Electronic Colloquium on Computational Complexity, Report No. 170 (2019)

A Quadratic Lower Bound for Algebraic Branching Programs

Prerona Chatterjee* Mrinal Kumar† Adrian She‡ Ben Lee Volk §

Abstract

We show that any Algebraic Branching Program (ABP) computing the polynomial $\sum_{i=1}^{n} x_i^n$ has at least $\Omega(n^2)$ vertices. This improves upon the lower bound of $\Omega(n \log n)$, which follows from the classical result of Baur and Strassen [Str73a, BS83], and extends the results in [Kum19], which showed a quadratic lower bound for *homogeneous* ABPs computing the same polynomial.

Our proof relies on a notion of depth reduction which is reminiscent of similar statements in the context of matrix rigidity, and shows that any small enough ABP computing the polynomial $\sum_{i=1}^{n} x_i^n$ can be depth reduced to essentially a homogeneous ABP of the same size which computes the polynomial $\sum_{i=1}^{n} x_i^n + \varepsilon(\mathbf{x})$, for a structured "error polynomial" $\varepsilon(\mathbf{x})$. To complete the proof, we then observe that the lower bound in [Kum19] is robust enough and continues to hold for all polynomials $\sum_{i=1}^{n} x_i^n + \varepsilon(\mathbf{x})$, where $\varepsilon(\mathbf{x})$ has the appropriate structure.

*Tata Institute of Fundamental Research, Mumbai, India. Email: prerona.chatterjee.tifr@gmail.com. Research supported by the Department of Atomic Energy, Government of India, under project no. 12-R&D-TFR-5.01-0500.

†Dept. of Computer Science & Engineering, IIT Bombay, India. Email: mrinal@cse.iitb.ac.in. A part of this work was done during a postdoctoral stay at University of Toronto.

‡Dept. of Computer Science, University of Toronto. Email : ashe@cs.toronto.edu.

§Center for the Mathematics of Information, California Institute of Technology, USA. Email:benleevolk@gmail.com.

ISSN 1433-8092

1 Introduction

Proving that there are explicit polynomials which are hard to compute is the template of many open problems in algebraic complexity theory. Various instances of this problem involve different definitions of explicitness, hardness and computation.

In the most general form, this is the well known VP vs. VNP question, which asks whether every "explicit" polynomial has a polynomial-size algebraic circuit. An algebraic circuit is a very natural (and the most general) algebraic computational model. Informally, it is a computational device which is given a set of indeterminates $\{x_1, \ldots, x_n\}$, and it can use additions and multiplications (as well as field scalars) to compute a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$. The complexity of the circuit is then measured by the number of operations the circuit performs.

It is trivial to give an explicit n-variate polynomial which requires circuits of size $\Omega(n)$. It is also not hard to show that a degree-d polynomial requires circuits of size $\Omega(\log d)$, since the degree can at most double in each operation. Thus, one trivially obtains a $\max\{n, \log d\} = \Omega(n + \log d)$ lower bound for an n-variate degree-d polynomial.

A major result of Baur and Strassen [Str73a, BS83] gives an explicit n-variate degree-d polynomial which requires circuits of size at least $\Omega(n \cdot \log d)$. On the one hand, this is quite impressive since when $d = \text{poly}(n)$, this gives lower bound which is super-linear in n. Such lower bounds for explicit functions in the analogous model of *boolean* circuits are a long-standing and important open problem in boolean circuit complexity. On the other hand, this lower bound is barely super-linear, whereas ideally one would hope to prove super-polynomial or even exponential lower bounds (indeed, it can be proved that "most" polynomials require circuits of size exponential in n).

Despite decades of work, this lower bound has not been improved, even though it has been reproved (using different techniques [Smo97, Ben83]). Most of the works thus deal with restricted models of algebraic computation. For some, there exist exponential or at least super-polynomial lower bounds. For other, more powerful models, merely improved polynomial lower bound. We refer the reader to [Sap15] for a comprehensive survey of lower bounds in algebraic complexity.

One such restricted model of computation for which we have better lower bounds is *algebraic formulas*. Formulas are simply circuits whose underlying graph is a tree. Kalorkoti [Kal85] has shown how to adapt Nechiporuk's method [Nec66], originally developed for boolean formulas, to prove an almost quadratic lower bound for an n-variate polynomial.[1] This is also the best lower bound obtainable using this technique.

[1] In his paper, Kalorkoti proves an $\Omega(n^3)$ lower bound for the $n \times n$ determinant, which has n^2 variables, so the lower bound not quadratic in the number of variables. However, it is possible to get the statement claimed here using a straightforward application of his techniques.

1.1 Algebraic Branching Programs

Algebraic Branching Programs (ABPs, for short), defined below, are an intermediate model between algebraic formulas and algebraic circuits. To within polynomial factors, algebraic formulas can be simulated by ABPs, and ABPs can be simulated by circuits. It is believed that each of the reverse transformations requires a super-polynomial blow-up in the size (for some restricted models of computation, this is a known fact [Nis91, Raz06, RY08, DMPY12, HY16]).

Polynomial families which can be efficiently computed by algebraic branching programs form the complexity class VBP, and the determinant is a complete polynomial for this class under an appropriate notion of reductions. Thus, the famous Permanent vs. Determinant problem, unbeknownst to many, is in fact equivalent to showing super-polynomial lower bound for ABPs. In this paper, we focus on the question of proving lower bounds on the size of algebraic branching programs for explicit polynomial families. We start by formally defining an algebraic branching program.

Definition 1.1 (Algebraic Branching Programs)**.** *An Algebraic Branching Program (ABP) is a layered graph where each edge is labeled by an affine linear form and the first and the last layer have one vertex each, called the "start" and the "end" vertex respectively.*

The polynomial computed by an ABP is equal to the sum of the weights of all paths from the start vertex to the end vertex in the ABP, where the weight of a path is equal to the product of the labels of all the edges on it.

The size of an ABP is the number of vertices in it. ◊

While Definition 1.1 is quite standard, there are some small variants of it in the literature which we now discuss. These distinctions make no difference as far as super-polynomial lower bounds are concerned, since it can be easily seen that each variant can be simulated by the other to within polynomial factors, and thus the issues described here are usually left unaddressed. However, it seems that we are very far from proving super-polynomial lower bounds for general algebraic branching programs, and in this paper we focus on proving polynomial (yet still super-linear) lower bounds. In this setting, those issues do affect the results.

Layered vs. Unlayered. In Definition 1.1, we have required the graph to be layered. We also consider in this paper ABPs whose underlying graphs are unlayered, which we call *unlayered ABPs*. We are able to prove super-linear (but weaker) lower bounds for this model as well.

One motivation for considering layered graph as the "standard" model is given by the following interpretation. From the definition, it can be observed that any polynomial computable by an ABP with d layers and ℓ_i vertices in the i-th layer can be written as the (only) entry of the 1×1 matrix given by the product $M := \prod_{i=1}^{d-1} M_i$, where M_i is an $\ell_i \times \ell_{i+1}$ matrix with affine forms as entries. One natural complexity measure of such a representation is the total number of non-zero entries in those matrices, which is the number of edges in the ABP. Another natural measure,

which can only be smaller, is the sums of dimensions of the matrices involved in the product, which is the same as the number of vertices in the underlying graph.

Branching programs are also prevalent in boolean complexity theory, and in particular in the context of derandomizing the class RL. In this setting again it only makes sense to talk about layered graphs.

Unlayered ABPs can also be thought of as (a slight generalization of) *skew circuits*. These are circuits in which on every multiplication gate, at least one of the operands is a variable (or more generally, a linear function).

Edge labels. In Definition 1.1 we have allowed each edge to be labeled by an arbitrary affine linear form in the variables. This is again quite standard, perhaps inspired by Nisan's characterization of the ABP complexity of a non-commutative polynomial as the rank of an associated coefficients matrix [Nis91], which requires this freedom. A more restrictive definition would only allow each edge to be labeled by a linear function in 1 variable. On the other hand, an even more general definition, which we sometimes adopt, is to allow every edge to be labeled by an *arbitrary* polynomial of degree at most Δ. In this case we refer to the model as an ABP with edge labels of degree at most Δ. Thus, the common case is $\Delta = 1$, but our results are meaningful even when $\Delta = \omega(1)$. Note that this is quite a powerful model, which is allowed to use polynomials with super-polynomial standard circuit complexity "for free".

We will recall some of these distinctions in Section 1.3, where we discuss previous results, some of which apply to several of the variants discussed here.

