Electronic Colloquium on Computational Complexity, Report No. 191 (2023)

On the Power of Homogeneous Algebraic Formulas

Hervé Fournier[1], Nutan Limaye *[2], Srikanth Srinivasan [3], and Sébastien Tavenas †[4]

[1]Université Paris Cité, IMJ-PRG
Email: herve.fournier@imj-prg.fr
[2]ITU Copenhagen,
Email: nuli@itu.dk
[3]University of Copenhagen ‡
Email: srinivasan.srikanth@gmail.com
[4]Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA
Email: sebastien.tavenas@univ-smb.fr

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 is 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 against 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 author is a member of Basic Algorithms Research Copenhagen(BARC), supported by VILLUM Foundation Grant 16582.
†The author is supported by ANR project - VONBICA - ANR-22-CE48-0007
‡The author holds a partial position at Aarhus University.

ISSN 1433-8092

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.

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 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* [NW97, Raz09, Raz06, RY08, Raz13, DMPY12, KS17a], which are formulas in which every subformula computes a multilinear polynomial.[1] While we have superpolynomial lower bounds against such formulas [Raz09], it remains an open question [SY10, 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 [NW97, GKKS14, KS15, FLMS13, KLSS17, KS17c, KST16, KS17b, LST21a, LST22, AGK+23]. Moreover, in recent work, it has been shown [LST21a, AGK+23], 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 [Hya79] 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 [Raz13] 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

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

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 [HY11] 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 [SW01] 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 [LST21a] 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 [LST21a] would work just as well if the first step was instead a *quasi*-homogenization.

- Finally, results of Kayal, Saha and Saptharishi [KSS14] and Amireddy, Garg, Kayal, Saha and Thankey [AGK+23] show how to prove lower bounds against homogeneous formulas of any depth, but with strong syntactic restrictions on the fan-ins of the gates [KSS14] or the multiplicative structure of the formula [AGK+23]. 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 [FLM+23] 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 to prove lower bounds against formulas of small depths [LST21a], 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 [Raz09], it seems hard to extend recent lower bounds [LST21a, AGK+23] in the low-degree setting to unbounded-depth formulas [LST22]. 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 [Raz09, DMPY12, LST22, KS23].

[2] Both the results of [HY11, SW01] 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.

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 [LST21a], 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 $\times$ 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 $\times$-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 [Hya79] that a circuit can be converted to a formula via a quasipolynomial blow-up. More formally,

Theorem 1 (Hyafil [Hya79])**.** *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.*

[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 [SY10]). However, this definition via "skew" circuits is equivalent up to polynomial blowups [MP08].

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), $\times$-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 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 (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 1 above, we have the following folklore corollary in the unweighted setting.

Corollary 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 for 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 $k, \ell \leq d$, define

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

Theorem 4 (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 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 [Raz13] and depth-3 formulas [SW01, HY11] 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 5 ((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 [SW01] and Hrubeš and Yehudayoff [HY11] whose results yield similar quasi-homogenization (with syntactic degree $O(d^2)$) and homogenization results for depth-3 formulas.

Corollary 6 (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 [Raz13] 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 [Raz09, Raz06, DMPY12]. 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 [FLM+23] we get the following result. This improves the size bound of $d^{O(\log s)}$ which follows from Hyafil's theorem above.

Corollary 7 (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 [DGI+23] 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 $d \leq n$) 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

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

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 [LST21a, TLS22]. 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 [Nis91]. 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 [LST21a], resulting in weaker bounds. If we could homogenize non-commutative formulas efficiently, then we could avoid this argument and lift the stronger results of [TLS22] to the inhomogeneous case.

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

Theorem 8 (Lower bounds for non-commutative homogenization)**.** *The following holds over any field. Let n, d, Δ be parameters.*

If $d \leq n^{0.99}$, 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 $d \leq n^{1-2/\log\log n}$, 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 5 in the commutative setting over fields of positive characteristic.

One of the main ingredients of Theorem 5 (and its precedents in the works of Shpilka and Wigderson [SW01] and Hrubeš and Yehudayoff [HY11]) 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 9 (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 $O((nd)^r) = \mathrm{poly}(n)$.

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

Theorem 10 (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 $< d$.

4 Proof Overview

4.1 Lower bound against weighted homogeneous formulas

Here we describe the proof ideas behind Theorem 4, 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 [Nis91] and Nisan and Wigderson [NW97]. 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 [NW97, Raz09, SY10, HY11]. 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 [HY11] 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 [Raz09]. 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 5 which holds over fields $\mathbb{F}$ of characteristic 0. As mentioned above, this result strengthens and generalizes the results of [SW01, HY11] who prove similar results for depth-3 formulas.

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. [LST21a]) which we sketch here. Consider a depth-3 $\Sigma\Pi\Sigma$ formula F. The formula F is a sum of terms, each of

[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.

[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 [Val79, JS82].

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 [SW01, HY11]) (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 [SW01, HY11] 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 [LST21a] that an analogous role at higher depths is played by a *weighted* generalization of these polynomials that we denote by WE_n^d. We define these polynomials formally in Section 6.1 below, but 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 [Sam42, MV99, MB19] 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 (these are harder to define and we postpone the definition to Section 6.1 below). 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_n^d. We have some understanding of the former from the works [SW01, HY11]. 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.

[8] It is not hard to see that this must be the case as the power-sum polynomials $\mathrm{P}_1^d, \ldots, \mathrm{P}_n^d$ are algebraically independent.

4.3 Lower bounds for non-commutative homogenization

The proof of Theorem 8 uses a lower bound technique introduced in [TLS22] (building on [NW97, LST21a]) 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_n^d 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_n^d. We call this polynomial $\mathrm{BE}_n^d(\mathcal{Y}_1, \ldots, \mathcal{Y}_d)$ and it is defined formally in Section 7 below. It is easy to show that if E_n^d has a small homogeneous non-commutative formula, then so does BE_n^d. Since the latter polynomial is set-multilinear, it is amenable to techniques introduced in [TLS22].

This technique is the partial derivative method of [NW97] 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 [TLS22] 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 [TLS22], 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 $\mathrm{BE}_n^d(\mathcal{Y}_1, \ldots, \mathcal{Y}_d)$, which does not have the rich combinatorial structure of the polynomial P from [TLS22], 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 $\mathrm{BE}_n^d(\mathcal{Y}_1, \ldots, \mathcal{Y}_d)$ 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 proof in Section 7 below. 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_n^d, we obtain the same lower bound for BE_n^d 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

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

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 32 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_n^d 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, $\mathrm{E}_n^D(a)$ 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*

[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 [TLS22] 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_n^d 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.

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.

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 4). 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 [HY11].

Lemma 11 (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 11)

The proof is similar to the proof of Hrubeš and Yehudayoff of a similar lemma for homogeneous formulas [HY11]. 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 $d^{\Omega(\log \log d)}$ (this bound is tight for this choice of ℓ by Lemma 18).

We will prove Theorem 4 using Lemma 11.

Proof. (Proof of Theorem 4) Let $H_{k,\ell,d}(Z)$ be computed by a weighted homogeneous formula of size s. Then by Lemma 11 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.

Observation 12. *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.

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

We will do that as follows.

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

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 13 below. The final inequality follows by observing that for any $j \in [t]$, $\Pr_I[\mathrm{wt}(I_j) = d_j] \leq 1/\sqrt{k-1}$ and that the events are independent for different j. To see that $\Pr_I[\mathrm{wt}(I_j) = d_j] \leq 1/\sqrt{k-1}$ 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 12. Therefore, the overall probability is upper bounded by this quantity as well.

Lemma 13. *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, $\ell(k-3)/2 \leq d - \ell < \ell(k-1)/2 \leq d$. 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 [MS77], we know that

$$\binom{\ell(k-1)}{d-\ell} \geq \binom{\ell(k-1)}{\ell(k-3)/2} > \sqrt{\frac{\ell(k-1)}{2\ell^2(k-3)(k+1)}} 2^{\ell(k-1)H((k-3)/(2k-2))}$$

where H is the binary entropy function:

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

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

$$\binom{\ell(k-1)}{d-\ell} / 2^{\ell(k-1)} > \sqrt{\frac{1}{2\ell(k-1)}} 2^{\ell(k-1)(H(\frac{k-3}{2k-2})-1)} \geq \frac{1}{\sqrt{4d}} 2^{-O(\ell/k)} \geq \Omega\left(\frac{1}{\sqrt{d}}\right). \qquad \square$$

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

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

6 (Quasi-)Homogenization in characteristic 0

Our main result in this section is Theorem 5.

Throughout this section, the field $\mathbb{F}$ will be assumed to be of characteristic 0. We will start with some preparatory notation and lemmas, and then prove Theorem 5 and its consequences in Section 6.2.

6.1 Preparatory work

We start with a straightforward structural lemma about formulas (proof omitted).

Lemma 14 (Syntactic degree under composition)**.** *Let $P(x_1, \ldots, x_r)$ be a weighted homogeneous polynomial of weighted degree d where x_i has weight j_i. Further, for each $i \in [r]$, let $Q_i(y_1, \ldots, y_t)$ be a set of weighted homogeneous polynomial of weighted degree j_i. Let $R(y_1, \ldots, y_t) = P(Q_1, \ldots, Q_r)$ be the composed weighted homogeneous polynomial of weighted degree d. Then we have the following:*

1. *If F is a weighted homogeneous formula for P and for each $i \in [r]$, G_i is a weighted homogeneous formula for Q_i, then the composed formula $F(G_1, \ldots, G_r)$ is a weighted homogeneous formula for R.*
2. *If F is a formula for P where the output gate has syntactic degree at most $d \cdot A$ and for each $i \in [r]$, G_i is a formula for Q_i of syntactic degree at most $j_i \cdot B$, then the the composed formula $F(G_1, \ldots, G_r)$ computes R and has syntactic degree at most $d \cdot (AB)$.*

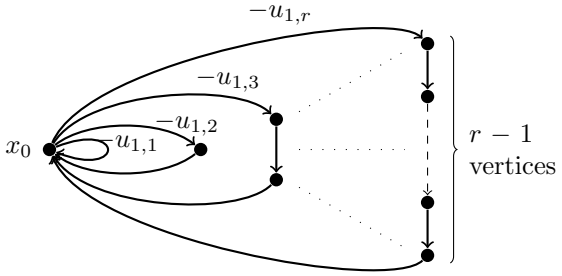
Important families of Weighted Homogeneous polynomials. We now introduce some important families of weighted homogeneous polynomials.

