Guest Column: A Paradigm for Arithmetic Circuit Lower Bounds[1]

Neeraj Kayal[2] ***Chandan Saha***[3]

Abstract

How many operations are needed to compute a given polynomial $f(x_1, x_2, \ldots, x_n)$? Answering questions of this form naturally leads us on a search for clever algorithmic techniques to reduce the number of operations required. Simultaneously, it also leads us towards the complementary task of finding techniques and paradigms for proving lower bounds on the minimum number of operations required. In this survey we describe one such paradigm for obtaining lower bounds.

1 Introduction

Arithmetic Models of computation. Polynomials feature in many different places in the mathematical sciences. Here we are interested in their complexity: how many operations are required to compute a given one. For example: how many operations are required to compute the determinant of an $n \times n$ matrix? Such questions are clearly natural from both mathematical and practical standpoints. These have been intensely investigated in the last few decades with much progress achieved in the form of upper bounds, i.e. finding clever ways to compute polynomials using only a few operations. For example, we know how to compute determinants (and therefore also to solve systems of linear equations) with significantly fewer operations compared to the well-known *Gaussian Elimination* algorithm. But very little progress has been made for the complementary task of proving lower bounds, i.e. proving that a large number of operations are required to compute certain polynomials of interest. In this direction, the limited success so far has been in the form of lower bounds for certain restricted classes of circuits. In this article we aim to convey the qualitative intuition behind most of the known lower bound proofs while eschewing some of the quantiative/technical details. In doing this, we make explicit a common paradigm underlying most of the known lower bounds with the hope that doing so might somehow help catalyze further progress on lower bounds. Alternatively, if this paradigm could be ruled out as a viable line of attack then that would be very helpful and insightful as well. We refer the reader to [Wig02] for motivating examples and connections to other areas in computer science and mathematics. A much more extensive treatment can be found in the books [BCS97, vzG88] while some recent surveys

[2]Microsoft Research Lab India, Bangalore 560001, India. neeraka@microsoft.com.

[3]Indian Institute of Science, Bangalore 560012, India. chandan@iisc.ac.in.

[SY10, CKW11, Sap14] give further details and proofs of some recent developments in the area. The rest of this article is organized as follows. We first formally capture arithmetic computation via the notion of *arithmetic circuits* in section 2. We then outline our paradigm/strategy in section 3. We then illustrate this via lower bounds for some restricted circuit classes in sections 4 and 5. The bounds presented here are based on the papers [KSS14] and [Raz09] which in turn build on a long series of works including [VSBR83, Nis91a, NW96, GKKS13].

2 Definitions and Notation

Arithmetic Circuits and Formulas. An arithmetic circuit computes a polynomial function over some underlying field $\mathbb{F}$ via a sequence of operations involving $+$ and $\times$ starting from its inputs $x_1, x_2, \ldots, x_n$. We typically allow arbitrary constants from $\mathbb{F}$ on the incoming edges to a $+$ gate so that a $+$ gate can in fact compute an arbitrary $\mathbb{F}$-linear combination of its inputs. The complexity of a circuit is measured in terms of its size (the number of edges in the corresponding graph) and depth (the maximum length of a path in the corresponding graph). Abusing notation, we will often refer to a family of polynomials $\{f_n(x_1, x_2, \ldots, x_n) \ : \ n \geq 1\}$ via its n-th member f_n. For example, we will say f is a n-variate of degree $d = d(n)$ to mean that f comes from a family of polynomials wherein the n-variate member has degree bounded by $d(n)$. When the underlying graph is tree (equivalently that every node has outdegree at most one), it is called an *arithmetic formula*.

Formal degree. The formal or syntactic degree of a circuit is the formal degree of its output node; the formal degree of a node being defined inductively in the natural manner - leaf nodes labelled with variables (resp. field constants) have formal degree 1 (resp. 0) and every internal $+$ gate (resp. $\times$ gate) is said to have formal degree equal to the maximum of (resp. the sum of) the formal degrees of its children.

Notation. We will denote tuples by boldfaced letters - for example we will typically use $\mathbf{x}$ to denote a tuple of variables and $\mathbf{a}$ to denote a tuple of field elements naturally interpreted as point in $|\mathbf{a}|$-dimensional space. Abusing notation, we will sometimes use $\mathbf{x}$ to also refer to the set of variables in the tuple $\mathbf{x}$. For example, we will say $\mathbf{y}$ *is a subset of the* $\mathbf{x}$ *variables* to refer to a tuple where each component is a variable that occurs in $\mathbf{x}$.

