Lower bounds Against Constant-Depth Algebraic Circuits

Nutan Limaye* Srikanth Srinivasan† Sébastien Tavenas‡

April 10, 2022

Abstract

Given a polynomial $f(x_1, \ldots, x_N)$, we can wonder how we can evaluate it with a minimal number of arithmetic operations. Indeed, the task of the algorithm designer is to find evolved processes which decrease this number. In the other direction, it could be interesting to be able to prove that there is no algorithm which does the computation in less than some number of operations. But how one could prove something like that? We will present here some methods which allow us to tackle the problem in the case of constant-depth algebraic circuits.

1 Definitions and notations

To show that P $\neq$ NP, it would be sufficient to show that a chosen NP-complete problem is hard to solve: for example, in a given undirected graph, find a Hamiltonian cycle – i.e., a cycle which passes once and only once through each vertex. Let us consider the polynomial

$$\mathrm{HamCyc}_n(x_{1,2}, x_{1,3}, \ldots, x_{n-1,n}) = \sum_{(i_1,i_2,\ldots,i_n) \text{ Hamiltonian cycle of } K_n} x_{i_1,i_2} x_{i_2,i_3} \cdots x_{i_n,i_1}$$

where K_n stands for the complete undirected graph over n vertices (and the variable $x_{i,j}$ can be interpreted as the edge between the vertices i and j). It is a homogeneous polynomial of degree n with $\binom{n}{2}$ variables. Moreover, on a $\{0,1\}$-input, it equals 0 if and only if the corresponding graph (where the edge (i,j) belongs to the graph exactly when $x_{i,j} = 1$) does not contain any Hamiltonian cycle. Consequently, we expect that such a polynomial should be hard to evaluate. A natural approach for doing such calculations is to only compute arithmetic operations on inputs and on constants. Would it be possible to show that, by restricting ourselves to these types of operations, we cannot evaluate such polynomials efficiently?

Algebraic circuits and formulas

We want a model of computation for evaluating polynomials over a field $\mathbb{F}$ where the elementary operations are the arithmetic ones. We will often use capital letters such as X to denote a vector or a set of variables. Let us mimic the constructions of the Boolean circuits and formulas.

An *algebraic circuit* is a directed acyclic graph. The leaves (with fan-in 0) are labelled by variables or constants from the field $\mathbb{F}$. The internal nodes are labelled by the arithmetic

*ITU-Copenhagen. Email: nuli@itu.dk.

†Department of Computer Science, Aarhus University. Email: srikanth@cs.au.dk. On leave from Department of Mathematics, IIT Bombay, India.

‡Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA. Email: sebastien.tavenas@univ-smb.fr.

operations: "+" or "×".[1] Recursively, each gate computes a polynomial. A polynomial is computed by the circuit if it is computed by one of its gates.

If the undirected graph is in fact a tree (i.e., there is no undirected cycle), we will call the model an *algebraic formula*. One can notice that such a representation corresponds exactly to the usual algebraic formulas (hence the name) where the tree structure is given by the operator precedences.

The complexity of the models will be measured by two parameters. First, the *size* of the circuit corresponds to the number of gates in it. It describes the number of elementary operations performed during the process. Then, the *depth* is the length of the longest path in the directed graph. An intuitive interpretation is as follows. Let's say we work with a parallel machine with enough processors, we will launch an elementary calculation of the process as soon as its inputs have been previously calculated. The depth then describes the time of this parallelized computation.

Classes

Following the Boolean approach, we will define classes of polynomials with respect to their hardness. As we still want to compare their asymptotic complexity, we will formally focus on sequences of polynomials. We will consider for example the sequence $(\mathrm{HamCyc}_n)_{n\in\mathbb{N}}$. We will say that the size-complexity of (HamCyc_n) is at most $(s_n) \in \mathbb{N}^{\mathbb{N}}$ if for any n, there is an algebraic circuit C_n of size at most s_n which computes the polynomial HamCyc_n.

In the following we will focus on sequences of polynomials $(P_n) \in \mathbb{F}[X]^{\mathbb{N}}$ where the number of variables ($|X|$) and the degree ($\deg(P_n)$) are polynomially bounded by n. To harmonize the notations, we will denote by N the number of variables. The restriction on the degree might seem artificial, but it avoids that algebraic models become superpolynomially stronger than their Boolean counterparts (indeed, the Boolean models can simulate the algebraic ones by doing the computations bit-wise, which requires us to maintain values with polynomial bit-sizes).

Particularly, we define the class VP to be the class of sequences of polynomials (f_n) such that there exists a polynomial $\pi \in \mathbb{Z}[n]$ upper-bounding for any $n \in \mathbb{N}$

1. the number N of variables of f_n,
2. the degree of f_n,
3. and the size of the smallest algebraic circuit computing f_n.

Then, we define the class VNP to be the class of sequences of polynomials (f_n) such that there exists a sequence $(g_n) \in \mathrm{VP}$ verifying

$$f_n(X) = \sum_{\varepsilon\in\{0,1\}^{\pi(n)}} g_n(X, \varepsilon).$$

More intuitively, a sequence of polynomials is in the class VNP if, given n (in unary) and a monomial m, we can efficiently[2] compute the coefficient of m in f_n. For example, this is the case for our polynomial HamCyc_n, so it belongs to the class VNP.

We define VF similarly to VP by replacing "circuit" by "formula". Clearly, since any formula is also a circuit, it implies that VF is a subset of VP.

[1] One can also allow the circuits to use divisions, but it is possible to show that divisions can be eliminated at a small cost. [Str73]

[2] The constraint here is stronger than required, since it is sufficient that the computation can be done in GapP/poly.

We will finally need a notion of reducibility. Let us consider a polynomial projection: (f_n) is reducible to (g_n) if there is a polynomial π and affine forms $\boldsymbol{\ell}_n \in \mathbb{F}[X]$ such that for any n and any entry X we have $f_n(X) = g_{\pi(n)}(\boldsymbol{\ell}_n(X))$.

One can notice that the defined classes are stable with respect to this reduction.

Let us introduce a polynomial which will interest us particularly in the following. We call it IMM for *Iterated Matrix Multiplication*.

$$\mathrm{IMM}_{n,d} = \sum_{1 \leqslant i_1, i_2, \ldots, i_{d-1} \leqslant n} x^{(1)}_{1,i_1} x^{(2)}_{i_1,i_2} \cdots x^{(d)}_{i_{d-1},1}.$$

It is a homogeneous polynomial of degree d with dn^2 variables. The semantic interpretation of this polynomial is quite clear: if we consider the d $(n \times n)$-matrices $X^{(i)}$ given by the entries $x^{(i)}_{u,v}$, then $\mathrm{IMM}_{n,d}$ is just the $(1,1)$-th entry of the matrix product $X^{(1)} \cdot X^{(2)} \cdots X^{(d)}$.

We can define the class VBP as the class of polynomials which are reducible to $\mathrm{IMM}_{n,n}$. This class is interesting since it contains several natural families of polynomials. For example, the polynomial Det_n obtained by taking the determinant of a matrix of $n \times n$ distinct variables is complete for VBP.

We know that VF $\subseteq$ VBP $\subseteq$ VP $\subseteq$ VNP. But it is still unknown if any one of these inequalities is strict, and even, if VF is strictly contained in VNP. Proving this requires us to prove a superpolynomial *lower bound* on the size of any algebraic formula computing a VNP-complete sequence of polynomials, such as HamCyc_n.

Finally, we will be particularly interested in circuits which have constant depth. We already noticed that formulas are, in particular, circuits. In the other direction, it is always possible to transform a circuit of size s and depth δ into a formula of depth δ by duplicating the subtrees as many times as necessary. The cost of this transformation is that the size blows up to at most s^δ. However, in the setting of constant-depth models, this blow-up is only polynomial. So, if we are looking for superpolynomial lower bounds, it makes no difference whether we consider constant-depth circuits or formulas. Furthermore, notice that it is always possible, up to multiplying the depth by at most two, to have a layered circuit where the top node (of depth 0) is an addition gate, all internal nodes at even depth are additions, and all internal nodes at odd depth are multiplications. In particular, such a depth-2 circuit is a sum of products of variables and constants. We call such a circuit a $\Sigma\Pi$-circuit. Similarly, we will use $\Sigma\Pi\Sigma$, $\Sigma\Pi\Sigma\Pi$, $\Sigma\Pi\Sigma\Pi\Sigma$, ... to denote such depth-3, depth-4, depth-5 circuits and so on.