- Let $U = \{u_{i,j} \mid i \in [m], j \in [r]\}$ be a set of weighted variables where $u_{i,j}$ has weight j for each $i \in [m]$ and $j \in [r]$.
  - The Weighted Elementary symmetric polynomial $\mathrm{WE}^d_{m,r}(u_{i,j} : i \in [m], j \in [r])$ is the sum of all monomials of weight exactly d containing at most one variable $u_{i,j}$ for each $i \in [m]$. The polynomial $\mathrm{WE}^0_{m,r}$ is the constant polynomial equal to one.

    If we set $u_{i,j} = 0$ for each $j > 1$, the polynomial $\mathrm{WE}^d_{m,r}$ restricts to the standard Elementary symmetric polynomial E^d_m in the remaining variables $u_{i,1}$ for $i \in [m]$.
  - The Weighted Power sum symmetric polynomial $\mathrm{WP}^d_{m,r}(u_{i,j} : i \in [m], j \in [r])$ is defined to be the sum, over $i \in [m]$, of $\mathrm{WP}^d_{1,r}(u_{i,j} : j \in [r])$.

    Defining the latter polynomial requires a little bit of work. Fix $i = 1$ without loss of generality. We define a graph G which is made up of r distinct *edge-disjoint* cycles $C_1, \ldots, C_r$ of lengths $1, 2, \ldots, r$ respectively (note that a cycle of length 1 is just a self-loop on a single vertex). The cycles are also vertex disjoint except for a single vertex x_0 that lies on all of them. We label the edges of the graph as follows: the first edge of each C_j after x_0 has label $-u_{1,j}$ and all the other edges of C_j have label 1. The graph G is illustrated in Figure 1.

Figure 1: Graph G



A *walk* of G is a sequence of vertices $W = (w_0, \ldots, w_l)$ such that for any $0 \le i \le l-1$, there is an edge in G from w_i to w_{i+1}. The walk is said *closed* if $w_0 = w_l$. Notice that by definition

a closed walk is rooted (in w_0). Given a walk W in G, we define the label of W — denoted $\mathrm{lab}(W)$ — to be the product of the labels of the edges that occur in W. Note that this label is a monomial.

Finally, $\mathrm{WP}^d_{1,r}(u_{1,j} : j \in [r])$ is defined to be the sum of the labels of all the (rooted) closed walks W of length exactly d in G.

It can be verified that $\mathrm{WP}^d_{m,r}$ is a weighted homogeneous polynomial of weighted degree d. As above, it can also be verified that setting all $u_{1,j}$ to 0 for $j > 1$ in the polynomial $\mathrm{WP}^d_{m,r}(u_{i,j} : i \in [m], j \in [r])$ restricts it (up to the sign $(-1)^d$) to the standard Power sum symmetric polynomial $\mathrm{P}^d_m(u_{i,1} : i \in [m])$.

- Let $Z = \{z_1, \ldots, z_k\}$ be a set of weighted variables with z_i having weight i. For parameters k, ℓ, d with $k, \ell \leq d$, define

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

  More generally, for parameters $p, q, \ell \leq d$, define

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

We now state some results about the above families of polynomials.

Reducing WE^d to WP^d via Girard-Newton Identities. The first is a weighted analogue of the Girard-Newton Identities linking the $\mathrm{WE}^d_{m,r}$ and $\mathrm{WP}^d_{m,r}$ polynomials. The following was shown in [LST21b] (ECCC version) using a result of Bera and Mukherjee [MB19].

Theorem 15 (Weighted Girard-Newton Identities)**.** *For any d, m, r with $d \leq mr$, we have the following identity linking the Weighted Elementary and Power sum symmetric polynomials.*

$$\mathrm{WE}^d_{m,r} = \frac{1}{d} \sum_{k=1}^{d} (-1)^{k-1} \mathrm{WE}^{d-k}_{m,r} \cdot \mathrm{WP}^k_{m,r},$$

where all the polynomials are defined over the weighted variable set U defined above. Applying this identity recursively, we see that for any d there is a polynomial $\mathrm{GN}^d(y_1, \ldots, y_d)$ such that

$$\mathrm{WE}^d_{m,r} = \mathrm{GN}^d(\mathrm{WP}^1_{m,r}, \ldots, \mathrm{WP}^d_{m,r}). \tag{5}$$

Moreover, it is verified that if y_i is assigned weight i, then GN^d is weighted homogeneous of weighted degree d.

Reducing GN^d to $H_{k,\ell,d}$. We now connect the Girard-Newton polynomials GN^d and the Weighted Power sum symmetric polynomials to the polynomial $H_{k,\ell,d}$ defined above. Along with Lemma 17, this will allow us to construct formulas for the Weighted Elementary symmetric polynomial by constructing formulas for $H_{k,\ell,d}$.

Lemma 16. *For any d, there exist constants $\alpha_1, \ldots, \alpha_d, \beta_1, \ldots, \beta_d \in \mathbb{Q} \subseteq \mathbb{F}$ such that*

$$\mathrm{GN}^d(y_1, \ldots, y_d) = \sum_{\ell=1}^{d} \beta_\ell \cdot H_{d,\ell,d}(\alpha_1 y_1, \ldots, \alpha_d y_d).$$

Proof. Note that the polynomial GN^d above is independent of the parameter r and hence to recover GN^d, we can consider (5) for any r we want. We will take $r = 1$ (i.e. the case of the standard Girard-Newton identities). In this case, the polynomial $\mathrm{WE}^d_{m,r}$ is just the Elementary symmetric polynomial

$\mathrm{E}^d_m(u_1,\ldots,u_m)$ (we use u_i to denote the variable $u_{i,1}$ for notational simplicity) and similarly $\mathrm{WP}^d_{m,r} = (-1)^d\mathrm{P}^d_m(u_1,\ldots,u_m)$.

Using the standard Girard-Newton identities, it follows that (see, e.g. [Mac15, Chapter 2] and also Shpilka and Wigderson [SW01, Proof of Theorem 5.3])

$$\mathrm{E}^d_m(u_1,\ldots,u_m) = \left[\exp\left(\sum_{i=1}^{d}\alpha_i \mathrm{P}^i_m(u_1,\ldots,u_m)\right)\right]_d$$

where $\exp(v)$ denotes the standard exponential power series $\sum_{\ell=0}^{\infty} v^\ell/\ell!$ and $\alpha_1,\ldots,\alpha_m$ are suitable constants from $\mathbb{F}$. Expanding out the definition of the power series, we get the identity

$$\begin{aligned}\mathrm{E}^d_m(u_1,\ldots,u_m) &= \sum_{\ell=1}^{d}\beta_\ell\left[\left(\sum_{i=1}^{d}\alpha_i\mathrm{P}^i_m(u_1,\ldots,u_m)\right)^\ell\right]_d \\ &= \sum_{\ell=1}^{d}\beta_\ell H_{d,\ell,d}(\alpha_1\mathrm{P}^1_m(u_1,\ldots,u_m),\ldots,\alpha_d\mathrm{P}^d_m(u_1,\ldots,u_m))\end{aligned}$$

where β_ℓ denotes $1/\ell!$ and the second equality follows from the definition of the polynomials $H_{k,\ell,d}$ above.

We have thus shown that

$$\mathrm{E}^d_m(u_1,\ldots,u_m) = G(\mathrm{P}^1_m(u_1,\ldots,u_m),\ldots,\mathrm{P}^d_m(u_1,\ldots,u_m))$$

where $G(y_1,\ldots,y_d) = \sum_{\ell=1}^{d}\beta_\ell\cdot H_{d,\ell,d}(\alpha_1 y_1,\ldots,\alpha_d y_d)$. Since the Power sum symmetric polynomials are algebraically independent in any field of characteristic 0 (see, e.g. (2.12) in [Mac98]), it follows that $G(y_1,\ldots,y_d) = \mathrm{GN}^d(-y_1,y_2,-y_3,\ldots,(-1)^d y_d)$, proving the lemma. □

Reducing WP **to** $H_{k,\ell,d}$**.** In the unweighted case, the Power sum symmetric polynomials have efficient representations as sums of monomials and hence trivially have small depth-2 homogeneous formulas. However, the weighted case is quite different. As in the case of the Girard-Newton polynomials, we construct weighted (quasi-)homogeneous formulas for these polynomials by reducing to computing $H_{k,\ell,d}$.

Lemma 17. *For any $d,m,r\geq 1$, we have the following identity.*

$$\mathrm{WP}^d_{m,r}(u_{i,j} : i\in[m], j\in[r]) = \sum_{i=1}^{m}\sum_{j=1}^{\min\{r,d\}} v_{i,j}\cdot\sum_{\ell=0}^{d-1} H_{r,\ell,d-j}(u_{i,1},\ldots,u_{i,r})$$

where $v_{i,j} := -ju_{i,j}$.

Proof. Since $\mathrm{WP}^d_{m,r}(u_{i,j} : i\in[m], j\in[r]) = \sum_{i=1}^{m}\mathrm{WP}^d_{1,r}(u_{i,j} : j\in[r])$, it suffices to prove the lemma in the case that $m=1$.