Sets of Polynomials. Let $A, B \subseteq \mathbb{F}[\mathbf{x}]$ be sets of polynomials. A polynomial naturally corresponds to a vector - its coefficient vector - and a set of polynomials A to a matrix whose rows are indexed by the polynomials in A and each row is the coefficient vector of that polynomial. We will denote by $\dim(A)$ the dimension of the vector space spanned by the coefficient vectors of polynomials in A. By the natural correspondence above, $\dim(A)$ also equals the rank of the matrix corresponding to A. We will denote by $A \cdot B$ the set of pairwise products, i.e.

$$A \cdot B \stackrel{\text{def}}{=} \{f(\mathbf{x}) \cdot g(\mathbf{x}) \ : \ f(\mathbf{x}) \in A, g(\mathbf{x}) \in B\} \subseteq \mathbb{F}[\mathbf{x}].$$

3 Overview

In this section we state the paradigm in a abstract way. In the next two sections we then make it concrete by instantiating this paradigm for two classes of arithmetic circuits where superpolynomial lower bounds are known. Let $\mathtt{C}$ be a (sub)class[4] of arithmetic circuits. Suppose we have a n-variate polynomial $f(\mathbf{x})$ of degree $d = d(n)$[5] which we wish to prove is hard for the circuit class $\mathtt{C}$.

1. **Depth Reduction/Simpler representation.** We wish to find representations for polynomials computed by $\mathtt{C}$ that are *easy to analyze*. This step typically involves proving a statement of the following form. If f is in $\mathtt{C}$ then there exists a representation of f of the form

$$f = T_1 + T_2 + \ldots + T_s, \tag{1}$$

   where each T_i is a product of *simple* polynomials and the number of summands s is not too many.

2. **Identifying a Geometric Property $\boldsymbol{\pi}$.** One then tries to identify a *weakness* of such representations by pinpointing interesting geometric properties of the geometric variety of a term T_i. Recall that the variety corresponding to a polynomial T, denoted $\mathbb{V}(T)$, is the the set of all zeroes of the polynomial T, i.e.

$$\mathbb{V}(T) = \{\mathbf{a} \in \mathbb{F}^n \ : \ T(\mathbf{a}) = 0\}.$$

   The intuition is that T_i being a product of simple polynomials should show up in the geometric properties[6] of $\mathbb{V}(T_i)$. We then try to exploit such geometric properties to obtain lower bounds on s via ranks of suitable matrices in the following way.

3. **Translating the property $\boldsymbol{\pi}$ into smallness of rank of a matrix**. Using π as an inspiration, we then try to associate a matrix $M(g)$ to any polynomial g such that the following two properties hold:

   (a) **Linearity**. For any two polynomials g and h and any two constants $\alpha, \beta \in \mathbb{F}$, it holds that $M(\alpha \cdot g + \beta \cdot h) = \alpha \cdot M(g) + \beta \cdot M(h)$, and

   (b) **Smallness of rank**. If the variety of any polynomial g has the property π identified above then the rank of $M(g)$ is significantly smaller than its size.

   In general its not clear that this can be done at all but suppose that it can be. Note that this would immediately imply a lower bound on the number of summands s in equation (1):

$$s \geq \frac{\text{rank}(M(f))}{r}, \quad \text{where } r \stackrel{\text{def}}{=} \max_{i \in [s]} \text{rank}(M(T_i)).$$

   It is even less clear that r will be small enough that this would provide a meaningful lower bound on s but fortunately, as we will see, this does happen for some restricted classes of circuits $\mathtt{C}$.

[4]For the purpose of this overview the reader could think of $\mathtt{C}$ as the class of polynomial-sized arithmetic circuits as this is the class for which we ultimately aspire to prove lower bounds against. In later sections we will take $\mathtt{C}$ to be certain restricted classes of arithmetic circuits.

[5]We are typically interested in the situation when $d(n)$ is (upper bounded by) a polynomial function of n.

[6]By this we refer to the consideration of the zeroes of T over the algebraic closure $\overline{\mathbb{F}}$ of $\mathbb{F}$ and properties therein.

4. **Finding an explicit polynomial f such that rank of $M(f)$ is large.** We finally find an explicit polynomial f such that $M(f)$ has large rank. The matrix $M(f)$ is typically very huge but remarkably one is often able to prove lower bounds on rank of $M(f)$ via two simple tools:

   (a) **Via existence of a large triangular submatrix.** If $M(f)$ contains an upper-triangular square submatrix U (with nonzero entries on the diagonal) then the size of U is a lower bound on the rank of $M(f)$.

