On the Power of Homogeneous Algebraic Formulas

Hervé Fournier, Nutan Limaye, Srikanth Srinivasan, Sébastien Tavenas

On the Power of Homogeneous Algebraic Formulas

Hervé Fournier
herve.fournier@imj-prg.fr
Université Paris Cité, IMJ-PRG
Paris, France

Nutan Limaye
nuli@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

Srikanth Srinivasan*
srsr@di.ku.dk
University of Copenhagen
Copenhagen, Denmark

Sébastien Tavenas
sebastien.tavenas@univ-smb.fr
Univ. Savoie Mont Blanc, CNRS, LAMA
Chambéry, France

ABSTRACT

Proving explicit lower bounds on the size of algebraic formulas is a long-standing open problem in the area of algebraic complexity theory. Recent results in the area (e.g. a lower bound against constant-depth algebraic formulas due to Limaye, Srinivasan, and Tavenas (FOCS 2021)) have indicated a way forward for attacking this question: show that we can convert a general algebraic formula to a *homogeneous* algebraic formula with moderate blow-up in size, and prove strong lower bounds against the latter model.

Here, a homogeneous algebraic formula F for a polynomial P is a formula in which all subformulas compute homogeneous polynomials. In particular, if P is homogeneous of degree d, F does not contain subformulas that compute polynomials of degree greater than d.

We investigate the feasibility of the above strategy and prove a number of positive and negative results in this direction.

(1) **Lower bounds against weighted homogeneous formulas**: We show the first lower bounds against homogeneous formulas *of any depth* in the *weighted* setting. Here, each variable has a given weight and the weight of a monomial is the sum of weights of the variables in it. This result builds on a lower bound of Hrubeš and Yehudayoff (Computational Complexity 2011) against homogeneous multilinear formulas. This result is strong indication that lower bounds against homogeneous formulas are within reach.

(2) **Improved (quasi-)homogenization for formulas**: A simple folklore argument shows that any formula F for a homogeneous polynomial of degree d can be homogenized with a size blow-up of $d^{O(\log s)}$. We show that this can be improved superpolynomially over fields of characteristic 0 as long as $d = s^{o(1)}$. Such a result was previously only known when $d = (\log s)^{1+o(1)}$ (Raz (J. ACM 2013)). Further, we show how to get rid of the condition on d at the expense of getting a *quasi-homogenization* result: this means that subformulas can compute polynomials of degree up to $\mathrm{poly}(d)$.

(3) **Lower bounds for non-commutative homogenization**: A recent result of Dutta, Gesmundo, Ikenmeyer, Jindal and Lysikov (2022) implies that to homogenize algebraic formulas of any depth, it suffices to homogenize *non-commutative* algebraic formulas of depth just 3. We are able to show strong lower bounds for such homogenization, suggesting barriers for this approach.

(4) **No Girard-Newton identities for positive characteristic**: In characteristic 0, it is known how to homogenize constant-depth algebraic formulas with a size blow-up of $\exp(O(\sqrt{d}))$ using the Girard-Newton identities. Finding analogues of these identities in positive characteristic would allow us, paradoxically, to show *lower bounds* for constant-depth formulas over such fields. We rule out a strong generalization of Girard-Newton identities in the setting of positive characteristic, suggesting that a different approach is required.

CCS CONCEPTS

• **Theory of computation** → **Algebraic complexity theory**; **Circuit complexity**.

KEYWORDS

Algebraic Formulas, Formula Homogenization

ACM Reference Format:
Hervé Fournier, Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 2024. On the Power of Homogeneous Algebraic Formulas. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3618260.3649760

1 INTRODUCTION

Given a multivariate polynomial $P(x_1, \ldots, x_n)$ over some field $\mathbb{F}$, an *Algebraic formula* for P is just an algebraic expression for P involving the variables $x_1, \ldots, x_n$ and field constants, which are combined using nested additions and multiplications. The size of the formula is the number of variables and field constants in the expression. The depth of the formula is the number of times additions and multiplications are nested within each other. (See Section 2 for a formal definition of the model.)

This paper is motivated by the problem of proving size lower bounds against algebraic formulas. More formally, we would like

*The author holds a partial position at Aarhus University.

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649760

to find explicit sequences of polynomials $P(x_1, \ldots, x_n)$ of degree $d = d(n) \leq \text{poly}(n)$ such that any algebraic formula for P has size $n^{\omega(1)}$. Proving such a result would imply a lower bound for the algebraic complexity class VF. It is worth noting that this is the algebraic analogue of the Boolean complexity class NC^1 and proving lower bounds against either of these classes is a long-standing open problem in complexity theory.

Several previous results in the area address these problems, especially the setting of *Mutlilinear formulas* [3, 12, 28–32], which are formulas in which every subformula computes a multilinear polynomial.[1] While we have superpolynomial lower bounds against such formulas [30], it remains an open question [35, Open question 14] as to whether these results can be used to obtain lower bounds against general formulas.

Another class of restricted formulas that has received quite some attention is the class of *Homogeneous formulas*, which is the main focus of this work. Here, we consider polynomials $P(x_1, \ldots, x_n)$ that are homogeneous of some degree $d = d(n)$. A formula is homogeneous if each of its subformulas computes a homogeneous polynomial. In particular, each subformula computes a polynomial of degree at most d. Relaxing this definition, we say that a formula is *quasi-homogeneous* if subformulas can compute polynomials of degree up to $\text{poly}(d)$. (Formal definition in Section 2 below.)

Lower bounds for homogeneous formulas of bounded depth have been the focus of many previous results, especially in the last decade [1, 4, 7, 11, 14–17, 19, 20, 28]. Moreover, in recent work, it has been shown [1, 19], in the setting of constant depth and fields of characteristic 0, that it is possible to prove lower bounds against *unrestricted formulas* using lower bounds against homogeneous formulas.

This suggests the following high-level approach to proving lower bounds against algebraic formulas.

(1) **Homogenization**: Show that a general algebraic formula can be converted to a homogeneous algebraic formula with a small size blow-up.
(2) **Homogeneous lower bounds**: Show lower bounds against homogeneous algebraic formulas. Ideally, these would be strong enough to imply lower bounds against general algebraic formulas. However, superpolynomial lower bounds against homogeneous algebraic formulas (without depth restrictions) would already be very interesting and are as yet not known.

Results of both kinds are known in various interesting special cases.

- A result of Hyafil [9] implies as a special case that any algebraic formula of size s can be homogenized with a size blow-up of $d^{O(\log s)}$. Unfortunately, this technique does not distinguish between formulas and more general computational models such as algebraic circuits. As known techniques do not seem capable of proving lower bounds against these stronger models, we do not believe that this result will be useful for the above approach.
- Raz [31] showed how to homogenize algebraic formulas computing polynomials of small degree. More precisely, the size blowup in this result is $\text{poly}(s) \cdot \binom{d+\log s}{d}$. In particular, if $d = O(\log s)$, this is only a polynomial blow-up. This implies that proving superpolynomial *homogeneous* formula lower bounds in this 'low-degree' setting implies superpolynomial lower bounds against general formulas.
  For $d \geq (\log s)^{\Omega(1)}$, however, this is essentially the same as the previous result.
- Hrubeš and Yehudayoff [8] showed lower bounds against algebraic formulas that are homogeneous and also multilinear. A notable feature of this result is that it holds for the *Elementary Symmetric polynomials*, which are intimately connected to homogenization. The result only holds for relatively high-degree polynomials (and in particular does not hold in the low-degree setting of Raz's result above). Further, the multilinearity condition means that it is unclear how to exploit this for general formula lower bounds, as mentioned above.
  This same paper also shows that depth-3 formulas computing polynomials of degree d can be homogenized with a size blow-up of $d^{O(\log d)}$. In particular, when $d = s^{o(1)}$, this is superpolynomially better than the consequence of Hyafil's result mentioned above. An earlier result of Shpilka and Wigderson [34] shows how to *quasi*-homogenize depth-3 formulas with only polynomial blowup.[2] Both these results are over fields of characteristic 0.
- The aforementioned result of [19] showed how to homogenize constant-depth formulas over fields of characteristic 0 with a size blow-up of $\exp(O(\sqrt{d}))$, which is small in the low-degree setting. It was also shown how to prove superpolynomial lower bounds against constant-depth homogeneous algebraic formulas over *any* characteristic, when the degree is low. This implies a lower bound for constant-depth (and otherwise unrestricted) algebraic formulas in characteristic 0, but falls short of proving this result in positive characteristic.
  It should be noted that these results of [19] would work just as well if the first step was instead a *quasi*-homogenization.
- Finally, results of Kayal, Saha and Saptharishi [13] and also Amireddy, Garg, Kayal, Saha and Thankey [1] show how to prove lower bounds against homogeneous formulas of any depth, but with strong syntactic restrictions on the fan-ins of the gates [13] or the multiplicative structure of the formula [1]. Like in the multilinear case, it seems unclear whether this will lead to lower bounds against general formulas.