1.2 Lower bounds for algebraic branching programs.

Our main result is an almost quadratic lower bound on the size of any algebraic branching program computing some explicit polynomial.

Theorem 1.2. *Let $\mathbb{F}$ be a field and $n \in \mathbb{N}$ such that $\mathrm{char}(\mathbb{F}) \nmid n$. Then any algebraic branching program over $\mathbb{F}$ computing the polynomial $\sum_{i=1}^{n} x_i^n$ is of size at least $\Omega(n^2)$.*

When the ABP's edge labels are allowed to be polynomials of degree at most Δ, our lower bound is $\Omega(n^2/\Delta)$.

For the unlayered case, we prove a weaker (but still superlinear) lower bound.

Theorem 1.3. *Let $\mathbb{F}$ be a field and $n \in \mathbb{N}$ such that $\mathrm{char}(\mathbb{F}) \nmid n$. Then any unlayered algebraic branching program over $\mathbb{F}$ with edge labels of degree at most Δ computing the polynomial $\sum_{i=1}^{n} x_i^n$ is of size at least $\Omega(n \log n / (\log \log n + \log \Delta))$.*

1.3 Previous work

The best lower bound known for ABPs prior to this work is a lower bound of $\Omega(n \log n)$ on the number of edges for the same polynomial $\sum_{i=1}^{n} x_i^n$. This follows from the classical lower bound of $\Omega(n \log n)$ by Baur and Strassen [Str73a, BS83] on the number of multiplication gates in any algebraic circuit computing the polynomial $\sum_{i=1}^{n} x_i^n$ and the observation that when converting an ABP to an algebraic circuit, the number of product gates in the resulting circuit is at most the number of edges in the ABP. Theorem 1.2 improves upon this bound quantitatively, and also qualitatively, since the lower bound is on the number of vertices in the ABP.

For the case of homogeneous ABPs,[2] a quadratic lower bound for the polynomial $\sum_{i=1}^{n} x_i^n$ was shown by Kumar [Kum19], and the proofs in this paper build on the ideas in [Kum19]. In a nutshell, the result in [Kum19] is equivalent to a lower bound for ABPs computing the polynomial $\sum_{i=1}^{n} x_i^n$ when the number of layers in the ABP is at most n. In this work, we generalize this to proving essentially the same lower bound as in [Kum19] for ABPs with an unbounded number of layers.

In general, an ABP computing an n-variate homogeneous polynomial of degree $\mathrm{poly}(n)$ can be homogenized with a polynomial blow-up in size. This is proved in a similar manner to the standard classical result which shows this statement for algebraic circuits [Str73b]. Thus, much like the discussion following Definition 1.1, homogeneity is not an issue when one considers polynomial vs. super-polynomial sizes, but becomes relevant when proving polynomial lower bounds. In other contexts in algebraic complexity this distinction is even more sharp. For example, exponential lower bounds for homogeneous depth-3 circuits are well known and easy to prove [NW97], but strong enough exponential lower bounds for non-homogeneous depth-3 circuits would separate VP from VNP [GKKS16].

For *unlayered* ABPs, the situation is more complex. If the edge labels are only functions of one variable, it is possible to adapt Nechiporuk's method [Nec66] in order to obtain a lower bound of $\tilde{\Omega}(n^{3/2})$ (for a different polynomial than we consider). This is an argument attributed to Pudlák and sketched by Karchmer and Wigderson [KW93] for the boolean model of parity branching programs, but can be applied to the algebraic setting. However, this argument does not extend to the case where the edge labels are arbitrary linear or low-degree polynomials in the n variables. The crux of Nechiporuk's argument is to partition the variables into m disjoint sets, to argue (using counting or dimension arguments) that the number of edges labeled by variables from each set must be somewhat large[3], and then to sum the contributions over all m sets. This is hard to

[2] An ABP is *homogeneous* if the polynomial computed between the start vertex and any other vertex is a homogeneous polynomial. This condition is essentially equivalent to assuming that the number of layers in the ABP is upper bounded by the degree of the output polynomial.

[3] This is usually guaranteed by constructing a function or a polynomial with the property that given a fixed set S in the partition, there are many subfunctions or subpolynomials on the variables of S that can be obtained by different restrictions of the variables outside of S.

implement in models where a single edge can have a "global" access to all variables, since it is not clear how to avoid over-counting in this case.

As mentioned above, the lower bound of Baur-Strassen does hold in the unlayered case, assuming the edge labels are linear functions in the variables. When we allow edge labels of degree at most Δ for some $\Delta \geq 2$, their technique does not seem to carry over. Indeed, even if we equip the circuit with the ability to compute such low-degree polynomials "for free", a key step in the Baur-Strassen proof is the claim that if a polynomial f has a circuit of size τ, then there is a circuit of size $O(\tau)$ which computes all its first order partial derivatives, and this statement does not seem to hold in this new model.

It is possible to get an $\Omega(n \log n / \log \Delta)$ lower bound for this model, for a different polynomial, by suitably extending the techniques of Ben-Or [Ben83, Ben94]. Our lower bounds are weaker by at most a doubly-logarithmic factor; however, the techniques are completely different. Ben-Or's proofs rely as a black-box on strong modern results in algebraic geometry, whereas our proofs are much more elementary.

1.4 Proof Overview

The first part in the proof of Theorem 1.2 is an extension of the lower bound proved in [Kum19] for ABPs with at most n layers. This straightforward but conceptually important adaptation shows that a similar lower bound holds for any polynomial of the form

$$\sum_{i=1}^{n} x_i^n + \varepsilon(\mathbf{x}) \,,$$

where the suggestively named $\varepsilon(\mathbf{x})$ should be thought of as an "error term" which is "negligible" as far as the proof of [Kum19] is concerned. The exact structure we require is that $\varepsilon(\mathbf{x})$ is of the form $\sum_{i=1}^{r} P_i Q_i + R$, where P_i, Q_i are polynomials with no constant term and $\deg(R) \leq n-1$. The parameter r measures the "size" of the error, which we want to keep small, and the lower bound holds if, e.g., $r \leq n/10$.

To argue about ABPs with d layers, with $d > n$, we show that unless the size τ of the ABP is too large to begin with (in which case there is nothing to prove), it is possible to find a small set of vertices (of size about $\eta = \tau/d$) whose removal adds a small error term $\varepsilon(\mathbf{x})$ as above with at most η summands, but also reduces the depth of the ABP by a constant factor. Repeatedly applying this operation $O(\log n)$ times eventually gives an ABP of depth at most n while ensuring that we have not accumulated too much "error",[4] so that we can apply the lower bound from the previous paragraph.

[4]It takes some care in showing that the total number of error terms accumulated is at most $n/10$ as opposed to the obvious upper bound of $O(n \log n)$. In particular, we observe that the number of error terms can be upper bounded by a geometric progression with first term roughly τ/n and common ratio being a constant less than 1.

In the full proof we have to be a bit more careful when arguing about the ABP along the steps of the proof above. The details are presented in Section 3.

The proof of Theorem 1.3 follows the same strategy, although the main impediment is that general undirected graphs can have much more complex structure then layered graphs. One of the main ingredients in our proof is (a small variant of) a famous lemma of Valiant [Val77], which shows that for every graph of depth 2^k with m edges, it is possible to find a set of edges, of size at most m/k, whose removal reduces the depth of the graph to 2^{k-1}. This lemma helps us identify a small set of vertices which can reduce the depth of the graph by a constant factor while again accumulating small error terms.

Interestingly, Valiant originally proved this lemma in a different context, where he showed that linear algebraic circuits of depth $O(\log n)$ and size $O(n)$ can be reduced to a special type of depth-2 circuits (and thus strong lower bounds on such circuits imply super-linear lower bounds on circuits of depth $O(\log n)$). This lemma can be also used to show that *boolean* circuits of depth $O(\log n)$ and size $O(n)$ can be converted to depth-3 circuits of size $2^{o(n)}$, and thus again strong lower bounds on depth-3 circuits will imply super-linear lower bounds on circuits of depth $O(\log n)$. Both of these questions continue to be well known open problems in algebraic and boolean complexity, and to the best of our knowledge, our proof is the first time Valiant's lemma is successfully used in order to prove circuit lower bounds for explicit functions or polynomials.

2 Notations and Preliminaries

All logarithms in the paper are base 2.