   (b) **Via near-orthogonality of the columns of the matrix.** A beautiful lemma commonly attributed to Noga Alon intuitively says that if the columns (or the rows) of a matrix are *almost orthogonal* then the matrix has *nearly full rank*. Specifically for any matrix M over the real numbers:

$$\text{rank}(M) \geq \frac{\text{Tr}(M^T \cdot M)^2}{\text{Tr}((M^T \cdot M)^2)}.$$

4 Regular Formulas

In this section we illustrate the paradigm of section 3 via superpolynomial lower bounds for a class of circuits that we refer to as *regular arithmetic formulas* (based on [KSS14]).

Definition (Regular Arithmetic Formulas). We say that an arithmetic circuit is a regular formula if:

1. The underlying graph is a tree consisting of alternating layers of $+$ and $\times$ gates, and
2. all the nodes at a layer have the same fanin, and
3. the formal degree of the output node is at most a constant factor (say twice) more than d, the degree of the polynomial computed by the formula.

A lower bound of $n^{\Omega(\log n)}$ against regular arithmetic formulas was obtained in [KSS14]. We restate the proof in the framework of section 3.

1. **Depth Reduction.** In the case of regular formulas one reduces to depth four - if Φ is a regular formula of size $2^{o(log^2 d)}$ computing a polynomial f of degree d then for some $t = \Omega(\log d)$ there exists a representation of f of the form

$$f = T_1 + T_2 + \ldots + T_s,$$

   where each T_i is a product of $O(\frac{d}{t})$-many polynomials of degree t and $s = 2^{o\left(\frac{d}{t} \cdot \log d\right)}$.

2. **Identifying a Geometric Property π.** The geometric intuition is best described by working in projective space over an algebraically closed field $\mathbb{F}$. The geometric property that we use is that when T is a product of many polynomials, say

$$T = g_1(\mathbf{x}) \cdot g_2(\mathbf{x}) \cdot \ldots \cdot g_r(\mathbf{x}) \tag{2}$$

then $\mathbb{V}(T)$ has lots of points vanishing with high multiplicity.[7] To see why this is so we first illustrate it with the case of $r = 2$. Assume that

$$T(\mathbf{x}) = g_1(\mathbf{x}) \cdot g_2(\mathbf{x}) \tag{3}$$

Intuitively, for any point $\mathbf{a} \in \mathbb{V}(g_1) \cap \mathbb{V}(g_2)$ the polynomial $T(\mathbf{x}) = g_1(\mathbf{x}) \cdot g_2(\mathbf{x})$ vanishes