Homogeneous, Multilinear, and Set-multilinear circuits

It often happens that the polynomials for which we expect to prove lower bounds have additional structure which we can hope to exploit to prove the lower bounds. This structure may be *homogeneity*, which means that all monomials have the same degree d; *multilinearity*, which means that no variable appears with degree more than 1 in any monomial; or *set-multilinearity*, which means that the underlying set of variables is partitioned into some d sets $X_1, \ldots, X_d$, and each monomial contains precisely one variable from each set.[3]

Notice that a set-multilinear polynomial P is both multilinear and homogeneous. The prototypical example of such a polynomial is the Iterated Matrix Multiplication polynomial $\mathrm{IMM}_{n,d}$ where the corresponding partition of the variable set is the partition into the d matrices $X^{(1)}, \ldots, X^{(d)}$. Another important example is the determinant Det_n of an $n \times n$ matrix X, which is set-multilinear w.r.t. the partition of the matrix into its rows, or its columns.

[3]Strictly speaking, we should speak about set-multilinear polynomials *with respect to a given variable partition.* But we assume that the variable partition is known from context.

Given the problem of computing a polynomial P that is structured (homogeneous, multilinear or set-multilinear), it is natural to consider circuits for this problem that are structured in a similar way. This leads to the definition of *homogeneous*, *multilinear* and *set-multilinear* circuits. A *homogeneous* circuit is one where each intermediate gate of the circuit computes a homogenous polynomial (of possibly a different degree than P). We can similarly define multilinear and set-multilinear circuits respectively.

The assumption that the circuit shares the same structure as the polynomial P is *not* without loss of generality. For instance, it is possible that the most efficient circuit of some form that computes a homogeneous polynomial P is *inhomogeneous*, i.e. it possibly computes some intermediate polynomials that are inhomogeneous, but by virtue of some cancellations, the final polynomial computed is indeed the homogeneous polynomial P. For an interesting example of this kind, we refer the reader to Ben-Or's construction described below, which gives quadratic-sized *inhomogeneous* $\Sigma\Pi\Sigma$ formulas of size for the elementary symmetric polynomial $E^n_{n/2}$. On the other hand, Nisan and Wigderson [NW97] showed that any homogeneous $\Sigma\Pi\Sigma$ formula for $E^n_{n/2}$ is of exponential size. Another important example is that of the $n \times n$ determinant polynomial Det_n which is a multilinear polynomial that is known to have (non-multilinear) algebraic circuits of polynomial size, but not known to have multilinear circuits of sub-exponential (i.e. $\exp(n^{o(1)})$) size.

Nevertheless, we can show many statements to the effect that the existence of an efficient (unstructured) algebraic circuit leads to the existence of a somewhat less efficient (in terms of both size and depth) structured algebraic circuit. We refer to such statements as *escalation results,* as they allow us to escalate a lower bound against a (seemingly) weaker family of algebraic circuits to a stronger one. Some examples of such escalation results from the literature are given below.

Theorem 1 (Escalation theorems)**.**

1. *[Str73] If P is a homogeneous polynomial of degree d that has a circuit of size s, then it has a homogeneous circuit of size* $\mathrm{poly}(s, d)$.
2. *[NW97] If P is a set-multilinear polynomial of degree d that has a circuit of size s, then it has a set-multilinear circuit of size* $\mathrm{poly}(s, 2^d)$.
3. *[Raz13] If P is a set-multilinear polynomial of degree d that has a formula of size s, then it has a set-multilinear formula of size* $\mathrm{poly}(s, d^d)$.
4. *[SW01, LST21b] Assume that the field $\mathbb{F}$ has characteristic 0. If P is a homogeneous polynomial of degree d that has a circuit of size s and depth p, then it has a homogeneous circuit of depth $2p - 1$ and size* $\mathrm{poly}(s, 2^{O(\sqrt{d})})$.
5. *[NW97, SW01, CKSV16, LST21b] Assume that the field $\mathbb{F}$ has characteristic 0. If P is a set-multilinear polynomial of degree d that has a circuit of size s and depth p, then it has a set-multilinear circuit of depth $2p - 1$ and size* $\mathrm{poly}(s, d^d)$.

These results imply that strong enough lower bounds against structured algebraic circuit models imply lower bounds against more general models as well. For instance, the first result says that if a sequence of homogeneous polynomials P_n cannot be computed by homogeneous algebraic circuits of polynomial size, then they cannot be computed by any kind of algebraic circuits of polynomial size. The latter three statements yield similar consequences for formulas and bounded-depth circuits, but under the additional condition that the degree d of the polynomials is sufficiently bounded: something like $d = O(\log N / \log \log N)$ is sufficient. For example, the last statement tells us that to prove superpolynomial lower bounds against general depth-3 algebraic circuits, it suffices to prove superpolynomial lower bounds against depth-5 set-multilinear algebraic circuits in this low-degree setting.

Polynomial Identity Testing Problem

We will consider the following algorithmic problem: Given a circuit which computes a polynomial P, determine whether P is identically zero or not. This problem, called *Polynomial Identity Testing (PIT)*, is a fundamental problem in algebraic complexity. Its importance comes from its numerous applications. Fore example, it appears in algorithms for finding perfect matchings in graphs [Lov79, MVV87, FGT19], for primality testing [AB03, AKS04], or for polynomial factoring [KSS14, DSS18]. Several other applications can be found in [SY10].

This problem has mainly two different settings. In the first setting, the algorithm can only access the polynomial P by querying the value of P on inputs of its choice. This is called the *blackbox* model. In this case, finding a PIT algorithm is equivalent to creating an efficient *hitting-set* H, i.e. a set verifying that if P is a non-zero polynomial, then there is an $a \in H$ such that $P(a) \neq 0$. In the second setting, the algorithm has access to the circuit which represents P. This is called the *whitebox* model.

In both settings, however, the problem seems to be quite hard. Until very recently, sub-exponential-time deterministic PIT algorithms were known only in restricted models such as noncommutative algebraic formulas [RS05, GGOW16], depth-3 circuits with bounded top fan-in [DS07, KS11, KS07, SS12] and some restricted versions of depth-4 circuits [KMSV13, SV18, PS20, PS21, DDS21].

The mention of determinism is important since even in the general case, there is a clever randomized algorithm which efficiently solves PIT (based on the DeMillo-Lipton-Schwartz–Zippel lemma [Zip79, Sch80, DL77]). On the other hand, as mentioned above, we struggle to obtain even sub-exponential time deterministic algorithms for restricted families of circuits. One reason for this hardness comes from its intimate connection to algebraic complexity. Indeed, Kabanets and Impagliazzo [KI04] showed that finding an efficient deterministic algorithm for PIT implies that either NEXP $\subsetneq$ P/poly or HamCyc_n has no polynomial-size algebraic circuits. But in the same paper, they also showed an implication in the other direction. More precisely, they showed that one can design quasi-polynomial time deterministic algorithms for PIT from strong enough lower bounds against algebraic circuits.

Moreover, PIT is important even if we restrict ourselves to constant-depth circuits. Indeed, Agrawal and Vinay [AV08] showed that polynomial-time deterministic algorithms for PIT for depth-4 circuits imply exponential lower bounds against arbitrary algebraic circuits (and again by [KI04], quasi-polynomial PIT for general circuits).

2 Power of constant-depth algebraic circuits

Let us consider the model of computation of depth-2 algebraic circuits. Let us search for a candidate for a hard polynomial for this class, i.e., one that does not have such circuits of small size. In fact, we can restrict ourselves to consider only irreducible polynomials (indeed, if $f(x_1, \ldots, x_n)$ is an interesting hard polynomial, then $g(x_1, \ldots, x_n, y) \stackrel{\text{def}}{=} f(x_1, \ldots, x_n) + y$ will be at least as hard than f). In particular, the output of our depth-2 circuit needs to be a sum and the gates at depth 1 are product gates. Hence, such a circuit computes a polynomial as the sum of its monomials. Consequently, given a polynomial with an exponential number of monomials (for example $\mathrm{IMM}_{n,n}$ or HamCyc_n), we can not compute it with a sub-exponential size depth-2 algebraic circuit.

