A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits

Suryajith Chillara
Department of CSE
IIT Bombay
Mumbai, India.
suryajith@cse.iitb.ac.in

Christian Engels
Department of CSE
IIT Bombay
Mumbai, India.
christian@cse.iitb.ac.in

Nutan Limaye
Department of CSE
IIT Bombay
Mumbai, India.
nutan@cse.iitb.ac.in

Srikanth Srinivasan
Department of Mathematics
IIT Bombay
Mumbai, India.
srikanth@math.iitb.ac.in

***Abstract*—We study the size blow-up that is necessary to convert an algebraic circuit of product-depth $\Delta + 1$ to one of product-depth Δ in the multilinear setting.**

We show that for every positive $\Delta = \Delta(n) = o(\log n/\log\log n)$, there is an explicit multilinear polynomial $P^{(\Delta)}$ on n variables that can be computed by a multilinear formula of product-depth $\Delta + 1$ and size $O(n)$, but not by any multilinear circuit of product-depth Δ and size less than $\exp(n^{\Omega(1/\Delta)})$. This result is tight up to the constant implicit in the double exponent for all $\Delta = o(\log n/\log\log n)$.

This strengthens a result of Raz and Yehudayoff (Computational Complexity 2009) who prove a quasipolynomial separation for constant-depth multilinear circuits, and a result of Kayal, Nair and Saha (STACS 2016) who give an exponential separation in the case $\Delta = 1$.

Our separating examples may be viewed as algebraic analogues of variants of the Graph Reachability problem studied by Chen, Oliveira, Servedio and Tan (STOC 2016), who used them to prove lower bounds for constant-depth *Boolean* circuits.

***Keywords*-Algebraic Circuit Complexity, Depth Hierarchy Theorem, Multilinear Circuits, Size Lower Bounds**

I. Introduction

This paper deals with a question in the area of *Algebraic Complexity,* which studies the Computational Complexity of any algorithmic task that can be cast as the problem of computing a fixed multivariate polynomial (or polynomials) $f \in \mathbb{F}[x_1, \ldots, x_N]$ on a given $a \in \mathbb{F}^N$. Many fundamental problems such as the Determinant, Permanent, the Fast Fourier Transform and Matrix Multiplication can be captured in this general paradigm. The natural computational model for solving such problems are Algebraic Circuits (and their close relations Algebraic Formulas and Algebraic Branching Programs) which use the algebraic operations of the polynomial ring $\mathbb{F}[x_1, \ldots, x_N]$ to compute the given polynomial.

Our main focus in this paper is on *Small-depth* Algebraic circuits, which are easily defined as follows.[1] Recall that any multivariate polynomial $f \in \mathbb{F}[x_1, \ldots, x_N]$ can be written as a linear combination of terms which are products of variables (i.e. monomials); we call such an expression a $\Sigma\Pi$ circuit for f. In general, such an expression for f could be prohibitively large. More compact representations for f may be obtained if we consider representations as linear combinations of products of linear functions, which we call $\Sigma\Pi\Sigma$ circuits, and subsequent generalizations such as $\Sigma\Pi\Sigma\Pi$ circuits and so on. We consider representations of the form $\Sigma\Pi \cdots O_d$ for some $d = d(N)$ where d is a slow growing function of the number of variables N. We consider such a representation of a polynomial f as a computation of f.

The efficiency of such a computation is captured by the following two complexity measures: the *size* of the corresponding expression, which captures the number of operations used in computing the polynomial; and the *product-depth* of the circuit, which is the number of Πs in the expression for the circuit class (e.g. 1 for $\Sigma\Pi\Sigma$ circuits, 2 for $\Sigma\Pi\Sigma\Pi$ circuits etc.) and which measures in some sense the inductive complexity of the corresponding computation.[2]

This paper is motivated by questions in a general body of results in the area that go by the name of *Depth-reduction.* Informally, the question is: what is the worst case blow-up in size required to convert a circuit of product-depth Δ to one of product-depth $\Delta' < \Delta$? This is an important question, since circuits of smaller depth are frequently easier to analyze and understand, and such results allow us to transfer this understanding to more complex (i.e. higher depth) classes of circuits. Consequently, there have been many results addressing this general question for various models of Algebraic (and also Boolean) computation including Algebraic and Boolean formulas [1], [2], Boolean circuits [3], and Algebraic circuits [4]–[8]. Recently, Tavenas [7] and Gupta, Kamath, Kayal and Saptharishi [8] (building on [4]–[6]) showed that a strong enough exponential lower bound for $\Sigma\Pi\Sigma$ circuits would imply a superpolynomial lower bound for general algebraic circuits. Interesting impossibility results are also known in this direction: for instance, it is known that the result of Tavenas [7], which converts a general circuit to a $\Sigma\Pi\Sigma\Pi$ circuit of subexponential size,

[1]We actually define algebraic *formulas* here, but the distinction is not too important for our results in this paper.

[2]It is also standard in the literature to consider the *depth* of the circuit, which is the number of Σ and Π terms in the defining expression for the circuit class, but the product-depth is frequently nicer (invariant under simple operations such as linear transformations etc.) and is essentially $\lfloor \text{depth}/2 \rfloor$. So we mostly use product-depth. We also state our results in terms of depth.

cannot be improved in the restricted model of *homogeneous*[3] $\Sigma\Pi\Sigma\Pi$ circuits [9]–[12].

Here, we study a more fine-grained version of the question of depth-reduction. We ask: what is the size blow-up in converting a circuit of product-depth $\Delta+1$ to one of product-depth Δ? A natural strategy to carry out such a depth-reduction for a given $(\Sigma\Pi)^{\Delta+1}\Sigma$ expression[4] F is to take some product terms in the expression and interchange them with the inner sum terms via the distributive law. This creates a blow-up in the size of the expression that is exponential in the number of sum terms. It is not hard to show that by choosing the sum terms carefully, one can limit this blow-up to $\exp(s^{1/\Delta+o(1)})$ for constant Δ,[5] where s is the size of the expression of depth $\Delta+1$.

Is this exponential[6] blow-up unavoidable for any Δ? The evidence we do have seems to suggest the answer is yes. For example, in the Boolean setting (where the Σ and Π are replaced by their Boolean counterparts $\bigvee$ and $\bigwedge$), such an exponential *Depth-hierarchy theorem* is a classical result of Håstad [13], with recent improvements by Rossman, Servedio, Tan and Håstad [14]–[16]. Even in the algebraic setting, there have been partial results in this direction [17]–[20]. For *homogeneous* circuits such a result was proved for $\Delta = 1$ in the work of Nisan and Wigderson [17].

We study this question in the *multilinear* setting, where the circuits are restricted to computing *multilinear* polynomials at each stage of computation (multilinear polynomials are polynomials where the degree of each variable is at most 1). This is a fairly natural model of computation for multilinear polynomials, and has been extensively studied in the literature [18], [19], [21]–[25].

Raz and Yehudayoff [18] considered the problem of separating product-depth $\Delta+1$ circuits from product-depth Δ circuits in the multilinear setting and proved a *superpolynomial* separation implying a superpolynomial depth-hierarchy theorem. More precisely, they showed that there are circuits of size s and product-depth $\Delta+1$ such that any product-depth Δ circuit computing the same polynomial must have size at least $s^{(\log s)^{\Omega(1/\Delta)}}$. While this result shows that some blow-up is unavoidable in the multilinear setting, this is still only a quasipolynomial separation and does not completely resolve our original question. More recently, Kayal, Nair and Saha [19] resolved the question completely in the case when $\Delta = 1$ by giving an optimal exponential separation between product-depth 2 and product-depth 1 multilinear circuits.

We extend both these results to prove a strong depth-hierarchy theorem for all small depths. The following is implied by Corollary 9.

Theorem 1. *For each $\Delta = \Delta(n) = o(\log n/\log\log n)$, there is an explicit multilinear polynomial $P^{(\Delta)}$ on n variables that can be computed by a multilinear formula of product-depth $(\Delta+1)$ and linear size, but not by any multilinear circuit of product-depth Δ and size less than $\exp(n^{\Omega(1/\Delta)})$.*

We also prove an analogous result for depth instead of product-depth (Corollary 10). Note that, from the above discussion, the above results are tight up to the constant implicit in the $\Omega(1/\Delta)$.

A. Related Work

We survey here some of the work on depth-hierarchy theorems in the Algebraic and Boolean settings.

As mentioned above, this is a well-known question in the Boolean setting, with a near-optimal separation between different depths due to Håstad [13] (building on [26]–[28]); the separating examples in Håstad's result are the *Sipser functions*, which are computed by linear-sized Boolean circuits of depth $\Delta+1$ but cannot be computed by subexponential-sized depth Δ circuits. A recent variant of this lower bound was proved by Chen et al. [29] for *Skew-Sipser functions*, which in turn was used to prove near-optimal lower bounds for constant-depth Boolean circuits solving variants of the Boolean Graph Reachability problem. Their construction motivates our hard polynomials, as we describe below.

In the algebraic setting, we have separations between fixed constant-depth circuits under the restriction of homogeneity. Nisan and Wigderson [17] show that converting a homogeneous $\Sigma\Pi\Sigma\Pi$ circuit to a homogeneous $\Sigma\Pi\Sigma$ circuit requires an exponential blow-up. A quasipolynomial separation between homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ and $\Sigma\Pi\Sigma\Pi$ circuits was shown by Kumar and Saptharishi [20]. However, as far as we know, nothing is known for larger depths.

When the algebraic circuits are instead restricted to be multilinear, more is known. Raz and Yehudayoff [18] were the first to study this question, and showed a quasipolynomial depth-hierarchy theorem for all constant product-depths. In the case of product-depth 1 vs. product-depth 2, this was strengthened to an exponential separation by Kayal, Nair and Saha [19].