Depth-reduction. (Quasi-)Homogeneous algebraic formulas are also easier to analyze for other reasons. For instance, it was shown recently [5] that quasi-homogeneous formulas computing polynomials of degree d could be converted to formulas of depth $O(\log d)$ with only a polynomial blow-up. This result implies that quasi-homogenization results for general formulas also imply that we can convert them to small-depth formulas. Given that it seems easier

[1] In particular, a multilinear formula can only compute a multilinear polynomial.

[2] Both the results of [8, 34] only state their results in terms of (quasi-)homogeneous upper bounds for the *Elementary symmetric polynomials*. However, this has the more general consequence noted here.

to prove lower bounds against formulas of small depths [19], this is an important step towards proving lower bounds.

The questions we address. In this paper, we investigate the feasibility of the above high-level approach towards formula lower bounds and prove many positive and negative results regarding homogeneous algebraic formulas and the process of homogenizing general algebraic formulas. In particular, we address the following questions:

(1) Are there techniques for proving lower bounds against homogeneous algebraic formulas of any depth? Note that this is not known, even over fields of characteristic 0. In our opinion, *this is the natural next question for algebraic complexity lower bounds.*
(2) Can we convert general formulas to (quasi-)homogeneous formulas efficiently even in the high-degree setting (say $d = s^{\Omega(1)}$)? While this is true in the low-degree setting [30], it seems hard to extend recent lower bounds [1, 19] in the low-degree setting to unbounded-depth formulas [20]. Having such a result in the high-degree setting would allow us to consider high-degree polynomials, which could be an advantage in proving lower bounds. This is indeed the case in various situations [3, 18, 20, 30].
(3) Can we convert constant-depth formulas efficiently to constant-depth homogeneous formulas in the low-degree setting over fields of positive characteristic? Note that this would immediately imply a lower bound for constant-depth formulas over positive characteristic by the result of [19], which would solve an important open problem.

2 PRELIMINARIES

Basic notation. Throughout, $\mathbb{F}$ will denote a field. In some of our results, we will have to assume that $\mathbb{F}$ has characteristic 0. We will mostly work over multivariate polynomial rings such as $\mathbb{F}[x_1, \ldots, x_n]$, but some of our results are related to the *non-commutative* polynomial ring $\mathbb{F}\langle x_1, \ldots, x_n\rangle$.

Given a polynomial $P(x_1, \ldots, x_n)$, we use $[P]_d$ to denote the homogeneous component of P of degree d. Further, we extend this to a *weighted* setting, where each variable x_i is associated to some positive integer weight w_i. The weighted degree of a monomial is then the sum of the weights of the variables in the monomial (with appropriate multiplicities) and the weighted degree of a polynomial P is the maximum degree of a monomial with non-zero coefficient in P. Again, we use $[P]_d$ to denote the homogeneous component of weighted-degree d (it will be clear from context what the weights are).

2.1 Algebraic Models of Computation

Algebraic formulas. We recall the basic model of Algebraic formulas.

An algebraic formula over the multivariate polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is a rooted, directed tree with edges directed towards the root. Leaves are labelled by variables $x_1, \ldots, x_n$ or by the constant 1 and edges by non-zero field constants. Internal nodes (i.e., gates) by + and × and compute linear combinations (based on the edge weights) or products of their children. We will assume, with loss of generality, that if a node α has for child a leaf labelled by 1, then α is a +-gate and that if a +-gate α has only children labelled by 1, then α is the output of the formula.[3] A *non-commutative* algebraic formula over the multivariate polynomial ring $\mathbb{F}\langle x_1, \ldots, x_n\rangle$ is defined similarly, with the additional assumption that the children of any ×-gate are linearly ordered, and the corresponding product is computed in this order.

Unless explicitly stated, the algebraic formulas we consider have *unbounded* fan-in (i.e., a gate can have any number of inputs). The *size* of F will denote the number of leaves,[4] the *depth* of F the longest leaf-to-root path. The *product-depth* and the *sum-depth* of F are defined to be the maximum number of product gates and sum gates encountered on a leaf-to-root path, respectively. If the product-depth of a formula is Δ, then its depth is betwen Δ and $2\Delta + 1$.

Algebraic Branching Programs and Circuits. An algebraic circuit is a generalization of an algebraic formula where the underlying graph is allowed to be a directed acyclic graph. An algebraic branching program (ABP) is a special case of an algebraic circuit where each multiplication gate has at most one input of syntactic degree greater than 1.[5]

Comparison between the models. Standard results in the literature show that formulas can be converted to equivalent ABPs with polynomial blow-up in size and a similar result for ABPs holds vis-a-vis algebraic circuits. Finally, it was shown by Hyafil [9] that a circuit can be converted to a formula via a quasipolynomial blow-up. More formally,

Theorem 2.1 (Hyafil [9]). *Let P be a polynomial of degree d computed by a circuit of size s. Then, P is also computed by a formula of size $s^{O(\log d)}$. In particular, this also holds for polynomials P that have an ABP of size s.*

Homogeneity. Each gate in an algebraic formula/circuit/ABP has a *syntactic degree* defined in a natural way. Leaves labelled by the constant 1 have syntactic degree 0, leaves labelled with a variable have syntactic degree 1 (or the weight of the variable if we are in the weighted setting), ×-gates have a syntactic degree that is the sum of the syntactic degrees of their children, and +-gates have a syntactic degree that is equal to the largest of the syntactic degrees of their children. The syntactic degree of a formula is defined as the syntactic degree of its output. Notice that in a formula the syntactic degree of any gate is bounded by the syntactic degree of the formula.

A formula/circuit/ABP is *homogeneous* if each gate in the formula computes a homogeneous polynomial. Equivalently, in terms of syntactic degrees, this means that all the children of a sum gate have the same syntactic degree. In particular, this implies that the output gate computes a polynomial whose degree *equals* its syntactic degree. Weakening this criterion, we say that a formula/circuit/ABP

[3] This ensures that a formula can compute polynomials with a constant term but forbids using many arithmetic operations just to compute constants.
[4] This is within a constant factor of the number of gates, as long as each gate has fan-in at least 2 each (which is without loss of generality).
[5] ABPs are typically defined using graphs in a slightly different way (see, e.g. Definition 3.1 in [35]). However, this definition via "skew" circuits is equivalent up to polynomial blowups [24].

is *quasi-homogeneous* if the syntactic degree of the output gate is at most a polynomial function of the degree of the output polynomial.

These definitions extend naturally to the weighted setting. However, to emphasize the difference, we will call such formulas/circuits/ABPs *weighted homogeneous* or *weighted quasi-homogeneous.*

It is well-known that circuits and ABPs can be *homogenized* with a small blow-up in the following sense.

Lemma 2.2 (Folklore). *If a (weighted or unweighted) homogeneous polynomial P of degree d is computed by an algebraic circuit (resp. ABP) of size s, then it is also computed by a (weighted or unweighted) homogeneous algebraic circuit (resp. ABP) of size $s \cdot \mathrm{poly}(d)$.*

Using the above lemma and Theorem 2.1 above, we have the following folklore corollary in the unweighted setting.

Corollary 2.3 (Folklore). *Any formula F of size s computing a (unweighted) homogeneous polynomial P of degree d can be homogenized in size $d^{O(\log s)}$.*

3 SUMMARY OF OUR RESULTS

Our results can be divided into two kinds. The first kind of results are positive results for the high-level proof approach towards lower bounds against algebraic formulas that was mentioned in the introduction. Here, we show non-trivial simulations of general algebraic formulas by homogeneous algebraic formulas, implying that a strong enough lower bound against the latter, more specialized, model implies a lower bound against the former model. We also show new lower bounds against variants of homogeneous algebraic formulas, indicating that lower bounds against the homogeneous model are within reach.