So constant-depth circuits seem to be a weak model of computation and so lower bounds against them should be easy to achieve. We will see that is not really the case. Indeed, even depth-3 circuits are surprisingly powerful.

Relation between constant-depth Boolean and algebraic circuits

Since the 80s, we have known that constant-depth *Boolean* circuits cannot efficiently compute some explicit Boolean functions. This is the case, for example, of the function Parity which evaluates the parity of the number of 1-bits of the input [FSS84, Ajt83, H̊86]. Generally, Boolean lower bounds are harder to achieve than their algebraic counterparts. The reason is that we can usually simulate an algebraic computation with Boolean models (doing the computation on the bits of the entries). However, let us consider the polynomial $f(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i$ which is just the sum of its variables. This computation can be achieved by a depth-1 formula with $n+1$ gates (one internal sum and n leaves). Now, if we want to simulate this computation for an input in $\{0,1\}^n$ with Boolean computations, we notice that the value of the least significant bit of the output is just the parity of the number of 1s of the input. Computing such a parity is exactly a problem which is not doable by sub-exponential size Boolean circuits! Consequently, constant-depth algebraic circuits cannot be efficiently simulated by Boolean constant-depth circuits and so, lower bounds against constant-depth circuits from the Boolean world are not sufficient to imply lower bounds against their algebraic counterparts.

Ben-Or's construction

We saw that constant-depth algebraic circuits can not be simulated by their Boolean analogs. But does it mean that there are natural, seemingly complex polynomials which can be computed by constant-depth algebraic circuits?

Let us consider the family of elementary symmetric polynomials

$$E_d^n(x_1, \ldots, x_n) = \sum_{1 \leqslant i_1 < i_2 < \cdots < i_d \leqslant n} x_{i_1} x_{i_2} \cdots x_{i_d}.$$

This polynomial is an irreducible polynomial that is a sum of $\binom{n}{d}$ monomials. Thus, for $d = n/2$ (say), this polynomial has exponentially many monomials. Hence, it does not have a small depth-2 circuit.

Surprisingly, however, this polynomial can be computed by a small depth-3 circuit. Let us consider this new family of polynomials where t is a fresh new variable

$$E^n(x_1, \ldots, x_n, t) = \prod_{j=1}^{n} (1 + t x_j).$$

Then, if we see E^n as a univariate polynomial in $\mathbb{F}[x_1, \ldots, x_n][t]$, we notice that the polynomial E_d^n is the coefficient of the monomial t^d in E^n. Moreover, the polynomial E^n is easy to compute (a linear-size product of constant-size sums). So the question rises: is it possible to interpolate efficiently the coefficient of an easy polynomial? The answer is yes!

Indeed, for any value of $t \in \{0, 1, 2, \ldots, n\}$,[4] we get an identity

$$E^n(x_1, \ldots, x_n, t) = \sum_{d=0}^{n} E_d^n(x_1, \ldots, x_n) t^d.$$

The matrix of the coefficients which appear in the system (where the $E_d^n(\boldsymbol{x})$ stand for the

[4] We assume here that the field $\mathbb{F}$ is the field of the rational numbers. However, the same idea works as long as the size of the field $\mathbb{F}$ is at least $n+1$.

variables) is just the Vandermonde matrix

$$\mathrm{Vand}_n \stackrel{\text{def}}{=} \mathrm{Vand}(0,1,\dots,n) = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 2^n \\ \vdots & \vdots & & \vdots \\ 1 & n & \cdots & n^n \end{pmatrix}$$

which is invertible. If we call $V_d^n(x_1,\dots,x_n)$ the matrix we get from Vand_n by replacing its dth column by the column vector ${}^t(E^n(\boldsymbol{x},0),E^n(\boldsymbol{x},1),\dots,E^n(\boldsymbol{x},n))$. By Cramer's formula, we obtain that

$$E_d^n = \det(V_d^n(\boldsymbol{x}))/\det(\mathrm{Vand}_n).$$

The denominator is just a constant from $\mathbb{F}$. The numerator is a linear combination of the $(E^n(\boldsymbol{x},i))_{0\leqslant i\leqslant n}$ with coefficients in $\mathbb{F}$. Consequently,

$$E_d^n(\boldsymbol{x}) = \sum_{i=0}^{n} \lambda_i^{n,d} E^n(\boldsymbol{x},i) = \sum_{i=0}^{n} \lambda_i^{n,d} \prod_{j=1}^{n} (1+ix_j)$$

where the $(\lambda_i^{n,d})$ are constants in $\mathbb{F}$.

In conclusion, we see that depth-3 algebraic circuits can compute elementary symmetric polynomials in just quadratic size.

Depth reduction

More generally, we will see now that if we allow constant-depth circuits of *sub-exponential* size, then we get a truly large class of polynomials, which includes the complexity class VP. This family of results goes by the name of *depth-reduction*.

Reduction to logarithmic depth

Let us consider the polynomial $\mathrm{IMM}_{n,d}$ we defined before. For simplicity, assume that d is a power of two. It is well-known that the product of matrices is associative. In particular, for computing a product of d matrices, we can start by computing the product of the $d/2$ first matrices, then the product of the $d/2$ remaining last matrices, and we finish by computing the product of the two matrices thus obtained.

The final product (of two matrices) can be computed by doing at most n^3 operations[5]. Hence, if we have a circuit of size $s_{n,d/2}$ for computing the inputs of a product of $d/2$ matrices, we can achieve a circuit of size at most $2s_{n,d/2}+n^3$ which computes the entries of a product of d matrices. Moreover, the depth of the new circuit is the one from the computation of the product of $d/2$ matrices increased by adding two. Clearly, $s_{n,1}=n^2$, and so by induction, $s_{n,2^k} \leqslant (2^{k+1}-1)n^3$.

Consequently, there exists an algebraic circuit of depth $2\log_2 d$ and of size at most $2dn^3$ which computes the entries of a product of d matrices. In particular, $\mathrm{IMM}_{n,n}$ can be computed by polynomial-size circuits of depth $O(\log n)$.

Consequently, any polynomial from VBP can be computed by algebraic circuits of polynomial size and logarithmic depth. In fact, it has been shown by Valiant, Skyum, Berkowitz,

[5]We only use a cubic bound since it is simpler and does not affect the message. However, it is well-known that this bound can be improved (for example by Strassen's algorithm). The current best bound is $O(n^{2.3728596})$ from [AW21].

and Rackoff [VSBR83] that it is also the case for any polynomial from the class VP. So, to show that a polynomial is not in VP, it is sufficient to show that it cannot be computed by a polynomial-size circuit which has logarithmic depth. Consequently, considering only polynomials where the degree is polynomially bounded, finding super-polynomial lower bounds against logarithmic-depth algebraic circuit is equivalent to getting similar bounds for general algebraic circuits.

Reduction to constant depth

In the previous section, we parallelized circuits to obtain polynomial-size circuits of logarithmic depth. But in fact, we do not even expect that VNP-complete polynomials (like HamCyc) have circuits of sub-exponential size. So, we can wonder if we can get even a stronger parallelization by allowing sub-exponential blow-up for the size (let us say $N^{o(d)}$ blow-ups). In fact, we can again easily see that is the case for $\mathrm{IMM}_{n,d}$.

This time, by grouping the matrices in blocks of size $\sqrt{d}$ (assume for simplicity that $\sqrt{d}$ is an integer), we can easily notice that

$$\mathrm{IMM}_{n,d} = \sum_{1 \leqslant i_1, i_2, \ldots, i_{\sqrt{d}-1} \leqslant n} (X^{(1)} X^{(2)} \cdots X^{(\sqrt{d})})_{1,i_1} \cdots (X^{(d-\sqrt{d}+1)} \cdots X^{(d)})_{i_{\sqrt{d}-1},1}.$$

The inputs of each block of matrix can be obtained by evaluating $\mathrm{IMM}_{n,\sqrt{d}}$. Furthermore, $\mathrm{IMM}_{n,d}$ can be obtained from the entries of the block matrices, by evaluating $\mathrm{IMM}_{n,\sqrt{d}}$ again. Overall, $\mathrm{IMM}_{n,d}$ can be seen as a composition of two layers of $\mathrm{IMM}_{n,\sqrt{d}}$.