One can ask if the methods of [18], [19] can be used to prove our result. Kayal et al. [19] prove their result by defining a suitable complexity measure for any polynomial $f \in \mathbb{F}[x_1, \ldots, x_N]$, and show that this measure is small for any subexponential-sized $\Sigma\Pi\Sigma$ circuit but large for some linear-sized $\Sigma\Pi\Sigma\Pi$ circuit. In particular, this means that this measure needs to be changed to prove lower bounds for larger depth circuits. However, it is not clear how to modify this measure to take into account the product-depth of the circuit class.

[3]I.e. a circuit where every intermediate expression computes a homogeneous polynomial.

[4]I.e. a $\Sigma\Pi\cdots\Sigma$ expression with exactly $(\Delta+1)$ many Πs.

[5]In fact, the upper bound is of the form $\exp(s^{1/\Delta}\log s)$, and is the $\exp(s^{1/\Delta+o(1)})$ as long as $\Delta = o(\log s/\log\log s)$.

[6]Strictly speaking, we get an exponential blow-up only for *constant* Δ. The careful reader should read "exponential" as "exponential in $s^{1/\Delta}$" for general Δ.

This is not a problem with the technique of Raz and Yehudayoff [18], which can indeed be used to prove exponential lower bounds on the sizes of small-depth circuits for computing certain polynomials. However, the polynomials used to witness the superpolynomial separation cannot give an exponential depth-hierarchy, for the reasons we now explain.

The depth-hierarchy theorem of [18] is obtained via an exponential lower bound $s(n, \Delta) \approx \exp(n^{1/\Delta})$ against product-depth Δ circuits computing polynomials from a certain "hard" class $\mathcal{P}$ of polynomials on n variables. Importantly, these lower bounds are tight in the sense that there also exist product-depth Δ circuits of size roughly $s(n, \Delta)$ computing polynomials from $\mathcal{P}$. Since $s(n, \Delta - 1)$ is superpolynomially larger than $s(n, \Delta)$, we obtain a superpolynomial separation between circuits of product-depth Δ and product-depth $\Delta - 1$. However, since we cannot improve on either the upper bound of $s(n, \Delta)$ or the lower bound of $s(n, \Delta - 1)$ for this class of polynomials, we cannot hope to improve this separation.

There is a striking parallel between this line of work and the setting of Boolean circuits, where also a similar quasipolynomial depth-hierarchy theorem can be obtained by appealing to easier lower bounds for explicit functions such as the "Parity" function [13]. It is known that the depth-Δ Boolean complexity of the Parity function is $\exp(\Theta(n^{1/\Delta-1}))$ and this yields a quasipolynomial depth-hierarchy theorem for constant-depth Boolean circuits as in the work of Raz and Yehudayoff described above. However, Håstad [13] was able to improve this to an exponential depth-hierarchy theorem by changing the candidate hard function to the Sipser functions and then proving a lower bound for these functions by a related, but more involved, technique [13], [29].

B. Proof Outline

As mentioned in the Related Work section above, the polynomials considered by Raz and Yehudyaoff [18] cannot be used to prove better than a quasipolynomial depth-hierarchy theorem. This parallels a quasipolynomial depth-hierarchy theorem in the Boolean setting. However, in the Boolean setting, there is a different family of explicit functions that can be used to prove an exponential depth-hierarchy theorem.

Our aim is to do something similar in the setting of multilinear small-depth circuits. While the methods for proving Boolean circuit lower bounds do not seem to apply in the algebraic setting, we can take inspiration in the matter of choosing the candidate hard polynomial. For this, we look to the recent result of Chen et al. [29], who observe that the Sipser functions (and also their "skew" variants) can be interpreted as special cases of the Boolean Graph Reachability problem. This is quite appealing for us, since Graph Reachability has a natural polynomial analogue, the Iterated Matrix Multiplication polynomial, which has been a source of many lower bounds in algebraic circuit complexity [10], [12], [17], [19], [30]–[32]. We therefore choose our lower bound candidate to be a restriction of the Iterated Matrix Multiplication polynomial, which we now describe.

1) The Hard Polynomials: All the polynomials we consider will be naturally defined in terms of Directed Acyclic Graphs (DAGs) with a unique source and sink. Given such a graph G with source s and sink t, we define a corresponding polynomial P_G as follows. Label each edge e of G with a distinct variable x_e. The polynomial P_G is defined to be the sum, over all paths π from s to t, of the monomial which is the product of edge labels along that path. This polynomial P_G is the algebraic analogue of the Boolean Computational problem of checking s-t reachability on subgraphs of G. (Informally, we think of each x_e as a Boolean variable that determines if e remains in the subgraph or not. Then the polynomial P_G on the Boolean input corresponding to a subgraph H of G counts the number of s-t paths in H.)

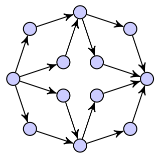
When G is a layered graph with d layers and all possible edges between consecutive layers, P_G is known as the Iterated Matrix Multiplication polynomial (for the connection to matrix product, see, e.g., [10]). This polynomial has been studied in many contexts in Algebraic circuit complexity. Unfortunately, it is not useful in our setting, since it does not have a subexponential-size constant-depth multilinear circuit, as was shown recently by some of the authors [32]. In particular, this means that it cannot be used to obtain the claimed separation between depths $\Delta + 1$ and Δ.

However, changing G can drastically reduce the complexity of the polynomial P_G. To obtain a G such that P_G has an efficient depth $\Delta + 1$ circuit, we use *series-parallel* graphs, as in the result of Chen et al. [29].

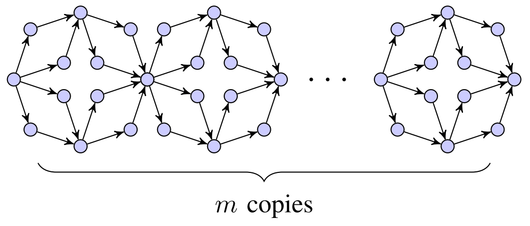
Given DAGs $G_1, \ldots, G_k$ with sources $s_1, \ldots, s_k$ and sinks $t_1, \ldots, t_k$, we can construct larger DAGs by composing these graphs in parallel (by identifying all the sources together as a new source and all the sinks together as a new sink) or in series (by identifying t_1 with s_2, t_2 with s_3 and so on until t_{k-1} with s_k) to get larger DAGs G_{par} or G_{ser} respectively. Note that the corresponding polynomials P_{par} and P_{ser} are the sums and products of the polynomials $P_1, \ldots, P_k$, respectively. In particular, if $P_1, \ldots, P_k$ have efficient circuits of product-depth at most Δ, then P_{par} (resp. P_{ser}) has an efficient circuit of product-depth at most Δ (resp. $\Delta + 1$).

In this way, we can inductively construct polynomials which, by their very definition, have efficient circuits of small depth. In particular, to keep the product-depth of the circuit bounded by $\Delta + 1$, it is sufficient to ensure that the total number of series compositions used in the construction of the graph is at most $\Delta + 1$.

2) The $\Sigma\Pi\Sigma$ lower bound: We motivate our proof with the solution to the simpler problem of separating product-



(a) $H^{(1)}$.



(b) $G^{(1)}$.

Figure 1.

depth 2 and product-depth 1 circuits. In fact, we will separate the power of $\Pi\Sigma\Pi$ and $\Sigma\Pi\Sigma$ circuits. While an exponential separation was already known in this case thanks to the work of Kayal et al. [19], we will outline a different proof that extends to larger depths.

Given the above discussion, a natural polynomial to witness the separation between product-depth 1 and product-depth 2 circuits is a polynomial corresponding to a series parallel graph obtained with exactly 2 series compositions. Unfortunately, our proof technique is not able to prove a lower bound for such a graph. However, we are able to prove such a separation with the slightly more complicated graph $G^{(1)}$ in Figure 1(b), which is made of a composition of m copies of the basic graph $H^{(1)}$ (see Figure 1(a)). Though the graph $G^{(1)}$ is constructed using three series compositions rather than two, the corresponding polynomial $P^{(1)}(X)$ nevertheless has a product-depth 2 circuit of small size (this is because the polynomial corresponding to $H^{(1)}$ depends only on a constant-number of variables and hence has a "brute-force" $\Sigma\Pi$ circuit).

We now describe how to prove that any $\Sigma\Pi\Sigma$ circuit for $P^{(1)}$ must have large size (the lower bound we obtain is $2^{\Omega(m)}$). Let F be a $\Sigma\Pi\Sigma$ circuit of size s computing $P^{(1)}$ and assume s is small. We can write F as $T_1 + \cdots + T_s$, where each T_i is a product of linear polynomials.

The lower bound is via a rank argument first used by Raz [21]. The idea is to associate a rank-based complexity measure with any polynomial f, and show that while the polynomial $P^{(1)}$ has rank as large as possible, the rank of F must be small.

This rank measure is defined as follows. We partition the variables X in our polynomial into two sets Y and Z and consider any polynomial $f(X)$ as a polynomial in the variables in Y with coefficients from $\mathbb{F}[Z]$. The rank of the space of coefficients (as vectors over the base field $\mathbb{F}$) is considered a measure of the complexity of f.

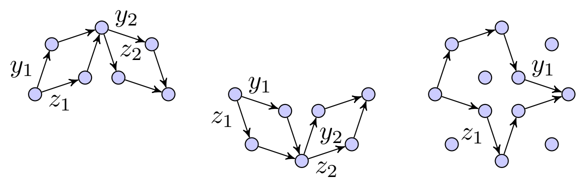


Figure 2. Restrictions applied to each copy of $H^{(1)}$. Edges that are not labelled have their variables set to 1. Edges that are not present have their variables set to 0.

It is easy to come up with a partition of the underlying variable set X into Y, Z so that the complexity of our polynomial $P^{(1)}$ is as large as it can be. Unfortunately, it is also easy to find $\Sigma\Pi\Sigma$ formulas that have maximum rank w.r.t. this partition. Hence, this notion of complexity is not by itself sufficient to prove a lower bound. At this point, we follow an idea of Raz [21] and show something stronger for $P^{(1)}$: we show that its complexity is *robust* in the sense that it is full rank w.r.t. many different partitions.