The second kind of results show negative results from the point of view of homogenization. Here, we obtain new lower bounds on the power of homogeneous algebraic formulas in simulating simple polynomials that have small inhomogeneous formulas of depth 3. In other settings, we show that new ideas are required to prove the kinds of homogenization results we would like.

3.1 Lower bounds against weighted homogeneous formulas

We show superpolynomial lower bounds against weighted homogeneous formulas *of any depth.*

The polynomial for which we prove the lower bound is quite simple to define, and understanding its complexity plays an important role in other results in the paper. It is the polynomial $H_{k,\ell,d}(z_1,\ldots,z_k)$ defined as follows. Let $z_1,\ldots,z_k$ be a weighted collection of variables, where z_i has weight i. For $k,\ell \le d$, define

$$H_{k,\ell,d}(z_1,\ldots,z_k) = \left[\left(\sum_{i=1}^{k} z_i\right)^{\ell}\right]_d.$$

Theorem 3.1 (Lower bounds against weighted homogeneous formulas). *The following holds over any field. Let d be a growing parameter. There exist $k = \Theta(d/\log d)$ and $\ell = \Theta(\log d)$ such that any weighted homogeneous formula F computing $H_{k,\ell,d}$ has size $d^{\Omega(\log\log d)}$.*

This gives the first explicit lower bound result in this variant of the model of homogeneous formulas and gives indication that lower bounds against homogeneous formulas are within reach. On the other hand, we notice that $H_{k,\ell,d}$ can be computed by interpolation by an inhomogeneous depth-3 formula of size $O(k^2\ell^2)$. This indicates that the suggested approach to prove lower bounds for generic models via homogenization is not sufficient for weighted formulas and that something more is required.

3.2 Improved bounds for (quasi-)homogenization in characteristic 0

The next question we consider is to understand the blow-up required for homogenization and quasi-homogenization of formulas. Let F be a formula computing a homogeneous polynomial P of degree d. The folklore result Corollary 2.3 above shows that F can be computed by a homogeneous formula of size $d^{O(\log s)}$. Unfortunately, as noted in the introduction, this does not distinguish between the case that F is a formula and the case that F is an algebraic circuit (for which lower bounds are probably much harder). Improvements over this are known in the setting where the degree is logarithmic [31] and depth-3 formulas [8, 34] in characteristic 0, as described in the introduction.

We show that the folklore homogenization result can be superpolynomially improved for all $d = s^{o(1)}$ in characteristic 0. Furthermore, we can remove any condition on d at the expense of turning the homogenization result to a quasi-homogenization. The main technical theorem is as follows, and the following corollary gives the improved homogenization result.

Theorem 3.2 ((Quasi-)Homogenization of algebraic formulas). *The following holds over fields of characteristic 0. Let s, d, Δ be parameters. Assume that F is an algebraic formula of size s and depth Δ computing a homogeneous polynomial P of degree d. Then P is also computed by a homogeneous formula F' of size $s \cdot d^{O(\Delta+\log d)}$. Further, for any fixed $\varepsilon > 0$, P is also computed by a quasi-homogeneous formula F'' of syntactic degree at most $d^{1+\varepsilon}$ and size $s \cdot d^{O(\Delta)}$.*

The above result considerably generalizes and strengthens results of Shpilka and Wigderson [34] and Hrubeš and Yehudayoff [8] whose results yield similar quasi-homogenization (with syntactic degree $O(d^2)$) and homogenization results for depth-3 formulas.

Corollary 3.3 (Superpolynomially better homogenization and quasi-homogenization). *The following holds over fields of characteristic 0. Let s, d be parameters. Assume that F is an (arbitrary, possibly inhomogeneous) algebraic formula of size s computing a homogeneous polynomial P of degree d. If $d = s^{o(1)}$, P is also computed by a homogeneous formula F' of size $d^{o(\log s)}$. Further, irrespective of d and for any fixed $\varepsilon > 0$, P is also computed by a quasi-homogeneous formula F'' of syntactic degree at most $d^{1+\varepsilon}$ and size $d^{o(\log s)}$.*

We note that the above results are exponentially better in terms of the allowable degree parameter than Raz's result [31] though they incur a superpolynomial blow-up in the size.

A consequence of this result is the following interesting implication: if a polynomial P of degree $d = \mathrm{poly}(n)$ in n variables has no quasi-homogeneous formula of size $n^{o(\log n)}$, then P also does not have any formula of size $\mathrm{poly}(n)$. Lower bounds of this quantitative form are known in the multilinear setting [3, 29, 30]. We

now know that obtaining such bounds in the quasi-homogeneous setting would result in general formula lower bounds.

As noted in the introduction, quasi-homogenization also has consequences for depth-reduction. Indeed putting the above corollary together with the depth-reduction of [5] we get the following result. This improves the size bound of $d^{O(\log s)}$ which follows from Hyafil's theorem above.

Corollary 3.4 (Superpolynomially better depth-reduction). *The following holds over fields of characteristic* 0. *Let s, d be parameters. If a homogeneous polynomial P of degree d is computed by an (arbitrary, possibly inhomogeneous) algebraic formula F of size s, then it is also computed by a homogeneous algebraic formula F of size $d^{o(\log s)}$ and depth $O(\log d)$.*

3.3 Homogenization in the non-commutative setting

We also consider the power of formula homogenization in the *non-commutative* setting where variables are not allowed to commute with each other. Non-commutative polynomials can be thought of as polynomials where the underlying variables take values in a non-commutative algebra (such as square matrices of some dimension over the field $\mathbb{F}$). There are two motivations for considering this question.

The principal motivation goes back to homogenizing *commutative* formulas. A recent result of Dutta, Gesmundo, Ikenmeyer, Jindal and Lysikov [2] shows the existence of a 'complete' polynomial $P_{n,d}(x_1, \ldots, x_n)$ for homogeneous algebraic formula computation in the following sense: if $P_{n,d}$ (which is a homogeneous polynomial of degree $d \le n$) has a homogeneous formula of size $\mathrm{poly}(n)$, then any formula can be homogenized with polynomial blow-up. While we do not want to recall the definition of $P_{n,d}$ here, it is worth noting that this polynomial is closely related to computing a simple polynomial in matrix variables. In particular, consider the Elementary symmetric polynomial E_n^d in non-commuting variables $x_1, \ldots, x_n$ defined by

$$\mathrm{E}_n^d(x_1, \ldots, x_n) = \sum_{1 \le i_1 < i_2 < \cdots < i_d \le n} x_{i_1} x_{i_2} \cdots x_{i_d}. \tag{1}$$

It is simple to show that if E_n^d has a *non-commutative* homogeneous formula of size $\mathrm{poly}(n)$, then so does $P_{n,d}$. Further, it is a standard fact that E_n^d has a depth-3 non-commutative inhomogeneous formula of polynomial size. So, the question of homogenizing general algebraic formulas reduces to this clean question of homogenizing depth-3 non-commutative formulas.

The second motivation comes from two results of Limaye, Srinivasan and Tavenas [19, 36]. The latter result shows a strong separation between Algebraic Branching Programs (ABPs) and homogeneous algebraic formulas of small-depths in the non-commutative setting, making progress towards an old question of Nisan [27]. On the other hand, we also have separations between ABPs and *inhomogeneous* constant-depth formulas, but we then have to go through the *commutative* setting of [19], resulting in weaker bounds. If we could homogenize non-commutative formulas efficiently, then we could avoid this argument and lift the stronger results of [36] to the inhomogeneous case.

We show the following strong no-go results for non-commutative homogenization.

Theorem 3.5 (Lower bounds for non-commutative homogenization). *The following holds over any field. Let n, d, Δ be parameters.*

If $d \le n^{0.99}$, the above polynomial E_n^d, which has an inhomogeneous non-commutative algebraic formula of product-depth 1 *(and depth* 3*), is such that any homogeneous non-commutative algebraic formula of product-depth Δ computing E_n^d must have size $n^{\Omega(d^{1/\Delta}/2^{\Delta})}$.*

Further, if $d \le n^{1-2/\log\log n}$, any homogeneous non-commutative algebraic formula (irrespective of depth) for E_n^d has size $(\log n)^{\Omega(\log d)}$. It gives the lower bound $n^{\Omega(\log\log n)}$ as soon as $d = n^{\Omega(1)}$.