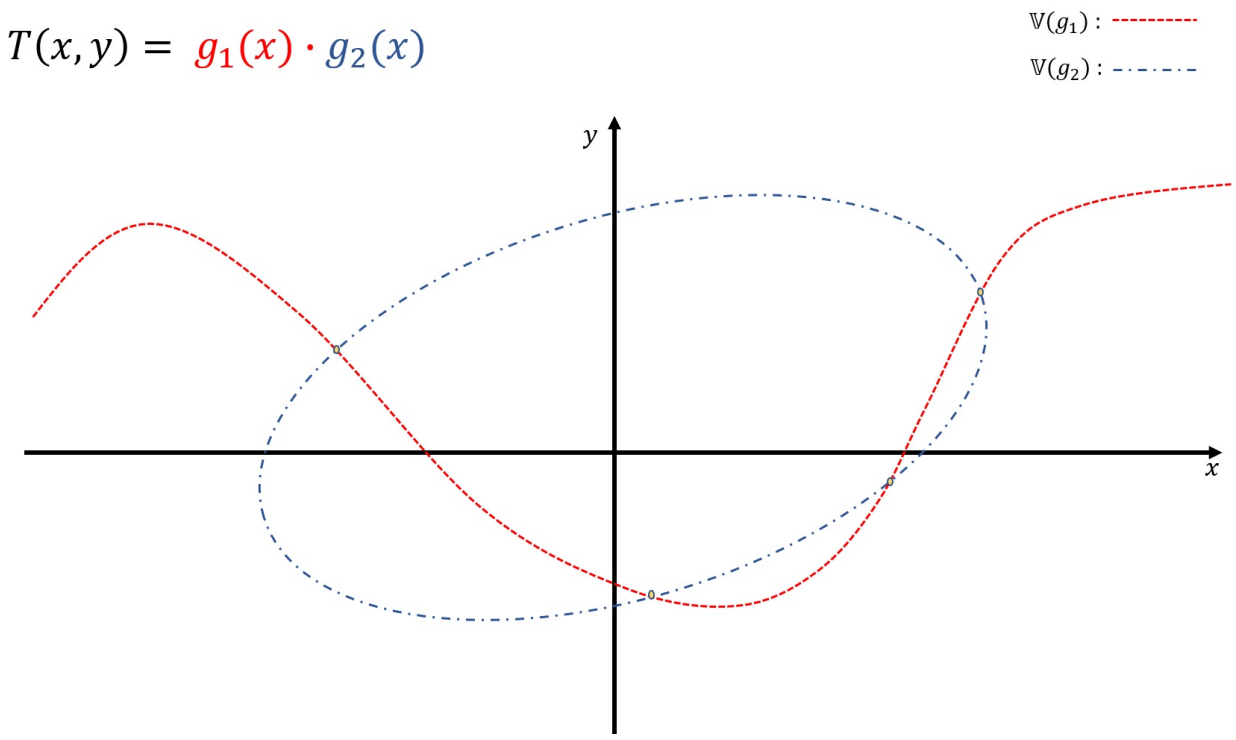

Figure 1: When $r = n = 2$. Intersection points (circled) are points of high multiplicity.

with multiplicity 2 at $\mathbf{a}$ as each of the factors vanish at $\mathbf{a}$. This can be seen pictorially for the case when the ambient space is a plane (i.e. $n = 2$) in figure 1. Multiplicity of a zero naturally corresponds to the vanishing of partial derivatives for which we use the following succinct notation.

Notation: set of partial derivatives and general varieties. For a polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$, we will denote the set of its k-th order partial derivatives by $\partial^{=k} f$, i.e.

$$\partial^{=k} f \stackrel{\text{def}}{=} \left\{ \frac{\partial^k f}{\partial m} \ : \ m \text{ is a monomial over } \mathbf{x} \text{ of degree exactly } k \right\} \subseteq \mathbb{F}[\mathbf{x}].$$

The (geometric) variety of a set of n-variate polynomials $A \subseteq \mathbb{F}[\mathbf{x}]$, denoted $\mathbb{V}(A)$ is defined as the set of points in $\overline{\mathbb{F}}^n$ where each polynomial in A vanishes, i.e.

$$\mathbb{V}(A) \stackrel{\text{def}}{=} \left\{ \mathbf{a} \in \overline{\mathbb{F}}^n \ : \ h(\mathbf{a}) = 0 \quad \text{for all } h(\mathbf{x}) \in A \right\}.$$

[7] Also called high-order singularities.

Our last observation can be formally verified by checking that every partial derivative of T vanishes at such a point $\mathbf{a}$. Differentiating equation (2) with respect to any x_i and applying product rule we have

$$\begin{aligned}
\frac{\partial T}{\partial x_i}(\mathbf{x}) &= \frac{\partial g_1}{\partial x_i}(\mathbf{x}) \cdot g_2(\mathbf{x}) + g_1(\mathbf{x}) \cdot \frac{\partial g_2}{\partial x_i}(\mathbf{x}) \\
\frac{\partial T}{\partial x_i}(\mathbf{a}) &= \frac{\partial g_1}{\partial x_i}(\mathbf{a}) \cdot g_2(\mathbf{a}) + g_1(\mathbf{a}) \cdot \frac{\partial g_2}{\partial x_i}(\mathbf{a}) \quad \text{(substituting } \mathbf{a} \text{ for } \mathbf{x}\text{)} \\
&= 0 \qquad (\text{as } g_1(\mathbf{a}) = g_2(\mathbf{a}) = 0)
\end{aligned}$$

Now since we are in projective n-dimensional space over an algebraically closed field, $\mathbb{V}(g_1) \cap \mathbb{V}(g_2)$ has dimenson at least[8] $(n-2)$. The above discussion can be summarized as saying that for $T = g_1 \cdot g_2$, the variety of first order partial derivatives, denoted $\mathbb{V}(\partial^{=1} T)$, has a large number of points. Specifically,

$$\dim(\mathbb{V}(\partial^{=1} T)) \geq (n-2).$$

This easily generalizes to larger values of r in the following way. For any positive integer $k \leq r$ we have

$$\dim(\mathbb{V}(\partial^{=k} T)) \geq (n-k-1).$$

This also formally captures our claim that when T is a product of many polynomials then $\mathbb{V}(T)$ contains many points of high multiplicity.

3. **Translating the property π into smallness of rank of a matrix**. We will use the correspondence between geometry and algebra (sometimes called the algebra-geometry dictionary) originating in the works of Hilbert to do this. First a piece of notation.