More precisely, we carefully design a large space of *restrictions* $\rho^{(1)} : X \to Y \cup Z \cup \{0, 1\}$ such that for any such restriction $\rho^{(1)}$, the resulting substitution of $P^{(1)}$, which we will call a *restriction of* $P^{(1)}$, continues to have full rank, but the rank of the restriction of F is small under many of them. We define these restrictions now.

The definition of the space of restrictions is motivated by the following easily verified observation (also used in many previous results [18], [22], [23], [25]): any polynomial of the form

$$\prod_{i\in[t]} (y_i + z_i) \tag{1}$$

is full-rank w.r.t. the natural partition $Y = \{y_1, \ldots, y_t\}$ and $Z = \{z_1, \ldots, z_t\}$. Given this, a possible option is to have the restriction $\rho^{(1)}$ set variables in each copy of $H^{(1)}$ in ways that ensure that after substitution, the polynomial is of the form in (1). We choose one among the three different (up to isomorphism) restrictions shown in Figure 2. It can be checked that each such restriction results in a polynomial of the form in (1) and hence is full rank. Since $P^{(1)}$ is a product of many disjoint copies of this basic polynomial, it remains full rank also.

Now, to show that F has small rank under some such restriction, we apply a *uniformly random* copy of such a restriction $\rho^{(1)}$ to F and show that with high probability each of its constituent terms $T_1, \ldots, T_s$ is very small in rank. Since rank is subadditive and s is assumed to be small, this implies that F cannot be full rank after the restriction and hence cannot have been computing the polynomial $P^{(1)}$.

Fix a term T_i which is a product of linear functions as mentioned above. We argue that it restricts to a small-rank term with high probability as follows.

- A standard argument [21] shows that, by multilinearity,[7] the rank of T_i is small if many of its constituent linear functions have smaller than "full-rank" after applying $\rho^{(1)}$. Here, a linear polynomial $L(Y', Z')$ ($Y' \subseteq Y$, $Z' \subseteq Z$) is said to be full-rank if its complexity (as defined above) is $2^{(|Y'|+|Z'|)/2}$.

- By a simple argument we can argue that, upon applying a random restriction $\rho^{(1)}$, any constituent linear function L of T_i becomes rank-deficient (i.e. short of full-rank) with good (constant) probability. This is by a case analysis on the set of variables X' that appear in L and proceeds roughly as follows.
  - Say X' is *structured* in that it is the union of the variable sets in some copies of $H^{(1)}$. In this case, we observe that with good probability, the total number of variables in L after restriction is larger than 2 and hence, the target full-rank is a number larger than 2. On the other hand, any linear function cannot have rank more than 2.
  - On the other hand if X' is not structured in the above sense, then it can be shown that with good probability, the number of restricted variables in L is odd and then it cannot be full-rank, since the target full-rank is $2^{t+1/2}$ for some integer t, while the rank of L can be at most 2^t.
- Given the above, we can easily argue using a concentration bound[8] that with high probability, T_i has *many* rank-deficient linear functions and consequently has small rank. This yields the proof of the $\Sigma\Pi\Sigma$ lower bound.

3) Separating Δ from $\Delta + 1$: To prove the lower bound for product-depth Δ circuits, we proceed by induction on Δ. Each step in the inductive proof is analogous to a step in the $\Sigma\Pi\Sigma$ lower bound so we will be brief.

The graph $H^{(\Delta)}$ is constructed from two copies of $G^{(\Delta-1)}$ by parallel composition and the graph $G^{(\Delta)}$ is constructed from many copies of $H^{(\Delta)}$ by series composition as shown in Figure 3. By construction $P^{(\Delta)}$ will have a circuit of product-depth $\Delta + 1$.

The lower bound is again proved via a random restriction argument. We define the random restriction $\rho^{(\Delta)}$ by restricting each copy of $H^{(\Delta)}$ independently by choosing one of these options at random.

- set it to a polynomial of the form $(y + z)$ by choosing a random path in each copy of $G^{(\Delta-1)}$ and setting a single variable in each to y or z (other variables in the path are set to 1 and off-path variables are set to 0),
- inductively applying the restriction $\rho^{(\Delta-1)}$ to one of the copies of $G^{(\Delta-1)}$ (setting the other one to 0).

As before, the polynomial $P^{(\Delta)}$ remains full-rank after the

[7]This essentially means that the linear functions that participate in T_i do not share any variables.

[8]By multilinearity, the events that the various linear functions are small rank become (roughly) independent.

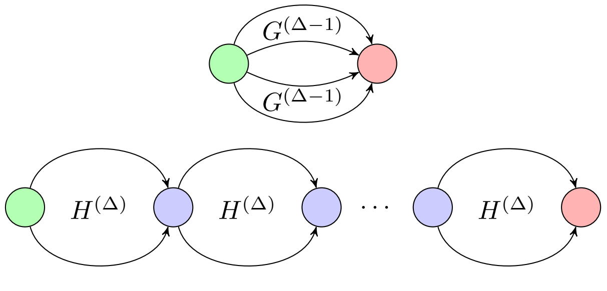


Figure 3. $H^{(\Delta)}$ (left) and $G^{(\Delta)}$.

restriction with probability 1, since it always transforms into a polynomial of the form (1) after the restrictions.

Let F be a small product-depth Δ circuit. To argue that it is low-rank w.h.p. after applying $\rho^{(\Delta)}$, we prove a variant of a decomposition lemma from the literature that allows us to write $F = T_1 + \ldots + T_s$ where each T_i now is a product of circuits each of which has small size and product depth at most $\Delta - 1$.

As before, we argue that each T_i is low-rank with high probability. Say $T_i = \prod_{j\in[t]} Q_{i,j}$. To show this, it suffices to show that each $Q_{i,j}$ is somewhat short of full-rank with good probability. This does indeed turn out to be true in one of two ways.

- If the variable set X' of $Q_{i,j}$ is structured (as defined above), then we can use induction and show that it is low-rank if we happen to apply the restriction $\rho^{(\Delta-1)}$ to the variables of $Q_{i,j}$. This happens with constant probability.
- On the other hand, if $Q_{i,j}$ is unstructured, then with good probability, $Q_{i,j}$ is left with an odd number of variables, which implies that it is rank-deficient exactly as before.

At an intuitive level, the above cases correspond to two different ways in which a small circuit can try to compute the hard polynomial $P^{(\Delta)}$. In the first case, it uses the fact that the polynomial $P^{(\Delta)}$ is a product of many polynomials and tries to compute each of them with a smaller circuit of depth $\Delta - 1$; in this case, we use the inductive hypothesis to show that this cannot happen. In the second case, the circuit tries to do some non-trivial (i.e. unexpected) computation at the top level and in this case, we argue directly that the circuit fails.

This finishes the sketch of an idealized version of the argument. While the above strategy can be carried out as stated, it would yield a sub-optimal but still exponential bound of the form $\exp(n^{\alpha_\Delta})$ for some $\alpha_\Delta > 0$ that depends on Δ. To obtain the near-optimal lower bound of $\exp(n^{\Omega(1/\Delta)})$, we need to make the following changes to the above idealized proof strategy.

- We expand the set of "structured" polynomials to allow for polynomials $Q_{i,j}$ (and inductively formulas computed at smaller depths) that compute polynomials over a non-trivial fraction, say ε, of the variables in many different copies, say M, of $H^{(\Delta-1)}$. The exact relationship between ε and M is somewhat delicate (we choose $\varepsilon \geq 1/M^{0.25}$) and must be chosen carefully for the proof to yield an optimal lower bound.

- Given this, we also need to handle (for the general inductive statement) the case that F does not depend on all the variables in $G^{(\Delta)}$ but rather at least δ-fraction of the variables in K many copies of $H^{(\Delta)}$ for some suitable δ and K. Given a single term $T_i = \prod_j Q_{i,j}$ in F as above, one of the following two cases occurs.
  - One of the $Q_{i,j}$s depends on at least a δ'-fraction of the variables in K' many copies of $H^{(\Delta)}$, where δ' and K' are not much smaller than δ and K respectively. This corresponds to the structured case above and in this case, we can actually use the induction hypothesis to show that we are done with high probability.
    To argue this, we have to use the fact that while δ has decreased to δ', the K' many copies of $H^{(\Delta)}$ contain many more copies of $H^{(\Delta-1)}$. Hence the number of copies of $H^{(\Delta-1)}$, say K'', and δ' still have the "correct" relationship so that the inductive statement is applicable.
  - Otherwise, in many copies of $H^{(\Delta)}$, the δ-fraction that are variables of F are further partitioned into parts of relative size $< \delta'$ by the variable sets of the $Q_{i,j}$. In this case, we have to argue that many of the $Q_{i,j}$ are rank-deficient with high probability. This is the most technical part of the proof and is done by a careful re-imagining of the sampling process that defines the random restriction.

The more general inductive statement (and some additional technicalities that force us to carry around auxiliary sets of Y and Z variables) results in a technically complicated theorem statement (see Theorem 11), which yields the near-optimal depth-hierarchy theorem.

Organization.: We start with some Preliminaries in Section II. We define the hard polynomials that we use in Section III. The general inductive statement (Theorem 11) is given in Section IV, which also contains the proof of the main depth-hierarchy theorem (Corollary 9) assuming Theorem 11. Owing to the space constraints, we omit the proof of Theorem 11 here and it can be found in the full version [33] of our paper, along with the other missing preliminaries and proofs.

II. Preliminaries

A. Polynomials and restrictions

Throughout, let $\mathbb{F}$ be an arbitrary field. A polynomial $P \in \mathbb{F}[X]$ is called *multilinear* if the degree of P in each variable $x \in X$ is at most 1.

Let X, Y, Z be disjoint sets of variables. An (X, Y, Z)-*restriction* is a function $\rho : X \to Y \cup Z \cup \{0, 1\}$. The sets X, Y and Z will sometimes be omitted when clear from context. We say that the restriction ρ is *multilinear* if no two variables in X are mapped to the same variable in $Y \cup Z$ by ρ. A *random* (X, Y, Z)-restriction is simply a random function $\rho : X \to Y \cup Z \cup \{0, 1\}$ (chosen according to some distribution).