Let G be the graph used in the definition of $\mathrm{WP}^d_{1,r}$. Note that any closed walk W of G traverses the edges of the cycles $C_1,\ldots,C_p$ with $p\leq\min(r,d)$ in some order (and possibly with repetition). More precisely, if the root of the cycle is different from x_0 (which is the unique vertex of G that lies on all the cycles), the cycle which is started at the end of the walk and finished at the beginning of the walk is considered as the last cycle. Hence, given W, there exists x, ℓ, $(j_1,\ldots,j_\ell)$ such that the walk W begins in x and traverses the ℓ cycles $C_{j_1},\ldots,C_{j_\ell}$ in this order. Notice that the vertex x belongs to the last cycle C_ℓ.

In fact the data of ℓ, $C_{j_1},\ldots,C_{j_\ell}$ and of the root $x\in C_{j_\ell}$ is sufficient to retrieve the walk W.

Then, by summing over the number of cycles and over the length j_ℓ of the last cycle C_{j_ℓ} we get

$$\begin{aligned}\mathrm{WP}^d_{1,r}(u_{1,j} : j\in[r]) &= \sum_{W\text{closed walk}}\mathrm{lab}(W) = \sum_{\substack{\ell,(C_1,\ldots,C_{j_\ell}),x\\ \text{with } x\in C_{j_\ell}}}\mathrm{lab}(W_{\ell,(C_{j_1},\ldots,C_{j_\ell}),x}) \\ &= \sum_{\ell=1}^{d}\sum_{j_\ell=1}^{\min(r,d)}\sum_{x\in C_{j_\ell}}(-u_{1,j_\ell})H_{r,\ell-1,d-j_\ell} \\ &= \sum_{\ell=0}^{d-1}\sum_{j=1}^{\min(r,d)}(-u_{1,j})j\cdot H_{r,\ell,d-j}.\end{aligned}$$

□

Formulas for $H_{k,\ell,d}$. To wrap up this subsection, we provide various formula constructions for $H_{k,\ell,d}$. This outlines how we will construct formulas for the Weighted Elementary symmetric polynomials, which will allow us to prove Theorem 5.

In the first two lemmas below, we will construct formulas for the more general family of polynomials $H_{p,q,\ell,d}$ defined above. We then show how to construct formulas for $H_{k,\ell,d}$ where we carefully control the number of occurrences of z_i as a function of i.

The first such construction is homogeneous, and follows the standard template for converting Algebraic Branching Programs to formulas.

Lemma 18 (Weighted homogeneous formulas for $H_{p,q,\ell,d}$)**.** *For any parameters p, q, ℓ, d, the polynomial $H_{p,q,\ell,d}$ has a weighted homogeneous formula of size $d^{O(\log \ell)}$.*

Proof. We employ the following recursion for $H_{p,q,\ell,d}$ that is easily verified

$$H_{p,q,\ell,d} = \sum_{j=1}^{d-1} H_{p,q,\lfloor \ell/2 \rfloor, j} \cdot H_{p,q,\lceil \ell/2 \rceil, d-j}.$$

Repeatedly applying this recursion yields a weighted homogeneous formula for $H_{p,q,\ell,d}$ of size $d^{O(\log \ell)}$. □

The second formula construction is inhomogeneous and follows the interpolation idea of Ben-Or (see, e.g. [SW01]). If p, q are not too far from each other, then this construction is only mildly inhomogeneous.

Lemma 19 (Inhomogeneous formulas for $H_{p,q,\ell,d}$)**.** *For any parameters $p, q, \ell \leq d$, the polynomial $H_{p,q,\ell,d}$ has a formula of size* $\mathrm{poly}(d)$ *and syntactic degree at most $d \cdot (q/p)$.*

Proof. We assume that $\ell \leq d/p$, since otherwise the polynomial $H_{p,q,\ell,d}$ is identically zero and hence the lemma holds trivially.

We introduce a new variable v and note that $H_{p,q,\ell,d}$ is precisely the coefficient of v^d in the following expression

$$\left(\sum_{i=p}^{q} v^i z_i \right)^{\ell}. \tag{6}$$

Treating the above expression as a univariate polynomial in v of degree $s = \ell \cdot q$, we note that $H_{p,q,\ell,d}$ can be interpolated by evaluating this polynomial at $s+1$ distinct values of v, say $\gamma_0, \ldots, \gamma_s$ and then taking a suitable linear combination. This gives us an expression for $H_{p,q,\ell,d}$ as follows.

$$H_{p,q,\ell,d} = \sum_{j=0}^{s} \xi_j \cdot \left(\sum_{i=p}^{q} (\gamma_j)^i z_i \right)^{\ell}$$

for suitable field constants $\xi_0, \ldots, \xi_s$. This yields a formula of size $\mathrm{poly}(d)$ where each summand has syntactic degree $q \cdot \ell \leq d \cdot (q/p)$ since we know that $\ell \leq d/p$. □

The final claim provides a pair of formula constructions for $H_{k,\ell,d}(z_1, \ldots, z_d)$, where we do not bound the size of the formula directly but rather the number of occurrences of each variable z_i in terms of i. As in the rest of the article, when the base of the logarithm is not specified, it is the base-2 logarithm.

Proposition 20. *For any $d \geq 1$ and any k, ℓ we have the following.*

1. *The polynomial $H_{k,\ell,d}(z_1, \ldots, z_k)$ has a weighted homogeneous formula such that the number of occurrences of each variable z_i is at most $d^{O(\log(d/i))}$ (where the $O(\cdot)$ hides absolute constants).*
2. *For any integer $t \geq 2$, $H_{k,\ell,d}(z_1, \ldots, z_k)$ has a formula of syntactic degree at most $d \cdot t$ such that the number of occurrences of each variable z_i is at most $d^{O(\log_t(d/i))+O(1)}$ (where the $O(\cdot)$ hides absolute constants).*

Proof. Both formula constructions exploit the recursive structure of the polynomial $H_{k,\ell,d}$ in a similar way. So we prove them together. We assume that $k, \ell \leq d$ since otherwise the polynomial $H_{k,\ell,d}$ is identically zero.

For any $t \geq 2$, we partition the positive integers into intervals $I_j = [t^{j-1}, t^j) \cap \mathbb{Z}$. Define $j(k) = \lfloor \log_t(k) \rfloor + 1$ to be the unique j such that $k \in I_j$. For item 1 of the proposition, we will take $t = 2$ but we need to consider the case of any integer $t \geq 2$ in item 2.

We prove the following statements by induction on $j(k)$. Here C denotes an absolute constant[12].

- When $t = 2$, the polynomial $H_{k,\ell,d}(z_1, \ldots, z_k)$ has a weighted homogeneous formula such that the number of occurrences of each variable z_i is at most $d^{C \log(d/i) + C(j(k) - j(i))}$.
- For any integer $t \geq 2$, $H_{k,\ell,d}(z_1, \ldots, z_k)$ has a formula of syntactic degree at most $d \cdot t$ such that the number of occurrences of each variable z_i is at most $d^{C(j(k)-j(i))+C}$.

Note that the above statements imply the proposition, since $j(k) - j(i) = O(\log_t(k/i)) = O(\log_t(d/i))$ for any $1 \leq i \leq k \leq d$.

The base case of the induction is when $j(k) = 1$, meaning that $k < t$. In this case, the two items follow directly from setting $p = 1$ and $q = k$ in Lemmas 18 and 19 respectively, since the formula construction in this case have the required homogeneity properties. Moreover, the number of occurrences of each z_i is bounded by the formula size, which is $d^{O(\log d)}$ for the first item (recall that $t = 2$ in this case) and $d^{O(1)}$ for the second item (for general $t \geq 2$).

Now assume that $j(k) > 1$. We use j to denote $j(k)$ for notational simplicity. Define the polynomials A and B by

$$A = \sum_{i=1}^{t^{j-1}-1} z_i \,, \qquad B = \sum_{i=t^{j-1}}^{k} z_i.$$

Note that we have

$$\begin{aligned}
H_{k,\ell,d} &= \left[(A+B)^\ell\right]_d \\
&= \sum_{\ell'=0}^{d/t^{j-1}} \binom{\ell}{\ell'} \left[A^{\ell-\ell'} B^{\ell'}\right]_d \\
&= \sum_{\ell'=0}^{d/t^{j-1}} \binom{\ell}{\ell'} \cdot \left(\sum_{d'=0}^{d} \left[A^{\ell-\ell'}\right]_{d-d'} \left[B^{\ell'}\right]_{d'} \right) \\
&= \sum_{\ell'=0}^{d/t^{j-1}} \sum_{d'=0}^{d} \binom{\ell}{\ell'} H_{t^{j-1}-1,\ell-\ell',d-d'}(z_1, \ldots, z_{t^{j-1}-1}) \cdot H_{t^{j-1},k,\ell',d'}(z_{t^{j-1}}, \ldots, z_k). \qquad (7)
\end{aligned}$$

We use the binomial theorem for the first identity, but restrict the maximum value of ℓ' to d/t^{j-1}, since $B^{\ell'}$ contains no monomials of degree at most d for larger ℓ'. Finally, in the expression from (7), we use the induction hypothesis to construct formulas for $H_{t^{j-1}-1,\ell-\ell',d-d'}$ and either Lemma 18 or Lemma 19 to construct formulas for $H_{t^{j-1},k,\ell',d'}$ to prove Item 1 or Item 2 respectively.

To analyze the above construction, consider any $i \leq k$. If $j(i) = j(k)$, then the variable z_i only occurs in the subformulas for $H_{t^{j-1},k,\ell',d'}$. Since $i < t^j$ it gives the bound $\ell' \leq d/t^{j-1} < td/i$. The number of occurrences of z_i in $H_{k,\ell,d}$ is at most $O(d^2)$ times the size of the formula for $H_{t^{j-1},k,\ell',d'}$. This size can be bounded by either $d^{O(\log(\ell'))} = d^{O(\log(d/i))}$ since for the first item $t = 2$ (if we use Lemma 18) or $d^{O(1)}$ (if we use Lemma 19).