   Notation: set of monomials of a given degree. We will denote by $\mathbf{x}^{=\ell}$ the set of monomials in the $\mathbf{x}$ variables *of degree exactly* ℓ, i.e. if $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ then

   $$\mathbf{x}^{=\ell} \stackrel{\text{def}}{=} \{x_1^{e_1} \cdot x_2^{e_2} \cdot \ldots \cdot x_n^{e_n} \ : \ \text{each } e_i \in \mathbb{Z}_{\geq 0} \text{ and } e_1 + e_2 + \ldots + e_n = \ell\} \subseteq \mathbb{F}[\mathbf{x}].$$

   We now observe that when $\mathbb{V}(T)$ has many points of high multiplicity then $\dim((\mathbf{x}^{=\ell}) \cdot (\partial^{=k} T))$ tends to be small. We give the qualitative intuition here while leaving the details and the quantitative bounds to [KSS14]. Let $V \subseteq \mathbb{F}^n$ be any set of points. For any integer $\ell \geq 1$, let

   $$H_\ell(V) \stackrel{\text{def}}{=} \{h(\mathbf{x}) \ : \ \deg(h) \leq \ell \text{ and } h(\mathbf{a}) = 0 \text{ for all } \mathbf{a} \in V\}.$$

   Note that $H_\ell(V)$ is a vector space whose dimension decreases as V increases, i.e. adding a point to V *might* reduce the dimension of $H_\ell(V)$ but can never increase this dimension. Thus if V is a large set of points then $H_\ell(V)$ ought to be low-dimensional. If we now set V to be the large set of points $\mathbb{V}(\partial^{=k} T)$ then $H_\ell(V)$ ought to be low-dimensional for all ℓ. Finally, note that the set of polynomials $(\mathbf{x}^{=\ell}) \cdot (\partial^{=k} T)$ is a subset of $H_{\ell + \deg(T) - k}(V)$, and intuitively therefore it should have a relatively low dimension as well.

[8] See [CLO07] for the definition of the dimension of a variety and the theorem that in projective space over an algebraically closed field, the intersection of two varieties of dimension $(n-a)$ and $(n-b)$ is at least $(n-a-b)$.

4. **Finding an explicit polynomial f such that rank of $M(f)$ is large.** It turns out that if we have a polynomial f which has a large number of monomials that are pairwise *almost coprime* (i.e. the degree of the gcd of any two monomials is small compared to the degree of f) then $M(f)$ tends to have a large number of nearly orthogonal columns. This leads to the desired lower bound on the rank of $M(f)$. Such an explicit f can now be obtained via known constructions of set systems with low pairwise intersection, such as the beautiful construction due to Nisan and Wigderson [NW94].

5 Multilinear Formulas

Many polynomials of interest such as the determinant (denoted Det_n) and the permanent (denoted Perm_n) have the property that the degree with respect to any variable in it is at most 1. Such polynomials are called multilinear polynomials. The best known circuit for computing the determinant has the property that many intermediate nodes compute non-multilinear polynomials but the non-multilinear terms generated at intermediate stages cancel out leaving only a multilinear polynomial at the end. It is natural to wonder if efficient computation of a multilinear polynomial like Det_n requires that we necessarily must compute intermediate polynomials which are not multilinear. We first formally capture this via notion of a (syntactic) multilinear circuit wherein the requirement of multilinearity of intermediate polynomials is enforced in a syntactic fashion. A circuit is said to be (syntactically) multilinear if the formal degree of any node with respect to any variable is at most 1.[9] We do not know an answer to the question posed above.

Open Problem 1. *Can the determinant Det_n be computed by a $\mathsf{poly}(n)$-sized multilinear circuit?*

A further motivation for multilinear circuits is that they are a natural generalization of monotone circuits[10] but unlike the latter a superpolynomial lower bound for multilinear circuits has remained elusive.

Open Problem 2. *Prove superpolynomial lower bounds for multilinear arithmetic circuits (for an explicit family of multilinear polynomials).*

A very significant piece of progress was obtained by Ran Raz in [Raz09] who showed a superpolynomial lower bound for multilinear *formulas*. We sketch[11] this result here in the form of the paradigm of section 3.

1. **Depth Reduction.** This reduction is best described using the notion of a *log-product* polynomial. We will say that a n-variate polynomial $T(\mathbf{x})$ is a log-product polynomial if it is a product of $r = \frac{\log n}{100}$ polynomials $g_1, g_2, \ldots, g_r$ so that the variable set $\mathbf{x}$ can be partitioned into r sets
$$\mathbf{x} = \mathbf{x}_1 \uplus \mathbf{x}_2 \uplus \ldots \uplus \mathbf{x}_r$$
each of size at least $n^{1/2}$ where every g_i is in the variables $\mathbf{x}_i$. Thus
$$T(\mathbf{x}) = g_1(\mathbf{x}_1) \cdot g_2(\mathbf{x}_2) \cdot \ldots \cdot g_r(\mathbf{x}_r).$$

[9] In other words, a circuit is syntactically multilinear if at every mutliplication node v, for every variable x_i, there is at most one child of v that has an incoming path from a leaf labelled with x_i.