Let X, Y, Z be as above and say we have for each $i \in [k]$, an (X_i, Y_i, Z_i)-restriction ρ_i where $X_i \subseteq X, Y_i \subseteq Y$ and $Z_i \subseteq Z$. If $X_1, \ldots, X_k$ form a (pairwise disjoint) partition of X, we define their *composition* — denoted $\rho_1 \circ \rho_2 \cdots \circ \rho_k$ — to be the (X, Y, Z)-restriction ρ such that $\rho(x)$ agrees with $\rho_i(x)$ for each $x \in X_i$. Further, if restrictions ρ_i are multilinear restrictions and the sets Y_i ($i \in [k]$) and Z_j ($j \in [k]$) are pairwise disjoint, then $\rho_1 \circ \cdots \circ \rho_k$ is also multilinear.

Let ρ be an (X, Y, Z)-restriction. Given a polynomial $f \in \mathbb{F}[X]$, the restriction ρ yields a natural polynomial in $\mathbb{F}[Y \cup Z]$ obtained from f by substitution; we denote this polynomial $f|_\rho$. Note, moreover, that if f is multilinear and ρ is multilinear, then so is $f|_\rho$.

B. Partial derivative matrices and relative rank

Let Y and Z be two disjoint sets of variables and let $g \in \mathbb{F}[Y \cup Z]$ be a multilinear polynomial. Define the $2^{|Y|} \times 2^{|Z|}$ matrix $M_{(Y,Z)}(g)$ whose rows and columns are labelled by distinct multilinear monomials in Y and Z respectively and the (m_1, m_2)th entry of $M_{(Y,Z)}(g)$ is the coefficient of the monomial $m_1 \cdot m_2$ in g. We will use the rank of this matrix as a measure of the complexity of g.

We define the *relative-rank* of g w.r.t. (Y, Z) (denoted by $\mathrm{relrk}_{(Y,Z)}(g)$) by

$$\mathrm{relrk}_{(Y,Z)}(g) = \frac{\mathrm{rank}(M_{(Y,Z)}(g))}{2^{(|Y|+|Z|)/2}}.$$

We note the following properties of relative-rank [18].

Proposition 2. *Let $g, g_1, g_2 \in \mathbb{F}[Y \cup Z]$ be multilinear polynomials.*

1) $\mathrm{relrk}_{(Y,Z)}(g) \leq 1$. *Further if $|Y| + |Z|$ is odd,* $\mathrm{relrk}_{(Y,Z)}(g) \leq 1/\sqrt{2}$.
2) $\mathrm{relrk}_{(Y,Z)}(g_1 + g_2) \leq \mathrm{relrk}_{(Y,Z)}(g_1) + \mathrm{relrk}_{(Y,Z)}(g_2)$.
3) *If Y is partitioned into Y_1, Y_2 and Z into Z_1, Z_2 with $g_i \in \mathbb{F}[Y_i \cup Z_i]$ ($i \in [2]$), then* $\mathrm{rank}(M_{(Y,Z)}(g)) = \mathrm{rank}(M_{(Y_1,Z_1)})(g_1) \cdot \mathrm{rank}(M_{(Y_2,Z_2)})(g_2)$. *In particular,* $\mathrm{relrk}_{(Y,Z)}(g_1 \cdot g_2) = \mathrm{relrk}_{(Y_1,Z_1)}(g_1) \cdot \mathrm{relrk}_{(Y_2,Z_2)}(g_2)$.

C. Multilinear models of computation

We refer the reader to the standard resources (e.g. [34], [35]) for basic definitions related to algebraic circuits and formulas. Having said that, we make a few remarks.

- All the gates in our formulas and circuits will be allowed to have *unbounded* fan-in.

- The size of a formula or circuit will refer to the number of gates (including input gates) in it, and the depth of the formula or circuit will refer to the maximum number of gates on a path from an input gate to the output gate.
- Further, the *product-depth* of the formula or circuit (as in [23]) will refer to the maximum number of product gates on a path from the input gate to output gate. Note that if a formula or circuit has depth Δ, we can assume without loss of generality that its product depth is between $\lfloor \Delta/2 \rfloor$ and $\lceil \Delta/2 \rceil$ (by collapsing sum and product gates if necessary).

An algebraic formula F (resp. circuit C) computing a polynomial from $\mathbb{F}[X]$ is said to be *multilinear* if each gate in the formula (resp. circuit) computes a multilinear polynomial. Moreover, a formula F is said to be *syntactic multilinear* if for each $\times$-gate Φ of F with children $\Psi_1, \ldots, \Psi_t$, we have $\mathrm{Supp}(\Psi_i) \cap \mathrm{Supp}(\Psi_j) = \emptyset$ for each $i \neq j$, where $\mathrm{Supp}(\Phi)$ denotes the set of variables that appear in the subformula rooted at Φ. Finally, for $\Delta \geq 1$, we say that a multilinear formula (resp. circuit) is a $(\Sigma\Pi)^{\Delta}\Sigma$ formula (resp. circuit) if the output gate is a sum gate and along any path from an input gate to the output gate, the sum and product gates alternate, with product gates appearing exactly Δ times and the bottom gate being a sum gate. We can define $(\Sigma\Pi)^{\Delta}, \Sigma\Pi\Sigma, \Sigma\Pi\Sigma\Pi$ formulas and circuits similarly.

An *Algebraic Branching Program* (ABP), over the set of variables X and field $\mathbb{F}$ is a layered (i.e. the edges are only between two consecutive layers) directed acyclic graph G with two special vertices called the *source* and the *sink*. The label of an edge is a linear polynomial in $\mathbb{F}[X]$. The weight of a path is the product of the labels of its edges. The polynomial computed by G is the sum of the weights of all the paths from s to t in G.

A *Multilinear ABP* (mlABP) is an algebraic branching program such that for any path from the source to sink, the labels of edges on that path are linear polynomials over pairwise-disjoint sets of variables.

Composing ABPs in series and in parallel.: We say that G is obtained by composing $G_1, \ldots, G_k$ *in series* if G is obtained by identifying t_i with s_{i+1} for each $i \in [k-1]$ to obtain an ABP with source s_1 and sink t_k. Note that the polynomial computed by G is the product of the polynomials computed by $G_1, \ldots, G_k$.

Let $G_1, \ldots, G_k$ be ABPs (on disjoint sets of vertices) with sources $s_1, \ldots, s_k$ and sinks $t_1, \ldots, t_k$. We say that G is obtained by composing $G_1, \ldots, G_k$ *in parallel* if G is obtained by identifying all sources $s_1, \ldots, s_k$ to obtain a single source s and all the sinks $t_1, \ldots, t_k$ to obtain a single sink t. Note that the polynomial computed by G is the sum of the polynomials computed by $G_1, \ldots, G_k$. Further if $G_1, \ldots, G_k$ are mlABPs over disjoint sets of variables, then G is also an mlABP.

D. Some structural lemmas

Given a syntactically multilinear formula F computing a polynomial $P \in \mathbb{F}[X]$, we define a *variable-set labelling* of F to be a labelling function that assigns to each gate Φ of F a set $\mathrm{Vars}(\Phi) \subseteq X$ with the following properties.

1) For any gate Φ in F, $\mathrm{Supp}(\Phi) \subseteq \mathrm{Vars}(\Phi)$.
2) If Φ is an sum gate, with children $\Psi_1, \Psi_2, \ldots, \Psi_k$, then $\forall i \in [k]$, $\mathrm{Vars}(\Psi_i) = \mathrm{Vars}(\Phi)$.
3) If Φ is a product gate, with children $\Psi_1, \Psi_2, \ldots, \Psi_k$, then $\mathrm{Vars}(\Phi) = \cup_{i=1}^{k} \mathrm{Vars}(\Psi_i)$ and the sets $\mathrm{Vars}(\Psi_i)$ ($i \in [k]$) are pairwise disjoint.

We call a syntactically multilinear formula F *variable-labelled* if it is equipped with a labelling function as above. For such an F, we define $\mathrm{Vars}(F)$ to be $\mathrm{Vars}(\Phi_0)$, where Φ_0 is the output gate of F.

The following proposition (cf. [33] for the proof) shows that we may always assume that a syntactically multilinear formula F is variable-labelled.

Proposition 3. *For each syntactically multilinear formula F computing $P \in \mathbb{F}[X]$, there is a variable-labelled syntactically multilinear formula F' of the same size as F computing P such that* $\mathrm{Vars}(\Phi) \subseteq X$ *for each gate Φ of F' and* $\mathrm{Vars}(F) = X$.

We will use the following structural result that converts any small-depth multilinear circuit to a small-depth syntactic multilinear formula without a significant blowup in size.

Lemma 4 (Raz and Yehudayoff [18], Lemma 2.1)**.** *For any multilinear circuit C of product-depth at most Δ and size at most s, there is a syntactic multilinear $(\Sigma\Pi)^{\Delta}\Sigma$ formula F of size at most $(\Delta+1)^2 \cdot s^{2\Delta+1}$ computing the same polynomial as C.*

III. The Hard polynomials and their restrictions

A. The Hard polynomials

The hard polynomial for circuits of product-depth Δ will be defined to be the polynomial computed by a suitable ABP. The definition is by induction on the product depth Δ. We also define some auxiliary variable sets that will be useful later when we restrict these polynomials.

Let $m \in \mathbb{N}$ be a growing parameter. We will define for each $\Delta \geq 1$ an ABP $G^{(\Delta)}$ on a variable set $X^{(\Delta)}$ of size n_{Δ}, along with some auxiliary variable sets $Y^{(\Delta)}$ and $Z^{(\Delta)}$. The parameter n_{Δ} itself is defined inductively as follows. Let $n_0 = 8$ and for each $\Delta \geq 1$, define

$$n_{\Delta} = 2 \cdot m \cdot n_{\Delta-1}. \tag{2}$$

Observe that $n_{\Delta} = 8 \cdot (2m)^{\Delta}$ for $\Delta \geq 1$.

The Variable sets.: For each $\Delta \geq 0$, we will define three disjoint variable sets $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$. We will have $|X^{(\Delta)}| = n_\Delta$ for each Δ.