As $\mathrm{IMM}_{n,\sqrt{d}}$ has $n^{\sqrt{d}-1}$ monomials, it can be computed by a depth-2 circuit of size at most $dn^{\sqrt{d}-1}$. By composing the circuits, we get a depth-4 circuit of size $(\sqrt{d}+1)(n^2)dn^{\sqrt{d}-1} = n^{O(\sqrt{d})}$ (as long as $n \geqslant 2$).

Furthermore, we can do a similar recursive construction to get a depth-6 algebraic circuit of size $n^{O(d^{1/3})}$, or even for any p, a depth-$2p$ algebraic circuit of size $n^{O(d^{1/p})}$.

Similar to what was done in the logarithmic-depth case, it was shown in a sequence of works ([AV08, Koi12, Tav15]) that such a parallelization can be achieved for any polynomial from VP. That is to say, if f is a polynomial of degree d in VP, then for any $p \in \mathbb{N}_{\geqslant 1}$, the polynomial f can be computed by a depth-$2p$ circuit of size $N^{O(d^{1/p})}$, where N denotes the number of variables of f.

Furthermore, such a parallelization maintains the homogeneity, multilinearity, and set-multilinearity of the underlying circuit. That is, if f has a homogeneous, multilinear, or set-multilinear circuit of polynomial size, then f also has a depth-$2p$ circuit with the same structure and size $N^{O(d^{1/p})}$.

In a breakthrough work, Gupta, Kamath, Kayal, and Saptharishi [GKKS16] showed that we can do even better, assuming that the underlying field $\mathbb{F}$ has characteristic 0 (e.g. when $\mathbb{F}$ is the field of rational numbers). If f is a polynomial of degree d in VP, then for any odd $p \in \mathbb{N}_{\geqslant 1}$, the polynomial f can be computed by a depth-$(p+1)$ circuit of size $N^{O(d^{1/p})}$. In particular, any polynomial from VP can be computed by depth-3 circuits of size $N^{O(\sqrt{d})}$.

However, this last transformation does not maintain at all the homogeneity, multilinearity, or set-multilinearity of the underlying circuit.

Consequently, proving strong enough lower bounds (i.e., of the form $N^{\omega(d^{1/p})}$) against algebraic circuits of depth $p+1$ already implies superpolynomial lower bounds against general circuits. In this sense, proving lower bounds against constant-depth circuits can be seen as a step to obtaining lower bounds against general circuits.

Finally, even the case of circuits of depth-3 seems very interesting. Indeed, we do not expect that a polynomial like HamCyc can be computed by circuits of size $N^{O(\sqrt{d})}$. On the

other hand, any polynomial from VP can be computed by depth-3 circuits of size $N^{O(\sqrt{d})}$. So, characterizing the class of polynomials computed by depth-3 circuits of size $N^{O(\sqrt{d})}$ should be sufficient to separate VP from VNP.

3 Lower bounds against constant-depth circuits

While the problem of proving depth-2 lower bounds is easy, we have seen in the previous section that the depth-3 case is already very interesting, with the potential of even being able to separate VP from VNP. Lower bounds in this setting were first investigated in a very influential work of Nisan and Wigderson [NW97], who proved such results for set-multilinear depth-3 circuits.

Depth-3 set-multilinear circuits and the Partial Derivative Method

We describe here the results of Nisan and Wigderson [NW97] which introduced a method for proving lower bounds against set-multilinear and homogeneous depth-3 algebraic circuit models. This method, called the *Partial Derivative method*, reduces the problem of proving lower bounds for computing the polynomial P to bounding the rank of a certain matrix associated to P (similar methods appeared in earlier work of Nisan [Nis91] and Hyafil [Hya77]). We illustrate this technique with the case of depth-3 (i.e. $\Sigma\Pi\Sigma$) set-multilinear circuits.

Assume we have a set of N variables X that is partitioned into d sets $X_1, \ldots, X_d$. Let P be a set-multilinear polynomial over the variables in X. A depth-3 set-multilinear circuit of size s is (up to polynomial factors) an expression of the following form

$$P = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell_{i,j} \tag{1}$$

where for each i and j, $\ell_{i,j}$ denotes a homogeneous *linear polynomial* in the variables of X_j.

In order to argue that s must be large for a given polynomial P, we analyze a certain matrix associated with P. We start with a toy case to illustrate the general idea. Assume that $d = 2$ and that $X_1 = \{x_1, \ldots, x_n\}$, $X_2 = \{y_1, \ldots, y_n\}$. Consider the polynomial P that is just the *inner-product* between the two variable sets. That is,

$$P = \sum_{i=1}^{n} x_i y_i.$$

Clearly, the expression above gives us a depth-3 (even depth-2) set-multilinear circuit of size $O(n)$ for P. We would like to show that any depth-3 expression for P of the form in (1) must have $s \geqslant n$ terms, proving that this construction is asymptotically tight.

This is easily proved via a matrix argument. For any set-multilinear polynomial Q over the variable sets X_1, X_2, we define the $n \times n$ matrix $M(Q)$ which has as the (i, j)th entry the coefficient of the monomial $x_i y_j$ in the polynomial Q. Then the matrix $M(P)$ is the $n \times n$ identity matrix and hence has rank n. On the other hand, for each summand Q on the right hand side of (1), it can be seen that the corresponding matrix $M(Q)$ has rank (at most) 1. As matrix rank is sub-additive (i.e. the rank of $M_1 + M_2$ is at most the sum of their ranks), we see that the number of summands must be at least n, concluding the proof.

This is a basic case of the partial derivative method of Nisan and Wigderson [NW97]. The reason for the terminology is that the columns of the matrix $M(Q)$ can be interpreted as the coefficients of the partial derivatives $\frac{\partial Q}{\partial y_j}$ $(j \in [n])$ of the polynomial Q.

This method easily generalizes to higher degrees. For any degree d and set-multilinear polynomial Q over $X_1, \ldots, X_d$, we can define a matrix $M(Q)$ analogously as follows. We partition

the variable sets $X_1, \ldots, X_d$ into two families. More precisely, for a subset R of $\{1, \ldots, d\}$, we call the sets X_i ($i \in R$) the *row variable sets* and we call the other sets the *column variable sets*. Note that every set-multilinear monomial m (which contains exactly one variable from each of $X_1, \ldots, X_d$) can be factored uniquely as $m = m_1 \cdot m_2$, where m_1 and m_2 are set-multilinear monomials over the row and column variable sets respectively. The rows and columns of the matrix $M(Q)$ are labelled by the set-multilinear monomials over the row-variable sets and column-variable sets respectively. Given a row labelled by monomial m_1 and a column labelled by monomial m_2, the corresponding entry of the matrix $M(Q)$ is the coefficient of the monomial $m_1 \cdot m_2$ in Q.

As in the toy example, it is easy to see that each summand Q on the right hand size of (1) is associated to matrix of rank 1. On the other hand, it is not hard to find polynomials P such that the matrix $M(P)$ is a high rank matrix of dimension $K \times K$ where $K = N^{\Omega(d)}$. This is true, say, when $P = \mathrm{IMM}_{n,d}$ and $d \leqslant n$. If we define the row-variable sets to be the X_i where i is odd and the column-variable sets to be the X_i where i is even, then the matrix $M(\mathrm{IMM}_{n,d})$ is a diagonal matrix with n^{d-1} many 1s along the diagonal. This implies the following *exponential* lower bound for this very explicitly described polynomial.

Theorem 2 ([NW97])**.** *Any set-multilinear depth-3 circuit for* $\mathrm{IMM}_{n,d}$ *must have size* n^{d-1}*.*

Extensions using Restrictions

The partial derivative method does not immediately extend even to proving lower bounds against set-multilinear circuits of depth 4. This was observed already by Nisan and Wigderson [NW97] who constructed the following example, called the *Product of Inner Products* (PIP) polynomial.