[10] In the sense that monotone circuits for multilinear polynomials are necessarily multilinear circuits as well.

[11] The sketch presented here is also partly based on the proof in the survey [SY10].

In the case of multilinear formulas one reduces to a sum of log-product polynomials - if Φ is a multilinear formula of size s computing a n-variate multilinear polynomial f then one can rewrite f as

$$f = T_1 + T_2 + \ldots + T_s,$$

where each T_i is a *log-product* polynomial.

2. **Identifying a Geometric Property π.** The geometric intuition is best described by working over an algebraically closed field $\mathbb{F}$. The geometric property that we use is that when T is a log-product polynomial, say

$$T = g_1(\mathbf{x}_1) \cdot g_2(\mathbf{x}_2) \cdot \ldots \cdot g_r(\mathbf{x}_r) \tag{4}$$

then $\mathbb{V}(T)$ has lots of axis-parallel affine subspaces.[12] To see why this is so we first explain it for the case of $r = n = 2$. Assume that

$$T(x, y) = g_1(x) \cdot g_2(y) \tag{5}$$

Suppose that $g_1(x)$ has roots $a_1, a_2, \ldots, a_{d_1}$ while $g_2(y)$ has roots $b_1, b_2, \ldots, b_{d_2}$. Then $\mathbb{V}(T)$

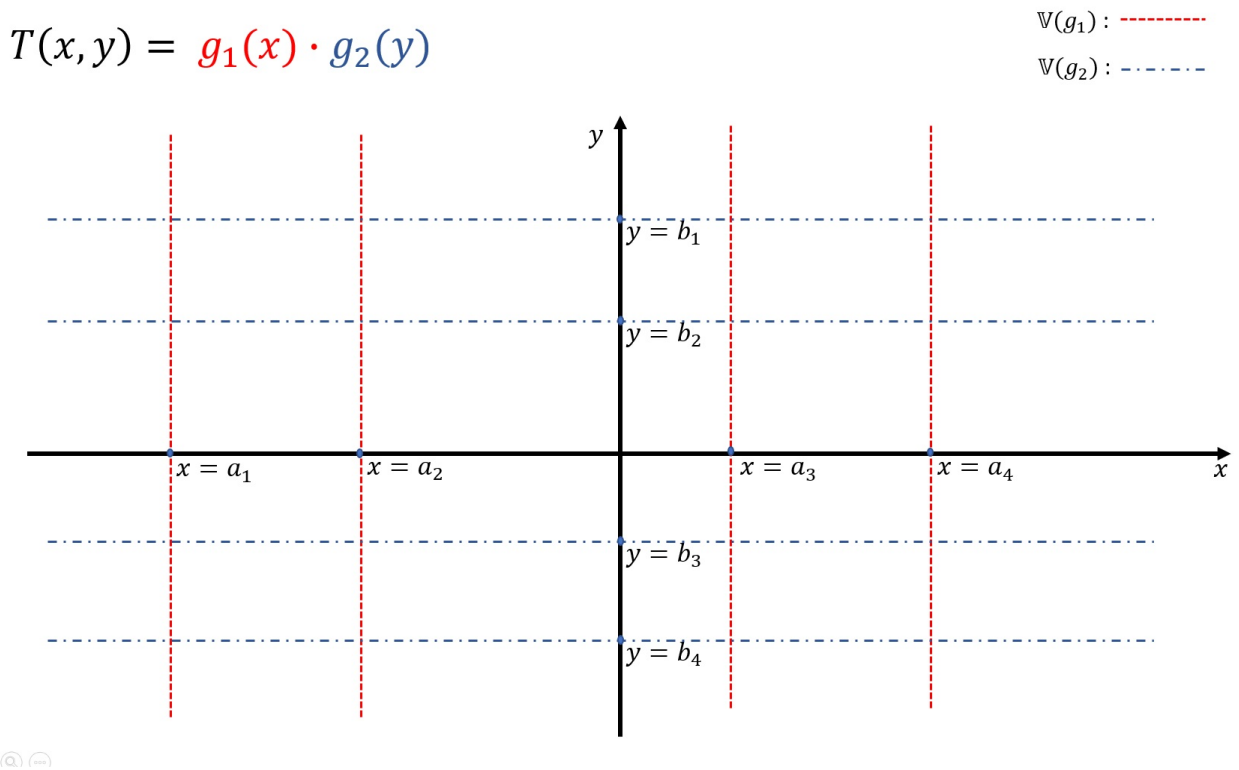

Figure 2: When $r = n = 2$. The variety is a union of horizontal and vertical lines.

is the union of d_1 lines parallel to the y-axis (the y-parallel lines are $\{\mathbb{V}(x - a_i) \ : \ i \in [d_1]\}$)

[12] Recall that an affine space is just a vector space translated by a fixed point. So for example, lines are one-dimensional affine subspaces, planes are two-dimensional affine subspaces and so on.

and d_2-lines parallel to the x-axis (the x-parallel lines are $\{\mathbb{V}(y - b_j) : j \in [d_2]\}$). This is illustrated visually in figure 1. This easily generalizes to larger values of r in the following way. But first a piece of notation.