Now, consider any i such that $j(i) < j(k)$. We note that the number of occurrences of z_i in the inductively constructed subformulas is

$$\begin{cases} d^{C\log(d/i)) + C(j(t^{j-1}-1) - j(i))} = d^{C\log(d/i)) + C(j(k)-j(i)) - C} & \text{in the item 1} \\ d^{C(j(t^{j-1}-1)-j(i)) + C} = d^{C(j(k)-j(i))} & \text{in the item 2.} \end{cases}$$

There are at most $O(d^2)$ summands in the expression in (7), hence the total number of occurrences of z_i can be bounded as in the inductive statement (as long as $C \geq 2$).

[12] In fact C can be chosen as 2 plus the maximum of the constants hidden in the $O(\cdot)$ of Lemmas 18 and 19.

Finally, while proving Item 1, we note that each of the individual subformulas substituted in (7) is a weighted homogeneous formula, meaning that the overall construction is also weighted homogeneous. For Item 2, we see that each subformula computing a weighted homogeneous polynomial of degree D has syntactic degree at most $D \cdot t$, implying that the overall formula has syntactic degree at most $d \cdot t$. □

Putting the above proposition together with Theorem 15 and Lemmas 16 and 17, we get the following corollary, which will be used directly in the proof of Theorem 5.

Corollary 21. *For any d, m, r, we have the following.*

1. *The polynomial $\mathrm{WE}^d_{m,r}(u_{i,j} : i \in [m], j \in [r])$ has a weighted homogeneous formula such that the number of occurrences of each variable $u_{i,j}$ is at most $d^{O(\log(d/j))+O(1)}$ (where the $O(\cdot)$ hides absolute constants).*
2. *For any $t \geq 2$, $\mathrm{WE}^d_{m,r}(u_{i,j} : i \in [m], j \in [r])$ has a formula of syntactic degree at most $t \cdot d$ such that the number of occurrences of each variable $u_{i,j}$ is at most $d^{O(\log_t(d/j))+O(1)}$ (where the $O(\cdot)$ hides absolute constants).*

6.2 Proof of Theorem 5

We are now ready to prove Theorem 5. Let F be a formula of size s and depth Δ computing a homogeneous polynomial P of degree d. We will show how to construct a homogeneous formula F' of size $s \cdot d^{O(\Delta+\log d)}$ for P, using Item 1 of Corollary 21. The construction of the quasi-homogeneous formula F'' is similar using Item 2 of the same corollary, and we will only sketch the differences.

Proof of Item 1. Let us start with the construction of F'. We show the following more general statement by induction on a parameter $h \leq \Delta$. We show that for any gate g at height h in F and any degree $j \leq d$, the degree-j component of the polynomial computed by g, denoted $[g]_j$ has a homogeneous formula of size at most $s_g \cdot d^{C\cdot(h+\log j)}$ where s_g denotes the size of the subformula rooted at g and C is an absolute constant that we will choose in the proof below. Applying this to the case when g is the output gate of F and $j = d$ yields the formula F'.

We now prove the inductive claim. The base case ($h = 0$) is immediate. We now assume that $h \geq 1$ and g is a gate of F at height h. The claim naturally breaks into two cases depending on whether g is a sum or a product gate. The case of the sum gate immediately follows by induction and we omit it.

We now consider the case when g is a product gate with children $g_1, \ldots, g_m$. Fix a $j \leq d$. By induction, for each $i \leq j$ and each $k \leq m$, the polynomial $[g_k]_i$ has a homogeneous formula $F_{i,k}$ of size $s_{g_k} \cdot d^{C\cdot(h-1+\log i)}$.

To get a formula for $[g]_j$, we note that

$$[g]_j = \left[\prod_{k=1}^{m}\left(\gamma_k + \sum_{i=1}^{j}[g_k]_i\right)\right]_j.$$

The reader will note that if $\gamma_k = 1$ for all k, then the above immediately implies that $[g]_j$ is the Weighted Elementary symmetric polynomial in the polynomials $([g_k]_i : k \in [m], i \in [j])$. A similar fact also holds if $\gamma_k \neq 0$ for all k (after scaling by suitable constants, we can assume $\gamma_k = 1$ for all k). To handle the case when some γ_k are 0, we use a simple interpolation idea (see also [LST21b, Proof of Lemma 20]).

Without loss of generality, we assume that $\gamma_k = 0$ for $k \leq \ell$ and $\gamma_k \neq 0$ otherwise. We introduce a new variable u and consider the polynomial

$$R = \left[\prod_{k=1}^{\ell}\left(u + \sum_{i=1}^{j}[g_k]_i\right) \cdot \prod_{k>\ell}\left(\gamma_k + \sum_{i=1}^{j}[g_k]_i\right)\right]_j.$$

It is easily verified that the degree of u in R is $\ell \leq j$. If we consider R to be a univariate polynomial in u, the coefficient of $u(0)$ is exactly the polynomial $[g]_j$. We can obtain this coefficient by interpolating R

at $\ell+1$ non-zero points $\upsilon_1,\dots,\upsilon_{\ell+1}\in\mathbb{F}$. We thus get constants $\theta_1,\dots,\theta_{\ell+1}\in\mathbb{F}$ such that

$$\begin{aligned}
[g]_j &= \sum_{q=1}^{\ell+1}\theta_q\cdot\left[\prod_{k=1}^{\ell}\left(\upsilon_q+\sum_{i=1}^{j}[g_k]_i\right)\cdot\prod_{k>\ell}\left(\gamma_k+\sum_{i=1}^{j}[g_k]_i\right)\right]_j\\
&= \sum_{q=1}^{\ell+1}\theta'_q\cdot\left[\prod_{k=1}^{m}\left(1+\sum_{i=1}^{j}\gamma'_{q,k,i}\cdot[g_k]_i\right)\right]_j\\
&= \sum_{q=1}^{\ell+1}\theta'_q\cdot \mathrm{WE}^{j}_{m,j}(\gamma'_{q,k,i}[g_k]_i : k\in[m], i\in[j]) \qquad (8)
\end{aligned}$$

where the second equality is obtained by re-scaling each non-zero constant term in the products to 1, resulting in some new coefficients $\gamma'_{q,k,i}$.

We use (8) to get a homogeneous formula for $[g]_j$. By Item 1 of Corollary 21, $\mathrm{WE}^{j}_{m,j}(u_{k,i} : k\in[m], i\in[j])$ has a weighted homogeneous formula where each $u_{k,i}$ appears $d^{O(\log(j/i))+O(1)}$ times. Using this formula in (8) and composing it with the inductively obtained homogeneous formulas for the polynomials $[g_k]_i$, we get a formula for $[g]_j$ of size at most

$$\begin{aligned}
&(\ell+1)\cdot\sum_{k=1}^{m}\sum_{i=1}^{j}\underbrace{d^{O(\log(j/i)+O(1))}}_{\text{Number of occurrences of }[g_k]_i}\cdot\underbrace{s_{g_k}\cdot d^{C(h-1+\log i)}}_{\text{Size of }[g_k]_i}\\
&\le \sum_{k=1}^{m}\sum_{i=1}^{j} d^{O(\log(j/i)+O(1))}\cdot s_{g_k}\cdot d^{C(h-1+\log i)} \qquad (\text{as } \ell\le d)\\
&\le \sum_{k=1}^{m}\sum_{i=1}^{j} d^{C(h+\log j)-1}\cdot s_{g_k} \qquad (\text{for large enough } C)\\
&\le \sum_{k=1}^{m} d^{C(h+\log j)}\cdot s_{g_k} \qquad (j\le d)\\
&= d^{C(h+\log j)}\cdot\sum_{k=1}^{m} s_{g_k} = d^{C(h+\log j)}\cdot s_g,
\end{aligned}$$

proving the inductive claim. Note that, as we are composing homogeneous formulas, the formula for $[g]_j$ thus obtained is homogeneous (Lemma 14 Item 1).

Proof Sketch of Item 2. The construction of the formula F'' proceeds in a similar way.

Fix the parameter $t=\lfloor d^{\varepsilon/\Delta}\rfloor$ where $\varepsilon>0$ is the absolute constant from the statement of the theorem. Note that we can assume that t is at least some large constant, since otherwise $\Delta=\Omega(\log d)$ and the statement immediately follows from Item 1 of the theorem. We prove the following inductive claim: for any gate g at height h in F and any degree $j\le d$, the degree-j component of the polynomial computed by g, denoted $[g]_j$ has a quasi-homogeneous formula of size at most $s_g\cdot d^{C\cdot(h+\log_t j)}$ and syntactic degree at most $t^h\cdot j$. In the special case that g is the output gate of the formula and $j=d$, we get the statement of the theorem.

The base case of the induction ($h=0$) is immediate. For the induction, we start with (8) and use Item 2 of Corollary 21 along with the inductively obtained quasi-homogeneous formulas for the polynomials $[g_k]_i$ to get a formula for $[g]_j$. This yields, as above, a formula for $[g]_j$ of the claimed size.

Furthermore, by induction, the syntactic degree of the formula computing $[g_k]_i$ is at most $t^{h-1}\cdot i$. Moreover, the quasi-homogeneous formulas for $\mathrm{WE}^{j}_{m,j}$ from Corollary 21 have syntactic degree at most $t\cdot j$. Hence, by Lemma 14 Item 2, the formula we have constructed for $[g]_j$ has syntactic degree at most $t^h\cdot j$, as desired.

6.3 Proofs of Corollaries

Proof of Corollary 6. By a standard result of Brent, Kuck, and Maruyama [BKM73], we can assume that F has fan-in bounded by 2 and depth $\Delta=O(\log s)$.

We cannot directly apply Theorem 5 to F as the depth of this formula is too large. Instead, we choose an integer parameter ℓ and define $\mathcal{G}_i$ $(0 \le i \le \Delta/\ell)$ to be the set of gates at depth exactly $i \cdot \ell$ in the formula. Given any gate $g \in \mathcal{G}_i$, we define the formula F_g to be the subformula of depth ℓ rooted at g with leaves in $\mathcal{G}_{i+1}$. Let $\hat{F}_g$ be the formula obtained by expressing the polynomial computed by F_g as a brute-force sum of monomials. Note that $\hat{F}_g$ computes a polynomial in at most 2^ℓ variables of degree at most 2^ℓ and hence has size at most $\exp(O(2^\ell))$.