For parameters n, d, assume that we have variable sets $X_1, \ldots, X_d$ where each X_i has size n^2. Assume $X_i = \{x_1^{(i)}, \ldots, x_{n^2}^{(i)}\}$. Define the polynomial $\mathrm{PIP}_{n,d}$ to be the polynomial $P_1 \cdot P_2 \cdots P_{d/2}$ where P_i is the inner product between the variables in X_{2i-1} and X_{2i}. More formally, we have

$$\mathrm{PIP}_{n,d} = \prod_{i=1}^{d/2} \left(\sum_{j=1}^{n^2} x_j^{(2i-1)} \cdot x_j^{(2i)} \right). \tag{2}$$

By construction, this yields a small ($O(N)$-sized) depth-4 circuit for $\mathrm{PIP}_{n,d}$.[6] On the other hand, it is an easy check that if we partition the variable sets as in the case of $\mathrm{IMM}_{n,d}$ above, the matrix $M(\mathrm{PIP}_{n,d})$ is similarly of large (in fact full) rank. Hence, it is not true that whenever the rank of $M(Q)$ is large, then Q does not have small depth-4 set-multilinear circuits.

More generally, for any way of partitioning the d variable sets $X_1, \ldots, X_d$ into two parts, there is a similar "PIP-type" example that constructs a small depth-4 set-multilinear circuit such that $M(Q)$ has large rank.

Many subsequent lower bound results for circuits of depth-4 and above can be seen as formulating ways of circumventing these examples.

Nisan and Wigderson [NW97] showed how to do this via the method of *random restrictions*, which has found a lot of purchase in the setting of Boolean circuits (see e.g. [Sub61, Ajt83, FSS84, H$\mathring{\text{a}}$86]). The way to restrict a set-multilinear polynomial P is to take a subset of the variable sets $X_1, \ldots, X_d$ and set all the variables in these sets to constants from the underlying field $\mathbb{F}$. If this is done in a suitable way, then the computational hardness of P is still preserved. For instance, if $P = \mathrm{IMM}_{n,d}$, we could take a subset S of (say) half the sets and set their variables so that the corresponding matrices become the identity matrix. Under such a restriction, the

[6] Strictly speaking, the construction yields a depth-3 $\Pi\Sigma\Pi$ circuit with a output gate that is a multiplication gate. However, we restrict our circuits to have output gates that are sum gates. We can achieve this by adding a trivial sum gate that has only one input. This yields a $\Sigma\Pi\Sigma\Pi$ circuit.

polynomial P would transform into the polynomial $\mathrm{IMM}_{n,d/2}$, which is not very different from the polynomial we started out with.

However, such restrictions could radically transform a PIP-type polynomial. For instance, if we choose to restrict each matrix with probability $1/2$ independently, it seems very unlikely that a PIP-type polynomial would preserve its structure.[7] More generally, it is possible to show that applying such a random restriction to a depth-4 set-multilinear circuit transforms it, with high probability, to a circuit that does not compute a polynomial of high rank. We can apply the partial derivative method after applying this restriction argument to get a lower bound.

Using this idea, Nisan and Wigderson were able to prove superpolynomial lower bounds against set-multilinear circuits of all constant depths. More precisely, they showed the following.

Theorem 3 ([NW97])**.** *For any constant p and any $n \geqslant 2$, any set-multilinear circuit of depth p for $\mathrm{IMM}_{n,d}$ has size at least $\exp(d^{1/p})$.*

Notice that the above bound is *exponential* when d approaches N. However, for $d \leqslant \log N$, the bound is not even linear in N, which is trivial. This is rather unfortunate in view of our escalation results in Theorem 1. However, we will see later that these bounds can be improved.

We now discuss some modifications of the partial derivative method to proving lower bounds against other kinds of circuits. Starting with a celebrated result of Raz [Raz09], there were many results [Raz06, RY09, RSY08, DMPY12, CLS19, CELS18] on extending the lower bounds of [NW97] to the more general setting of *multilinear* circuits.

The partial derivative method can be extended to the setting of general multilinear polynomials $P(x_1, \ldots, x_N)$ as follows. Recall that a multilinear polynomial is one where no variable appears with degree more than 1. Equivalently, each monomial in P is a product of a subset of its variables. To define the matrix $M(P)$ in this case, we divide the variable set into two parts Y and Z of equal size $N/2$. Every multilinear monomial m factors uniquely as $m_1 \cdot m_2$ where m_1 and m_2 are multilinear monomials in the variables of Y and Z respectively. The partial derivative matrix $M(P)$ is now similarly defined, with the rows labelled by multilinear monomials over Y and the columns by multilinear monomials in Z; the (m_1, m_2)th entry of $M(P)$ is the coefficient of the monomial $m_1 \cdot m_2$ in P. The complexity of the polynomial P is now captured by the rank of the matrix $M(P)$.

By combining this complexity measure with the method of random restrictions, many lower bound results have been shown against multilinear circuits. In particular, in the setting of constant-depth circuits, Raz and Yehudayoff showed the following exponential lower bound.

Theorem 4 ([RY09])**.** *For any constant p, any multilinear circuit of depth p for the $n \times n$ determinant polynomial Det_n must have size $\exp(n^{\Omega(1/p)})$.*

Similar lower bounds have also been shown for variants of $\mathrm{IMM}_{n,d}$ [DMPY12, CLS19, CELS18].

Augmented with restrictions, the partial derivative can also be applied to prove lower bounds against other kinds of constant-depth circuits. To see this, we need to (again) reformulate the partial derivative method in the setting of general polynomials. Let $P(x_1, \ldots, x_N)$ be a polynomial of degree at most d. We think of such a polynomial as a list of its coefficients, that is, as a vector of suitable length $L_{N,d}$[8] with entries from the underlying field $\mathbb{F}$. We define $\partial_k(P)$ to be the set of all partial derivatives of P of order k. This is a set of homogeneous polynomials of degree at most $d - k$, each of which is a vector of dimension $L_{N,d-k}$. We will define $M(P)$

[7] Intuitively, about half the inner products would turn into linear polynomials, which would be quite "un-PIP-like".

[8] More precisely, we have $L_{N,d} = \binom{N+d}{d}$, though we do not need that here.

to be the matrix whose columns are exactly the vectors in $\partial_k(P)$ and the rank of $M(P)$ will measure the complexity of P.

This general version of the partial derivative method was first used by Nisan and Wigderson [NW97] to show exponential lower bounds against *homogeneous* depth-3 circuits computing the elementary symmetric polynomials defined in Section 2. However, the construction of Ben-Or shows that these same polynomials can be computed by (inhomogenous) depth-3 circuits of polynomial size! So, clearly the partial derivative method, as formulated above, cannot be used to prove lower bounds against general depth-3 circuits.

Grigoriev and Karpinski [GK98] showed how to overcome this bottleneck in the setting of a finite field $\mathbb{F}$ of *constant size*, using again the idea of restrictions (of a different kind from the ones described above) along with the partial derivative technique. Here, they proved the following exponential lower bound.

Theorem 5. *Let $\mathbb{F}$ be a finite field of size q, which is a constant. Then, any depth-3 circuit computing the $n \times n$ determinant over $\mathbb{F}$ has size* $\exp(\Omega(n))$.

Their ideas can also be extended to the case of elementary symmetric polynomials (see [Sap15, Chapter 10]).[9] For the case of large or infinite fields (e.g. the rational numbers), Shpilka and Wigderson [SW01] used similar ideas to prove an $\Omega(N^2)$ lower bound against depth-3 circuits computing certain elementary symmetric polynomials.

Homogeneous Circuits and Shifted Partial derivatives

As already noted, the partial derivative method can be used to prove lower bounds against homogeneous circuits of depth 3 [NW97]. However, once again, the PIP construction prevents this method from working at depths 4 and above. Breakthrough results of Kayal [Kay12] and Gupta, Kamath, Kayal and Saptharishi [GKKS14] formulated a modification of this method to prove strong lower bounds against depth-4 circuits. This technique, called the *Shifted Partial Derivative* technique, has since proven to be useful in a variety of contexts.

To define this, we start with the set $\partial_k(P)$ as defined above. For a parameter ℓ, we let $\mathbf{x}^{\leqslant \ell}$ to be the set of all monomials in $x_1, \ldots, x_N$ of degree at most ℓ. We now consider the set

$$\mathbf{x}^{\leqslant \ell} \cdot \partial_k(P) = \{m \cdot Q \mid m \text{ a monomial of degree at most } \ell,\, Q \in \partial_k(P)\}.$$

I.e., each polynomial in the above set is obtained by "shifting" a polynomial $Q \in \partial_k(P)$ by a monomial of degree at most ℓ. The above is a set of polynomials of degree at most $d-k+\ell$. The shifted partial derivative complexity of the polynomial P (with respect to the chosen parameters k and ℓ) is measured by the rank of the matrix with these column vectors.