3.4 Girard-Newton identities in positive characteristic

Finally, we investigate possible analogues of Theorem 3.2 in the commutative setting over fields of positive characteristic.

One of the main ingredients of Theorem 3.2 (and its precedents in the works of Shpilka and Wigderson [34] and Hrubeš and Yehudayoff [8]) is the family of *Girard-Newton Identities* that allow us to express the Elementary symmetric polynomials of degree at most d in terms of Power Sum symmetric polynomials of degree at most d *in fields of characteristic* 0. Here, the Elementary symmetric polynomial is the polynomial E_n^d as defined in (1) (except that the variables now commute), and the Power sum symmetric polynomial P_n^d is the sum of the dth powers of all the variables $x_1, \ldots, x_n$. Note that the Power sum symmetric polynomials P_n^d have *support* 1, in the sense that each monomial depends on at most 1 variable. To be more formal, we introduce some notation.

Definition 3.6 (Support of a polynomial). The support-size of a polynomial $Q \in \mathbb{F}[w_1, \ldots, w_m]$ is the maximum number of distinct variables in a single monomial.

Observe that if the support-size of $Q(w_1, \ldots, w_m)$ is at most r then Q has a depth-2 formula of size at most $(md)^r$, where d denotes the degree of Q. This implies, in particular, that the Power sum symmetric polynomials trivially have small formulas of depth 2. This last fact is what makes the Girard-Newton identities useful. For example, since

$$\mathrm{E}_n^d = Q_d(\mathrm{P}_n^1, \ldots, \mathrm{P}_n^d) \tag{2}$$

for some polynomial Q_d, this immediately implies that E_n^d has a depth-4 homogeneous formula of size exponential in d but polynomial in n. In particular, for slowly growing d, this allows us to homogenize depth-3 formulas without blowing up size or depth significantly.

In positive characteristic, it is easy to see that there is no identity as in (2).[6] However, we could hope for weaker analogues, expressing the Elementary symmetric polynomials in terms of symmetric polynomials of 'small' support, i.e. polynomials where each monomial involves at most $r = O(1)$ variables, implying that the polynomial has a depth-2 formula of size $O((nd)^r) = \mathrm{poly}(n)$.

[6] This follows, for example, from the fact that the Power sum symmetric polynomials are algebraically dependent in positive characteristic, while the Elementary symmetric polynomials remain algebraically independent.

We rule out even such weak analogues of Girard-Newton identities in small positive characteristic.

Theorem 3.7 (No Girard-Newton Identities in positive characteristic). *Fix a constant prime $p > 0$. For any d that is a power of p and $n \geq d$, there is no polynomial $Q_d(w_1, \ldots, w_m)$ such that the Elementary symmetric polynomial E_n^d can be expressed as*

$$\mathrm{E}_n^d = Q_d(P_1, \ldots, P_m)$$

where $P_1, \ldots, P_m$ are symmetric polynomials of support-size $< d$.

Organization. We start with a proof overview of all the four results described above in the next section. However, due to space constraints, we only include the proofs of the first and fourth results, and point the reader to the full version for all the proofs [6].

4 PROOF OVERVIEW

4.1 Lower bound against weighted homogeneous formulas

Here we describe the proof ideas behind Theorem 3.1, which shows a superpolynomial lower bound against weighted homogeneous formulas computing the weighted homogeneous polynomial $H_{k,\ell,d}$.

Most lower bounds for strong models of algebraic computation use linear algebraic methods based on rank techniques going back to the work of Nisan [27] and Nisan and Wigderson [28]. In contrast, our proof is surprisingly simple. We use a *covering argument*, which shows a lower bound for computing *any* polynomial containing all monomials of weighted degree d, which in particular implies a lower bound for computing $H_{k,\ell,d}$.

More precisely, we show that any weighted homogeneous formula of small size can be written as a sum of a few terms, each of which is a product of many polynomials. Such 'product lemmas' offer a standard route to proving lower bounds in many different settings [8, 28, 30, 35]. In our setting, we show that each product term can only compute a small fraction of all monomials of weighted degree d. This implies the lower bound.

Such arguments are usually only useful in the monotone setting.[7] Note that our lower bounds do not assume monotonicity of any form, but we are nonetheless able to use this argument here, which we think is strong indication that homogeneous formula lower bounds are within reach. Our proof is inspired by a result of Hrubeš and Yehudayoff [8] who also use a covering argument to prove a lower bound against homogeneous *multilinear* formulas. Multilinearity is a strong condition and we know how to prove lower bounds even against *inhomogeneous* multilinear formulas [30]. Here, we remove the multilinearity condition at the expense of considering the weighted setting.

4.2 (Quasi-)Homogenization in characteristic 0

We now turn to the proof of Theorem 3.2 which holds over fields $\mathbb{F}$ of characteristic 0. As mentioned above, this result strengthens and generalizes the results of [8, 34] who prove similar results for depth-3 formulas.

[7] In the setting of monotone algebraic computation, the underlying field is $\mathbb{R}$ and all the coefficients of the polynomials that are computed by the gates of the formula/ABP/circuit are non-negative. This implies that there can be no cancellations in the underlying computation, making the models quite weak [10, 37].

Quick sketch of the depth-3 case. As they are stated, these results yield quasi-homogeneous and homogeneous formulas of size $\mathrm{poly}(n, d)$ and $\mathrm{poly}(n) \cdot d^{O(\log d)}$ respectively for a very concrete family of polynomials: the Elementary symmetric polynomial E_n^d defined above. From this very concrete result, we get a similar result for general depth-3 formulas via the following standard argument (see, e.g. [19]) which we sketch here. Consider a depth-3 $\Sigma\Pi\Sigma$ formula F. The formula F is a sum of terms, each of which is a product of linear polynomials. After some manipulation, one can show that without loss of generality, each such term T has the form

$$T = \alpha \cdot \prod_{i=1}^{n} (1 + \ell_i)$$

where $\alpha \in \mathbb{F}$ and each ℓ_i is a homogeneous linear polynomial. Note that the homogeneous degree-d component of T is given by $\mathrm{E}_n^d(\ell_1, \ldots, \ell_n)$. Thus, if we have efficient (as obtained in [8, 34]) (quasi-)homogeneous formulas for E_n^d, we can use these to get similarly efficient (quasi-)homogeneous formulas for the degree-d component of T and by extension for the polynomial computed by F (assuming that it is homogeneous of degree d).

To prove the above results for E_n^d, the two works [8, 34] use a common idea: the *Girard-Newton identities* that allow us to write the Elementary symmetric polynomials in terms of the Power sum symmetric polynomials P_n^d defined above. The latter family of polynomials is homogeneous and sparse. Hence, they trivially have depth-2 homogeneous formulas of small size. So, it suffices to analyze the complexity of the 'composing' weighted homogeneous polynomial GN such that

$$\mathrm{E}_n^d = \mathrm{GN}^d(\mathrm{P}_1^d, \ldots, \mathrm{P}_n^d).$$

By designing small weighted (quasi-)homogeneous formulas for GN^d, we get (quasi-)homogeneous formulas for E_n^d.

Extending to higher depths. We extend these results to higher depths and using this, we are able to get a superpolynomial improvement over previously known (quasi-)homogenization results. This result builds on a series of elementary but non-trivial steps, resulting in a somewhat intricate argument. We sketch the high-level ideas here.

The depth-3 strategy is tied to the fact that computing the family of Elementary symmetric polynomials (quasi-)homogeneously captures the complexity of (quasi-)homogenizing depth-3 formulas. Unfortunately, this is not true for higher depths. However, it was observed in [19] that an analogous role at higher depths is played by a *weighted* generalization of these polynomials that we denote by WE_n^d. Informally, the underlying variable set is divided into n buckets, each containing one variable each of weights $1, \ldots, d$. The polynomial WE_n^d is the sum of all monomials of weighted degree d that contain at most one variable per bucket. Setting variables of weight greater than 1 to zero in WE_n^d returns E_n^d.