Let $\hat{F}$ be the formula obtained by replacing each F_g by $\hat{F}_g$ as computed above. The size of $\hat{F}$ can be bounded by

$$s' = \prod_{i=0}^{\Delta/\ell} \max_{g \in \mathcal{G}_i} \text{size}(\hat{F}_g) \le \exp(O(\Delta 2^\ell/\ell)).$$

The depth of $\hat{F}$ is at most Δ/ℓ.

We now apply Theorem 5 to $\hat{F}$. We see that the polynomial P computed by $\hat{F}$ also has a weighted homogeneous formula F' of size

$$\exp(O(\Delta 2^\ell/\ell)) \cdot d^{O(\log d + \Delta/\ell)} = \exp(O(\Delta \log d/\log\log d)) \cdot d^{O(\log d)}$$

for $\ell = \log\log d$. For $d = s^{o(1)}$, we see that this is bounded by $d^{o(\log s)}$.

Similarly, we also see that the polynomial P has a quasi-homogeneous formula F'' of syntactic degree at most $d^{1+\varepsilon}$ and size

$$\exp(O(\Delta 2^\ell/\ell)) \cdot d^{O(\Delta/\ell)} = \exp(O(\Delta \log d/\log\log d)). \qquad \square$$

Corollary 7 follows directly by using Corollary 6 and [FLM+23, Theorem 7].

7 Lower bound for non-commutative homogenization

In the following A is an integral domain. We will denote by latin letters constants and fixed parameters and we will keep greek letters for the maps.

7.1 Definitions

7.1.1 BE polynomial

Let us consider the polynomial

$$\mathrm{BE}_n^d(\mathcal{Y}) = \sum_{1 \le j_1 \le \cdots \le j_d \le n} \prod_{i=1}^{d} y_{i,j_i}.$$

BE_n^d is a polynomial of degree d with nd variables. Moreover, BE_n^d is set-multilinear with respect to the partition $\mathcal{Y}_i = \{y_{i,j} \mid j \in [1,n]\}$.

The point is that if the ordered elementary symmetric polynomials have a small homogeneous non-commutative formula, then BE has a small set-multilinear formula.

Proposition 22. *If the non-commutative polynomial E_n^d is computed by a homogeneous non-commutative formula of size s and product-depth Δ, then BE_{n-d+1}^d is also computed by a (non-commutative) set-multilinear formula of size s and product-depth Δ.*

Proof. If F is a homogeneous non-commutative formula, we can associate to each node $\mathfrak{n}$ a sub-interval of $[1,d]$, $I(\mathfrak{n})$ such that $|I(\mathfrak{n})| = \deg(\mathfrak{n})$ as follows:

- If $\mathfrak{n}$ is the output gate of the formula of degree d then $I(\mathfrak{n}) = [1,d]$.
- If $\mathfrak{n}$ is a sum-gate of degree δ of children $\mathfrak{n}_1, \ldots, \mathfrak{n}_p$, then for all $1 \le i \le p$, we set $I(\mathfrak{n}_i) = I(\mathfrak{n})$. This is accurate since by homogeneity, each node $\mathfrak{n}_i$ has degree δ.
- If $\mathfrak{n}$ is a product-gate of degree δ of interval $[a, a+\delta-1]$ which have p children $\mathfrak{n}_1, \ldots, \mathfrak{n}_p$ of respective degrees $\delta_1, \ldots, \delta_p$, then we know by homogeneity that $\delta = \delta_1 + \cdots + \delta_p$. We associate to $\mathfrak{n}_i$ the interval $[a+\delta_1+\cdots+\delta_{i-1}, a+\delta_1+\cdots+\delta_i-1]$.

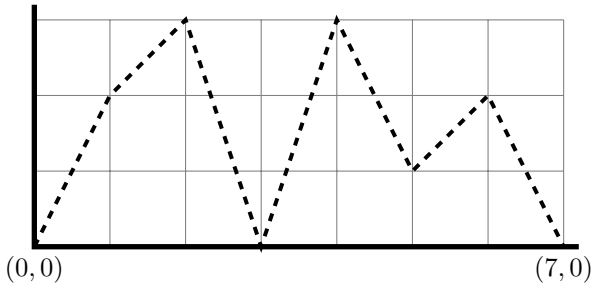


Figure 2: Path of Example 23

It is easily seen by induction that renaming all variables of the leaves of a homogeneous non-commutative formula computing E_n^d as follows: if the variable X_i is on a leaf associated to the interval $[a, a]$, it is replaced by the variable $z_{a,i}$, then the new formula computes the polynomial $\sum_{1\leq i_1<\cdots<i_d\leq n}\prod_{j=1}^d z_{j,i_j}$. The computation is set-multilinear. We finally obtain the polynomial $\mathrm{BE}_{n-d+1}^d(\mathcal{Y})$ by replacing the variables $z_{j,i}$ by the variables

$$\begin{cases} y_{j,i-j+1} & \text{if } j \leq i \text{ and } i \leq n+j-d \\ 0 & \text{otherwise.} \end{cases} \qquad \square$$

Consequently, to show lower bounds on the size of homogeneous non-commutative formulas which compute E_n^d, it is enough to show lower bounds on the size of set-multilinear formulas which compute BE_{n-d+1}^d. To do this, we will follow the proofs of Theorems 1 and 4 of [TLS22].

7.1.2 Paths and p-multilinearity

We will consider paths in the grid $[0, d] \times [0, k]$. The paths we consider start at $(0, 0)$, end at $(d, 0)$ and at each step, increases its abscissa by 1 and modifies its ordinate. More formally, we will describe these paths by elements of the set

$$\mathcal{P}_{k,d} = \left\{ p \in ([-k, k] \setminus \{0\})^d \;\middle|\; \sum_{i=1}^d p_i = 0 \text{ and } \forall j < d, 0 \leq \sum_{i=1}^j p_i \leq k \right\}.$$

When k and d are implicit, we will only write $\mathcal{P}$. Given $p \in \mathcal{P}$, we will denote by $p_{\leq t}$ the partial sum $\sum_{i=1}^t p_i$ (we will also naturally identify $p_{\leq 0}$ to 0). A vector $p \in \mathcal{P}$ corresponds to the path $(0, 0), (1, p_1), \ldots, (i, p_{\leq i}), \ldots, (d, 0)$.

For a path $p \in \mathcal{P}$, we say that a polynomial (with coefficients in A) is p-multilinear if it is set-multilinear with respect to a partition $\mathcal{X} = \biguplus_{i=1}^d \mathcal{X}_i$ which satisfies $|\mathcal{X}_i| = 2^{|p_i|}$. We will denote by $\mathrm{A}_p[\mathcal{X}]$ the set of these polynomials.

Moreover, a variable $x \in \mathcal{X}_i$ is called *positive* (resp. *negative*) if $p_i > 0$ (resp. $p_i < 0$). A monomial is called *positive* (resp. *negative*) if it only contains positive variables (resp. negative variables). Notice that any set-multilinear monomial m of $\mathcal{X}$ can be decomposed into its positive part m_+ and its negative part m_-, and then, we have $m = m_+ \cdot m_-$. The set of such positive monomials m_+ is denoted by $\mathcal{M}_{>0}$. Similarly, $\mathcal{M}_{<0}$ is the set of the negative parts. Notice that both sets have same cardinality since

$$|\mathcal{M}_{>0}| = \prod_{i, p_i>0} 2^{p_i} = \prod_{i, p_i<0} 2^{-p_i} = |\mathcal{M}_{<0}|.$$

The second equality holds since $p_{\leq d} = 0$.

Example 23. *Let $k = 3$ and $d = 7$. Let choose for instance the path $p = (2, 1, -3, 3, -2, 1, -2) \in \mathcal{P}_{k,d}$ represented in Figure 2.*

The monomial $x_{1,4}x_{2,2}x_{3,8}x_{4,8}x_{5,4}x_{6,2}x_{7,4}$ is in $\mathrm{A}_p[\mathcal{X}]$. Its positive part is $x_{1,4}x_{2,2}x_{4,8}x_{6,2}$ and its negative part is $x_{3,8}x_{5,4}x_{7,4}$.

Let $f \in \mathrm{A}_p[\mathcal{X}]$. We associate to f its 'partial derivatives matrix' $\mathrm{PDM}(f)$. $\mathrm{PDM}(f)$ is the square matrix with coefficients in A where rows are indexed by elements of $\mathcal{M}_{>0}$, columns by elements of $\mathcal{M}_{<0}$ and defined by

$$\mathrm{PDM}(f)_{m_+,m_-} = (\text{coefficient of } m_+ \cdot m_- \text{ in } f).$$

A polynomial $f \in \mathrm{A}_p[\mathcal{X}]$ is said to be *full-rank* if $\mathrm{PDM}(f)$ is full-rank. We will also use the useful notation: $\mathrm{relrk}(f) = \frac{\mathrm{rk}(\mathrm{PDM}(f))}{\sqrt{|\mathcal{M}_{>0}||\mathcal{M}_{<0}|}}$.

The main proposition which will be proved in this section is

Proposition 24. *Let $n = 2^k$ and A of cardinal at least $d^2 2^{k(d+1)}$. For all $p \in \mathcal{P}_{k,d}$, there exists a polynomial H_p in $\mathrm{A}_p[\mathcal{X}]$ which is a set-multilinear projection of BE^d_{nd} such that H_p is full-rank.*

7.1.3 Proof of Theorem 8

We start by showing how Proposition 24 can be used for proving Theorem 8. As we said earlier, the approach will follow the proofs of Theorems 4 and 1 in [TLS22]. So we will directly use Lemma 10, 13 and 14 and Proposition 15 of [TLS22].

Let us start by proving that if $d \leq n^{0.99}$, then any homogeneous non-commutative formula of product-depth Δ computing E^d_n has size $n^{\Omega(d^{1/\Delta}/2^\Delta)}$ (the proof mimics the one of Corollary 16 in [TLS22]). We assume that $d \geq 2^{\Delta^2} 80^\Delta$ since otherwise the result is trivial. We now split the analysis into two cases.

