suppose such a summand $T = Q_1 \cdots Q_t$ where each Q_i is now a set-multilinear depth-5 circuit.

Case 1: $\deg(Q_1)$ is large. In this case, just use the upper-bound we just proved for set-multilinear depth-5 circuits to bound $\mu(Q_1)$ and hence $\mu(T)$.

Case 2: All $\deg(Q_i)$'s are small. Use the same argument to show that each $M(Q_i)$ is non-square-ish to get the upper bound on $\mu(Q_i)$ and hence $\mu(T)$.

The only technical hurdle is that earlier we had chosen our k depending on what degree we were working with etc. but now the choice needs to be flexible. If we think about it, all we want is to make sure that if Alice parts are of size αk and Bob parts are of size k, then we want $|a\alpha - b|$ to be large when a and b are not too large. Here is an easy fix — take α to be an irrational number, like $\sqrt{2}$! But then, we want $2^{\alpha k}$ to be an integer as well. Turns out, taking Alice parts to be of size $2^{\lceil \alpha k \rceil}$ and Bob parts to be of size 2^k is sufficient to make the proof work.

That's basically the proof for arbitrary constant depth as well.

RP: Fill in the details

28.4 Getting the lower bound for IMM

As mentioned earlier, an explicit hard polynomial with respect to the above measure is to just choose a polynomial P such that $M(P)$ looks like the identity matrix. With this definition, we might just be able to say that $P \in \mathsf{VNP}$ but with a bit more care, we can find a hard-polynomial that is actually computable by a polynomial sized set-multilinear ABP and hence extend the above lower bounds to IMM as well.

28.4.1 Re-interpreting the polynomial

Let us consider a more general setting where $Y_i = 2^{w_i}$ and $Z_i = 2^{v_i}$ for some positive integers w_i and v_i. This allows us to just name the variables in each of these parts using binary strings of the appropriate length:

$$Y_i = \left\{ y_a^{(i)} \ : \ a \in \{0,1\}^{w_i} \right\} \qquad\qquad Z_i = \left\{ z_b^{(i)} \ : \ b \in \{0,1\}^{v_i} \right\}.$$

Hence, asking for $M(P) = I$ is the same as saying that P must be a sum of monomials of the form

$$P = \sum_{\substack{\mathbf{a},\mathbf{b} \text{ such that} \\ a_1 \cdots a_r = b_1 \cdots b_{r'}}} y_{a_1}^{(1)} \cdots y_{a_r}^{(r)} \cdot z_{b_1}^{(1)} \cdots z_{b_{r'}}^{(r')}$$

(If the $\prod |Y_i| \neq \prod |Z_i|$, then one can work with monomials where the indices concatenated on the *smaller* side is a prefix of the indices concatenated on the *larger* side so that $M(P)$ becomes a full-rank matrix).

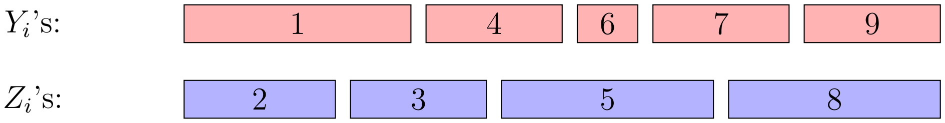
The key point is that, as long as $|Y_i|, |Z_i| = \operatorname{poly}(n)$, the above polynomial is computable by a set-multilinear ABP of polynomial size.

Lemma 28.8. *Let $Y_1, \ldots, Y_r, Z_1, \ldots, Z_{r'}$ be sets as above with $|Y_i|, |Z_i| \leq m$, and P defined as above. Then there is a* $\operatorname{poly}(m, (r + r'))$*-sized set-multilinear ABP computing P.*

Proof. Intuitively, the polynomial P looks like the *inner-product* or the *doubling* polynomial in the non-commutative setting. Typically, an ABP cannot "remember" the entire concatenated Y indices to currently evolve the Z-part. However, we can choose to carefully interleave the parts so that the ABP will never have to "remember" too much. Here is how we are going to build the order in which variables are going to be read:

> Suppose we have chose $Y_1, \ldots, Y_t$ and $Z_1, \ldots, Z_{t'}$ (in some order). Let $\ell = \sum_{i \leq t} w_i - \sum_{j \leq t'} v_i$; the difference in length of the current Alice-part and Bob-part. If $\ell \leq 0$, add Y_{t+1} next in the order. Else, add $Z_{t'+1}$ next in the order.

Here is an example of such an ordering:



Notice that given the way we have defined the order, the parameter ℓ would never exceed $\lceil \log_2 m \rceil$ as any intermediate imbalance is bounded by just one part.