Let $X^{(0)}$ be the set $\{x_{i,j}^{(0)} \mid i \in [4], j \in [2]\}$, $Y^{(0)} = \{y_1^{(0)}, y_2^{(0)}\}$ and $Z^{(0)} = \{z_1^{(0)}, z_2^{(0)}\}$. Note that $|X^{(0)}| = n_0 = 8$.

Given $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$, we now define $X^{(\Delta+1)}, Y^{(\Delta+1)}$ and $Z^{(\Delta+1)}$ as follows.

Clones, segments and half-segments.: For each $i \in [m]$ and $j \in [2]$, let $X_{i,j}^{(\Delta+1)}$, $Y_{i,j}^{(\Delta+1)}$ and $Z_{i,j}^{(\Delta+1)}$ be pairwise disjoint copies of $X^{(\Delta)}, Y^{(\Delta)}$ and $Z^{(\Delta)}$ respectively. Define $X^{(\Delta+1)} = \bigcup_{i,j} X_{i,j}^{(\Delta+1)}$; note that $|X^{(\Delta+1)}| = 2m|X^{(\Delta)}| = n_{\Delta+1}$ as desired. Define $Y^{(\Delta+1)}$ to be $\bigcup_i Y_{i,1}^{(\Delta+1)} \cup Y_{i,2}^{(\Delta+1)} \cup \{y_i^{(\Delta+1)}\}$ and $Z^{(\Delta+1)}$ to be $\bigcup_i Z_{i,1}^{(\Delta+1)} \cup Z_{i,2}^{(\Delta+1)} \cup \{z_i^{(\Delta+1)}\}$, where $y_i^{(\Delta+1)}$ and $z_i^{(\Delta+1)}$ are fresh variables.

By the inductive definition of the sets $X^{(\Delta+1)}$ above, we see that $X^{(\Delta+1)}$ is made up of $2m$ copies of $X^{(\Delta)}$, which we denote by $X_{i,j}^{(\Delta+1)}$ for $i \in [m]$ and $j \in [2]$. Using this fact inductively, we see that for any $t \in \{0, \ldots, \Delta+1\}$, $X^{(\Delta+1)}$ contains $(2m)^t$ copies of $X^{(\Delta+1-t)}$. Each such copy is uniquely labelled by a tuple $\omega = ((i_1, j_1), \ldots, (i_t, j_t)) \in ([m] \times [2])^t$; we denote this copy by $X_\omega^{(\Delta+1)}$ and call this an $X^{(\Delta+1-t)}$*-clone*. In a similar way, we see that for each $\omega \in ([m] \times [2])^t$, $Y^{(\Delta+1)}$ and $Z^{(\Delta+1)}$ contain copies of $Y^{(\Delta+1-t)}$ and $Z^{(\Delta+1-t)}$ respectively, which we denote as $Y_\omega^{(\Delta+1)}$ and $Z_\omega^{(\Delta+1)}$ respectively, and call $Y^{(\Delta+1-t)}$-clones and $Z^{(\Delta+1-t)}$-clones respectively.

Say $X_\omega^{(\Delta')}$ is an $X^{(\Delta)}$-clone for some $\Delta' \geq \Delta \geq 1$. Then we refer to $X_{(\omega,(i,j))}^{(\Delta')}$ (which is an $X^{(\Delta-1)}$-clone) as the (i,j)th *half-segment* of $X_\omega^{(\Delta')}$. Further, we refer to $X_{(\omega,(i,1))}^{(\Delta')} \cup X_{(\omega,(i,2))}^{(\Delta')}$ as the ith *segment* of $X_\omega^{(\Delta')}$, which we will denote $X_{(\omega,i)}^{(\Delta')}$; note that $X_\omega^{(\Delta')}$ is a union of its m segments. Similarly a $Y^{(\Delta)}$-clone $Y_\omega^{(\Delta')}$ (resp. a $Z^{(\Delta)}$-clone $Z_\omega^{(\Delta')}$) is a disjoint union of m segments $Y_{(\omega,i)}^{(\Delta')}$ (resp. $Z_{(\omega,i)}^{(\Delta')}$) for $i \in [m]$, the ith of which is made up of two half-segments $Y_{(\omega,(i,1))}^{(\Delta')}$ and $Y_{(\omega,(i,2))}^{(\Delta')}$ (resp. $Z_{(\omega,(i,1))}^{(\Delta')}$ and $Z_{(\omega,(i,2))}^{(\Delta')}$) and an additional fresh variable that we denote $y_{(\omega,i)}^{(\Delta')}$ (resp. $z_{(\omega,i)}^{(\Delta')}$). We refer to a set of the form $X_{(\omega,i)}^{(\Delta')}$ as an $X^{(\Delta)}$-segment[9] and a set of the form $X_{(\omega,(i,j))}^{(\Delta')}$ as an $X^{(\Delta)}$-half-segment (similarly $Y^{(\Delta)}$-segment, $Y^{(\Delta)}$-half-segment, $Z^{(\Delta)}$-segment and $Z^{(\Delta)}$-half-segment).

To summarize, given any $\Delta' \geq \Delta \geq 0$ the set $X^{(\Delta')}$ contains $(2m)^{\Delta'-\Delta}$ many $X^{(\Delta)}$-clones which are indexed by elements of the set $([m] \times [2])^{\Delta'-\Delta}$. If $\Delta \geq 1$, each such $X^{(\Delta)}$-clone contains m many $X^{(\Delta)}$-segments, which are indexed by the set $([m] \times [2])^{\Delta'-\Delta} \times [m]$; and each segment contains two $X^{(\Delta-1)}$-clones, also referred to as $X^{(\Delta)}$-half-segments. Finally, these definitions extend naturally to $Y^{(\Delta')}$- and $Z^{(\Delta')}$-clones.

[9]The reader may want to read "$X^{(\Delta)}$-segment" as "a segment of (a clone of) $X^{(\Delta)}$" and similarly for other segments and half-segments.

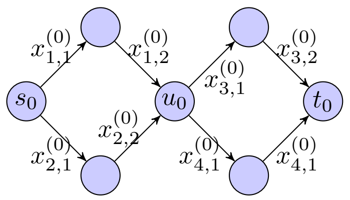


Figure 4. The ABP $G^{(0)}$. All edges go from left to right.

The Hard polynomial.: The hard polynomial for product-depth Δ is defined by induction on Δ.

Define $G^{(0)}$ to be an ABP as follows. Assume that $G^{(0)}$ has source vertex s_0, sink vertex t_0 and an intermediate vertex u_0. The graph of the ABP consists of two disjoint paths π_1 and π_2 of length 2 each from s_0 to u_0 and two disjoint paths π_3 and π_4 of length 2 each from u_0 to t_0 (see Figure 4). The edges of π_i are labelled by the distinct variables $x_{i,1}^{(0)}$ and $x_{i,2}^{(0)}$. Let $X^{(0)}$ be the set $\{x_{i,j}^{(0)} \mid i \in [4], j \in [2]\}$ of variables that appear in $G^{(0)}$. Note that $G^{(0)}$ computes a polynomial over the variable set $X^{(0)}$ defined above.

Now fix any $\Delta \geq 0$. Given $G^{(\Delta)}$, an ABP on variable set $X^{(\Delta)}$, we inductively define the ABP $G^{(\Delta+1)}$ an ABP on variable set $X^{(\Delta+1)}$ as follows.

- For each $i \in [m]$ and $j \in [2]$, let $G_{i,j}^{(\Delta+1)}$ be a copy of $G^{(\Delta)}$ on the variable set $X_{i,j}^{(\Delta+1)}$.
- Let $H_i^{(\Delta+1)}$ be the ABP obtained by composing $G_{i,1}^{(\Delta+1)}$ and $G_{i,2}^{(\Delta+1)}$ in parallel. This ABP is defined over variable set $X_i^{(\Delta+1)}$.
- Finally, let $G^{(\Delta+1)}$ be the ABP obtained by composing $H_1^{(\Delta+1)}, \ldots, H_m^{(\Delta+1)}$ in series, in that order.

From the definition of $G^{(\Delta)}$ it is easy to observe the following properties.

Proposition 5. *For any $\Delta \geq 0$, $G^{(\Delta)}$ has a unique source, say s_Δ, and a unique sink, say t_Δ. Also, each edge in $G^{(\Delta)}$ appears on some source to sink path.*

We define $P^{(\Delta)}$ to be the multilinear polynomial in $\mathbb{F}[X^{(\Delta)}]$ computed by the ABP $G^{(\Delta)}$. We can also note the following properties of the polynomial computed by $G^{(\Delta)}$.

Proposition 6 (Properties of $P^{(\Delta)}$)**.**
1) *$P^{(0)}$ is the polynomial $\sum_{i_1 \in \{1,2\}, i_2 \in \{3,4\}} x_{i_1,1}^{(0)} \cdot x_{i_1,2}^{(0)} \cdot x_{i_2,1}^{(0)} \cdot x_{i_2,2}^{(0)}$.*
2) *For each $\Delta \geq 0$, $P^{(\Delta+1)}(X^{(\Delta+1)})$ is equal to*
$$\prod_{i \in [m]} \left(P^{(\Delta)}(X_{i,1}^{(\Delta+1)}) + P^{(\Delta)}(X_{i,2}^{(\Delta+1)}) \right).$$
3) *The polynomial $P^{(0)}$ is computed by a $\Sigma\Pi$ multilinear formula of size $O(1)$. For each $\Delta \geq 1$, $P^{(\Delta)}$ can be*

computed by a syntactic multilinear $(\Pi\Sigma)^{\Delta}\Pi$ *formula of size* $O(n_{\Delta})$.