We refer the reader to a survey of Kayal and Saha [KS18] for a description of this complexity measure in geometric terms.

Using this measure, a result of Gupta, Kamath, Kayal and Saptharishi [GKKS14] showed lower bounds against certain families of homogeneous depth-4 circuits computing the determinant. A series of extensions of this result followed [FLMS15, KLSS17, KS17, KS19, KS16, BC15], including to the family of all homogeneous depth-4 circuits and restricted families of homogeneous depth-5 circuits. We state below two of the strongest such results, due to Kayal and Saha [KS16] and Bera and Chakrabarti [BC15]. While the first lower bound is stronger, the polynomial for which the lower bound is proved is harder to describe.

Theorem 6. *1. [KS16] There is some explicitly described sequence of polynomials P_N of degree $d = N^{\Omega(1)}$ such that any homogeneous depth-5 (i.e. $\Sigma\Pi\Sigma\Pi\Sigma$) circuit for P_N with the bottom-most sum gates having fan-ins at most $N^{0.99}$ must have size $N^{\Omega(\sqrt{d})}$.*

[9] This shows the difference between the case of large fields and fields of constant size.

2. *[BC15] For some $d = N^{\Omega(1)}$, any homogeneous depth-5 (i.e. $\Sigma\Pi\Sigma\Pi\Sigma$) circuit for $\mathrm{IMM}_{n,d}$ with the bottom-most sum gates having fan-ins at most $N^{0.49}$ must have size $N^{\Omega(\sqrt{d})}$.*

(Here, the constants 0.99 and 0.49 can be replaced by any constants less than 1 and 0.5 respectively.)

Note that the above lower bounds require non-trivial upper bounds on the fan-ins of the bottom sum gates of the depth-5 circuits. If we did not have this restriction, along with an escalation result of Shpilka and Wigderson [SW01] (see Item 4 of Theorem 1), we would get exponential lower bounds against depth-3 circuits without any other restrictions. Using this idea, however, Kayal and Saha [KS16] were able to prove lower bounds against depth-3 circuits where the bottom sum gates have restricted fan-ins.

Further, the shifted partial derivative technique does yield some stronger results in the unrestricted depth-3 case. Building on the work of Shpilka and Wigderson [SW01], Kayal, Saha and Tavenas [KST16] showed nearly cubic lower bounds against depth-3 circuits computing a certain explicitly described multilinear polynomial. This was the best explicit lower bound against this model (for non-constant-sized finite fields) until very recently.

Partial Derivatives Strike Back

While the theorems of the previous sections proved strong (exponential) lower bounds against various restricted families of constant-depth circuits, the question of proving superpolynomial lower bounds against even depth-3 circuits (without any other restrictions) over non-constant-sized finite fields remained open until recently. The authors were able to answer this question (for fields of characteristic 0) by proving such lower bounds for the Iterated Matrix Multiplication and determinant polynomials, among others.

Theorem 7. *[LST21b] Assume that the field $\mathbb{F}$ has characteristic 0. Then, for any $d \leqslant \log n$, any depth-3 circuit C computing $\mathrm{IMM}_{n,d}$ has size $n^{\Omega(\sqrt{d})}$. Any depth-3 circuit C computing Det_n has size $n^{\Omega(\sqrt{\log n})}$.*

By the depth reduction result of [GKKS16] described in Section 2, the above lower bound for $\mathrm{IMM}_{n,d}$ is nearly tight. The result for Det_n, however, is far from the best known upper bound, which is $n^{O(\sqrt{n})}$ [GKKS16]. Also, while many of the lower bounds from the previous sections were exponential, the above theorem can only prove a quasi-polynomial lower bound. Nevertheless, it is the first superpolynomial lower bound for general $\Sigma\Pi\Sigma$ circuits.

We sketch the proof of the above theorem, which goes via the strategy of escalation. By Theorem 1, it suffices to prove a lower bound of $n^{\Omega(\sqrt{d})}$ against set-multilinear circuits of depth-5. So we focus on this latter class of circuits.

The method for proving this lower bound harks back to the simplest version of the partial derivative method for set-multilinear polynomials. As there, we work with set-multilinear polynomials P over variable sets $X_1, \ldots, X_d$. To measure the complexity of such a polynomial P, we partition the sets $X_1, \ldots, X_d$ into two categories, the row-variable sets $R \subseteq [d]$ and the column-variable sets $S = [d] \backslash R$, and consider the matrix $M(P)$ as described above. The rank of $M(P)$ will be our measure of the complexity of P.

While the basic set-up is the same, some idea is required to circumvent the PIP-type examples that cause a bottleneck in the Nisan-Wigderson lower bound. Our idea for doing this is surprisingly simple: just ensure that no row-variable set has the same size as a column-variable set! This ensures that we can never take an inner-product between the variables from a row-variable set and those from a column-variable set. More generally, the condition we need is the

following: for any 'small' set R' of row-variable sets and any 'small' set S' of column-variable sets, we have

$$\max\left\{\frac{\prod_{i\in R'}|X_i|}{\prod_{j\in S'}|X_j|}, \frac{\prod_{j\in S'}|X_j|}{\prod_{i\in R'}|X_i|}\right\} \text{ is large.} \tag{3}$$

This ensures that there are no small 'perturbations' of the PIP-type examples that are close to full-rank. Finally, we also ensure that overall the matrix $M(P)$ is square:

$$\prod_{i\in R}|X_i| = \prod_{j\in S}|X_j|.$$

We need the above condition as without it, we would not be able to find *any* polynomial P such that $M(P)$ is full rank. Note that there is some tension between this condition and the one in (3). However, both can be met by setting each $|X_i|$ ($i \in R$) to n and $|X_j|$ ($j \in S$) to n^α where α is a suitably chosen number.[10]

With this idea in place, we now outline the lower bound against set-multilinear depth-5 circuits. A depth-5 set-multilinear circuit of size s can be written as follows.

$$C = \sum_{i=1}^{s} C_{i,1}\cdots C_{i,d_i} \tag{4}$$

where each $C_{i,j}$ is a set-mutilinear depth-3 circuit (w.r.t. to some subset $X_1,\ldots,X_d$) of size at most s. As before, the idea is to show that each summand on the right hand side of (4) has low rank. Fix such a summand, say corresponding to $i = 1$. The argument splits naturally into two cases.

The first case is when each term in the product has small degree, say at most r. In this case, each $C_{1,j}$ depends on at most r of row-variable sets and at most r column variable sets. We consider the partial derivative matrix of each $M(C_{1,j})$ w.r.t. the same partition into row and column variable sets restricted to the variable sets involved in $C_{1,j}$. The property informally described in (3) allows us to infer that this matrix is highly 'imbalanced': i.e. it has many more rows than columns, or vice versa. This allows us to show that the matrix $M(C_{1,j})$ is low-rank. Further, we can show that the matrix $M(C_{1,1}\cdots C_{1,d_1})$ is low-rank because it is the tensor product of $M(C_{1,j})$ for $j \in \{1,\ldots,d_1\}$.

The second case is some term in the product, say $C_{1,1}$, has degree greater than r. In this case, as $C_{1,1}$ is a set-multilinear depth-3 circuit, we simply use the Nisan-Wigderson argument to argue that $M(C_{1,1})$ has very low rank (much lower than what we would expect a general degree-r polynomial to have). This is enough to infer that the matrix $M(C_{1,1}\cdots C_{1,d_1})$ has low rank as well, concluding the proof.

Finally, to derive a lower bound for a fixed polynomial P, we need to show that $M(P)$ has full rank. This can be done quite easily for a suitable choice of P. In fact, we can ensure that P is just a restriction of $\mathrm{IMM}_{n,d}$. This way, we are able to derive a lower bound for $\mathrm{IMM}_{n,d}$. The lower bound for Det_n is derived by reducing $\mathrm{IMM}_{n,d}$ to it using a reduction due to Valiant [Val79].