The description of the ABP should now be intuitively clear — the ABP remembers the "excess" string at any intermediate point and picks a variable with this excess string as prefix. For each i, let ℓ_i be current difference in length (just like in the above ordering algorithm) and the i-th layer will have 2^{ℓ_i} different vertices, one for each string $a \in \{0,1\}^{\ell_i}$. If the excess string at the i-th layer is $a \in \{0,1\}^{\ell_i}$, and multiplying with a variable x_k from the $i+1$-th part results in an excess string $b \in \{0,1\}^{\ell_{i+1}}$, there is an edge from $v_a^{(i)}$ to $v_b^{(i+1)}$ with weight x_k on it. It can be readily seen that this ABP indeed computes the polynomial P. Furthermore, this ABP is a *set-multilinear* projection of IMM as all edges between layer i and $i+1$ only involve variables □

28.5 TODO: Subsequent results

There were some subsequent results proved by the Limaye, Srinivasan and Tavenas that ought to be added here. Some of these include:

- Super-polynomial separation between depth Δ and depth $\Delta + 1$
- Super-polynomial separation between homogeneous formulas and algebraic branching programs in the non-commutative setting

Bibliography

[Agr05] Manindra Agrawal. Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, pages 92–105, 2005.

[AJMV98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-Commutative Arithmetic Circuits: Depth Reduction and Size Lower Bounds. *Theoretical Computer Science*, 209(1-2):47–86, 1998.

[AKV18] Noga Alon, Mrinal Kumar, and Ben Lee Volk. Unbalancing Sets and an Almost Quadratic Lower Bound for Syntactically Multilinear Arithmetic Circuits. In Servedio [Ser18], pages 11:1–11:16.

[Alo09] Noga Alon. Perturbed Identity Matrices Have High Rank: Proof and Applications. *Combinatorics, Probability and Computing*, 18(1-2):3–15, March 2009.

[AV08] Manindra Agrawal and V. Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 67–75, 2008.

[BC15] Suman K. Bera and Amit Chakrabarti. A Depth-Five Lower Bound for Iterated Matrix Multiplication. In *Conference on Computational Complexity (CCC)*, pages 183–197, 2015.

[BCS97] Peter Bürgisser, Michael Clausen, and Mohammad A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1997.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18(3):147 – 150, 1984.

[BLS16] Nikhil Balaji, Nutan Limaye, and Srikanth Srinivasan. An Almost Cubic Lower Bound for ΣΠΣ Circuits Computing a Polynomial in VP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:143, 2016. `eccc:TR16-143`.

[Bre74] Richard P. Brent. The Parallel Evaluation of General Arithmetic Expressions. *Journal of the ACM*, 21(2):201–206, April 1974.

[BS83] Walter Baur and Volker Strassen. The Complexity of Partial Derivatives. *Theoretical Computer Science*, 22:317–330, 1983.

[Bür00] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Algorithms and Computation in Mathematics. Springer, 2000. Online version: `http://math-www.uni-paderborn.de/agpb/work/habil.ps`.

[Cai90] Jin-Yi Cai. A note on the determinant and permanent problem. *Information and Computation*, 84(1):119–127, 1990.

[CCL08] Jin-Yi Cai, Xi Chen, and Dong Li. A quadratic lower bound for the permanent and determinant problem over any characteristic$\neq 2$. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008)*, pages 491–498. ACM, 2008.

[CILM18] Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin. Hardness Amplification for Non-Commutative Arithmetic Circuits. In Servedio [Ser18], pages 12:1–12:16.

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial Derivatives in Arithmetic Complexity (and beyond). *Foundation and Trends in Theoretical Computer Science*, 2011.

[CLO07] David A. Cox, John B. Little, and Donal O'Shea. *Ideals, Varieties and Algorithms*. Undergraduate texts in mathematics. Springer, 2007.

[CM14] Suryajith Chillara and Partha Mukhopadhyay. Depth-4 Lower Bounds, Determinantal Complexity : A Unified Approach. *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, 2014.

[DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 615–624, 2012. eccc:TR11-134.

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM Journal of Computing*, 39(4):1279–1293, 2009. Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*.

[EGOW18] Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for Rank Methods in Arithmetic Complexity. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 1:1–1:19, 2018.

[Ell69] W. J. Ellison. A 'Waring's Problem' for homogeneous forms. *Proceedings of the Cambridge Philosophical Society*, 65:663–672, 1969.

[Fis94] Ismor Fischer. Sums of like powers of multivariate linear forms. *Mathematics Magazine*, 67(1):59–61, 1994.