As with the variable sets, we see that for any $\Delta' \geq \Delta \geq 1$, and for each $\omega \in ([m] \times [2])^{\Delta'-\Delta}$, the ABP $G^{(\Delta')}$ contains a corresponding copy of $G^{(\Delta)}$ on the variable set $X_{\omega}^{(\Delta')}$. We denote this copy of $G^{(\Delta)}$ by $G_{\omega}^{(\Delta')}$. The ABP $G_{\omega}^{(\Delta')}$ is obtained by composing in series the ABPs $H_{(\omega,1)}^{(\Delta')}, \ldots, H_{(\omega,m)}^{(\Delta')}$ (defined on the segments $X_{(\omega,1)}^{(\Delta')}, \ldots, X_{(\omega,m)}^{(\Delta')}$ respectively) in that order.

B. Restrictions

We define a random multilinear[10] $(X^{(\Delta+1)}, Y^{(\Delta+1)}, Z^{(\Delta+1)})$-restriction $\rho^{(\Delta+1)}$ by an inductively defined sampling process.

Base case.: Let $\rho^{(0)}$ be defined as follows: $\rho^{(0)}(x_{1,1}^{(0)}) = y_1^{(0)}, \rho^{(0)}(x_{2,1}^{(0)}) = z_1^{(0)}$, $\rho^{(0)}(x_{3,1}^{(0)}) = y_2^{(0)}, \rho^{(0)}(x_{4,1}^{(0)}) = z_2^{(0)}$ (see Figure 5). That is, we set the first variable in each of the paths $\pi_1, \pi_2, \pi_3, \pi_4$ to a distinct variable in $Y^{(0)} \cup Z^{(0)}$. Also, we set $\rho^{(0)}(x_{i,2}^{(0)}) = 1$ for each $i \in [2]$. That is, we set all the remaining variables to the constant 1. (Note that $\rho^{(0)}$ is in fact a *deterministic* $(X^{(0)}, Y^{(0)}, Z^{(0)})$-restriction.) It can be checked that $\rho^{(0)}$ is multilinear.

Inductive case.: Let us now assume that we have defined the process for sampling the random multilinear $(X^{(\Delta)}, Y^{(\Delta)}, Z^{(\Delta)})$-restriction $\rho^{(\Delta)}$. We define $\rho^{(\Delta+1)}$ by the following sampling process.

For each $i \in [m]$, we sample a random multilinear $(X_i^{(\Delta+1)}, Y_i^{(\Delta+1)}, Z_i^{(\Delta+1)})$-restriction $\rho_i^{(\Delta+1)}$ independently using the general sampling process described below. We then define $\rho^{(\Delta+1)}$ as a composition of these restrictions, i.e. $\rho^{(\Delta+1)} = \rho_1^{(\Delta+1)} \circ \cdots \circ \rho_m^{(\Delta+1)}$ (as defined in Section II-A). Clearly, $\rho^{(\Delta+1)}$ is multilinear since each $\rho_i^{(\Delta+1)}$ is multilinear.

We now give a general sampling process to sample a random multilinear (X, Y, Z)-restriction where X is an $X^{(\Delta+1)}$-segment and Y and Z are the corresponding $Y^{(\Delta+1)}$- and $Z^{(\Delta+1)}$-segments respectively.

For the remainder of this section, let $\Delta' \geq \Delta + 1 \geq 1$ be arbitrary and for some $\omega \in ([m] \times [2])^{\Delta'-\Delta-1}$ and $i \in [m]$, let X denote the ith segment $X_{(\omega,i)}^{(\Delta')}$ of the $X^{(\Delta+1)}$-clone $X_{\omega}^{(\Delta')}$ and let $Y = Y_{(\omega,i)}^{(\Delta')}$, $Z = Z_{(\omega,i)}^{(\Delta')}$. For any $j \in [2]$, let X_j, Y_j, and Z_j denote the $X^{(\Delta)}$-clones $X_{(\omega,(i,j))}^{(\Delta')}, Y_{(\omega,(i,j))}^{(\Delta')}$ and $Z_{(\omega,(i,j))}^{(\Delta')}$ respectively, and let G_j denote the ABP $G_{(\omega,(i,j))}^{(\Delta')}$.

Let y, z denote the variables $y_{(\omega,i)}^{(\Delta')}$ and $z_{(\omega,i)}^{(\Delta')}$ in the sets Y and Z respectively (recall that we have $Y = Y_1 \cup Y_2 \cup \{y\}$ and $Z = Z_1 \cup Z_2 \cup \{z\}$). Let H denote the ABP $H_{(\omega,i)}^{(\Delta')}$ (recall that H is the parallel composition of G_1 and G_2).

[10]See Section II-A for the definition.

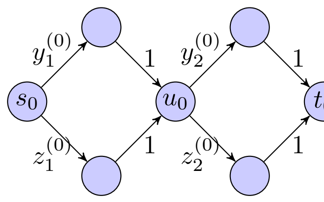


Figure 5. The effect of $\rho^{(0)}$ on $G^{(0)}$.

We now show how to sample for any such X, Y, Z a random multilinear (X, Y, Z)-restriction ρ.

Sampling Algorithm $\mathcal{A}$ for (X, Y, Z)-restriction ρ.:

E_1: Set all the variables from the set X_2 to 0. For the variables in X_1, sample a random (X_1, Y_1, Z_1)-restriction ρ_1 using the sampling procedure for $\rho^{(\Delta)}$. (Recall that X_1, Y_1, Z_1 are $X^{(\Delta)}$-, $Y^{(\Delta)}$-, and $Z^{(\Delta)}$-clones respectively.) Set variables in X_1 according to ρ_1.

E_2: This is the same as E_1 except that the roles of X_1 and X_2 are exchanged. Formally, we set all the variables from the set X_1 to 0 and apply a random (X_2, Y_2, Z_2)-restriction ρ_2, sampled using the sampling procedure for $\rho^{(\Delta)}$, to X_2 .

E_3: Choose variables x_1 and x_2 independently and uniformly at random from X_1 and X_2 respectively. The variable x_j ($j \in [2]$) labels a unique edge, say e_j, in ABP G_j; let π_{e_j} denote the lexicographically smallest source to sink path in $G_{i,j}^{(\Delta+1)}$ containing e_j. For any variable $x \in X$ which does not label an edge on either of the paths π_{e_1} or π_{e_2}, set x to the constant 0.
Set x_1 to y and x_2 to z.
For any edge e other than e_1, e_2 which lies on either π_{e_1} or π_{e_2}, set the variable labelling e to 1.

C. Properties of $\rho^{(\Delta)}$

Fix some $\Delta \geq 1$. For a given random choice of $\rho^{(\Delta)}$, we use $\tilde{Y}$ and $\tilde{Z}$ to denote $\mathrm{Img}(\rho^{(\Delta)}) \cap Y^{(\Delta)}$ and $\mathrm{Img}(\rho^{(\Delta)}) \cap Z^{(\Delta)}$ respectively. Note that these are *random* sets.

Given any $f \in \mathbb{F}[X^{(\Delta)}]$, the polynomial $f|_{\rho^{(\Delta)}}$ belongs to the set $\mathbb{F}[\tilde{Y} \cup \tilde{Z}]$. By the multilinearity of $\rho^{(\Delta)}$, if f is multilinear, then so is $f|_{\rho^{(\Delta)}}$.

Lemma 7. *With probability* 1, $\mathrm{relrk}_{(\tilde{Y},\tilde{Z})}(P^{(\Delta)}|_{\rho^{(\Delta)}}) = 1$.

Proof: We prove the lemma by induction on Δ.

The base case corresponds to $\Delta = 0$, in which case $\tilde{Y} = Y^{(0)} = \{y_1^{(0)}, y_2^{(0)}\}$ and $\tilde{Z} = Z^{(0)} = \{z_1^{(0)}, z_2^{(0)}\}$ deterministically. From the definition of $P^{(0)}$ and $\rho^{(0)}$, we know that $P^{(0)}|_{\rho^{(0)}} = (y_1^{(0)} + z_1^{(0)})(y_2^{(0)} + z_2^{(0)})$, which can easily be seen to have relative-rank 1 w.r.t. the partition $(\tilde{Y}, \tilde{Z})$.

Recall (Proposition 6) that for each $\Delta \geq 1$,

$$P^{(\Delta)}(X^{(\Delta)}) = \prod_{i\in[m]} \left(P^{(\Delta-1)}(X_{i,1}^{(\Delta)}) + P^{(\Delta-1)}(X_{i,2}^{(\Delta)})\right).$$

Let $Q_i^{(\Delta)}$ be equal to $P^{(\Delta-1)}(X_{i,1}^{(\Delta)}) + P^{(\Delta-1)}(X_{i,2}^{(\Delta)})$. The random restriction $\rho^{(\Delta)}$ is defined to be the composition $\rho_1^{(\Delta)} \circ \cdots \circ \rho_m^{(\Delta)}$ where each $\rho_i^{(\Delta)}$ is a random multilinear $(X_i^{(\Delta)}, Y_i^{(\Delta)}, Z_i^{(\Delta)})$-restriction sampled according to the algorithm $\mathcal{A}$ described in Section III-B. Thus, we have for any fixing of $\rho_i^{(\Delta)}$ $(i \in [m])$,

$$P^{(\Delta)}|_{\rho^{(\Delta)}} = \prod_{i\in[m]} Q^{(\Delta)}|_{\rho_i^{(\Delta)}}$$

and hence by Proposition 2, we have

$$\mathrm{relrk}_{(\tilde{Y},\tilde{Z})}(P^{(\Delta)}|_{\rho^{(\Delta)}}) = \prod_{i\in[m]} \mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}})$$

where $\tilde{Y}_i := \mathrm{Img}(\rho_i^{(\Delta)}) \cap Y_i^{(\Delta)}$ and $\tilde{Z}_i := \mathrm{Img}(\rho_i^{(\Delta)}) \cap Z_i^{(\Delta)}$. So it suffices to argue that each term in the above product is 1 with probability 1.

Fix an $i \in [m]$ and consider $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}})$. There are three possibilities for the sampling algorithm $\mathcal{A}$ in choosing $\rho_i^{(\Delta)}$.