We use some standard graph theory terminology: If G is a directed graph and (u, v) is an edge, v is called the *head* of the edge and u the *tail*. Our directed graphs are always acyclic with designated source vertex s and sink vertex t. The *depth* of a vertex v, denoted $\mathsf{depth}(v)$, is the length (in edges) of a longest path from s to v. The depth of the graph, denoted by $\mathsf{depth}(G)$, is the depth of t.

For any two vertices u and v in an ABP, the polynomial computed between u and v is the sum of weights of all paths between u and v in the ABP. We denote this by $[u, v]$.

The formal degree of a vertex u in an ABP denoted $\mathrm{fdeg}(u)$, is defined inductively as follows: If s is the start vertex of the ABP, $\mathrm{fdeg}(s) = 0$. If u is a vertex with incoming edges from $u_1, \ldots, u_k$, labeled by non-zero polynomials $\ell_1, \ldots, \ell_k$, respectively, then

$$\mathrm{fdeg}(u) = \max_{i \in [k]} \{\deg(\ell_i) + \mathrm{fdeg}(u_i)\}.$$

It follows by induction that for every vertex u, $\deg([s, u]) \leq \mathrm{fdeg}(u)$ (however, cancellations can allow for arbitrary gaps between the two). The formal degree of the ABP is the maximal formal

degree of any vertex in it.

We sometimes denote by $\mathbf{x}$ the vector of variables $(x_1, \ldots, x_n)$, where n is understood from the context. Similarly we use $\mathbf{0}$ to denote the n-dimensional vector $(0, 0, \ldots, 0)$.

2.1 A decomposition lemma

The following lemma gives a decomposition of a (possibly unlayered) ABP in terms of the intermediate polynomials it computes. Its proof closely resembles that of Lemma 3.5 of [Kum19]. For completeness we prove it here for a slightly more general model.

Lemma 2.1 ([Kum19])**.** *Let $\mathcal{B}$ be a (possibly unlayered) algebraic branching program whose edge labels are arbitrary polynomials of degree at most $\Delta \leq n/10$, which computes a degree d polynomial $P \in \mathbb{F}[x_1, \ldots x_n]$, and has formal degree d. Set $d' = \lfloor d/\Delta \rfloor$.*

For any $i \in \{1, 2, \ldots, d' - 1\}$, let $S_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,m}\}$ be the set of all vertices in $\mathcal{B}$ whose formal degree is in the interval $[i\Delta, (i+1)\Delta)$.

Then, there exist polynomials $Q_{i,1}, Q_{i,2}, \ldots, Q_{i,m}$ and R_i, each of degree at most $d - 1$ such that

$$P = \sum_{j=1}^{m} [s, u_{i,j}] \cdot Q_{i,j} + R_i \,.$$

Proof. Fix i as above and set $S_i = \{u_{i,1}, u_{i,2}, \ldots, u_{i,m}\}$ as above (observe that since each edge label is of degree at most Δ, S_i is non empty). Further suppose, without loss of generality, that the elements of S_i are ordered such that there is no directed path from $u_{i,j}$ to $u_{i,j'}$ for $j' > j$.

Consider the unlayered ABP $\mathcal{B}_1$ obtained from $\mathcal{B}$ by erasing all incoming edges to $u_{i,1}$, and multiplying all the labels of the outgoing edges from $u_{i,1}$ by a new variable y_1. The ABP $\mathcal{B}_1$ now computes a polynomial of the form

$$P'(y_1, x_1, \ldots, x_n) = y_1 \cdot Q_{i,1} + R_{i,1}$$

where $P = P'([s, u_{i,1}], x_1, \ldots, x_n)$. $R_{i,1}$ is the polynomial obtained from $\mathcal{B}_1$ by setting y_1 to zero, or equivalently, removing $u_{i,1}$ and all its outgoing edges. We continue in the same manner with $u_{i,2}, \ldots, u_{i,m}$ to obtain

$$P = \sum_{j=1}^{m} [s, u_{i,j}] \cdot Q_{i,j} + R_i.$$

Indeed, observe that since there is no path from $u_{i,j}$ to $u_{i,j'}$ for $j' > j$, removing $u_{i,j}$ does not change $[s, u_{i,j'}]$. The bound on the degrees of $Q_{i,j}$ is immediate from the fact that the formal degree of the ABP is at most d and $\mathrm{fdeg}(u_{i,j}) \geq 1$. It remains to argue the $\deg(R_i) \leq d - 1$.

The polynomial R_i is obtained from $\mathcal{B}$ by erasing all the vertices in S_i and the edges touching them. We will show that every path in the corresponding ABP computes a polynomial of degree

at most $d-1$. Let $s = v_1, v_2, \ldots, v_r = t$ be such a path, which is also a path in $\mathcal{B}$. Let v_k be the minimal vertex in the path whose degree (in $\mathcal{B}$) is at least $(i+1)\Delta$ (if no such v_k exists, the proposition follows). As $v_{k-1} \notin S_i$, the formal degree of v_{k-1} is at most $i\Delta - 1$. The degree of the polynomial computed by this path is thus at most $i\Delta - 1 + \Delta + D = (i+1)\Delta - 1 + D$, where D is the degree of product of the labels on the path $v_k, v_{k+1}, \ldots, t$. To complete the proof, it remains to be shown that $D \leq d - (i+1)\Delta$.

Indeed, if $D \geq d - (i+1)\Delta + 1$ then since the degree of v_k is at least $(i+1)\Delta$, there would be in $\mathcal{B}$ a path of formal degree at least $(i+1)\Delta + D \geq d+1$, contradicting the assumption on $\mathcal{B}$. □

3 A lower bound for Algebraic Branching Programs

In this section we prove Theorem 1.2. We start by restating it.

Theorem 3.1. *Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be a field such that $\mathrm{char}(\mathbb{F}) \nmid n$. If $\mathcal{A}$ is an algebraic branching program with edge labels of degree at most Δ that computes the polynomial $\sum_{i=1}^{n} x_i^n$, then the size of $\mathcal{A}$ is at least*

$$\Omega\left(\frac{n^2}{\Delta}\right).$$

For technical reasons, we work with a slightly more general model which we call *multilayered* ABPs, which we now define.

Definition 3.2 (Multilayered ABP)**.** *Let $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be k ABPs with $d_1, \ldots, d_k$ layers and $\tau_1, \ldots, \tau_k$ vertices, respectively. A multilayered ABP $\mathcal{A}$, denoted by $\mathcal{A} = \sum_{i=1}^{k} \mathcal{A}_i$, is the ABP obtained by placing $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_k$ in parallel and identifying their start and end vertices respectively. Thus, the polynomial computed by $\mathcal{A}$ is $\sum_{i=1}^{k} [\mathcal{A}_i]$, where $[\mathcal{A}_i]$ is the polynomial computed by $\mathcal{A}_i$.*

The number of layers of $\mathcal{A}$ is $d := \max\{d_1, \ldots, d_k\}$. The size of $\mathcal{A}$ is the number of vertices in $\mathcal{A}$, and thus equals

$$|\mathcal{A}| := 2 + \sum_i (\tau_i - 2).$$

◊

This model is an intermediate model between (layered) ABPs and unlayered ABPs: given a multilayered ABP of size τ it is straightforward to construct an unlayered ABP of size $O(\tau)$ which computes the same polynomial.

3.1 A robust lower bound for ABPs of formal degree at most n

In this section, we prove a lower bound for the case where the formal degree of every vertex in the ABP is at most n. In fact, Kumar [Kum19] has already proved a quadratic lower bound for this case.

Theorem 3.3 ([Kum19]). *Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be a field such that $\mathrm{char}(\mathbb{F}) \nmid n$. Then any algebraic branching program of formal degree at most n which computes the polynomial $\sum_{i=1}^{n} x_i^n$ has at least $\Omega(n^2)$ vertices.*

However, to prove Theorem 3.1, we need the following more "robust" version of Theorem 3.3, which gives a lower bound for a larger class of polynomials. For completeness, we also sketch an argument for the proof which is a minor variation of the proof of Theorem 3.3.

Theorem 3.4. *Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be field such that $\mathrm{char}(\mathbb{F}) \nmid n$. Let $A_1(\mathbf{x}), \ldots, A_r(\mathbf{x}), B_1(\mathbf{x}), \ldots, B_r(\mathbf{x})$ and $R(\mathbf{x})$ be polynomials such that for every i, $A_i(\mathbf{0}) = B_i(\mathbf{0}) = 0$ and R is a polynomial of degree at most $n-1$. Then, any algebraic branching program over $\mathbb{F}$, of formal degree at most n and edge labels of degree at most $\Delta \leq n/10$, which computes the polynomial*

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R$$

has at least $\frac{(n/2-r)n}{2\Delta}$ vertices.