This basic proof idea of the depth-3 lower bound is simple enough that it extends to larger (but constant) depths in a straightforward manner. We work with low-degree polynomials, where again it suffices to prove lower bounds against constant-depth set-multilinear circuits by Theorem 1. To do this, we use an inductive argument based on the depth. At each stage of the induction, the argument again splits into two cases. In the first case, we use an imbalance

[10] Taking α to be an irrational number such as $1/\sqrt{2}$ is enough to get a slightly weaker lower bound than the one stated in Theorem 7.

condition on the partial derivative matrix to derive an upper bound on its rank. In the second case, we do this via the inductive hypothesis. This idea yields the following result.

Theorem 8. *Assume that the field $\mathbb{F}$ has characteristic 0 and p is a constant. Then, for any $d \leqslant \log n$, any depth-p circuit C computing $\mathrm{IMM}_{n,d}$ has size $n^{d^{\varepsilon}}$ where ε is a constant that depends only on p. Any depth-p circuit C computing Det_n has size $n^{(\log n)^{\varepsilon}}$.*

Applications to PIT

As mentioned in Section 1 above, the problem of obtaining deterministic algorithms for PIT is closely related to the proving lower bounds against algebraic circuits. In particular, it is known that strong enough lower bounds against *general* algebraic circuits would lead to deterministic polynomial-time algorithms for PIT [KI04, GKSS19].

What about the lower bounds of the previous sections? Do they have implications for PIT, at least in the constant-depth setting?

This line of questioning was initiated by Dvir, Shpilka and Yehudayoff [DSY09], which showed that deterministic algorithms for PIT for *some* depth-p circuits would follow from lower bounds against depth-$(p+5)$ circuits. A follow-up result of Chou, Kumar and Solomon [CKS19] proved a similar statement connecting PIT for unrestricted depth-p circuits to lower bounds against depth-$(p+5)$ circuits. With the results of the previous section, we can now apply these theorems to get deterministic PIT algorithms in the setting of constant-depth circuits.

We state below the result of Chou, Kumar and Solomon, which is best suited to our setting.

Theorem 9 ([CKS19])**.** *Let p be any constant. Assume that $d = o(\log N)$ and there is some explicit sequence of degree-d polynomials that do not have depth-$(p+5)$ circuits of size $\mathrm{poly}(N)$. Then, there is a deterministic algorithm for PIT for circuits of depth-p that runs in subexponential (i.e. $\exp(N^{o(1)})$) time on circuits of size $\mathrm{poly}(N)$.*

Applying the above theorem along with the lower bounds for $\mathrm{IMM}_{n,d}$ given by Theorem 11 yields a deterministic subexponential-time algorithm for PIT of all constant-depth circuits.

While there are considerably faster PIT algorithms known for various special families of circuits (see, e.g. the surveys [SY10, Sax09, Sax13]), no non-trivial PIT algorithms were known even for general depth-3 circuits of polynomial size before these results. So this represents a considerable improvement.

4 Lower bounds in the non-constant depth setting

Most lower bounds from Section 3 were proved for computing either the Iterated Matrix Multiplication or the determinant polynomials, both of which lie in the algebraic complexity class VBP. These lower bounds (for example Theorem 8) thus separate VBP from the complexity class of polynomial-sized constant-depth circuits or equivalently, the class of polynomial-sized constant-depth formulas.[11] This leads to the following natural question: can we separate the power of arbitrary formulas (of any depth) from VBP? Equivalently, can we separate the complexity classes VF and VBP?

There are many ways to attack this problem. One natural approach is to try the same proof idea we used for proving constant-depth circuit lower bounds. That is,

(a) Prove an escalation theorem to convert general algebraic formulas into set-multilinear formulas. Specifically, show that if P is a set-multilinear polynomial that has a formula of size $\mathrm{poly}(N)$, then P also has a set-multilinear formula of size $\mathrm{poly}(N)$.

[11] Recall that we can convert circuits to formulas with only a polynomial blow-up in the constant-depth setting.

(b) Prove a superpolynomial lower bound against set-multilinear formulas (of any depth) for a set-multilinear sequence of polynomials from VBP.

Both these steps have been attempted in the works of [Raz09, Raz06, Raz13, DMPY12].

As mentioned earlier in Theorem 1 Item 3, Raz [Raz13] gave an escalation result suitable for (a). The size blow-up in the process is $\text{poly}(N, d^d)$. So if a set-multilinear polynomial P has a (not necessarily set-multilinear) formula of size $\text{poly}(N)$, then it also has a set-multilinear formula of size $\text{poly}(N)$ as long as the degree of P is $O(\log N / \log\log N)$.

This means that it is sufficient to prove a superpolynomial separation between set-multilinear formulas and VBP in part (b) as long as we can do it with respect to a polynomial of degree $O(\log N / \log\log N)$. Unfortunately, this is not yet known. However, many interesting attempts have been made in this direction as well.

Raz [Raz09] studied multilinear formulas of arbitrary depth and proved the first strong lower bound for this model of computation. Specifically, [Raz09] showed the following.

Theorem 10. *Any multilinear (and in particular set-multilinear) formula (of any depth) computing the determinant polynomial* Det_n *must have size* $n^{\Omega(\log n)}$.

In particular, this separates the power of set-multilinear formulas from VBP. Unfortunately, this lower bound does not escalate, as the determinant is a relatively high degree polynomial (Det_n has degree $n = \sqrt{N}$ where N is the number of variables). Therefore, this does not give a separation between general (not necessarily set-multilinear) formulas and VBP.

Dvir, Malod, Perifel, and Yehudayoff [DMPY12] also proved a separation between multilinear formulas and VBP. Their hard polynomial is more special, in that it lies in a *multilinear* version of the complexity class VBP. Unfortunately, this polynomial, like the determinant, also has relatively high degree.

In general, there does not seem to be any easy way to convert these results from the high-degree setting into lower bounds in the low-degree setting. But in some cases, this may be possible. To illustrate this point, let us consider the polynomial $\text{IMM}_{n,n}$. It is a folklore result that $\text{IMM}_{n,n}$ has algebraic formulas of size $n^{O(\log n)}$.[12] Assume that we are able to prove an optimal $n^{\Omega(\log n)}$ formula lower bound for $\text{IMM}_{n,n}$. This is a lower bound for a high-degree polynomial. But such a lower bound would imply something interesting even for $\text{IMM}_{n,d}$, where $d < n$. Specifically, it is known that if $\text{IMM}_{n,d}$ can be computed by set-multilinear formulas of size $n^{o(\log d)}$, then $\text{IMM}_{n,n}$ can also be computed by set-multilinear formulas of size $n^{o(\log n)}$. In the contrapositive, this means that an $n^{\Omega(\log n)}$ lower bound for $\text{IMM}_{n,n}$ is enough to prove an $n^{\Omega(\log d)}$ lower bound for $\text{IMM}_{n,d}$, even for a small value of d. This combined with Raz's escalation would give us the lower bound we desire. As far as we know, this is still open.

Set-multilinear formula lower bounds for $\text{IMM}_{n,n}$ are therefore very interesting. Unfortunately, the results mentioned above [Raz09, DMPY12] do not seem to yield such a lower bound. While the hard polynomials from these results belong to VBP and thus reduce to $\text{IMM}_{n,n}$, these reductions do not preserve the set-multilinearity of the underlying formula.

In a recent result, we proved the first superpolynomial set-multilinear formula lower bound for $\text{IMM}_{n,n}$.

Theorem 11 ([LST21a])**.** *Any set-multilinear formula for* $\text{IMM}_{n,n}$ *requires size* $n^{\Omega(\log\log n)}$.

As mentioned above, improving this bound to $n^{\Omega(\log n)}$ for $\text{IMM}_{n,n}$ would separate general algebraic formulas from ABPs. While we may not be close to proving this yet, the above result gives an interesting corollary for *non-commutative* formulas. A non-commutative formula is an

[12] This can be shown, for example, using the divide-and-conquer argument from the depth-reduction results in Section 2.

algebraic formula over a set of variables that do not commute. One can define non-commutative ABP in a similar way. The above result gives the following corollary.

Corollary 12 ([LST21a]). *Any non-commutative homogeneous formula computing* $\mathrm{IMM}_{n,n}$ *must have size* $n^{\Omega(\log\log n)}$.