Say $\mathcal{A}$ picks option E_1. Then, all variables in $X^{(\Delta)} \setminus X_{i,1}^{(\Delta)} = X_{i,2}^{(\Delta)}$ are set to 0 and $\rho_i^{(\Delta)}$ is simply a copy of $\rho^{(\Delta-1)}$ defined w.r.t. the sets $(X_{i,1}^{(\Delta)}, Y_{i,1}^{(\Delta)}, Z_{i,1}^{(\Delta)})$. Thus, by induction $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}}) = \mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(P^{(\Delta-1)}|_{\rho_i^{(\Delta)}}) = 1$. A similar reasoning works when $\mathcal{A}$ picks option E_2.

In case $\mathcal{A}$ picks E_3 then $Q_i^{(\Delta)} = (y_i^{(\Delta)} + z_i^{(\Delta)})$ and $\tilde{Y}_i = \{y_i^{(\Delta)}\}, \tilde{Z}_i = \{z_i^{(\Delta)}\}$. It is then easily checked that $\mathrm{relrk}_{(\tilde{Y}_i,\tilde{Z}_i)}(Q^{(\Delta)}|_{\rho_i^{(\Delta)}}) = 1$. This completes the induction and proves the lemma. ■

IV. The Main Result

The main result of this paper is the following.

Theorem 8. *Let $m, \Delta \in \mathbb{N}$ be growing parameters with $\Delta = m^{o(1)}$.*[11] *Assume that the polynomials $P^{(\Delta)}(X^{(\Delta)})$ are as defined in Section III-A. Then any multilinear circuit C of product-depth Δ computing $P^{(\Delta)}$ must have a size of at least $\exp(m^{\Omega(1)}) = \exp(n_\Delta^{\Omega(1/\Delta)})$.*

The following corollary is immediate from the lower bound in Theorem 8 and the formula upper bound in Proposition 6.

Corollary 9. *Assume $\Delta = \Delta(n) = o(\log n/\log\log n)$. For all large enough $n \in \mathbb{N}$, there is an explicit multilinear polynomial on n variables that has a multilinear formula of size $O(n)$ and product-depth $\Delta(n) + 1$ but no multilinear circuit of size at most $\exp(n^{\Omega(1/\Delta)})$ and product-depth at most $\Delta(n)$.*

Proof: We can find $m = m(n)$ so that that $\Delta = m^{o(1)}$ and the corresponding $n_\Delta \in [\sqrt{n}, n]$ for large enough n.

[11]Since $n_\Delta = O(m)^\Delta$, this is equivalent to requiring that $\Delta = o(\log n_\Delta/\log\log n_\Delta)$.

We now apply Proposition 6 and Theorem 8 to obtain the result. ■

We also have a similar result for depth instead of product-depth.

Corollary 10. *Assume $\Delta = \Delta(n) = o(\log n/\log\log n)$. For all large enough $n \in \mathbb{N}$, there is an explicit multilinear polynomial on n variables that has a multilinear formula of size n and depth $\Delta + 1$ but no multilinear circuit of size $\exp(n^{\Omega(1/\Delta)})$ and depth at most Δ.*

Proof: Let $\Delta' = \lfloor(\Delta+1)/2\rfloor$. Fix $m = m(n)$ so that that $\Delta = m^{o(1)}$ and the corresponding $n_{\Delta'} \in [\sqrt{n}, n/2]$ for large enough n. We define the explicit polynomial to be either $P^{(\Delta')}$ or the sum of two copies of $P^{(\Delta'-1)}$ depending on whether Δ is even or odd.

Assume that Δ is even. Then, $\Delta = 2\Delta'$. In this case, the explicit polynomial is $P^{(\Delta')}$ which has a $(\Pi\Sigma)^{\Delta'}\Pi$ formula of size $O(n_{\Delta'}) = O(n)$ by Proposition 6. Note that a $(\Pi\Sigma)^{\Delta'}\Pi$ formula is of depth $2\Delta'+1 = \Delta+1$. This gives the upper bound.

For the lower bound, we use Theorem 8. Any circuit C of size at most s and depth at most $\Delta = 2\Delta'$ can be converted to one of size at most s and product-depth at most Δ' as follows. If C contains two $\times$-gates Ψ_1 and Ψ_2 where Ψ_1 feeds into Ψ_2, we merge Ψ_1 and Ψ_2. Repeated applications of this procedure yields a circuit of depth at most $2\Delta'$ in which no input-to-output path can contain consecutive $\times$-gates. This circuit must have product-depth at most Δ'. Clearly, this process can only reduce the size of the circuit. By Theorem 8, we see that if C computes $P^{(\Delta')}$ it must have a size of at least $\exp(n_{\Delta'}^{\Omega(1/\Delta')}) = \exp(n^{\Omega(1/\Delta)})$.

Now consider the case when Δ is odd. Then $\Delta = 2\Delta'-1$. In this case, we define the explicit polynomial to be $Q = P^{(\Delta'-1)}(X_1) + P^{(\Delta'-1)}(X_2)$ where X_1 and X_2 are disjoint copies of $X^{(\Delta'-1)}$. Note that the number of variables in Q is $2n_{\Delta'-1} \le n_{\Delta'} \le n$ and by Proposition 6, Q has a $\Sigma(\Pi\Sigma)^{\Delta'-1}\Pi$ formula of size $O(n_{\Delta'-1}) = O(n)$. Note that a $\Sigma(\Pi\Sigma)^{\Delta'-1}\Pi$ formula is of depth $2(\Delta'-1)+2 = \Delta+1$. This gives the upper bound.

For the lower bound, we proceed as was done in the case where Δ is even. Like before, we can assume that the most efficient circuit C of depth Δ for Q has the property that no path in C contains consecutive $\times$-gates. Further, since Q is easily seen to be an irreducible polynomial, we can also assume that the output gate is a $+$-gate. This implies that the product-depth of C is at most $\lfloor\Delta/2\rfloor \le \Delta' - 1$. Applying Theorem 8 now yields the lower bound as in the even case. ■

Theorem 8 is proved by induction on the parameter Δ. For the purposes of induction, we need to prove a more technical statement from which we can easily infer Theorem 8. We give this technical statement below.

Recall that for a random mutlilinear (X, Y, Z)-restriction ρ, $\tilde{Y} = \mathrm{Img}(\rho) \cap Y$ and $\tilde{Z} = \mathrm{Img}(\rho) \cap Z$. Note that these are *random* sets. Now, given a multilinear polynomial $f \in \mathbb{F}[X' \cup Y' \cup Z']$ where $X' \subseteq X$ and Y', Z' are disjoint sets of variables, the polynomial $f|_\rho$ is a multilinear polynomial in $\mathbb{F}[\tilde{Y} \cup \tilde{Z} \cup Y' \cup Z']$.

We now state the following crucial theorem (Theorem 11), without proof due to the lack of space[12], and deduce Theorem 8, our main result, from Theorem 11 below.

Theorem 11. *Let $m, k, \Delta, M \in \mathbb{N}$ be growing positive integer parameters with $k = m^{0.05}$ and $M \geq m/10$. Let $\varepsilon > 0$ be such that $\varepsilon \geq 1/M^{0.25}$. Let $\Delta' \geq \Delta$ be such that $\Delta' \leq m^{0.001}$.*

Let $X^{(\Delta')}_{(\omega_1,j_1)}, \ldots, X^{(\Delta')}_{(\omega_M,j_M)}$ be arbitrary distinct $X^{(\Delta)}$-segments. Let $X = \bigcup_{i\in[M]} X^{(\Delta')}_{(\omega_i,j_i)}, Y = \bigcup_{i\in[M]} Y^{(\Delta')}_{(\omega_i,j_i)}$, and $Z = \bigcup_{i\in[M]} Z^{(\Delta')}_{(\omega_i,j_i)}$.

Assume that $X' = \bigcup_{i\in[M]} X'_i$ where for each $i \in [M]$, $X'_i \subseteq X^{(\Delta')}_{(\omega_i,j_i)}$ satisfying $|X'_i| \geq \varepsilon \cdot |X^{(\Delta')}_{(\omega_i,j_i)}|$. Let F be any $(\Sigma\Pi)^\Delta\Sigma$ syntactially multilinear variable-labelled[13] formula with $\mathrm{Vars}(F) = X' \dot\cup Y' \dot\cup Z'$ where Y' and Z' are arbitrary sets of variables that are disjoint from X'. Assume that the size of F is $s \leq \exp(k^{0.1}/\Delta^2)$.

For each $i \in [M]$, let ρ_i be an independent *multilinear random restriction obtained by using the sampling algorithm $\mathcal{A}$ described in Section III-B to sample a $(X^{(\Delta')}_{(\omega_i,j_i)}, Y^{(\Delta')}_{(\omega_i,j_i)}, Z^{(\Delta')}_{(\omega_i,j_i)})$-restriction. Let $\rho = \rho_1 \circ \cdots \circ \rho_M$ be the resulting (X, Y, Z)-random restriction. Let $\tilde{Y}' = \rho(X') \cap Y$ and $\tilde{Z}' = \rho(X') \cap Z$.*[14]

Then, we have

$$\Pr_\rho[\mathrm{relrk}_{(Y' \cup \tilde{Y}', Z' \cup \tilde{Z}')}(F|_\rho) \geq \exp\left(\frac{-k^{0.1}}{\Delta}\right)] \leq \frac{\Delta}{\exp(\frac{k^{0.1}}{\Delta})} .$$

Remark 12. *Note that the $X^{(\Delta)}$-segments that appear in the statement of Theorem 11 are $X^{(\Delta)}$-segments contained in $X^{(\Delta')}$ for some $\Delta' \geq \Delta$. This general form of the theorem is needed when we are proving lower bounds for multilinear formulas of product-depth Δ'. During the course of the inductive proof, the depth of the circuit reduces iteratively and at some intermediate point in the proof when the depth is down to Δ, we end up in the situation described in Theorem 11.*

Proof of Theorem 8 assuming Theorem 11: Let C be any multilinear circuit of product-depth at most Δ computing $P^{(\Delta)}$. Let s denote the size of C. By Lemma 4, there is a $(\Sigma\Pi)^\Delta\Sigma$ syntactic multilinear formula F of size $s' = s^{O(\Delta)}$ computing $P^{(\Delta)}$. We can assume that F does not use any variables outside $X^{(\Delta)}$ since such variables may be safely set to 0 without affecting the polynomial being computed.