The proof of the theorem follows from Lemma 2.1 and the following lemma which is a slight generalization of Lemma 3.1 in [Kum19]. We include a proof for completeness.

Lemma 3.5. *Let $n \in \mathbb{N}$, and let $\mathbb{F}$ be an algebraically closed field such that $\mathrm{char}(\mathbb{F}) \nmid n$. Let $\{P_1, \ldots, P_m, Q_1, \ldots, Q_m, A_1, \ldots, A_r, B_1, \ldots, B_r\}$ be a set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ such that the set of their common zeros*

$$V = \mathbb{V}(P_1, \ldots, P_m, Q_1, \ldots, Q_m, A_1, \ldots, A_r, B_1, \ldots, B_r) \subseteq \mathbb{F}^n$$

is non-empty. Finally, suppose R is a polynomial in $\mathbb{F}[\mathbf{x}]$ of degree at most $n-1$, such that

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j = R + \sum_{i=1}^{m} P_i \cdot Q_i.$$

Then, $m \geq \frac{n}{2} - r$.

Proof. Since $V \neq \emptyset$, $\dim(V) \geq n - 2m - 2r$ (see, e.g., Section 2.8 of [Smi14]). Thus, the set of zeros with multiplicity two of

$$\sum_{i=1}^{n} x_i^n - R = \sum_{i=1}^{m} P_i \cdot Q_i - \sum_{j=1}^{r} A_j \cdot B_j,$$

has dimension at least $n - 2m - 2r$. In other words, if S is the set of common zeros of the set of all first order partial derivatives of $\sum_{i=1}^{n} x_i^n - R$, then $\dim(S) \geq n - 2m - 2r$. Up to scaling by n (which is non-zero in $\mathbb{F}$, by assumption), the set of all first order partial derivatives of $\sum_{i=1}^{n} x_i^n - R$

is given by

$$\left\{ x_i^{n-1} - \frac{1}{n}\partial_{x_i} R \right\}_{i\in[n]}.$$

Thus, the statement of this lemma immediately follows from the following claim.

Claim 3.6 (Lemma 3.2 in [Kum19]). *Let $\mathbb{F}$ be an algebraically closed field, and D a positive natural number. For every choice of polynomials $g_1, g_2, \ldots g_n \in \mathbb{F}[\mathbf{x}]$ of degree at most $D-1$, the dimension of the variety*

$$\mathbb{V}(x_1^D - g_1, x_2^D - g_2, \ldots, x_n^D - g_n)$$

is zero.

Indeed, the above claim shows that $0 = \dim(S) \geq n - 2m - 2r$, and so $m \geq \frac{n}{2} - r$. This completes the proof of Lemma 3.5. □

We now use Lemma 2.1 and Lemma 3.5 to complete the proof of Theorem 3.4.

Proof of Theorem 3.4. Let $\mathcal{B}$ be an algebraic branching program of formal degree at most n, edge labels of degree at most $\Delta \leq n/10$, and with start vertex s and end vertex t, which computes

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R.$$

We may assume without loss of generality that $\mathbb{F}$ is algebraically closed, by interpreting $\mathcal{B}$ as an ABP over the algebraic closure of $\mathbb{F}$, if necessary.

Let $n' = \lfloor n/\Delta \rfloor$, fix $k \in \{1, 2, \ldots, n'-1\}$, and let $V_k = \{v_{k,1}, v_{k,2}, \ldots, v_{k,m}\}$ be the set of all vertices in $\mathcal{B}$ whose formal degree lies in the interval $[k\Delta, (k+1)\Delta)$. Letting $P'_j = [s, v_{k,j}]$, by Lemma 2.1, there exist polynomials $Q'_1, Q'_2, \ldots, Q'_m$ and R', each of degree at most $n-1$ such that

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R = \sum_{j=1}^{m} P'_j \cdot Q'_j + R'.$$

Let α_j, β_j be the constant terms in P'_j, Q'_j respectively. Then by defining

$$P_j = P'_j - \alpha_j \qquad \text{and} \qquad Q_j = Q'_j - \beta_j,$$

we have that

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j = R'' + \sum_{j=1}^{m} P_j \cdot Q_j.$$

Here, $R'' = -R + R' + \sum_{j=1}^{m} (\alpha_j \cdot Q'_j + \beta_j \cdot P'_j + \alpha_j \beta_j)$. We now have that for every i, the constant

terms of P_i, Q_i are zero and $\deg(R'') \leq n-1$. Let

$$\mathcal{V} = \mathbb{V}(P_1, \ldots, P_m, Q_1, \ldots, Q_m, A_1, \ldots, A_r, B_1, \ldots, B_r).$$

Then $\mathbf{0} \in \mathcal{V}$, and so $\mathcal{V} \neq \emptyset$. Thus by Lemma 3.5, we know that $m \geq \frac{n}{2} - r$.

Finally, for $k \neq k' \in \{1, 2, \ldots, n'-1\}$, $V_k \cap V_{k'} = \emptyset$ if $d \neq d'$. Thus, the number of vertices in $\mathcal{B}$ must be at least

$$\left(\frac{n}{2} - r\right) \cdot (n'-1) \geq \left(\frac{n}{2} - r\right) \cdot \frac{n}{2\Delta}. \qquad \square$$

3.2 A lower bound for the general case

The following lemma shows how we can obtain, given an ABP with d layers which computes a polynomial F, a multilayered ABP, whose number of layers is significantly smaller, which computes F plus a small "error term".

Lemma 3.7. *Let $\mathcal{A}$ be an ABP over a field $\mathbb{F}$ with d layers, which computes the polynomial F and has m vertices. Let s and t be the start and end vertices of $\mathcal{A}$ respectively, and let $L = \{u_1, u_2, \ldots, u_{|L|}\}$ be the set of vertices in the ℓ-th layer of $\mathcal{A}$. For every $i \in \{1, 2, \ldots, |L|\}$, let α_i and β_i be the constant terms of $[s, u_i]$ and $[u_i, t]$ respectively. Furthermore, let P_i and Q_i be polynomials such that $[s, u_i] = P_i + \alpha_i$ and $[u_i, t] = Q_i + \beta_i$.*

Then, there is a multilayered ABP $\mathcal{A}'$, with at most $\max\{\ell, d - \ell + 1\}$ layers and size at most $|\mathcal{A}|$ that computes the polynomial

$$F - \sum_{i=1}^{|L|} P_i \cdot Q_i + \sum_{i=1}^{|L|} \alpha_i \cdot \beta_i.$$

Proof. Let $u_1, u_2, \ldots, u_{|L|}$ be the vertices in L as described, so that

$$F = [s, t] = \sum_{i=1}^{|L|} [s, u_i] \cdot [u_i, t].$$

Further, for every $i \in \{1, 2, \ldots, |L|\}$, $[s, u_i] = P_i + \alpha_i$ and $[u_i, t] = Q_i + \beta_i$, where the constant terms of P_i and Q_i are zero (by definition). Having set up this notation, we can thus express the polynomial F computed by $\mathcal{A}$ as

$$F = [s, t] = \sum_{i=1}^{|L|} (P_i + \alpha_i) \cdot (Q_i + \beta_i).$$

On further rearrangement, this gives

$$F - \left(\sum_{i=1}^{|L|} P_i \cdot Q_i\right) + \left(\sum_{i=1}^{|L|} \alpha_i \cdot \beta_i\right) = \left(\sum_{i=1}^{|L|} \alpha_i \cdot (Q_i + \beta_i)\right) + \left(\sum_{i=1}^{|L|} (P_i + \alpha_i) \cdot \beta_i\right).$$

This is equivalent to the following expression.

$$F - \left(\sum_{i=1}^{|L|} P_i \cdot Q_i\right) + \left(\sum_{i=1}^{|L|} \alpha_i \cdot \beta_i\right) = \left(\sum_{i=1}^{|L|} \alpha_i \cdot [u_i, t]\right) + \left(\sum_{i=1}^{|L|} [s, u_i] \cdot \beta_i\right) .$$

Now, observe that the polynomial $\sum_{i=1}^{|L|} [s, u_i] \cdot \beta_i$ is computable by an ABP $\mathcal{B}$ with $\ell + 1$ layers, obtained by just keeping the vertices and edges within first ℓ layers of $\mathcal{A}$ and the end vertex t, deleting all other vertices and edges, and connecting the vertex u_i in the ℓ-th layer to t by an edge of weight β_i. Similarly, the polynomial $\sum_{i=1}^{|L|} \alpha_i \cdot [u_i, t]$ is computable by an ABP $\mathcal{C}$ with at most $(d - \ell + 1) + 1$ layers, whose set of vertices is s along the vertices in the layers $\ell, \ell + 1, \ell + 2, \ldots, d$ of $\mathcal{A}$. From the definition of $\mathcal{B}$ and $\mathcal{C}$, it follows that the multilayered ABP $\tilde{\mathcal{A}}$ obtained by taking the sum of $\mathcal{B}$ and $\mathcal{C}$ has at most $\max\{\ell + 1, d - \ell + 2\}$ layers.