This separates non-commutative homogeneous formulas from the non-commutative version of VBP.

5 Summary and future directions

Escalation and lower bounds. Our new lower bounds against constant-depth circuits crucially use escalation results (such as those in Theorem 1) to convert set-multilinear circuit lower bounds to lower bounds against general circuits. Further, it is known that improving these escalation statements can lead to qualitative improvements in our lower bounds. Below are some questions regarding possible improvements of these results. We state the questions and also mention how they can lead to better lower bounds.

- Can we prove a version of Theorem 1 Item 4 above which holds over all fields? If we are able to do that, then it would also imply a similar improvement in Theorem 1 Item 5. These improvements would then show that the lower bound from Theorem 7 holds over all characteristics.
- Can we prove Theorem 1 Item 5 with a size of $\mathrm{poly}(s, 2^{\sqrt{d}})$ instead of $\mathrm{poly}(s, d^d)$? If we are able to prove this, then the lower bound in Theorem 7 would hold for a larger range of d. Specifically, the statement would hold true for $d \leqslant N^{0.99}$, and we would therefore get *exponential* lower bounds against depth-3 algebraic circuits.

Lower bounds and PIT. As discussed above, we can show that any $\Sigma\Pi\Sigma$ circuit computing $\mathrm{IMM}_{n,d}$ must have size $n^{\Omega(\sqrt{d})}$ (as long as $d \leqslant \log n$). We also know that this lower bound is tight for $\Sigma\Pi\Sigma$ circuits. But how about higher depths? More generally, we know that for any constant odd integer p, $\mathrm{IMM}_{n,d}$ has depth-p circuits of size $n^{O(d^{1/(p-1)})}$. Unfortunately, our lower bound is not tight for $p > 3$. To pin-down the complexity of IMM at higher depths is an interesting open question arising from these results. We conjecture that a lower bound of $n^{\Omega(d^{1/(p-1)})}$ holds at depth p for every odd p.

Proving the above conjecture would help us understand the complexity of IMM, which is by itself a fundamental computational problem. Another motivation is the connection between lower bounds and PIT. We saw that constant-depth circuit lower bounds imply deterministic PIT algorithms for constant-depth circuits. Here, the quality of the lower bound dictates the performance of the deterministic algorithm for PIT; the stronger the lower bound, the better the parameter regime of the deterministic PIT algorithm.

As stated above, our current lower bound gives PIT for circuits for all constant depths. In fact, we also get PIT for circuits of depth $o(\log\log d)$. If we prove the conjecture, then we will be able to get deterministic subexponential-time PIT algorithm for circuits of larger depth, in fact for circuits of depth $o(\log d)$.

VP vs. VNP. As we saw earlier, any N-variate, degree-d polynomial in VP can be computed by a $\Sigma\Pi\Sigma$ circuit of size $N^{O(\sqrt{d})}$. So, to prove that VP is not equal to VNP, it suffices to prove that there is a polynomial in VNP such that any $\Sigma\Pi\Sigma$ polynomial for it must have size $N^{\omega(\sqrt{d})}$.

Presently, we have a $\Sigma\Pi\Sigma$ lower bound of $N^{\Omega(\sqrt{d})}$ for $\mathrm{IMM}_{n,d}$. As we noted above, this is optimal. In fact, this bound is optimal for any polynomial in VP. But proving an asymptotically stronger, say $N^{\omega(\sqrt{d})}$, lower bound for a polynomial in VNP is not ruled out, perhaps even using variations of the Partial Derivative method.

References

[AB03] Manindra Agrawal and Somenath Biswas. Primality and identity testing via Chinese remaindering. *Journal of the ACM (JACM)*, 50(4):429–443, 2003.

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

[AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *proceedings of Foundations of Computer Science (FOCS)*, pages 67–75, 2008.

[AW21] Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021.

[BC15] Suman K. Bera and Amit Chakrabarti. A depth-five lower bound for Iterated Matrix Multiplication. In *Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 183–197. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[CELS18] Suryajith Chillara, Christian Engels, Nutan Limaye, and Srikanth Srinivasan. A near-optimal depth-hierarchy theorem for small-depth multilinear circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 934–945. IEEE Computer Society, 2018.

[CKS19] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019.

[CKSV16] Suryajith Chillara, Mrinal Kumar, Ramprasad Saptharishi, and V. Vinay. The chasm at depth four, and tensor rank : Old results, new insights. *CoRR*, abs/1606.04200, 2016.

[CLS19] Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for Iterated Matrix Multiplication with applications. *SIAM J. Comput.*, 48(1):70–92, 2019.

[DDS21] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[DL77] Richard A DeMillo and Richard J Lipton. A probabilistic remark on algebraic program testing. Technical report, Georgia Inst. of Tech Atlanta School of Information and Computer Science, 1977.

[DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In *proceedings of Symposium on Theory of Computing (STOC)*, pages 615–624, 2012.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

[DSS18] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1152–1165, 2018.

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009.

[FGT19] Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. *SIAM Journal on Computing*, 50(3):STOC16–218, 2019.

[FLMS15] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM Journal on Computing*, 44(5):1173–1201, 2015.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984.

[GGOW16] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117. IEEE, 2016.

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 577–582, 1998.

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, December 2014.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM Journal of Computing*, 45(3):1064–1079, 2016.

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness: Treading the borders. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 147–157. IEEE Computer Society, 2019.

[Hå86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, page 6–20, 1986.

[Hya77] Laurent Hyafil. The power of commutativity. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 171–174. IEEE, 1977.

[Kay12] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electron. Colloquium Comput. Complex.*, 19:81, 2012.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complex.*, 13(1-2):1–46, 2004.

[KLSS17] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. *SIAM Journal on Computing*, 46(1):307–335, 2017.

[KMSV13] Zohar S Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM Journal on Computing*, 42(6):2114–2131, 2013.

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012.

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *computational complexity*, 16(2):115–138, 2007.

[KS11] Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011.

[KS16] Neeraj Kayal and Chandan Saha. Lower bounds for depth-three arithmetic circuits with small bottom fanin. *computational complexity*, 25(2):419–454, 2016.

[KS17] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. *SIAM Journal on Computing*, 46(1):336–387, 2017.

[KS18] Neeraj Kayal and Chandan Saha. Guest column: A paradigm for arithmetic circuit lower bounds. *SIGACT News*, 49(1):55–65, 2018.

[KS19] Mrinal Kumar and Ramprasad Saptharishi. The computational power of depth five arithmetic circuits. *SIAM J. Comput.*, 48(1):144–180, 2019.

[KSS14] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 169–180. IEEE, 2014.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An Almost Cubic Lower Bound for Depth Three Arithmetic Circuits. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:15, 2016.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, volume 79, pages 565–574, 1979.

[LST21a] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. *To appear in STOC 2022. Electron. Colloquium Comput. Complex.*, page 94, 2021.

[LST21b] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *FOCS*, 2021.

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

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

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[PS20] Shir Peleg and Amir Shpilka. A generalized sylvester-gallai type theorem for quadratic polynomials. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 8:1–8:33. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[PS21] Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials. In *STOC '21—Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 259–271. ACM, New York, [2021] ©2021.

[Raz06] Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(1):121–135, 2006.

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *Journal of the ACM*, 56(2):8:1–8:17, 2009.

[Raz13] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *Journal of the ACM*, 60(6):40:1–40:15, 2013.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005.

[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.

[RY09] Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009.

[Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

[Sax13] Nitin Saxena. Progress on polynomial identity testing - II. *Electron. Colloquium Comput. Complex.*, page 186, 2013.

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

[SS12] Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012.

[Str73] Volker Strassen. Vermeidung von divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

[Sub61] Bella A. Subbotovskaya. Realization of linear functions by formulas using $\wedge$, $\vee$, $\neg$. In *Doklady Akademii Nauk*, volume 136–3, pages 553–555. Russian Academy of Sciences, 1961.

[SV18] Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. *Combinatorica*, 38(5):1205–1238, 2018.

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[SY10] Amir Shpilka and Amir Yehudayoff. *Arithmetic circuits: A survey of recent results and open questions*. Now Publishers Inc, 2010.

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015.

[Val79] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC)*, pages 249–261. ACM, 1979.

[VSBR83] Leslie G. Valiant, Sven Skyum, Stuart J. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979.