Previous results [23, 26, 33] have shown how to generalize the Girard-Newton identities to express WE_n^d in terms of an analogous weighted generalization of the Power sum symmetric polynomials that we denote WP_n^d. In fact, the composing polynomial here again

is the same polynomial GN^d from the Girard-Newton identities.[8] Having these identities is the first crucial step in our proof.

The next step is to understand the complexity of computing the weighted homogeneous polynomials GN^d and WP^d_n. We have some understanding of the former from the works [8, 34]. However, the power sums turn out to be quite a bit more complicated in the weighted setting. Nevertheless, we are able to show that the complexity of both polynomials are closely related to the complexity of the polynomial $H_{k,\ell,d}$ defined above (and a more general variant). This is not obvious as the two families of polynomials are not similar at all at first sight.

The final step is to construct weighted (quasi-)homogeneous formulas for the polynomial $H_{k,\ell,d}$ and compose these formulas together to (quasi-)homogenize a depth-Δ formula F. It is not straightforward to do this. First, we show how to construct formulas for $H_{k,\ell,d}(z_1,\ldots,z_k)$ where the number of copies of z_i is inversely related to its weight i. At a high-level, this is useful for the following reason. Let us imagine that we have a formula using gates that compute the polynomial $H_{k,\ell,d}(z_1,\ldots,z_k)$. Replacing this gate by the formulas constructed above results in a large blow-up for inputs of small weighted degree (which intuitively have small formulas since they have small weighted degree) but only a small blow-up for inputs of large weighted degree. We use this high-level idea to show how to compose these formulas together to (quasi-)homogenize a depth-Δ formula F efficiently.

4.3 Lower bounds for non-commutative homogenization

The proof of Theorem 3.5 uses a lower bound technique introduced in [36] (building on [19, 28]) where it was used to prove lower bounds for non-commutative homogeneous formulas computing a different polynomial.[9] This technique is suited to proving lower bounds for *set-multilinear* polynomials which are special kinds of homogeneous polynomials. More precisely, the variables in a set-multilinear polynomial of degree d are partitioned into d sets $\mathcal{X}_1,\ldots,\mathcal{X}_d$, each monomial contains exactly one variable per set.

While the polynomial E^d_n is *not* set-multilinear, in the non-commutative setting, the complexity of this polynomial is equivalent to the set-multilinear polynomial essentially obtained by 'set-mutilinearizing' each monomial of E^d_n. We call this polynomial $\mathrm{BE}^d_n(\mathcal{Y}_1,\ldots,\mathcal{Y}_d)$. It is easy to show that if E^d_n has a small homogeneous non-commutative formula, then so does BE^d_n. Since the latter polynomial is set-multilinear, it is amenable to techniques introduced in [36].

This technique is the partial derivative method of [28] combined with a restriction argument. Fix a set-multilinear polynomial $H(\mathcal{X}_1,\ldots,\mathcal{X}_d)$. We divide the underlying variable sets into two families, say $\{\mathcal{X}_{i_1},\ldots,\mathcal{X}_{i_r}\}$ and $\{\mathcal{X}_{j_1},\ldots,\mathcal{X}_{j_{d-r}}\}$, and analyze the rank of the 'partial derivative' matrix M with rows and columns labelled by set-multilinear monomials in the two sets of variables. The coefficient of the (m_1,m_2)-th entry of M is the coefficient in H of the monomial m that has exactly the variables of m_1 and m_2 (in the right order).

It was shown in [36] that for any polynomial with a small non-commutative homogeneous formula, the matrix M has small rank, as long as the sizes of the variable sets $|\mathcal{X}_1|,\ldots,|\mathcal{X}_d|$ are sufficiently 'different'. In the setting of the hard polynomial $P(\mathcal{Y}_1,\ldots,\mathcal{Y}_d)$ from [36], it is possible to find a 'projection' from P to a set-multilinear polynomial $H(\mathcal{X}_1,\ldots,\mathcal{X}_d)$ where $|\mathcal{X}_1|,\ldots,|\mathcal{X}_d|$ are different (in the sense required) while maintaining the property that the partial derivative matrix M is the identity matrix, and hence full rank. We thus get a lower bound from H, which implies a lower bound for P.

Here, we instead have to work with the polynomial $\mathrm{BE}^d_n(\mathcal{Y}_1,\ldots,\mathcal{Y}_d)$, which does not have the rich combinatorial structure of the polynomial P from [36], making the argument for that polynomial inapplicable.[10] Nevertheless, we show that for essentially any choice of $|\mathcal{X}_1|,\ldots,|\mathcal{X}_d|$,[11] there is a projection from $\mathrm{BE}^d_n(\mathcal{Y}_1,\ldots,\mathcal{Y}_d)$ to a set-multilinear $H(\mathcal{X}_1,\ldots,\mathcal{X}_d)$ whose partial derivative matrix is upper-triangular with non-zero entries along the diagonal. This is an involved combinatorial argument that we postpone to the full version. The end result is that the polynomial H has a full-rank partial derivative matrix, implying a lower bound for computing H. Since H is a projection of BE^d_n, we obtain the same lower bound for BE^d_n as well.

4.4 No Girard-Newton identities in positive characteristic

The proof of this theorem is based on a more general functional lower bound. We show in fact that there is no function $f:\mathbb{F}^m\to\mathbb{F}$ such that

$$\mathrm{E}^d_n = f(P_1,\ldots,P_m) \tag{3}$$

where the above equality is an equality of functions mapping Boolean inputs (i.e. inputs in $\{0,1\}^n$) to $\mathbb{F}$.

The proof uses a theorem of Lucas (see Theorem 6.1 below), which has also found many applications in Boolean complexity. Lucas' theorem gives a nice functional interpretation to the Elementary symmetric polynomials on Boolean inputs. More precisely, if $d=p^k$, then the evaluation of the polynomial E^d_n on input $a\in\{0,1\}^n$ is the $(k+1)$th least significant digit of the Hamming weight w of a. More generally, for a degree parameter D that is not a power of p, $\mathrm{E}^D_n(a)$ is a function of the $\lceil\log_p(D+1)\rceil$ least significant digits of w.

Looking at (3), since $d=p^k$, we thus see that the left hand side is functionally the $(k+1)$th least significant digit of the Hamming weight w of the input a.

On the right hand side, each of the polynomials $P_1,\ldots,P_m$ are symmetric polynomials of support-size less than d. However, as functions on Boolean inputs, they are functional equivalent to *multilinear* symmetric polynomials of support-size less than d, which are simply linear combinations of Elementary symmetric polynomials of degree less than d. Again, by Lucas' theorem, we see that the right hand side depends functionally only on the k least significant digits of w.

[8] It is not hard to see that this must be the case as the power-sum polynomials $\mathrm{P}^d_1,\ldots,\mathrm{P}^d_n$ are algebraically independent.

[9] The 'Iterated Matrix Multiplication' polynomial $\mathrm{IMM}_{n,d}$ which is the top left entry of a product of d $n\times n$ generic matrices.

[10] The crucial fact about P used in [36] is that it is *complete* for the class of polynomials computed by small Algebraic Branching Programs. It is unclear if this is true for the polynomial BE^d_n we consider here.

[11] Slightly more precisely, we only consider $|\mathcal{X}_1|,\ldots,|\mathcal{X}_d|$ where each $|\mathcal{X}_i|$ is a power of 2 and the underlying partial derivative matrix is square.

Thus, we cannot have a functional equivalence between the two sides.

5 LOWER BOUND AGAINST WEIGHTED HOMOGENEOUS FORMULAS

In this section, we prove the lower bound against weighted homogeneous formulas (Theorem 3.1). Throughout this section, the set $Z = \{z_1, \ldots, z_k\}$ will denote a weighted set of variables where z_i has weight i. As defined also above, we define the weighted homogeneous polynomial $H_{k,\ell,d}$ as follows.

$$H_{k,\ell,d} = \left[\left(\sum_{i=1}^{k} z_i\right)^{\ell}\right]_d.$$

We will first prove a *product lemma* for weighted homogeneous formulas. The product lemma is very similar to one for homogeneous formulas [8].

Lemma 5.1 (Product Lemma for Weighted Homogeneous Formulas). *Let $P(Z)$ be a weighted homogeneous polynomial of weighted degree $d \geq 1$ such that $P(Z)$ is computed by a weighted homogeneous formula of size s. Then we can write*

$$P(Z) = \sum_{i=1}^{s} \prod_{j=1}^{t} g_{i,j}(Z),$$

where $t = \lceil \log_3(d/k) \rceil$ and $g_{i,j}$s are weighted homogeneous polynomials of weighted degree at least one.