If $2^\Delta \geq 0.001 \log n$, then we need to argue a lower bound of $\exp(\Omega(d^{1/\Delta}))$. For this, we appeal to a result of Hrubeš and Yehudayoff [HY11] which yields such a lower bound for commutative homogeneous multilinear formulas of product-depth Δ. This also implies a lower bound for the non-commutative case, as we can just treat any non-commutative set-multilinear formula for E^d_n as a commutative formula for the same polynomial. Hence, we are done.

Now assume that $2^\Delta < 0.001 \log n$. Let us fix integers $k = 2\lfloor 0.001 \log n \rfloor$ and $k' = \lfloor \log(n^{0.001/2^\Delta}) \rfloor \geq 1$. Since, $k' 2^\Delta \leq k/2$, there is a balanced word $w \in \mathbb{Z}^{d-2}$ as guaranteed by Proposition 15 of [TLS22] that is $k/2$-unbiased. Let p be the path $(k/2, w_1, \ldots, w_{d-2}, -k/2)$ in $\mathcal{P}_{k,d}$.

Since $2^k d \leq n - d + 1$, by Propositions 22 and 24, if E^d_n has a set-multilinear formula F of size s and product-depth Δ, then so do the polynomials BE^d_{n-d+1} and H_p. H_p is set-multilinear along the ordered partition $\mathcal{X} = \biguplus_{i=1}^d \mathcal{X}_i$. Let $x_{1,a}$ and $x_{d,b}$ be one variable from $\mathcal{X}_1$ and one from $\mathcal{X}_d$. Let $\mathrm{H}_{p,a,b}$ be the polynomial we obtain by substituting in H_p the variables $x_{1,a}$ and $x_{d,b}$ by 1 and all other variables from $\mathcal{X}_1 \cup \mathcal{X}_d$ by 0. The polynomial $\mathrm{H}_{p,a,b}$ is a set-multilinear in $\mathrm{A}_w[\uplus_{2 \leq i \leq d-1} \mathcal{X}_i]$. By Proposition 15 of [TLS22], $\mathrm{relrk}(\mathrm{H}_{p,a,b}) \leq s 2^{-k'(d-2)^{1/\Delta}/10}$. However, since the matrix $\mathrm{PDM}(\mathrm{H}_p)$ can be seen as a $2^{k/2} \times 2^{k/2}$ blocks-matrix where each block is some $\mathrm{PDM}(\mathrm{H}_{p,a,b})$ (for some choice of variables $(x_{1,a}, x_{d,b})$), we also get

$$1 = \mathrm{relrk}(\mathrm{H}_p) \leq 2^{k/2} \max_{a,b}(\mathrm{relrk}(\mathrm{H}_{p,a,b})).$$

It implies that $s \geq 2^{(d^{1/\Delta} \log n)/(40000 \cdot 2^\Delta)}$. This finishes the proof of the first part of the theorem.

Let us prove now the second part of Theorem 8. Let us recall the definition of a bias with respect to a partition [TLS22]:

Definition 25 (Bias of a word w.r.t. a partition)**.** *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be an ordered partition of $[d]$ (each $S_i \subseteq [d]$ is non-empty). We assume that the S_is are ordered with respect to their maximal elements (i.e., $i < j \implies \max(S_i) < \max(S_j)$).*

*Let $w \in \mathbb{Z}^d$ be arbitrary. Given a partition $\mathcal{S} = (S_1, S_2, \ldots, S_\ell)$ of $[d]$, we define the $\mathcal{S}$-*bias of w — $\mathrm{bias}(\mathcal{S}, w)$ — *to be the quantity $\sum_{j \in [\ell]} |w_{S_j}|$ where $w_{S_j} = \sum_{i \in S_j} w_i$.*

Assume E^d_n has a homogeneous non-commutative formula of size s and product-depth Δ. Then, by Proposition 22, BE^d_{n-d+1} has a set-multilinear formula of size s and product-depth Δ.

Using the Product Lemma (Lemma 10 in [TLS22]), we have

$$\mathrm{BE}^d_{n-d+1} = \sum_{i=1}^{s} \prod_{j=1}^{\ell} F_{i,j} \tag{9}$$

where $\ell \geq \log d$ and for each $i \in [s]$, there is a partition $\mathcal{S}_i = (S_{i,1}, \ldots, S_{i,\ell})$ of $[1, d]$ such that $F_{i,j}$ is a set-multilinear polynomial in the variables $(\mathcal{Y}_p : p \in S_{i,j})$.

As we aim for a lower bound of the form $(\log n)^{\Omega(\ell)}$, if $\ell \leq \frac{10 \log n}{\log\log n}$ then the result is trivial, so we assume this is not the case.

For each partition $\mathcal{S}_i$ of $[1,d]$, we introduce the partition $\mathcal{T}_i$ of $[2,d-1]$ which is the restriction of $\mathcal{S}_i$ to the set $[2,d-1]$.

Let us take $\varepsilon = (\log\log n)^2/\log n$. We have $\varepsilon\ell \geq 10$. Let us fix 2^k be the largest power of two which is at most $(n-d+1)/d$. In particular, $(n-d+1)/2d < 2^k \leq (n-d+1)/d$. We will consider paths of $\mathcal{P}_{k,d}$. Since $d \leq n^{1-2/\log\log n} \leq n/2$ for n sufficiently large, we have that $\varepsilon k \geq \log\log n$.

By Lemma 14 in [TLS22], there is a probability distribution $\mathcal{D}$ over $k/2$-unbiased words $w = (w_2, \ldots, w_{d-1}) \in \mathbb{Z}^{d-2}$ such that we have

$$\Pr_w[\exists i \in [s] : \mathrm{bias}(\mathcal{T}_i, w) \leq \frac{\varepsilon k \ell}{2}] \leq (10\varepsilon)^{\ell/2} \cdot s$$

where the inequality uses a union bound and each $\mathcal{T}_i$ is the partition of $[2,d-1]$ defined above.

If $s \geq (1/10\varepsilon)^{\ell/2}$, then the lower bound of the theorem holds trivially and we are done. So we assume $s < (1/10\varepsilon)^{\ell/2}$. In particular, we see that there is a w such that $\mathrm{bias}(\mathcal{T}_i, w) > \varepsilon k\ell/2$ for each $i \in [s]$ and fix such a w. We will consider the path $p = (k/2, w_2, \ldots, w_{d-1}, -k/2 - \sum_i w_i)$. Notice that since w is $k/2$-unbiased, it implies that p is in $\mathcal{P}_{k,d}$. Moreover, we get $\mathrm{bias}(\mathcal{S}_i, p) > k(\varepsilon\ell - 3)/2 \geq k\varepsilon\ell/4$ for each $i \in [s]$.

We know by Proposition 24 that there is a polynomial H_p which is a set-multilinear restriction of BE^d_{n-d+1} (since $2^k d \leq n-d+1$) and which is full-rank. Thus, by applying this linear substitution to both sides of (9) we get

$$\mathrm{H}_p = \sum_{i=1}^{s} \prod_{j=1}^{\ell} \mathrm{H}_{i,j}$$

where $\mathrm{H}_{i,j}$ is the result of applying the linear substitutions to all the variables of $F_{i,j}$. Note in particular that $\mathrm{H}_{i,j}$ is a set-multilinear polynomial in just the variables of $\biguplus_{t \in \mathcal{S}_{i,j}} \mathcal{Y}_t$. Hence, by Lemma 13 in [TLS22], we have for each $i \in [s]$,

$$\mathrm{relrk}_p\left(\prod_{j=1}^{\ell} \mathrm{H}_{i,j}\right) \leq 2^{-\varepsilon k\ell/8}.$$

On the other hand, by the sub-additivity of relrk we have

$$1 \leq \mathrm{relrk}_p(\mathrm{H}_p) \leq \sum_{i=1}^{s} \mathrm{relrk}_p\left(\prod_{j=1}^{\ell} \mathrm{H}_{i,j}\right) \leq s \cdot 2^{-\varepsilon k\ell/8}.$$

This implies that $s \geq (\log n)^{\Omega(\ell)}$ finishing the proof.

7.1.4 Tree associated to a path

Let p be a path in $\mathcal{P}$, we associate a tree T_p. Intuitively, T_p is the tree we obtain from the path p (seen inside the grid) by 'pushing' the decreasing edges on the left to make them coincide with the last increasing edge of the same level. The tree T_p is more formally defined by:

- the nodes of the tree are the elements of

$$\{(0,0)\} \cup \{(i,j) \in [1,d] \times [1,k] \mid p_{\leq i-1} < j \leq p_{\leq i}\},$$

- $(0,0)$ is the root of the tree,
- there is an edge from (i_1, j_1) to (i_2, j_2) (two nodes of the tree) if and only if $j_2 = j_1 + 1$, $i_2 \geq i_1$, and for all $i_1 < i < i_2$, $p_{\leq i} \geq j_1$.

Given $p \in \mathcal{P}$, we define L_p as the set of the leaves of T_p. Notice that L_p corresponds exactly to the set of the couples $(i, p_{\leq i})$ which verify $p_i > 0$ and $p_{i+1} < 0$. Consequently, we order the elements of

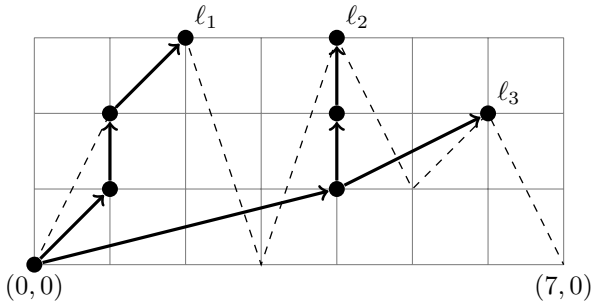


Figure 3: Tree T_p associated to the path p from Example 23 of leaves $\{\ell_1, \ell_2, \ell_3\}$

L_p according to their projection on the first coordinate. Moreover, we also notice that L_p induces the following intervals partition on $[1, d] = \biguplus_{\ell \in L_p} I_\ell$ where

$$I_{(i,j)} = \{t \in [1, i] \mid \forall t \leq u \leq i, p_u > 0\} \cup \{t \in [i+1, d] \mid \forall i+1 \leq u \leq t, p_u < 0\}.$$

We define the function $\lambda : [1, d] \to L_p$ which associates to each $i \in [1, d]$ the corresponding leaf ℓ verifying $i \in I_\ell$.