[FLMS15] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower Bounds for Depth-4 Formulas Computing Iterated Matrix Multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015. Preliminary version in the *46th Annual ACM Symposium on Theory of Computing (STOC 2014)*.

[FS13] Michael A. Forbes and Amir Shpilka. Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 243–252, 2013.

[GK98] Dima Grigoriev and Marek Karpinski. An Exponential Lower Bound for Depth 3 Arithmetic Circuits. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 577–582, 1998.

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *Journal of the ACM*, 61(6):33:1–33:16, 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Preliminary version in the *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*.

[GLSS12] Dmitry Gavinsky, Shachar Lovett, Michael Saks, and Srikanth Srinivasan. A Tail Bound for Read-k Families of Functions. *CoRR*, abs/1205.1478, 2012.

[GR00] Dima Grigoriev and Alexander A. Razborov. Exponential Lower Bounds for Depth 3 Arithmetic Circuits in Algebras of Functions over Finite Fields. *Appl. Algebra Eng. Commun. Comput.*, 10(6):465–487, 2000. Preliminary version in the *39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998)*.

[Gro] Joshua Grochow. `http://cstheory.stackexchange.com/questions/19261/degree-restriction-for-polynomials-in-mathsfvp/19268#19268`.

[Hås90] Johan Håstad. Tensor Rank is NP-complete. *J. Algorithms*, 11(4):644–654, December 1990. Preliminary version in the *16th International Colloquium on Automata, Languages and Programming (ICALP 1989)*.

[HR18] G. H. Hardy and Srinavasa Ramanujan. Asymptotic Formulæ in Combinatory Analysis. *Proceedings of the London Mathematical Society*, s2-17(1):75–115, 1918.

[HWY10] Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. Non-commutative circuits and the sum-of-squares problem. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC 2010)*, pages 667–676, 2010.

[HY11] Pavel Hrubeš and Amir Yehudayoff. Homogeneous Formulas and Symmetric Polynomials. *Computational Complexity*, 20(3):559–578, 2011.

[HY16] Pavel Hrubeš and Amir Yehudayoff. On Isoperimetric Profiles and Computational Complexity. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, pages 89:1–89:12, 2016. Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR15-164.

[Hya79] Laurent Hyafil. On the Parallel Evaluation of Multivariate Polynomials. *SIAM Journal of Computing*, 8(2):120–123, 1979. Preliminary version in the *10th Annual ACM Symposium on Theory of Computing (STOC 1978)*.

[JS82] Mark Jerrum and Marc Snir. Some Exact Complexity Results for Straight-Line Computations over Semirings. *Journal of the ACM*, 29(3):874–897, 1982.

[Kal85] Kyriakos Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM Journal of Computing*, 14(3):678–687, 1985.

[Kal00] D. Kalman. A Matrix Proof of Newton's Identities. *Mathematics Magazine*, 73(4):313–315, 2000.

[Kay09] Neeraj Kayal. The Complexity of the Annihilating Polynomial. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC 2009)*, pages 184–193, July 2009.

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

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[KLSS17] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Formulas. *SIAM J. Comput.*, 46(1):307–335, 2017. Preliminary version in the *55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*.

[Koi12] Pascal Koiran. Arithmetic Circuits: The Chasm at Depth Four Gets Wider. *Theoretical Computer Science*, 448:56–65, 2012.

[Kou08] Ioannis Koutis. Faster Algebraic Algorithms for Path and Packing Problems. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 575–586, 2008.

[KS14a] Neeraj Kayal and Ramprasad Saptharishi. A selection of lower bounds for arithmetic circuits. In *Perspectives in Computational Complexity*. Birkhäuser, Basel, 2014.

[KS14b] Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: it's all about the top fan-in. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 136–145, 2014.

[KS14c] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, 2014.

[KS15a] Neeraj Kayal and Chandan Saha. Lower Bounds for Depth Three Arithmetic Circuits with Small Bottom Fanin. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC 2015)*, pages 158–208, 2015.

[KS15b] Neeraj Kayal and Chandan Saha. Lower Bounds for Sums of Products of Low arity Polynomials. In *Electronic Colloquium on Computational Complexity (ECCC)TR15-073*, 2015.

[KS15c] Neeraj Kayal and Chandan Saha. Multi-k-ic Depth Three Circuit Lower Bound. In *Proceedings of the 32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015)*, volume 30, pages 527–539, 2015.

[KS16a] Mrinal Kumar and Shubhangi Saraf. Arithmetic circuits with locally low algebraic rank. In *Proceedings of the 31st Annual Computational Complexity Conference (CCC 2016)*, 2016.