Notation: Varieties corresponding to restrictions of subsets of variables. For $\mathbf{y} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ subset of $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and a point $\mathbf{a} = (a_1, a_2, \ldots, a_m) \in \overline{\mathbb{F}}^m$, $\mathbb{V}(\mathbf{y} = \mathbf{a})$ shall denote the variety $\mathbb{V}(\{x_{i_1} - a_1, x_{i_1} - a_1, \ldots, x_{i_m} - a_m\})$. Note that for any $\mathbf{a} \in \overline{\mathbb{F}}^m$, $\mathbb{V}(\mathbf{y} = \mathbf{a})$ is an affine subspace in $\overline{\mathbb{F}}^n$ parallel to the (linear) subspace $\mathbb{V}(\mathbf{y} = \mathbf{0})$ which is spanned by the coordinate axes corresponding to variables in $\mathbf{x} \setminus \mathbf{y}$.

Now when $T(\mathbf{x})$ is of the form given by equation (4) then each zero $\mathbf{a}_i$ of a factor $g_i(\mathbf{x}_i)$ corresponds to an affine subspace $\mathbb{V}(\mathbf{x}_i = \mathbf{a}_i)$ inside $\mathbb{V}(T)$ and so $\mathbb{V}(T)$ contains a lot of axis-parallel affine subspaces.

3. **Translating the property π into smallness of rank of a matrix**. Let $\mathbf{x} = \mathbf{y} \uplus \mathbf{z}$ be some partition of our variable set $\mathbf{x}$ into disjoint sets $\mathbf{y}$ and $\mathbf{z}$. Now write polynomial $T(\mathbf{x}) = T(\mathbf{y}, \mathbf{z})$ as

$$T(\mathbf{y}, \mathbf{z}) = \sum_{i=1}^{s} p_i(\mathbf{y}) \cdot q_i(\mathbf{z}), \tag{6}$$

where the p_i's and q_i's are polynomials over the indicated variable sets. Such a representation always exists[13] but is not unique. Lets fix any one such representation. Now an affine subspace $\mathbb{V}(\mathbf{y} = \mathbf{a})$ parallel to the $\mathbf{z}$-axes is contained in $\mathbb{V}(T)$ if and only if $T(\mathbf{a}, \mathbf{z}) = 0$ identically as a polynomial. Or equivalently, that

$$\sum_{i=1}^{s} p_i(\mathbf{a}) \cdot q_i(\mathbf{z}) = 0,$$

Since each $p_i(\mathbf{a})$ is in the underlying field $\mathbb{F}$, we obtain a $\mathbb{F}$-linear dependence among the q_i's. This means that intuitively, *a lot of* $\mathbf{z}$-parallel subspaces correspond to *a lot of* linear dependencies among the q_i's in any repesentation of the form (6). Stated differently, *a lot of* $\mathbf{z}$-parallel subspaces correspond to only a few $\mathbb{F}$-linearly independent q_i's. This is captured by the following vector space of polynomials derived out of $T(\mathbf{y}, \mathbf{z})$.

Definition 1. *For any polynomial $T(\mathbf{y}, \mathbf{z}) \in \mathbb{F}[\mathbf{y}, \mathbf{z}]$, define*

$$\mathsf{evalDim}_{\mathbf{y}}(T) = \dim\left(\left\{T(\mathbf{a}, \mathbf{z}) \ : \ \mathbf{a} \in \mathbb{F}^{|\mathbf{y}|}\right\}\right).$$

‘ $\mathsf{evalDim}_{\mathbf{y}}(T)$ is always finite and it has the following equivalent definitons.

Proposition 3. *For any polynomial $T(\mathbf{y}, \mathbf{z}) \in \mathbb{F}[\mathbf{y}, \mathbf{z}]$, the following quantities are equal:*

(a) $\mathsf{evalDim}_{\mathbf{y}}(T)$

(b) $\mathsf{evalDim}_{\mathbf{z}}(T)$

[13] To see the existence of a representation as given in equation (6), note that one can simply choose p_i's and q_i's to be scalar multiples of all monomials of the appropriate degree over the indicated variable sets.

(c) Number of linearly independent q_i's in any representation of T of the form (6).