Example (Continued from Example 23)**.** *The tree T_p associated with the path p of Example 23 is given in Figure 3. This tree has three leaves $\ell_1 < \ell_2 < \ell_3$ and $[1, d]$ is partitioned by $I_{\ell_1} = [1, 3]$, $I_{\ell_2} = [4, 5]$, and $I_{\ell_3} = [6, 7]$.*

7.1.5 Hard polynomial

The hard polynomial will be defined as the evaluation of BE on a particular matrix $\mathcal{H}$. We need to start to introduce the ordered set $\mathcal{C}_p$ of the columns of $\mathcal{H}$.

Given a word $w \in \{0,1\}^k$, we will denote its subwords by $w_{[a:b]} \stackrel{\text{def}}{=} (w_a, \ldots, w_b)$ (where $a \leq b$).

Let $\mathcal{C}_p = L_p \times \{0,1\}^k$. We define the following total order on $\mathcal{C}_p$:

$$(\ell, w) \prec (\ell', w') \Leftrightarrow \begin{cases} w_{[1:j]} <_{\text{lex}} w'_{[1:j]} \\ \text{or } (w_{[1:j]} = w'_{[1:j]} \text{ and } \ell < \ell') \\ \text{or } (w_{[1:j]} = w'_{[1:j]} \text{ and } \ell = \ell' \text{ and } w <_{\text{lex}} w'). \end{cases}$$

where j is the second coordinate of the first commun ancestor of ℓ and ℓ'.

We will also use the notation $(\ell, w) \preccurlyeq (\ell', w')$ when $(\ell, w) \prec (\ell', w')$ or $(\ell, w) = (\ell', w')$. Let us call by ψ be the unique isomorphism of ordered sets from $\mathcal{C}_p$ to $[1, 2^k|L_p|]$.

For any path $p \in \mathcal{P}$, we consider the matrix $\mathcal{H}_p$ with d rows and $|\mathcal{C}_p|$ columns defined by

$$\mathcal{H}_{i,(\ell,w)} = \begin{cases} t^{\psi(\ell,w)} x_{i, w_{[p_{\leq i-1}+1 : p_{\leq i}]}} & \text{if } \ell = \lambda(i) \text{ and } p_i > 0, \\ t^{\psi(\ell,w)} x_{i, w_{[p_{\leq i}+1 : p_{\leq i-1}]}} & \text{if } \ell = \lambda(i) \text{ and } p_i < 0, \\ 0 & \text{otherwise.} \end{cases}$$

We define $\mathrm{H}_p(t, \mathcal{X}) \stackrel{\text{def}}{=} \mathrm{BE}^d_{2^k|L_p|}(\mathcal{H})$. Notice that for any $a \in \mathrm{A}$, $\mathrm{H}_p(a, \mathcal{X}) \in \mathrm{A}_p[\mathcal{X}]$ is a set-multilinear projection of BE^d_{nd}.

Example (Continued from Example 23)**.** *Continuing the example when $p = (2, 1, -3, 3, -2, 1, -2)$, we obtain the ordered set $\mathcal{C}_p$:*

$$\begin{aligned} &(\ell_1, 000) \prec \cdots \prec (\ell_1, 111) \prec (\ell_2, 000) \prec \cdots \prec (\ell_2, 011) \prec (\ell_3, 000) \prec \cdots \prec (\ell_3, 011) \\ &\prec (\ell_2, 100) \prec \cdots \prec (\ell_2, 111) \prec (\ell_3, 100) \prec \cdots \prec (\ell_3, 111). \end{aligned}$$

Notice that $(4, 1)$ is the first commun ancestor of ℓ_2 and ℓ_3, so $(\ell_3, 011) \prec (\ell_2, 100)$ by the first rule of the ordering and $(\ell_2, 111) \prec (\ell_3, 100)$ by the second one.

The transpose of the matrix $\mathcal{H}_{(2,1,-3,3,-2,1,-2)}$ *is given by*

$$\begin{bmatrix}
t^1x_{1,000} & t^1x_{2,000} & t^1x_{3,000} & & & & & \\
t^2x_{1,001} & t^2x_{2,001} & t^2x_{3,001} & & & & & \\
t^3x_{1,010} & t^3x_{2,010} & t^3x_{3,010} & & & & & \\
t^4x_{1,011} & t^4x_{2,011} & t^4x_{3,011} & & & & & \\
t^5x_{1,100} & t^5x_{2,100} & t^5x_{3,100} & & & & & \\
t^6x_{1,101} & t^6x_{2,101} & t^6x_{3,101} & & & & & \\
t^7x_{1,110} & t^7x_{2,110} & t^7x_{3,110} & & & & & \\
t^8x_{1,111} & t^8x_{2,111} & t^8x_{3,111} & & & & & \\
 & & & t^9x_{4,000} & t^9x_{5,000} & & & \\
 & & & t^{10}x_{4,001} & t^{10}x_{5,001} & & & \\
 & & & t^{11}x_{4,010} & t^{11}x_{5,010} & & & \\
 & & & t^{12}x_{4,011} & t^{12}x_{5,011} & & & \\
 & & & & & t^{13}x_{6,000} & t^{13}x_{7,000} & \\
 & & & & & t^{14}x_{6,001} & t^{14}x_{7,001} & \\
 & & & & & t^{15}x_{6,010} & t^{15}x_{7,010} & \\
 & & & & & t^{16}x_{6,011} & t^{16}x_{7,011} & \\
 & & & t^{17}x_{4,100} & t^{17}x_{5,100} & & & \\
 & & & t^{18}x_{4,101} & t^{18}x_{5,101} & & & \\
 & & & t^{19}x_{4,110} & t^{19}x_{5,110} & & & \\
 & & & t^{20}x_{4,111} & t^{20}x_{5,111} & & & \\
 & & & & & t^{21}x_{6,100} & t^{21}x_{7,100} & \\
 & & & & & t^{22}x_{6,101} & t^{22}x_{7,101} & \\
 & & & & & t^{23}x_{6,110} & t^{23}x_{7,110} & \\
 & & & & & t^{24}x_{6,111} & t^{24}x_{7,111} &
\end{bmatrix}.$$

Proposition 24 is a direct consequence of the next proposition.

Proposition 26. *Let* $p \in \mathcal{P}$. *There exists* $a \in \mathrm{A}$ *such that* $\mathrm{H}_p(a, \mathcal{X})$ *is full-rank.*

7.2 Proof of Proposition 26

7.2.1 Existence of a

Let $f \in \mathrm{A}[t]$ be a nonzero polynomial, we denote by $\mathrm{val}(f)$ (valuation of f) the minimal integer v such that the coefficient of t^v in f is non-zero. We extend the definition for the zero polynomial by $\mathrm{val}(0) = +\infty$.

For proving Proposition 26, we will need the two following lemmas that we will prove in the next sections.

Lemma 27. *Let* $p \in \mathcal{P}$. *Let* PDM *be the matrix* $\mathrm{PDM}(\mathrm{H}_p(t, .))$ *when* $\mathrm{H}_p(t, \mathcal{X})$ *is seen as a polynomial in* $(\mathrm{A}[t])_p[\mathcal{X}]$.

For any $m_+ \in \mathcal{M}_{>0}$, *there exists a unique* m_- *in* $\mathcal{M}_{<0}$ *such that for all* m'_- *in* $\mathcal{M}_{<0}$, *if* $\mathrm{val}(\mathrm{PDM}_{m_+,m'_-}) \leq \mathrm{val}(\mathrm{PDM}_{m_+,m_-})$ *then* $m'_- = m_-$.

Let us denote by θ the application which sends m_+ to m_-.

Lemma 28. *The application* θ *is a bijection of* $\mathcal{M}_{>0}$ *to* $\mathcal{M}_{<0}$.

Proof of Proposition 26 from Lemmas 27 and 28. Let

$$\Delta(t) = \det(\mathrm{PDM}(\mathrm{H}_p(t, .))) \in \mathrm{A}[t].$$

By definition of θ, for any map σ from $\mathcal{M}_{>0}$ to $\mathcal{M}_{<0}$,

$$\mathrm{val}\left(\prod_{m_+ \in \mathcal{M}_{>0}} \mathrm{PDM}_{m_+,\sigma(m_+)}\right) \geq \mathrm{val}\left(\prod_{m_+ \in \mathcal{M}_{>0}} \mathrm{PDM}_{m_+,\theta(m_+)}\right) \stackrel{\text{def}}{=} \tau$$

and the equality holds only if $\sigma = \theta$. Consequently, the coefficient of the monomial t^τ in Δ is just the product $\prod_{m_+ \in \mathcal{M}_{>0}} \mathrm{PDM}_{m_+,\sigma(m_+)}$ evaluated at $t = 1$. As A is an integral domain, this product is nonzero. Consequently Δ is not identically zero.

Then, we can notice that the degree of Δ is bounded by $|\mathcal{M}_{>0}|d2^k|L_p| \leq d^2 2^{k(d+1)}$. As A is an integral domain of cardinality larger than $d^2 2^{k(d+1)}$, there exists $a \in \mathrm{A}$ such that $\Delta(a) \neq 0$. Since $\det(\mathrm{PDM}(\mathrm{H}_p(a,.))) = \Delta(a)$, it proves the proposition. □

7.2.2 Proof of Lemma 27

Let us recall that λ is a function from $[1,d]$ to L_p which was defined at the end of Subsection 7.1.4.