We are almost done with the proof of the lemma, except for the upper bound on the number of vertices of the resulting multilayered ABP $\tilde{\mathcal{A}}$, and the fact that the upper bound on the depth is slightly weaker than claimed. Both these issues can be solved simultaneously.

The vertices in L appear in both the ABP $\mathcal{B}$ and the ABP $\mathcal{C}$ and are counted twice in the size of $\tilde{\mathcal{A}}$. However, every other vertex is counted exactly once. Hence,

$$|\mathcal{B}| + |\mathcal{C}| = |\mathcal{A}| + |L| . \tag{3.8}$$

In order to fix this issue, we first observe that the edges between the vertices in the ℓ-th layer of $\mathcal{B}$ and the end vertex t are labeled by $\beta_1, \beta_2, \ldots, \beta_{|L|}$, all of which are field constants. In the following claim, we argue that for ABPs with this additional structure, the last layer is redundant and can be removed.

Claim 3.9. *Let $\mathcal{M}$ be an ABP over $\mathbb{F}$ with $k + 1$ layers and edge labels of degree at most Δ such that the labels of all the edges between the k-th layer of $\mathcal{M}$ and its end vertex are scalars in $\mathbb{F}$. Then, there is an ABP $\mathcal{M}'$ with k layers computing the same polynomial as $\mathcal{M}$, with edge labels of degree at most Δ, such that*

$$\left|\mathcal{M}'\right| \leq |\mathcal{M}| - |V| ,$$

where V is the set of vertices in the k-th layer of $\mathcal{M}$.

An analogous statement, with an identical proof, is true if we assume that all edge labels between the first and second layer are scalars in $\mathbb{F}$.

We first use Claim 3.9 to complete the proof of the lemma. As observed above, the edge labels between the last layer L of $\mathcal{B}$ and its end vertex are all constants. Hence, by Claim 3.9, there is an ABP $\mathcal{B}'$ which computes the same polynomial as $\mathcal{B}$ such that $|\mathcal{B}'| \leq |\mathcal{B}| - |L|$, and $\mathcal{B}'$ has only ℓ layers. Similarly, we can obtain an ABP $\mathcal{C}'$ with at most $d - \ell + 1$ layers.

We consider the multilayered ABP $\mathcal{A}'$ by taking the sum of $\mathcal{B}'$ and $\mathcal{C}'$. Clearly, the number of layers in $\mathcal{A}'$ is at most $\max\{\ell, d-\ell+1\}$ and the size is at most

$$|\mathcal{A}'| \leq |\mathcal{B}'| + |\mathcal{C}'| \leq (|\mathcal{B}| - |L|) + (|\mathcal{C}| - |L|) \leq |\mathcal{A}| \,.$$

Here, the second inequality follows by Claim 3.9 and the last one follows by Equation 3.8. To complete the proof of the lemma, we now prove Claim 3.9. □

Proof of Claim 3.9. For the proof of the claim, we focus on the k-th and $(k-1)$-st layer of $\mathcal{M}$. To this end, we first set up some notation. Let $\{v_1, v_2, \ldots, v_r\}$ be the set of vertices in the k-th layer of $\mathcal{M}$, $\{u_1, u_2, \ldots, u_{r'}\}$ be the set of vertices in $(k-1)$-st layer of $\mathcal{M}$, and a, b denote the start and the end vertices of $\mathcal{M}$ respectively. Then, the polynomial computed by $\mathcal{M}$, can be decomposed as

$$[a, b] = \sum_{i=1}^{r} [a, v_i] \cdot [v_i, b] \,.$$

Note that (v_i, b) is an edge in the ABP. Similarly, the polynomial $[a, v_i]$ can be written as

$$[a, v_i] = \sum_{j=1}^{r'} [a, u_j] \cdot [u_j, v_i] \,.$$

Combining the two expressions together, we get

$$[a, b] = \sum_{i=1}^{r} [v_i, b] \cdot \left(\sum_{j=1}^{r'} [u_j, v_i] \cdot [a, u_j] \right) ,$$

which on further rearrangement, gives us

$$[a, b] = \sum_{j=1}^{r'} \left(\sum_{i=1}^{r} [v_i, b][u_j, v_i] \right) \cdot [a, u_j] \,. \tag{3.10}$$

From the hypothesis of the claim, we know that for every $i \in [r]$, the edge label $[v_i, b]$ is a field constant, and the edge label $[u_j, v_i]$ is a polynomial of degree at most Δ. Thus, for every $j \in [r']$, the expression $\left(\sum_{i=1}^{r} [u_i, b][u_j, v_i]\right)$ is a polynomial of degree at most Δ.

This gives us the following natural construction for the ABP $\mathcal{M}'$ from $\mathcal{M}$. We delete the vertices $v_1, v_2, \ldots, v_r$ in $\mathcal{M}$ (and hence, all edges incident to them), and for every $j \in \{1, 2, \ldots, r'\}$, we connect the vertex u_j with the end vertex b using an edge with label $\left(\sum_{i=1}^{r} [v_i, b][u_j, v_i]\right)$. The upper bound on the size and the number of layers of $\mathcal{M}'$ is immediate from the construction, and that it computes the same polynomial as $\mathcal{M}$ follows from Equation 3.10. □

We now state and prove a simple generalization of Lemma 3.7 for a multilayered ABP.

Lemma 3.11. *Let $\mathcal{A} = \sum_{i=1}^{m} \mathcal{A}_i$ be a multilayered ABP with d layers over a field $\mathbb{F}$ computing the polynomial F, such that each $\mathcal{A}_i$ is an ABP with d_i layers. Also, let $\ell_{i,j}$ be the number of vertices in the j-th layer of $\mathcal{A}_i$ ($\ell_{i,j} = 0$ if $\mathcal{A}_i$ has fewer than j layers), and $\ell = \min_{j \in (d/3, 2d/3)} \left\{\sum_{i=1}^{m} \ell_{i,j}\right\}$.*

Then, there is a multilayered ABP with at most $2d/3$ layers and size at most $|\mathcal{A}|$ that computes a polynomial of the form

$$F - \sum_{i=1}^{\ell} P_i \cdot Q_i + \delta\,,$$

where $\{P_1, \ldots, P_\ell, Q_1, \ldots, Q_\ell\}$ is a set of non-constant polynomials with constant term zero and $\delta \in \mathbb{F}$.

Proof. Let $j_0 \in (d/3, 2d/3)$ be the natural number which minimizes the quantity $\sum_{i=1}^{m} \ell_{i,j}$, and let $S \subseteq [m]$ be the set of all indices i such that $\mathcal{A}_i$ has at least j_0 layers. Let $\mathcal{A}' = \sum_{i \in S} \mathcal{A}_i$ and $\mathcal{A}'' = \sum_{i \notin S} \mathcal{A}_i$. Thus,

$$\mathcal{A} = \mathcal{A}' + \mathcal{A}''.$$

Here, $\mathcal{A}'' = \sum_{i \notin S} \mathcal{A}_i$ is a multilayered ABP with at most $2d/3$ layers. Moreover, $|\mathcal{A}| = |\mathcal{A}'| + |\mathcal{A}''|$.