Recall that $X^{(\Delta)}$ is the disjoint union of its segments $X^{(\Delta)}_1, \ldots, X^{(\Delta)}_m$. The random restriction $\rho^{(\Delta)}$ from Section III-B is defined to be $\rho^{(\Delta)} = \rho^{(\Delta)}_1 \circ \cdots \circ \rho^{(\Delta)}_m$ where each $\rho^{(\Delta)}_i$ is an independent $(X^{(\Delta)}_i, Y^{(\Delta)}_i, Z^{(\Delta)}_i)$-restriction sampled using the algorithm $\mathcal{A}$. Let $\tilde{Y}$ and $\tilde{Z}$ denote $\mathrm{Img}(\rho^{(\Delta)}) \cap (\bigcup_i Y^{(\Delta)}_i)$ and $\mathrm{Img}(\rho^{(\Delta)}) \cap (\bigcup_i Z^{(\Delta)}_i)$ respectively.

Consider the restricted polynomials $P' = P^{(\Delta)}|_\rho$ and $F' = F|_\rho$. By Lemma 7, we know that $\mathrm{relrk}_{(\tilde{Y},\tilde{Z})}(P') = 1$ with probability 1. On the other hand, applying Theorem 11 with $M = m$, $\varepsilon = 1$, $\Delta' = \Delta = m^{o(1)}$ and $Y' = Z' = \emptyset$, we get

$$\Pr_\rho[\mathrm{relrk}_{(\tilde{Y},\tilde{Z})}(F') \geq \exp(-k^{0.1}/\Delta)] \leq \Delta \exp(-k^{-0.1}/\Delta)$$

which is $o(1)$ if $s' \leq \exp(k^{0.1}/\Delta^2)$. The final inequality above follows from the fact that $\Delta \leq m^{o(1)}$ and hence $\Delta \exp(-k^{-0.1}/\Delta) = \Delta \exp(-m^{\Omega(1)}/\Delta) = o(1)$.

Since the formula F' computes P', we must therefore have $s' > \exp(k^{0.1}/\Delta^2) = \exp(m^{\Omega(1)}/\Delta^2)$ which yields $s = \exp(m^{\Omega(1)}/\Delta^3)$. Since $\Delta = m^{o(1)}$, this means that $s = \exp(m^{\Omega(1)}) = \exp((n_\Delta)^{\Omega(1/\Delta)})$. ∎

[12]See [33] for the full proof.

[13]See Section II-D for the definition of variable-labelled formulas.

[14]Note that these are *random* sets. Also note that for each fixing of ρ, the restricted formula $F|_\rho$ computes a multilinear polynomial in $\mathbb{F}[\tilde{Y}' \cup \tilde{Z}' \cup Y' \cup Z']$

References

[1] R. P. Brent, "The parallel evaluation of general arithmetic expressions," *Journal of the ACM*, vol. 21, no. 2, pp. 201–206, Apr. 1974.

[2] P. M. Spira, "Computation times of arithmetic and boolean functions in (d, r) circuits," *IEEE Transactions on Computers*, vol. 100, no. 6, pp. 552–555, 1973.

[3] J. E. Hopcroft, W. J. Paul, and L. G. Valiant, "On time versus space," *J. ACM*, vol. 24, no. 2, pp. 332–337, 1977. [Online]. Available: http://doi.acm.org/10.1145/322003.322015

[4] L. G. Valiant, S. Skyum, S. J. Berkowitz, and C. Rackoff, "Fast Parallel Computation of Polynomials Using Few Processors," *SIAM Journal of Computing*, vol. 12, no. 4, pp. 641–644, 1983.

[5] M. Agrawal and V. Vinay, "Arithmetic circuits: A chasm at depth four," in *proceedings of Foundations of Computer Science (FOCS)*, 2008, pp. 67–75.

[6] P. Koiran, "Arithmetic circuits: The chasm at depth four gets wider," *Theor. Comput. Sci.*, vol. 448, pp. 56–65, 2012.

[7] S. Tavenas, "Improved bounds for reduction to depth 4 and depth 3," *Information and Computation*, vol. 240, pp. 2–11, 2015.

[8] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, "Arithmetic circuits: A chasm at depth 3," *SIAM Journal of Computing*, vol. 45, no. 3, pp. 1064–1079, 2016. [Online]. Available: https://doi.org/10.1137/140957123

[9] ——, "Approaching the chasm at depth four," *J. ACM*, vol. 61, no. 6, pp. 33:1–33:16, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2629541

[10] H. Fournier, N. Limaye, G. Malod, and S. Srinivasan, "Lower bounds for depth 4 formulas computing iterated matrix multiplication," in *proceedings of Symposium on Theory of Computing (STOC)*, 2014, pp. 128–135. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591824

[11] N. Kayal, N. Limaye, C. Saha, and S. Srinivasan, "An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Circuits," in *proceedings of Foundations of Computer Science (FOCS)*, 2014.

[12] M. Kumar and S. Saraf, "On the power of homogeneous depth 4 arithmetic circuits," in *proceedings of Foundations of Computer Science (FOCS)*, 2014.

[13] J. Håstad, *Computational limitations of small-depth circuits*, ser. ACM doctoral dissertation award. MIT Press, 1987. [Online]. Available: https://books.google.co.in/books?id=_h0ZAQAAIAAJ

[14] B. Rossman, R. A. Servedio, and L. Tan, "An average-case depth hierarchy theorem for boolean circuits," in *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, 2015, pp. 1030–1048. [Online]. Available: https://doi.org/10.1109/FOCS.2015.67

[15] J. Håstad, "An average-case depth hierarchy theorem for higher depth," in *FOCS*. IEEE Computer Society, 2016, pp. 79–88.

[16] J. Håstad, B. Rossman, R. A. Servedio, and L. Tan, "An average-case depth hierarchy theorem for boolean circuits," *J. ACM*, vol. 64, no. 5, pp. 35:1–35:27, 2017.

[17] N. Nisan and A. Wigderson, "Lower bounds on arithmetic circuits via partial derivatives," *Computational Complexity*, vol. 6, no. 3, pp. 217–234, 1997.

[18] R. Raz and A. Yehudayoff, "Lower bounds and separations for constant depth multilinear circuits," *Computational Complexity*, vol. 18, no. 2, pp. 171–207, 2009.

[19] N. Kayal, V. Nair, and C. Saha, "Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth three circuits," in *proceedings of Symposium on Theoretical Aspects of Computer Science (STACS)*, 2016, pp. 46:1–46:15. [Online]. Available: https://doi.org/10.4230/LIPIcs.STACS.2016.46

[20] M. Kumar and R. Saptharishi, "Finer separations between shallow arithmetic circuits," in *FSTTCS*, ser. LIPIcs, vol. 65. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 38:1–38:12.

[21] R. Raz, "Separation of multilinear circuit and formula size," *Theory of Computing*, vol. 2, no. 1, pp. 121–135, 2006.

[22] ——, "Multilinear-$\mathsf{NC}^2 \neq$ multilinear-NC^1," in *proceedings of Foundations of Computer Science (FOCS)*, 2004, pp. 344–351. [Online]. Available: https://doi.org/10.1109/FOCS.2004.42

[23] R. Raz and A. Yehudayoff, "Balancing syntactically multilinear arithmetic circuits," *Computational Complexity*, vol. 17, no. 4, pp. 515–535, 2008.

[24] R. Raz, A. Shpilka, and A. Yehudayoff, "A lower bound for the size of syntactically multilinear arithmetic circuits," *SIAM Journal of Computing*, vol. 38, no. 4, pp. 1624–1647, 2008.

[25] Z. Dvir, G. Malod, S. Perifel, and A. Yehudayoff, "Separating multilinear branching programs and formulas," in *proceedings of Symposium on Theory of Computing (STOC)*, 2012, pp. 615–624. [Online]. Available: http://doi.acm.org/10.1145/2213977.2214034

[26] M. Ajtai, "σ_1^1-formulae on finite structures," *Annals of Pure and Applied Logic*, vol. 24, no. 1, pp. 1 – 48, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0168007283900386

[27] M. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Mathematical systems theory*, vol. 17, no. 1, pp. 13–27, Dec 1984. [Online]. Available: https://doi.org/10.1007/BF01744431

[28] A. C.-C. Yao, "Separating the polynomial-time hierarchy by oracles," in *Proc. 26th Annual Symposium on Foundations of Computer Science*. Piscataway, NJ, USA: IEEE Press, 1985, pp. 1–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=4479.4487

[29] X. Chen, I. C. Oliveira, R. A. Servedio, and L. Tan, "Near-optimal small-depth lower bounds for small distance connectivity," in *STOC*. ACM, 2016, pp. 612–625.

[30] S. K. Bera and A. Chakrabarti, "A depth-five lower bound for iterated matrix multiplication," in *Conference on Computational Complexity*, ser. LIPIcs, vol. 33. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, pp. 183–197.

[31] N. Kayal, C. Saha, and S. Tavenas, "On the size of homogeneous and of depth four formulas with low individual degree," in *proceedings of Symposium on Theory of Computing, STOC*, 2016, pp. 626–632. [Online]. Available: http://doi.acm.org/10.1145/2897518.2897550

[32] S. Chillara, N. Limaye, and S. Srinivasan, "Small-depth multilinear formula lower bounds for iterated matrix multiplication, with applications," in *STACS*, ser. LIPIcs, vol. 96. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 21:1–21:15.

[33] S. Chillara, C. Engels, N. Limaye, and S. Srinivasan, "A near-optimal depth-hierarchy theorem for small-depth multilinear circuits," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 25, p. 62, 2018.

[34] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, pp. 207–388, March 2010. [Online]. Available: http://dx.doi.org/10.1561/0400000039

[35] R. Saptharishi, "A survey of lower bounds in arithmetic circuit complexity," 2015, github survey. [Online]. Available: https://github.com/dasarpmar/lowerbounds-survey/releases/