(d) Number of linearly independent p_i's in any representation of T of the form (6).

This definition also means that $\mathsf{evalDim}_{\mathbf{y}}(T)$ is equal to the rank of a corresponding matrix - call it $M(T)$ - whose rows correspond to the polynomials in the set $\left\{T(\mathbf{a},\mathbf{z}) \ : \ \mathbf{a} \in \mathbb{F}^{|\mathbf{y}|}\right\}$. The above discussion suggests that if $\mathbb{V}(T)$ has lots of $\mathbf{z}$-parallel subspaces then rank of $M(T)$ ought to be small. This matrix was used by Nisan [Nis91b] and by Nisan and Wigderson [NW94] to prove lower bounds for some simpler models of computation such as non-commutative branching programs. Inspired by their work, Raz [Raz09] observed that if T is a log-product polynomial then for a randomly chosen subset of variables $\mathbf{y}$, $\mathsf{evalDim}_{\mathbf{y}}(T)$ is relatively small. Specifically,

Proposition 4. *Let $T(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a n-variate multilinear log-product polynomial. Then for a random choice of subset of variables $\mathbf{y} \subseteq \mathbf{x}$ of size $\frac{n}{2}$, the probability that $\mathsf{evalDim}_{\mathbf{y}}(T) \geq 2^{n/2-n^{1/16}}$ is at most $n^{-\Omega(r)} = n^{-\Omega(\log n)}$.*

Combined with the depth reduction for multilinear formulas as in step 1 above, it means that if a n-variate multilinear polynomial f satisfies

$$\mathsf{evalDim}_{\mathbf{y}}(f) = 2^{n/2} \quad \text{for every } \mathbf{y} \subseteq \mathbf{x}, |\mathbf{y}| = n/2 \tag{7}$$

then any multilinear formula for f must have size at least $n^{\Omega(\log n)}$.

4. **Finding an explicit polynomial f such that rank of $M(f)$ is large.** Raz also showed that both the determinant Det and the permanent Perm satisfy the property (7) (in a certain, more general, sense). This implies that any multilinear formula for Det or Perm must have superpolynomial size.

6 Discussion

We made explicit here a paradigm for arithmetic circuit lower bounds that captures most such lower bounds that are known. We outlined how this can be implemented for two subclasses of arithmetic circuits. We hope that this can be successfully implemented for more general and interesting classes of circuits. A very recent piece of work [EGOW17] indicates that the absolutely best possible lower bounds might be unattainable via rank-based methods such as the one outlined in section 3. Specifically, they consider tensors in d sets of n variables each (so a total of nd variables) and *unconditionally* show that any rank-based method such as the ones espoused here cannot prove a lower bound of more than $2^d \cdot n^{\lfloor d/2 \rfloor}$ on the rank of such tensors. In comparison a random tensor of these dimensions has rank $\frac{n^{(d-1)}}{d}$. This indicates that rank-based methods such as the ones described here might not be powerful enough to prove optimal lower bounds for various classes of circuits. One hopes however that the paradigm described might be powerful enough to prove a *mere* superpolynomial lower bound for arithmetic circuits.

References

[BCS97] P. Bürgisser, M. Clausen, and A. Shokrollahi. *Algebraic Complexity Theory*. Springer, 1997.

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial Derivatives in Arithmetic Complexity and Beyond. *Foundations and Trends in Theoretical Computer Science*, 6(1-2):1–138, 2011.

[CLO07] D.A. Cox, J.B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Undergraduate texts in mathematics. Springer, 2007.

[EGOW17] Klim Efremenko, Ankit Garg, Rafael Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, TR17:162, 2017.

[GKKS13] Ankit Gupta, Neeraj Kayal, Pritish Kamath, and Ramprasad Saptharishi. Approaching the chasm at depth four. In *Conference on Computational Complexity (CCC)*, pages 65–73, 2013.

[KSS14] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *STOC*, pages 146–153, 2014.

[Nis91a] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *STOC*, pages 410–418, 1991.

[Nis91b] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *STOC*, pages 410–418, 1991.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[NW96] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1996.

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009.

[Sap14] Ramprasad Saptharishi. Recent progress on arithmetic circuit lower bounds. *Bulletin of the EATCS*, 114, 2014.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[VSBR83] L.G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM Journal on Computing*, 12(4):641–644, 1983.

[vzG88] Joachim von zur Gathen. Annual review of computer science: vol. 3, 1988. chapter Algebraic complexity theory, pages 317–347. Annual Reviews Inc., Palo Alto, CA, USA, 1988.

[Wig02] Avi Wigderson. Arithmetic complexity -a survey. 2002. Available at `http://www.math.ias.edu/avi/books`.