To complete the proof of this lemma, we will now apply Lemma 3.7 to every ABP in $\mathcal{A}'$. For every $i \in S$, we know that there exist some polynomials $P_{i,1}, \ldots, P_{i,\ell_{i,j_0}}, Q_{i,1}, \ldots, Q_{i,\ell_{i,j_0}}$ with constant terms zero and a constant δ_i, such that

$$F_i - \sum_{r=0}^{\ell_{i,j_0}} P_{i,r} Q_{i,r} + \delta_i$$

can be computed by a multilayered ABP. Let us denote this multilayered ABP by $\mathcal{B}_i$. From Lemma 3.7, we know that $\mathcal{B}_i$ has at most $\max\{j_0, d_i - j_0 + 1\} \leq 2d/3$ layers and size at most $|\mathcal{A}_i|$. Taking a sum over all $i \in S$ and re-indexing the summands, we get that there exist polynomials $P_1, \ldots, P_\ell, Q_1, \ldots, Q_\ell$ with constant terms zero and a constant δ such that the polynomial

$$\sum_{i \in S} F_i - \sum_{r=0}^{\ell} P_r Q_r + \delta$$

is computable by a multilayered ABP $\mathcal{B} = \sum_{i \in S} \mathcal{B}_i$ with at most $2d/3$ layers and size at most $\sum_{i \in S} |\mathcal{A}_i| \leq |\mathcal{A}'|$. Now, by combining the multilayered ABPs $\mathcal{B}$ and $\mathcal{A}''$, we get that the polynomial

$$F - \sum_{r=0}^{\ell} P_r Q_r + \delta$$

is computable by a multilayered ABP with at most $2d/3$ layers and size at most $|\mathcal{A}|$. □

We now use Lemma 3.11 to prove Theorem 3.1.

Proof of Theorem 3.1. Let $\mathcal{A}$ be a multilayered ABP with d_0 layers which computes the polynomial

$\sum_{i=1}^{n} x_i^n$. As before we may assume without loss of generality that the underlying field $\mathbb{F}$ is algebraically closed. Note that if d_0 is at most n/Δ, then by Theorem 3.4, we know that $|\mathcal{A}|$ is at least $\Omega(n^2)$ and we are done. Also, if $d_0 > n^2/\Delta$, then again we have our lower bound since each layer of $\mathcal{A}$ must have at least one vertex. Thus, we can assume that $n/\Delta \leq d_0 \leq n^2/\Delta$.

The proof idea is to iteratively make changes to $\mathcal{A}$ till we get a multilayered ABP $\mathcal{A}'$ of formal degree at most n that computes a polynomial of the type

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R$$

where $r \leq n/10$ and $A_1(\mathbf{x}), \ldots, A_r(\mathbf{x}), B_1(\mathbf{x}), \ldots, B_r(\mathbf{x}), R(\mathbf{x})$ are polynomials such that for every i, $A_i(\mathbf{0}) = B_i(\mathbf{0}) = 0$ and R has degree at most $n-1$. Once we have this, we can invoke Theorem 3.4 and get the required lower bound.

We now explain how to iteratively obtain $\mathcal{A}'$ from $\mathcal{A}$. In one step, we ensure the following.

Claim 3.12. *Let $\mathcal{A}_k$ be a multilayered ABP with edge labels of degree at most Δ, $d_k \geq n/\Delta$ layers and size at most τ that computes a polynomial of the form $\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R$ where $A_1(\mathbf{x}), \ldots, A_r(\mathbf{x}), B_1(\mathbf{x}), \ldots, B_r(\mathbf{x}), R(\mathbf{x})$ are polynomials such that for every j, $A_j(\mathbf{0}) = B_j(\mathbf{0}) = 0$ and R has degree at most $n-1$.*

If $\tau \leq 0.001n^2/\Delta$, then there exists a multilayered ABP $\mathcal{A}_{k+1}$ with at most $2d_k/3$ layers and size at most τ which computes a polynomial of the form

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r'} A'_j \cdot B'_j + R',$$

such that $r' \leq r + 0.005\frac{n^2}{\Delta \cdot d_k}$ and $A'_1(\mathbf{x}), \ldots, A'_{r'}(\mathbf{x}), B'_1(\mathbf{x}), \ldots, B'_{r'}(\mathbf{x}), R'(\mathbf{x})$ are polynomials such that for every i, $A'_i(\mathbf{0}) = B'_i(\mathbf{0}) = 0$ and R' has degree at most $n-1$.

Before moving on to the proof of Claim 3.12, we first use it to complete the proof of Theorem 3.1. Let us set $\mathcal{A}_0 = \mathcal{A}$. Then, $\mathcal{A}_0$ is a multilayered ABP with d_0 layers and size at most τ that computes the polynomial $\sum_{i=1}^{n} x_i^n$.

If $\tau \geq 0.001n^2/\Delta$, the statement of the theorem follows. Otherwise, we apply Claim 3.12 iteratively K times, as long as the number of layers is more than n/Δ, to eventually get a multilayered ABP $\mathcal{A}' = \mathcal{A}_K$ with $d' \leq n/\Delta$ layers. Let $d_0, \ldots, d_{K-1}, d_K$ denote the number of layers in each ABP in this sequence, so that $d_{K-1} > n/\Delta$, and $d_k \leq 2d_{k-1}/3$ for $k \in [K]$. $\mathcal{A}'$ is an ABP with at most n/Δ layers and size at most τ, which by induction, computes a polynomial of the form

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R,$$

where $A_1(\mathbf{x}), \ldots, A_r(\mathbf{x}), B_1(\mathbf{x}), \ldots, B_r(\mathbf{x}), R(\mathbf{x})$ are polynomials such that for every i, $A_i(\mathbf{0}) = B_i(\mathbf{0}) = 0$ and R has degree at most $n-1$. Further, the number of error terms, r, is at most

$$\frac{0.005n^2}{\Delta}\left(\frac{1}{d_{K-1}} + \frac{1}{d_{K-2}} + \cdots + \frac{1}{d_0}\right).$$

Since $d_k \leq \frac{2}{3} \cdot d_{k-1}$, we have that $\frac{1}{d_{k-1}} \leq \frac{2}{3} \cdot \frac{1}{d_k}$ for all $k \in [K]$, so that

$$r \leq \frac{0.005n^2}{\Delta} \cdot \frac{1}{1-2/3} \cdot \frac{1}{d_{K-1}} \leq \frac{n}{10}$$

as $d_{K-1} \geq n/\Delta$.

At this point, since the formal degree is at most n, using Theorem 3.4 we get

$$\tau \geq |\mathcal{A}'| \geq \frac{(n/2-r)n}{2\Delta} = \Omega\left(\frac{n^2}{\Delta}\right). \qquad \square$$

To complete the proof of Theorem 3.1, we now prove Claim 3.12.

Proof of Claim 3.12. Let $\mathcal{A}_k = \sum_{i=1}^{m} \mathcal{A}_{k,i}$, and for $j \in [d_k]$, let $\ell_{i,j}$ be the number of vertices in layer j of $\mathcal{A}_{k,i}$. Recall that if the number of layers in $\mathcal{A}_{k,i}$ is strictly less than j, then we set $\ell_{i,j} = 0$. Let ℓ be the total number of vertices in the middle layers of $\mathcal{A}_k$, defined as

$$\ell = \sum_{i=1}^{m}\left(\sum_{j \in (d_k/3, 2d_k/3)} \ell_{i,j}\right).$$

Since $\ell \leq \tau \leq \frac{0.001n^2}{\Delta}$, by averaging, we know that there is a $j_0 \in (d_k/3, 2d_k/3)$, such that

$$\ell_{j_0} = \sum_{i=1}^{m} \ell_{i,j_0} \leq \frac{\ell}{d_k/3} \leq \frac{0.001n^2}{\Delta} \cdot \frac{1}{d_k/3} \leq 0.005\frac{n^2}{\Delta \cdot d_k}.$$

This condition, together with Lemma 3.11, tells us that there is a multilayered ABP $\mathcal{A}'_{k+1}$ with at most $2d_k/3$ layers and size at most τ that computes a polynomial of the form

$$\sum_{i=1}^{n} x_i^n + \sum_{j=1}^{r} A_j \cdot B_j + R - \sum_{i=1}^{\ell_{j_0}} P_i \cdot Q_i + \delta,$$

where $P_1, \ldots, P_\ell, Q_1, \ldots, Q_\ell$ are a set of non-constant polynomials with constant term zero and $\delta \in \mathbb{F}$. Since $\ell_{j_0} \leq 0.005\frac{n^2}{\Delta \cdot d_k}$, the claim follows. $\square$

4 Unlayered Algebraic Branching Programs

In this section, we prove Theorem 1.3. We begin with the following definition.