Proof. (Proof of Lemma 5.1)

The proof is similar to the proof of Hrubeš and Yehudayoff of a similar lemma for homogeneous formulas [8]. The proof proceeds by induction on s and d.

The base cases: If $d \leq 3k$, then the product lemma is trivially true, as we can get $t = 1$ by simply defining $g_{1,1} = P(Z)$. Suppose $s = 1$, then it means that $d \leq k$ and the statement holds again by the previous argument.

Now, let us assume that $s > 1$ and $d > 3k$. Let F be the formula computing $P(Z)$. Without blowing up the size of the formula, we may assume that each gate of F has fan-in at most 2.

For any node u in the formula F, let F_u be the formula rooted at u. Let $f_u(Z)$ be the polynomial computed by F_u. Let s_u denote the size of F_u. Let $F_{u=0}$ be the formula obtained by substituting $u = 0$ in F and let s'_u be the size of $F_{u=0}$. Let $h_u(Z)$ be the polynomial computed by $F_{u=0}$. Notice that $s \geq s_u + s'_u$ and that $s_u, s'_u < s$.

Given a formula F for $P(Z)$, there exists a node u in the formula such that the weighted degree of the polynomial f_u is at least $d/3$ and at most $2d/3$. It is easy to see that we can express $P(Z)$ in terms of $f_u(Z)$ and $h_u(Z)$. Specifically, $P(Z) = g_0(Z) \cdot f_u(Z) + h_u(Z)$, for some non-constant homogeneous polynomial $g_0(Z)$.

We apply the induction hypothesis to $h_u(Z)$ and $f_u(Z)$ to obtain the following expressions.

$$h_u(Z) = \sum_{i=1}^{s'_u} \prod_{j=1}^{t} h_{i,j}(Z),$$

where $t = \lceil \log_3(d/k) \rceil$ and the $h_{i,j}$s are weighted homogeneous polynomials. Similarly, using the fact that $\deg(f_u) \geq d/3$, we see that

$$f_u(Z) = \sum_{i=1}^{s_u} \prod_{j=1}^{t'} f_{i,j}(Z),$$

where $t' \geq \lceil \log_3(d/3k) \rceil = t - 1$ and the $f_{i,j}$s are weighted homogeneous polynomials.

Therefore, overall we get

$$P(Z) = g_0(Z) \cdot \sum_{i=1}^{s_u} \prod_{j=1}^{t-1} f_{i,j}(Z) + \sum_{i=1}^{s'_u} \prod_{j=1}^{t} h_{i,j}(Z)$$

By distributivity of multiplication and using the fact that $s \geq s_u + s'_u$, we get the claimed expression for $P(Z)$. □

From now, let $\ell = 2\lfloor \log(d) \rfloor$ and $k = 2\lfloor d/\ell \rfloor + 1$ (with d large enough). Our aim is to show that any weighted homogeneous formula F computing $H_{k,\ell,d}$ has size $d^{\Omega(\log \log d)}$ (it is not hard to see that this bound is tight).

We will prove Theorem 3.1 using Lemma 5.1.

Proof. (Proof of Theorem 3.1) Let $H_{k,\ell,d}(Z)$ be computed by a weighted homogeneous formula of size s. Then by Lemma 5.1 we can write

$$H(Z) = \sum_{i=1}^{s} \prod_{j=1}^{t} g_{i,j}(Z)$$

with $t = \lceil \log_3(d/k) \rceil$.

Fix a specific product term $T = g_1 \cdot g_2 \ldots \cdot g_t$. We say that a monomial is *covered* by such a product term if the monomial appears in T after T is simplified as a sum of monomials. To prove the lower bound, we will show that any such product term can only cover a few monomials of $H_{k,\ell,d}(Z)$. This will show that we need s to be large to cover all the monomials of the polynomial. We will do this by using a probabilistic argument.

Let $i_1, i_2, \ldots, i_\ell$ be chosen randomly from $[k]$. The distribution is given by the following random experiment.

Random experiment to generate $i_1, i_2, \ldots, i_\ell$. For every $j \in [\ell]$, let $Y_{j,1}, Y_{j,2}, \ldots, Y_{j,k-1}$ be independent Bernoulli random variables that take values 0, 1 with probability 1/2 each. Let $Y_j = \sum_{p=1}^{k-1} Y_{j,p}$ and let $i_j = Y_j + 1$. Note that, $i_j \in [k]$ and $\mathbb{E}[Y_j] = (k-1)/2$.

Here is a simple property about the random variable Y_j, which will be useful later.

Lemma 5.2. *Let $Y_{j,1}, \ldots, Y_{j,k-1}$ and Y_j be as defined above and let $r \in [k-1]$. Then,*

$$\Pr_{Y_{j,1},\ldots,Y_{j,k-1}}[Y_j = r] \leq 1/\sqrt{k-1}.$$

Proof. As Y_j is distributed as per the binomial distribution, it is easy to see that

$$\Pr_{Y_{j,1},\ldots,Y_{j,k-1}}[Y_j = r] = \frac{\binom{k-1}{r}}{2^{k-1}}.$$

Here, the numerator is maximised when $r = (k-1)/2$ and for this value of r, the ratio is upper bounded by $1/\sqrt{k-1}$. □

Let I denote the set of these indices $\{i_1, \ldots, i_\ell\}$ and let $\mathcal{M}_I$ denote the monomial $z_{i_1} z_{i_2} \cdots z_{i_\ell}$. Conditioned on the event that the weighted degree of $\mathcal{M}_I$ is exactly d, the monomial appears in the polynomial $H_{k,\ell,d}$. On the other hand, conditioned on this event, we will show that the probability that the product term T covers $\mathcal{M}_I$ is upper bounded by $1/d^{\Omega(\log\log d)}$. This will imply the lower bound.

Let $g_1, g_2, \ldots, g_t$ be polynomials of positive weighted degrees $d_1, d_2, \ldots, d_t$, respectively. If T covers $\mathcal{M}_I$ then there exists a partition of I into t parts, say $\pi = (I_1, I_2, \ldots, I_t)$, such that $\mathrm{wt}(I_j) = d_j$ for $j \in [t]$, where $\mathrm{wt}(S)$ for a set S is the sum of the elements of that set.

We will now bound the probability of T covering $\mathcal{M}_I$ for a randomly chosen I. Let $\mathcal{E}_I$ be the event that there exists a partition $\pi_I = (I_1, I_2, \ldots, I_t)$ of I such that $\mathrm{wt}(I_i) = d_i$. In order to bound the probability that T covers $\mathcal{M}_I$, it suffices to bound the following probability.

$$\Pr_I[\mathcal{E}_I \mid \mathrm{wt}(I) = d]$$

We will do that as follows.

$$\begin{aligned}
&\Pr_I[\mathcal{E}_I \mid \mathrm{wt}(I) = d] \\
&= \Pr_I[\exists \pi_I = (I_1, I_2, \ldots, I_t) : \forall j \in [t], \mathrm{wt}(I_j) = d_j \mid \mathrm{wt}(I) = d] \\
&\le t^\ell \cdot \Pr_I[\forall j \in [t], \mathrm{wt}(I_j) = d_j \mid \mathrm{wt}(I) = d] \\
&= t^\ell \cdot \frac{\Pr_I[\forall j \in [t], \mathrm{wt}(I_j) = d_j \text{ AND } \mathrm{wt}(I) = d]}{\Pr_I[\mathrm{wt}(I) = d]} \\
&\le t^\ell \cdot O(\sqrt{d}) \cdot \Pr_I[\forall j \in [t], \mathrm{wt}(I_j) = d_j \text{ AND } \mathrm{wt}(I) = d] \qquad (4) \\
&= t^\ell \cdot O(\sqrt{d}) \cdot \Pr_I[\forall j \in [t], \mathrm{wt}(I_j) = d_j] \\
&= t^\ell \cdot O(\sqrt{d}) \cdot \prod_{j \in [t]} \Pr_I[\mathrm{wt}(I_j) = d_j] \\
&\le t^\ell \cdot O(\sqrt{d}) \cdot \left(\frac{1}{\sqrt{k-1}}\right)^t.
\end{aligned}$$

The first inequality is by applying the union bound. Here, t^ℓ is an upper bound on the total number of partitions. The inequality (4) above uses Lemma 5.3 below. The final inequality follows by observing that for any $j \in [t]$, $\Pr_I[\mathrm{wt}(I_j) = d_j] \le 1/\sqrt{k-1}$ and that the events are independent for different j. To see that $\Pr_I[\mathrm{wt}(I_j) = d_j] \le 1/\sqrt{k-1}$ for every j, observe that if all the elements of the partition are fixed, but the last one, and the sum is say $d_i - r$ for some r, then the probability that the final element equals r is upper bounded by $1/\sqrt{k-1}$ by Observation 5.2. Therefore, the overall probability is upper bounded by this quantity as well.