[KS16b] Mrinal Kumar and Shubhangi Saraf. Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing. In Ran Raz, editor, *Proceedings of the 31st Annual Computational Complexity Conference (CCC 2016)*, volume 50 of *LIPIcs*, pages 35:1–35:29. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[KS17] Mrinal Kumar and Ramprasad Saptharishi. An Exponential Lower Bound for Homogeneous Depth-5 Circuits over Finite Fields. In Ryan O'Donnell, editor, *Proceedings of the 32nd Annual Computational Complexity Conference (CCC 2017)*, volume 79 of *LIPIcs*, pages 31:1–31:30. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. `arXiv:1507.00177`.

[KSS14] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 146–153, 2014.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An Almost Cubic Lower Bound for Depth Three Arithmetic Circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. `eccc:TR16-006`.

[LST21] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 28:81, 2021. `eccc:TR21-081`.

[MB18] Sajal Kumar Mukherjee and Sudip Bera. Combinatorial proofs of the Newton-Girard and Chapman-Costas-Santos identities, 2018. Pre-print available at `arXiv:1807.11749`.

[Mes89] Roy Meshulam. On two extremal matrix problems. *Linear Algebra and its Applications*, 114:261–271, 1989.

[MP08] Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *J. Complexity*, 24(1):16–38, 2008. Preliminary version in the *31st Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2006)*.

[MR04] Thierry Mignon and Nicolas Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notes*, 2004(79):4241–4253, 2004. Available on `citeseer:10.1.1.106.4910`.

[MV97] Meena Mahajan and V. Vinay. A Combinatorial Algorithm for the Determinant. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1997)*, pages 730–738, 1997. Available on `citeseer:10.1.1.31.1673`.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 410–418, 1991. Available on `citeseer:10.1.1.17.5067`.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. Available on `citeseer:10.1.1.83.8416`.

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. Available on `citeseer:10.1.1.90.2644`.

[Pam85] P. Pamfilos. On the maximum rank of a tensor product. *Acta Mathematica Hungarica*, 45(1):95–97, 1985.

[PSS18] Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low-algebraic-rank circuits. *Computational Complexity*, 27(4):617–670, 2018.

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[Raz06] Ran Raz. Separation of Multilinear Circuit and Formula Size. *Theory of Computing*, 2(1):121–135, 2006. Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*.

[Raz09] Ran Raz. Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size. *Journal of the ACM*, 56(2), 2009. Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*.

[Raz10a] Ran Raz. Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory of Computing*, 6(1):135–177, 2010. Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*.

[Raz10b] Ran Raz. How to fool people to work on circuit lower bounds. Invited talk at Microsoft Research, 2010.

[Raz10c] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC 2010)*, pages 659–666, 2010.

[RSY08] Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM Journal of Computing*, 38(4):1624–1647, 2008. Preliminary version in the *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*.

[RY08] Ran Raz and Amir Yehudayoff. Balancing Syntactically Multilinear Arithmetic Circuits. *Computational Complexity*, 17(4):515–535, 2008.

[RY09] Ran Raz and Amir Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity*, 18(2):171–207, 2009. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[Rys63] Herbert J. Ryser. Combinatorial Mathematics. *Mathematical Association of America*, 14, 1963.

[Sap14] Ramprasad Saptharishi. Recent Progress on Arithmetic Circuit Lower Bounds. *Bulletin of the EATCS*, 114, 2014.

[Sax08] Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 60–71, 2008.

[Ser18] Rocco A. Servedio, editor. *Proceedings of the 33rd Annual Computational Complexity Conference (CCC 2018)*, volume 102 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[Sri13] Srikanth Srinivasan. personal communication, 2013.

[SV14] Ramprasad Saptharishi and V. Vinay. An alternate proof of Tavenas' depth reduction. Unpublished manuscript, 2014.

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001. Preliminary version in the *14th Annual IEEE Conference on Computational Complexity (CCC 1999)*.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. Preliminary version in the *38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*.

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 249–261, 1979.

[Val82] Leslie G. Valiant. Reducibility by algebraic projections. *L'Enseignement Mathematique: Logic and Algorithmic, Geneva*, 2:365–380, 1982.

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*.

[vzG86] Joachim von zur Gathen. Permanent and Determinant. In *Foundations of Computer Science (FOCS)*, pages 398–401. IEEE Computer Society, 1986.

[Yeh18] Amir Yehudayoff. Separating Monotone VP and VNP. *Electronic Colloquium on Computational Complexity (ECCC)*, 2018. `eccc:TR18-124`.