Definition 4.1. *Let $\mathcal{A}$ be an unlayered ABP over $\mathbb{F}$. Let s and t denote the start and end vertices of $\mathcal{A}$, respectively, and let $v \neq s, t$ be a vertex in $\mathcal{A}$. Denote by $\alpha \in \mathbb{F}$ the constant term of $[s, v]$ and by $\beta \in \mathbb{F}$ the constant term of $[v, t]$.*

The cut *of $\mathcal{A}$ with respect to v, denoted $\mathsf{cut}(\mathcal{A}, v)$, is the unlayered ABP obtained from $\mathcal{A}$ using the following sequence of operations:*

1. *Duplicate the vertex v (along with its incoming and outgoing edges). Let v_1, v_2 denote the two copies of v.*
2. *Erase all outgoing edges of v_1, and connect v_1 to t by a new edge labeled β.*
3. *Erase all incoming edges of v_2, and connect s to v_2 by a new edge labeled α.* ◊

We now prove some basic properties of the construction in Definition 4.1.

Claim 4.2. *Let $\mathcal{A}$ be an unlayered ABP over $\mathbb{F}$ computing a polynomial F, and let v be a vertex in $\mathcal{A}$. Denote $\mathcal{A}' = \mathsf{cut}(\mathcal{A}, v)$. Denote by d the depth of $\mathcal{A}$ and by d_v the depth of v in $\mathcal{A}$. Then the following properties hold:*

1. *$\mathcal{A}'$ has 1 more vertex and 2 more edges than $\mathcal{A}$.*
2. *The depth of $\mathcal{A}'$ is at most*

$$\max\left\{\mathsf{depth}(\mathcal{A} \setminus \{v\}), d_v + 1, d - d_v + 1\right\},$$

   where $\mathcal{A} \setminus \{v\}$ is the ABP obtained from $\mathcal{A}$ by erasing v and all of its adjacent edges.

3. *$\mathcal{A}'$ computes a polynomial of the form $F - P \cdot Q - \delta$ where P and Q have no constant term, and $\delta \in \mathbb{F}$.*

Proof. The first property is immediate from the construction. The second property follows from the following reasoning: each path in $\mathcal{A}'$ is of exactly one of the following types: (a) misses both v_1 and v_2, (b) passes through v_1, or (c) passes through v_2. In case (a), the path also appears in the graph of $\mathcal{A} \setminus \{v\}$. In case (b), the only edge going out of v_1 is to t, and all other edges in the path appear in $\mathcal{A}$, hence the length is at most $d_v + 1$. In case (c), the only edge entering v_2 is from s, hence similarly the path is of length at most $d - d_v + 1$.

It remains to show the last property. Let $P' = [s, v]$ and $Q' = [v, t]$ (as computed in $\mathcal{A}$). Denote $P' = P + \alpha$ where P has no constant term and $\alpha \in \mathbb{F}$ and similarly $Q' = Q + \beta$. One may write

$F = P' \cdot Q' + R = (P + \alpha)(Q + \beta) + R$ where R is the sum over all paths in $\mathcal{A}$ which do not pass through v. In $\mathcal{A}'$, we have that $[s, v_1] = P'$ and $[v_2, t] = Q'$, and thus $\mathcal{A}'$ computes the polynomial

$$R + \alpha \cdot Q' + P' \cdot \beta = F - P \cdot Q + \alpha\beta. \qquad \square$$

Our goal is to perform cuts on a strategically chosen set of vertices. In order to select them, will use the following well known lemma of Valiant [Val77], simplifying and improving an earlier result of Erdős, Graham and Szemerédi [EGS75]. For completeness, we also sketch a short proof.

Lemma 4.3 ([Val77])**.** *Let G be a directed acyclic graph with m edges and depth $d \geq \sqrt{n}$. Then, there exists a set E' of at most $4m/\log n$ edges such that removing E' from G results in a graph of depth at most $d/2$.*

Proof. Let $d' \geq d \geq \sqrt{n}$ be a smallest power of 2 larger than d, so that $d' \leq 2d$. Let $k = \log d'$. A *valid labeling* of a directed graph $G = (V, E)$ is a function $f : V \to \{0, \ldots, N-1\}$ such that whenever (u, v) is an edge, $f(u) < f(v)$. Clearly if G had depth d then there is a valid labeling with image $\{0, \ldots, N-1\} = \{0, \ldots, d-1\}$ by labeling each vertex by its depth. Conversely, if there is a valid labeling with image $\{0, \ldots, N-1\}$ then $\mathsf{depth}(G) \leq N$.

Let f be a valid labeling of G with image $\{0, \ldots, d'-1\}$ and for $i \in [k]$ let E_i be the set of edges such that the most significant bit in which the binary encoding of the labels of their endpoints differ is i. If E_i is removed, we can obtain a valid relabeling of the graph with image $\{0, \ldots, d'/2 - 1\}$ by removing the i-th bit from all labels.

The two smallest sets among the E_i-s have size at most $2m/k \leq 4m/\log n$ (since $k = \log d' \geq \log n/2$), and removing them gives a valid labeling with image $\{0, \ldots, d'/4 - 1\}$, and therefore a graph with depth at most $d'/4 \leq d/2$. $\square$

We need a slight variation of this lemma, in which we do not pick edges whose endpoints have too small or too large a depth in the graph.

Lemma 4.4. *Let G be a directed acyclic graph with m edges and depth $d \geq \sqrt{n}$. Then, there exist a set U of vertices, of size at most $4m/\log n$, such for every $v \in U$ we have that $d/9 \leq \mathsf{depth}(v) \leq 8d/9$, and removing U (and the edges touching those vertices) results in a graph of depth at most $3d/4$.*

Proof. Let E denote the set of edges of G and $E' \subseteq E$ be the set of edges guaranteed by Lemma 4.3. Let $E_1 \subseteq E'$ be the edges in E' whose heads have depth at most $d/9$, and E_2 be the edges in E' whose heads have depth at least $8d/9$. Let $E'' = E' \setminus (E_1 \cup E_2)$. Clearly, $|E''| \leq |E'| \leq 4m/\log n$. Let U be the set of heads of vertices in E''.

Consider now any path in the graph obtained from G by removing U (and hence in particular E''). Given such a path, let e_1 be the last edge from E_1 in the path which appears before all edges from E_2 (if there exists such an edge), and let e_2 the first edge from E_2 (if any) in the path. We partition the path into three (possibly empty) parts: the first part is all the edges which appear

until e_1 (including e_1); the second part is all the edges after e_1 and before e_2; the last part consists of all the edges which appear after e_2 (including e_2). Because the head of e_1 is a vertex of depth at most $d/9$, the first part can contribute at most $d/9$ edges. The second part includes only edges from $E \setminus E'$, and thus its length is at most $d/2$. The last part again has depth at most $d/9+1$, as any path leaving a vertex of depth at least $8d/9$ can have at most that many edges (here we add 1 to account for the edge e_2 itself, since the assumption is on the depth of the head of e_2). Thus, the total length of the path is at most

$$d/9 + d/2 + d/9 + 1 \leq 3d/4\,. \qquad \square$$

The set of vertices given by the lemma above will be the vertices according to which we will cut the ABP. We describe it in the following lemma, and prove some properties of this operation.

Lemma 4.5. *Let $\mathcal{A}$ be an ABP over a field $\mathbb{F}$ of depth $d \geq \sqrt{n}$ computing a polynomial F. Let τ be the number of vertices and m be the number of edges in $\mathcal{A}$. Then, there exist an unlayered ABP $\mathcal{A}'$, with at most $\tau + 4m/\log n$ vertices, at most $m + 8m/\log n$ edges, and depth at most $9d/10$, computing a polynomial of the form $F - \sum_{i=1}^{r} P_i Q_i - \delta$ where $\delta \in \mathbb{F}$ is a field constant, the P_i, Q_i's have no constant term, and $r \leq 4m/\log n$.*

Proof. Let G be the underlying graph of the ABP $\mathcal{A}$. Let $U = \{u_1, \ldots, u_r\}$ be the set of vertices guaranteed by Lemma 4.4, such that $r \leq 4m/\log n$. We perform the following sequence of cuts on $\mathcal{A}$. Set $\mathcal{A}_0 := \mathcal{A}$ and for $i \in [r]$, $\mathcal{A}_i = \mathsf{cut}(\mathcal{A}_{i-1}, u_i)$. Finally $\mathcal{A}' = \mathcal{A}_r$.

The statements of the lemma now follow from the properties of cuts as proved in Claim 4.2. The bound on the number of vertices and edges in $\mathcal{A}'$ is immediate. The claim on the polynomial computed by $\mathcal{A}'$ follows by induction on i.