Lemma 5.3. *For the choice of parameter ℓ and for the random experiment defined above*

$$\Pr[wt(I) = d] = \Omega\left(\frac{1}{\sqrt{d}}\right).$$

Proof. Note that $\mathrm{wt}(I) = \sum_{j=1}^{\ell} i_j = \sum_{j=1}^{\ell}(Y_j + 1) = \left(\sum_{j=1}^{\ell} Y_j\right) + \ell$. We have $\ell(k-1)$ random variables. Note that from our choice of parameters, $\ell(k-3)/2 \le d - \ell < \ell(k-1)/2 \le d$. So we want to estimate what is the probability that $d - \ell$ of these random variables are set to 1 (getting k and ℓ as integers as we did, implies $d - \ell$ is not exactly half of the random variables and we need to be precise enough so that the approximation does not become too large).

Using estimate of Lemma 7, Chapter 10 in [22], we know that

$$\begin{aligned}
\binom{\ell(k-1)}{d-\ell} &\ge \binom{\ell(k-1)}{\ell(k-3)/2} \\
&> \sqrt{\frac{\ell(k-1)}{2\ell^2(k-3)(k+1)}}\, 2^{\ell(k-1)H((k-3)/(2k-2))}
\end{aligned}$$

where H is the binary entropy function:

$$\begin{aligned}
&H\left(\frac{k-3}{2k-2}\right) \\
&= -\frac{k-3}{2(k-1)}\log_2\left(\frac{k-3}{2(k-1)}\right) - \frac{k+1}{2(k-1)}\log_2\left(\frac{k+1}{2(k-1)}\right) \\
&\ge 1 - \frac{k-3}{2(k-1)}\log_2\left(1 - \frac{2}{k-1}\right) - \frac{k+1}{2(k-1)}\log_2\left(1 + \frac{2}{k-1}\right) \\
&\ge 1 - O(1/k^2).
\end{aligned}$$

Consequently, the probability that $\mathrm{wt}(I)$ equals d is bounded by below by

$$\begin{aligned}
\binom{\ell(k-1)}{d-\ell}/2^{\ell(k-1)} &> \sqrt{\frac{1}{2\ell(k-1)}}\, 2^{\ell(k-1)(H(\frac{k-3}{2k-2})-1)} \\
&\ge \frac{1}{\sqrt{4d}} 2^{-O(\ell/k)} \ge \Omega\left(\frac{1}{\sqrt{d}}\right).
\end{aligned}$$

□

Now, by using the values of k, ℓ, t the probability that the term T covers $\mathcal{M}_I$ is upper bounded by

$$\begin{aligned}
&t^\ell \cdot O(\sqrt{d}) \cdot \frac{1}{\sqrt{(k-1)^t}} \\
&= \exp\left(\ell \log t + \frac{1}{2}\log d - \frac{1}{2} t \log(k-1) + O(1)\right) \\
&\le \exp\left(-\frac{1}{2}\log d \log\log d + O(\log d \log\log\log d)\right) \\
&= d^{-\Omega(\log\log d)}.
\end{aligned}$$

□

6 NO GIRARD-NEWTON IDENTITIES IN POSITIVE CHARACTERISTIC

The proof is a consequence of Lucas' theorem (see, e.g. [25]), which is a standard result in combinatorial number theory. We recall this result below. Throughout this section, fix a constant prime p and let $\mathbb{F}$ be any field of characteristic p.

Theorem 6.1 (Lucas' theorem). *Let p be any prime and $a, b \in \mathbb{N}$. Let $a_1, \ldots, a_\ell \in \{0, \ldots, p-1\}$ and $b_1, \ldots, b_\ell \in \{0, \ldots, p-1\}$ be the digits in the p-ary expansion of a and b, i.e., $a = \sum_{j \in [\ell]} a_j p^{j-1}$ and $b = \sum_{j \in [\ell]} b_j p^{j-1}$. Then, we have*

$$\binom{a}{b} \equiv \prod_{i \le \ell} \binom{a_i}{b_i} \pmod{p}$$

where $\binom{a_i}{b_i}$ is defined to be 0 if $a_i < b_i$.

This has the following well-known corollary (see, e.g. [21, Proposition 1] for a similar statement when $p = 2$).

Corollary 6.2. *Let $d = p^k$ and $n \geq d$. Then, for any function $f : \mathbb{F}^{d-1} \to \mathbb{F}$, there is an $a \in \{0,1\}^n$ such that*

$$\mathrm{E}_n^d(a) \neq f(\mathrm{E}_n^1(a), \ldots, \mathrm{E}_n^{d-1}(a)).$$

Proof. On any input $a \in \{0,1\}^n$ of Hamming weight w, we note that $\mathrm{E}_n^d(a)$ is in the base field $\mathbb{F}_p$ and takes the value $\binom{w}{d} \pmod p$. Since $d = p^k$, by Lucas' theorem (Theorem 6.1), this is the $(k+1)$th least significant digit of w written in base p.

On the other hand, again by Theorem 6.1, each of $\mathrm{E}_n^1(a), \ldots, \mathrm{E}_n^{d-1}(a)$ depend on the k least significant digits of w.

Consider inputs $a^{(0)}$ and $a^{(1)}$ of weights $w_0 = 0$ and $w_1 = p^k$ respectively (such an $a^{(1)}$ exists as $n \geq p^k$). The two Hamming weights have the same k least significant digits but the kth digit is different. Thus, for $a = a^{(0)}$ or $a = a^{(1)}$ we have the statement of the corollary. □

We now prove the main result of this section.

Proof of Theorem 3.7. Assume that $d = p^k$ and $n \geq d$. For the sake of contradiction, assume that

$$\mathrm{E}_n^d = Q_d(P_1, \ldots, P_m) \tag{5}$$

where $P_1, \ldots, P_m$ are symmetric polynomials of support-size at most $d-1$. We consider the above as an equality of *functions* on Boolean inputs $a \in \{0,1\}^n$. On Boolean inputs, we also have the simple functional equality $x_i^2 = x_i$. This implies that the function computed by any symmetric polynomial P_i of support-size at most $d-1$ is also computed by a symmetric *multilinear* polynomial $\tilde{P}_i$ of degree at most $d-1$.

Note that any multilinear symmetric polynomial of degree at most $d-1$ is a linear combination of elementary symmetric polynomials of degree at most $d-1$. This shows that from (5) we get the functional equality

$$\mathrm{E}_n^d = f(\mathrm{E}_1^d, \ldots, \mathrm{E}_n^{d-1}).$$

However, Corollary 6.2 implies that such a functional inequality cannot hold. This proves the theorem. □

ACKNOWLEDGEMENTS

The authors would like to thank Guillaume Malod for many helpful discussions. Part of this work was done while NL and SS were visiting the Simons Institute for the Theory of Computing, UC Berkeley for the Meta-Complexity program. NL was also supported by funding from the Independent Research Fund Denmark (grant agreement No. 10.46540/3103-00116B) and the Basic Algorithms Research Copenhagen(BARC), supported by VILLUM Foundation Grant 16582. SS is also grateful for a research visit sponsored by the Guest researchers faculty program at Université Paris Cité in summer 2022, where the work was initiated. ST is supported by ANR project - VONBICA - ANR-22-CE48-0007.

REFERENCES

[1] Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. 2023. Low-Depth Arithmetic Circuit Lower Bounds: Bypassing Set-Multilinearization. In *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany (LIPIcs, Vol. 261)*, Kousha Etessami, Uriel Feige, and Gabriele Puppis (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 12:1–12:20. https://doi.org/10.4230/LIPIcs.ICALP.2023.12

[2] Pranjal Dutta, Fulvio Gesmundo, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. 2023. De-bordering and Geometric Complexity Theory for Waring rank and related models. arXiv:2211.07055 [cs.CC]

[3] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. 2012. Separating Multilinear Branching Programs and Formulas. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, Howard J. Karloff and Toniann Pitassi (Eds.). ACM, 615–624. https://doi.org/10.1145/2213977.2214034

[4] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. 2014. Lower Bounds for Depth 4 Formulas Computing Iterated Matrix Multiplication. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 128–135.

[5] Hervé Fournier, Nutan Limaye, Guillaume Malod, Srikanth Srinivasan, and Sébastien Tavenas. 2023. Towards Optimal Depth-Reductions for Algebraic Formulas. In *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK (LIPIcs, Vol. 264)*, Amnon Ta-Shma (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 28:1–28:19. https://doi.org/10.4230/LIPIcs.CCC.2023.28

[6] Hervé Fournier, Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 2023. On the Power of Homogeneous Algebraic Formulas. *Electron. Colloquium Comput. Complex.* TR23-191 (2023). ECCC:TR23-191 https://eccc.weizmann.ac.il/report/2023/191

[7] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2014. Approaching the Chasm at Depth Four. *J. ACM* 61, 6 (2014), 33:1–33:16. https://doi.org/10.1145/2629541

[8] Pavel Hrubes and Amir Yehudayoff. 2011. Homogeneous Formulas and Symmetric Polynomials. *Comput. Complex.* 20, 3 (2011), 559–578. https://doi.org/10.1007/s00037-011-0007-3

[9] Laurent Hyafil. 1979. On the Parallel Evaluation of Multivariate Polynomials. *SIAM J. Comput.* 8, 2 (1979), 120–123. https://doi.org/10.1137/0208010

[10] Mark Jerrum and Marc Snir. 1982. Some Exact Complexity Results for Straight-Line Computations Over Semirings. *Journal of the ACM (JACM)* 29, 3 (1982), 874–897.

[11] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. 2017. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Formulas. *SIAM J. Comput.* 46, 1 (2017), 307–335.

[12] Neeraj Kayal and Chandan Saha. 2017. Multi-k-ic Depth Three Circuit Lower Bound. *Theory Comput. Syst.* 61, 4 (2017), 1237–1251. https://doi.org/10.1007/s00224-016-9742-9

[13] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. 2014. A Super-polynomial Lower Bound for Regular Arithmetic Formulas. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, David B. Shmoys (Ed.). ACM, 146–153. https://doi.org/10.1145/2591796.2591847

[14] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. 2016. On the Size of Homogeneous and of Depth Four Formulas with Low Individual Degree. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, Daniel Wichs and Yishay Mansour (Eds.). ACM, 626–632. https://doi.org/10.1145/2897518.2897550

[15] Mrinal Kumar and Ramprasad Saptharishi. 2017. An Exponential Lower Bound for Homogeneous Depth-5 Circuits over Finite Fields. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia (LIPIcs, Vol. 79)*, Ryan O'Donnell (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 31:1–31:30. https://doi.org/10.4230/LIPIcs.CCC.2017.31

[16] Mrinal Kumar and Shubhangi Saraf. 2015. The Limits of Depth Reduction for Arithmetic Formulas: It's All About the Top Fan-In. *SIAM J. Comput.* 44, 6 (2015), 1601–1625. https://doi.org/10.1137/140999220

[17] Mrinal Kumar and Shubhangi Saraf. 2017. On the Power of Homogeneous Depth 4 Arithmetic Circuits. *SIAM J. Comput.* 46, 1 (2017), 336–387. https://doi.org/10.1137/140999335

[18] Deepanshu Kush and Shubhangi Saraf. 2023. Near-Optimal Set-Multilinear Formula Lower Bounds. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference* (Warwick, United Kingdom) *(CCC '23)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, Article 15, 33 pages. https://doi.org/10.4230/LIPIcs.CCC.2023.15

[19] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 2021. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 804–814. https://doi.org/10.1109/FOCS52979.2021.00083

[20] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 2022. On the Partial Derivative Method Applied to Lopsided Set-Multilinear Polynomials. In *Proceedings of the 37th Computational Complexity Conference* (Philadelphia, Pennsylvania) *(CCC '22)*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, DEU, Article 32, 23 pages. https://doi.org/10.4230/LIPIcs.CCC.2022.32

[21] Chi-Jen Lu. 2001. An exact characterization of symmetric functions in $\mathrm{qAC}^0[2]$. *Theor. Comput. Sci.* 261, 2 (2001), 297–303. https://doi.org/10.1016/S0304-3975(00)00145-6

[22] Florence Jessie MacWilliams and Neil James Alexander Sloane. 1977. *The theory of error-correcting codes*. Vol. 16. Elsevier.

[23] Meena Mahajan and V Vinay. 1999. Determinant: Old Algorithms, New Insights. *SIAM journal on Discrete Mathematics* 12, 4 (1999), 474–490.
[24] Guillaume Malod and Natacha Portier. 2008. Characterizing Valiant's Algebraic Complexity Classes. *J. Complex.* 24, 1 (2008), 16–38. https://doi.org/10.1016/j.jco.2006.09.006
[25] Romeo Mestrovic. 2014. Lucas' theorem: its generalizations, extensions and applications (1878–2014). arXiv:1409.3820 [math.NT]
[26] Sajal Kumar Mukherjee and Sudip Bera. 2019. Combinatorial Proofs of the Newton–Girard and Chapman-Costas-Santos Identities. *Discrete Mathematics* 342, 6 (2019), 1577–1580.
[27] Noam Nisan. 1991. Lower Bounds for Non-Commutative Computation (Extended Abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, Cris Koutsougeras and Jeffrey Scott Vitter (Eds.). ACM, 410–418. https://doi.org/10.1145/103418.103462
[28] Noam Nisan and Avi Wigderson. 1997. Lower Bounds on Arithmetic Circuits Via Partial Derivatives. *Comput. Complex.* 6, 3 (1997), 217–234. https://doi.org/10.1007/BF01294256
[29] Ran Raz. 2006. Separation of Multilinear Circuit and Formula Size. *Theory Comput.* 2, 6 (2006), 121–135. https://doi.org/10.4086/toc.2006.v002a006
[30] Ran Raz. 2009. Multi-linear Formulas for Permanent and Determinant are of Super-polynomial Size. *J. ACM* 56, 2 (2009), 8:1–8:17. https://doi.org/10.1145/1502793.1502797
[31] Ran Raz. 2013. Tensor-Rank and Lower Bounds for Arithmetic Formulas. *J. ACM* 60, 6 (2013), 40:1–40:15. https://doi.org/10.1145/2535928
[32] Ran Raz and Amir Yehudayoff. 2008. Balancing Syntactically Multilinear Arithmetic Circuits. *Comput. Complex.* 17, 4 (2008), 515–535. https://doi.org/10.1007/s00037-008-0254-0
[33] Paul A Samuelson. 1942. A Method of Determining Explicitly the Coefficients of the Characteristic Equation. *The Annals of Mathematical Statistics* 13, 4 (1942), 424–429.
[34] Amir Shpilka and Avi Wigderson. 2001. Depth-3 Arithmetic Circuits Over Fields of Characteristic Zero. *Comput. Complex.* 10, 1 (2001), 1–27. https://doi.org/10.1007/PL00001609
[35] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic Circuits: A Survey of Recent Results and Open Questions. *Found. Trends Theor. Comput. Sci.* 5, 3-4 (2010), 207–388. https://doi.org/10.1561/0400000039
[36] Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. 2022. Set-multilinear and Non-commutative Formula Lower Bounds for Iterated Matrix Multiplication. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, Stefano Leonardi and Anupam Gupta (Eds.). ACM, 416–425. https://doi.org/10.1145/3519935.3520044
[37] Leslie G. Valiant. 1980. Negation can be Exponentially Powerful. *Theor. Comput. Sci.* 12 (1980), 303–314. https://doi.org/10.1016/0304-3975(80)90060-2

Received 13-NOV-2023; accepted 2024-02-11