Finally, by induction on i, we have that the depth of $\mathcal{A}'$ is at most

$$\max\{\mathsf{depth}(\mathcal{A} \setminus U), \mathsf{depth}(u_1)+1, \ldots, \mathsf{depth}(u_r)+1, d - \mathsf{depth}(u_1)+1, \ldots, d - \mathsf{depth}(u_r)+1\}\,,$$

where $\mathcal{A} \setminus U$ is the ABP obtained by removing all vertices in U.

By the choice of U as in Lemma 4.4, for every $i \in [r]$ we have that $d/9 \leq \mathsf{depth}(u_i) \leq 8d/9$, and $\mathsf{depth}(\mathcal{A} \setminus U) \leq 3d/4$, which implies the required upper bound on the depth of $\mathcal{A}'$ (assuming n, and hence d, are large enough). $\square$

Repeated applications of Lemma 4.5 give the following statement.

Corollary 4.6. *Let $\mathcal{A}$ be an ABP over a field $\mathbb{F}$, with edge labels of degree at most $\Delta = n^{o(1)}$, computing an n-variate polynomial F. Further suppose $\mathcal{A}$ has depth at least $\sqrt{n}$, and that the number of edges in $\mathcal{A}$ is at most $n \log n/(1000(\log\log n + \log\Delta))$. Let τ denote the number of vertices in $\mathcal{A}$.*

Then, there exists an ABP $\mathcal{A}'$, whose depth is at most n/Δ, which computes a polynomial of the form $F - \sum_{i=1}^{r} P_i Q_i - \delta$, such that P_i, Q_i are all polynomials without a constant term, $\delta \in \mathbb{F}$ is a field constant, and $r \leq n/10$. The number of vertices in $\mathcal{A}'$ is at most $\tau + n/10$.

Proof. Observe that the depth of $\mathcal{A}$ is at most $d := n \log n$. As long as the depth is at least $\sqrt{n}$, apply Lemma 4.5 repeatedly at most $k := 7(\log\log n + \log\Delta)$ times, to obtain an ABP of depth at most $(0.9)^k \cdot d \leq n/\Delta$.

The upper bound on the number of summands $P_i Q_i$ and the number of vertices after each application is given as a function of the number of edges, which increases in the process. Hence, we first provide a crude estimate on the number of edges at each step. For $i \in [k]$, let $\mathcal{A}_i$ denote the ABP obtained after the i-th application of Lemma 4.5, and let m_i be the number of edges in that ABP.

We claim that by induction on i, $m_i \leq m_0 \cdot (1 + 8/\log n)^i$. This is true for $i = 0$ by definition. For $i \geq 1$, since we maintain the invariant that the depth is at least $\sqrt{n}$, it follows from Lemma 4.5 that

$$m_i \leq m_{i-1} + 8m_{i-1}/\log n = m_{i-1}(1 + 8/\log n) \leq m_0(1 + 8/\log n)^{i-1} \cdot (1 + 8/\log n)\,,$$

where the last inequality uses the induction hypothesis. Thus, the final ABP has at most

$$m_k \leq m_0(1 + 8/\log n)^k \leq 2m_0 = n\log n/(500(\log\log n + \log\Delta)) =: M$$

assuming n is large enough (recall that by assumption we have that $\log\Delta = o(\log n)$, so that $\lim_{n\to\infty}(1 + 8/\log n)^{o(\log n)} = 1$). It is convenient to now use M as a uniform upper bound on the number of edges in all stages of this process, so that each step adds at most $4M/\log n$ summands and vertices. It now follows that r is at most

$$\frac{4kM}{\log n} \leq \frac{7(\log\log n + \log\Delta) \cdot 4n}{500(\log\log n + \log\Delta)} \leq n/10,$$

and similarly the total number of vertices added throughout the process is at most $n/10$. □

The lower bound given in Theorem 1.3 now follows by a simple win-win argument. For convenience, we restate the theorem.

Corollary 4.7. *Let $\mathcal{A}$ be an ABP over a field $\mathbb{F}$, with edge labels of degree at most $\Delta = n^{o(1)}$, computing $\sum_{i=1}^{n} x_i^n$. Then $\mathcal{A}$ has at least $\Omega(n\log n/(\log\log n + \log\Delta)$ edges.*

Proof. Let τ denote the number of vertices in $\mathcal{A}$. If the number of edges is at least $n\log n/(1000(\log\log n + \log\Delta))$, then we already have our lower bound. Else, the number of edges is at most $n\log n/(1000(\log\log n + \log\Delta))$. Now, by Corollary 4.6, there exists an ABP $\mathcal{A}'$,

with $\tau + n/10$ vertices and depth at most n/Δ, computing $\sum_{i=1}^{n} x_i^n - \sum_{j=1}^{r} P_i Q_i - \delta$, such that P_j, Q_j have no constant term, $r \leq n/10$, and $\delta \in \mathbb{F}$.

It thus follows that $\mathcal{A}'$ has formal degree at most n. By Theorem 3.4, it has $\Omega(n^2/\Delta)$ vertices, thus $\tau = \Omega(n^2/\Delta)$, so that the number of edges is also $\Omega(n^2/\Delta)$. □

5 Open problems

We conclude with some open problems.

- A natural open question here is to prove an improved lower bound for unlayered algebraic branching programs. In particular, in the absence of an obvious non-trivial upper bound, it seems reasonable to conjecture that any unlayered ABP computing the polynomial $\sum_{i=1}^{n} x_i^n$ has size at least $\Omega(n^{2-o(1)})$.
- Yet another question which is natural in the context of this work and remains open is to prove stronger lower bounds for ABPs. As a first step towards this, the question of proving super-quadratic lower bound for homogeneous algebraic formulas might be more approachable.

Acknowledgements

We are thankful to Ramprasad Saptharishi for helpful discussions at various stages of this work.

References

[Ben83] Michael Ben-Or. Lower Bounds for Algebraic Computation Trees (Preliminary Report). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC 1983)*, pages 80–86. ACM, 1983.

[Ben94] Michael Ben-Or. Algebraic Computation Trees in Characteristi p>0 (Extended Abstract). In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 534–539. IEEE Computer Society, 1994.

[BS83] Walter Baur and Volker Strassen. The Complexity of Partial Derivatives. *Theoretical Computer Science*, 22:317–330, 1983.

[DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 615–624, 2012.

[EGS75] Paul Erdős, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. *Computers & Mathematics with Applications*, 1(3):365 – 369, 1975.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016.

[HY16] Pavel Hrubeš and Amir Yehudayoff. On Isoperimetric Profiles and Computational Complexity. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, pages 89:1–89:12, 2016. `eccc:TR15-164`.

[Kal85] Kyriakos Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM Journal of Computing*, 14(3):678–687, 1985.

[Kum19] Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. *Computational Complexity*, 28(3):409–435, 2019.

[KW93] Mauricio Karchmer and Avi Wigderson. On Span Programs. In *Proceedings of the 8th Annual Structure in Complexity Theory Conference (Structures 1993)*, pages 102–111. IEEE Computer Society, 1993.

[Nec66] Eduard Ivanovich Nechiporuk. On a Boolean function. *Dokl. Akad. Nauk SSSR*, 169:765–766, 1966.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 410–418, 1991. Available on `citeseer:10.1.1.17.5067`.

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. Available on `citeseer:10.1.1.90.2644`.

[Raz06] Ran Raz. Separation of Multilinear Circuit and Formula Size. *Theory of Computing*, 2(1):121–135, 2006. Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*. Pre-print available at `eccc:TR04-042`.

[RY08] Ran Raz and Amir Yehudayoff. Balancing Syntactically Multilinear Arithmetic Circuits. *Computational Complexity*, 17(4):515–535, 2008.

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

[Smi14] Justin R. Smith. *Introduction to Algebraic Geometry*. Textbooks in Mathematics. Taylor & Francis, 2014.

[Smo97] Roman Smolensky. Easy Lower Bound for a Strange Computational Model. *Computational Complexity*, 6(3):213–216, 1997.

[Str73a] Volker Strassen. Die Berechnungskomplexität Von Elementarsymmetrischen Funktionen Und Von Interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, June 1973.

[Str73b] Volker Strassen. Vermeidung von Divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

[Val77] Leslie G. Valiant. Graph-Theoretic Arguments in Low-Level Complexity. In *Proceedings of the 2nd International Symposium on the Mathematical Foundations of Computer Science (MFCS 1977)*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il