Let us recall that

$$\mathrm{H}_p = \sum_{\substack{\sigma:[1,d]\to\mathcal{C}_p \\ \sigma \text{ is non-decreasing}}} \prod_{i=1}^{d} \mathcal{H}_{i,\sigma(i)} = \sum_{\substack{\omega:[1,d]\to\{0,1\}^k \\ (\lambda,\omega) \text{ non-decreasing}}} \prod_{i=1}^{d} \mathcal{H}_{i,(\lambda,\omega)(i)} \tag{10}$$

The second equality stands since a summand is nonzero if and only if for all $i \in [1,d]$, $\sigma(i)$ is of the form $(\lambda(i), w)$. Let us fix a map ω such that $\sigma = (\lambda, \omega)$ is non-decreasing, the corresponding summand of H_p is the monomial

$$t^{\sum_i \psi(\sigma(i))} \prod_{i|p_i>0} x_{i,\omega(i)_{[p_{\leq i-1}+1:p_{\leq i}]}} \prod_{i|p_i<0} x_{i,\omega(i)_{[p_{\leq i}+1:p_{\leq i-1}]}}.$$

The positive part of this monomial is $m_+ = \prod_{i|p_i>0} x_{i,\omega(i)_{[p_{\leq i-1}+1:p_{\leq i}]}}$. Hence, it is fixed by the data of $\omega(i)_{[p_{\leq i-1}+1:p_{\leq i}]}$ for all i such that $p_i > 0$.

We can also notice that if two such maps σ, σ' are associated to two nonzero summands and verify $\sigma \preccurlyeq \sigma'$ (in the sense that for all i, $\sigma(i) \preccurlyeq \sigma'(i)$), then we have (since ψ is an isomorphism of ordered sets)

$$\sum_i \psi(\sigma(i)) \leq \sum_i \psi(\sigma'(i)).$$

It means that the t-valuation of $\prod_i \mathcal{H}_{i,\sigma(i)}$ is at most the t-valuation of $\prod_i \mathcal{H}_{i,\sigma'(i)}$. Furthermore the equality holds only if $\sigma = \sigma'$.

Let m_+ be a monomial in $\mathcal{M}_{>0}$, let us denote by $\mathfrak{S}_{m_+}$ the set of non-decreasing maps $\sigma = (\lambda, \omega)$ from $[1,d]$ to $\mathcal{C}_p$ such that the positive part of the associated summand is exactly m_+.

First, we note that for any m_+ in $\mathcal{M}_{>0}$, the set $\mathfrak{S}_{m_+}$ is non empty. Indeed, let us construct ω by induction on i with the property that for each i with $p_i > 0$, we have that the suffix $\omega(i)_{[p_{\leq i}+1:k]}$ is the zeros-word. When $i = 1$, we know that $p_1 = p_{\leq 1} > 0$. Let $x_{1,w}$ be the variable in $m_+ \cap \mathcal{X}_1$. We set $\omega(1) \in \{0,1\}^k$ to be the concatenation of the word $w \in \{0,1\}^{p_1}$ and of the zeros-word $(0,\ldots,0) \in \{0,1\}^{k-p_1}$. We get that $\mathcal{H}_{1,(\lambda,\omega)(1)} = t^{\psi((\lambda,\omega)(1))} x_{1,w}$ as required. Assume now that for some $i < d$ we have defined $\omega(i)$ and let us define $\omega(i+1)$. In the first case, $p_{i+1} < 0$. In this case, we have $\lambda(i+1) = \lambda(i)$ so we can set $\omega(i+1) = \omega(i)$. We clearly have $(\lambda,\omega)(i) \preccurlyeq (\lambda,\omega)(i+1)$. In the second case, $p_{i+1} > 0$. Let $x_{i+1,w}$ be the variable in $m_+ \cap \mathcal{X}_{i+1}$. We set $\omega(i+1)$ be the concatenation of $\omega(i)_{[1:p_{\leq i}]} \in \{0,1\}^{p_{\leq i}}$, of $w \in \{0,1\}^{p_{i+1}}$, and of the zeros-word $(0,\ldots,0) \in \{0,1\}^{k-p_{\leq i+1}}$. By construction $\mathcal{H}_{i+1,(\lambda,\omega)(i+1)}$ is of the form $t^{\psi((\lambda,\omega)(i+1))} x_{i+1,w}$ as required. So we just need to show that the order $(\lambda,\omega)(i) \preccurlyeq (\lambda,\omega)(i+1)$ holds. If $\lambda(i+1) = \lambda(i)$, then $p_i > 0$ and by hypothesis $\omega(i)_{[p_{\leq i}+1:k]}$ is the zeros-word, which implies that $\omega(i) \leq_{\mathrm{lex}} \omega(i+1)$ and the order is verified. Otherwise $\lambda(i+1) > \lambda(i)$. It means that $p_i < 0$ and that $p_{\leq i}$ is the second coordinate of the first commun ancestor of $\lambda(i)$ and $\lambda(i+1)$. Hence the order is again verified

Finally, we can notice that if σ, σ' are two maps from $\mathfrak{S}_{m_+}$, then the map $\min(\sigma, \sigma')$ also belongs to $\mathfrak{S}_{m_+}$. Consequently, $\mathfrak{S}_{m_+}$ has a minimal element σ_{m_+} associated to a unique minimal t-valuation. Choosing m_- as the negative part of $\prod \mathcal{H}_{i,\sigma_{m_+}(i)}$ proves Lemma 27.

7.2.3 Proof of Lemma 28

We want to show here that θ is a bijection. The lemma will easily follow from some lemmas on the structure of $\min(\mathfrak{S}_{m_+})$. Let us start by identifying $\sigma_{m_+}(i)$ when i verifies $p_i < 0$. To get the minimal t-valuation, it is enough to select the smallest possible output (the only limitation comes from the fact that σ_{m_+} is non-decreasing)

Lemma 29. *Let $m_+ \in \mathcal{M}_{>0}$ and $\sigma_{m_+} = \min\left(\mathfrak{S}_{m_+}\right)$. For all i we have $\sigma_{m_+}(i) = \sigma_{m_+}(i-1)$ as soon as $p_i < 0$.*

Proof. Assume this is not the case. Let i be an index where $\sigma_{m_+}(i) \succ \sigma_{m_+}(i-1)$ and $p_i < 0$. Let us consider the map

$$\sigma : j \mapsto \begin{cases} \sigma_{m_+}(j) \text{ if } j \neq i \\ \sigma_{m_+}(j-1) \text{ if } j = i. \end{cases}$$

Clearly σ is still non-decreasing and its associated positive part is m_+. Moreover for $j \neq i$, the first projection of $\sigma(j)$ is $\lambda(j)$. So, the only thing to show to get that σ is in $\mathfrak{S}_{m_+}$, is to notice that $\lambda(i-1) = \lambda(i)$. But, since $p_i < 0$, it directly follows from the definition of λ. Consequently σ is also in $\mathfrak{S}_{m_+}$, which contradicts the minimality of σ_{m_+}. □

Let us focus now on the behavior when i verifies $p_i > 0$.

Lemma 30. *Let $m_+ \in \mathcal{M}_{>0}$ and $\sigma_{m_+} = (\lambda, \omega) = \min\left(\mathfrak{S}_{m_+}\right)$. For all i such that $p_i > 0$, we have*

i) $\omega(i)_{[1:p_{\leq i-1}]} = \omega(i-1)_{[1:p_{\leq i-1}]}$,

ii) $\omega(i)_{[p_{\leq i-1}+1:p_{\leq i}]}$ *is such that* $x_{i,\omega(i)_{[p_{\leq i-1}+1:p_{\leq i}]}}$ *is the* $\mathcal{X}_i$ *variable of* m_+,

iii) $\omega(i)_{[p_{\leq i}+1:k]} = \{0\}^{k-p_{\leq i}}$.

.

Notice that in the case $i = 1$, $\omega(0)$ is not formally well defined, but since $p_{\leq 0} = 0$, i) is just the equality of two empty strings.

Proof. The point ii) is directly implied from the fact that the positive part of a monomial associated with a map $\sigma \in \mathfrak{S}_{m_+}$ has to be m_+.

Assume that there exists i such that i) or iii) is not satisfied and then let us choose such an i minimal.

Assume first that only iii) is not satisfied for i. We consider

$$\sigma : j \mapsto \begin{cases} \sigma_{m_+}(j) \text{ if } j \neq i \\ \left(\lambda(j), \omega(i)_{[1:p_{\leq i}]} \parallel \{1\}^{k-p_{\leq i}}\right) \text{ if } j = i \end{cases}$$

where $w_1 \parallel w_2$ is the concatenation of w_1 with w_2. Obviously, $\sigma \prec \sigma_{m_+}$ and the positive part of the monomial associated to σ is again m_+. Let us show that σ is non-decreasing. As $\sigma(i) \prec \sigma_{m_+}(i)$, the only thing to check is that $\sigma(i-1) \preccurlyeq \sigma(i)$ when $i > 1$. We already know (since i) is satisfied) that $\omega(i)_{[1:p_{\leq i-1}]} = \omega(i-1)_{[1:p_{\leq i-1}]}$. If $p_{i-1} > 0$, then by minimality of i, we know that $\omega(i-1)_{[p_{\leq i-1}+1:k]} = \{1\}^{k-p_{\leq i-1}}$, and so, $\omega(i-1) \preccurlyeq \omega(i)$. If $p_{i-1} < 0$, then $\lambda(i-1) < \lambda(i)$ and their first common ancestor is $(i-1, p_{\leq i-1})$. It directly implies that $\sigma(i-1) \preccurlyeq \sigma(i)$. Consequently, σ is also in $\mathfrak{S}_{m_+}$ which contradicts the minimality of σ_{m_+}.

Then, assume that i) is not satisfied for i. Clearly, we have $i > 1$. We consider

$$\sigma : j \mapsto \begin{cases} \sigma_{m_+}(j) \text{ if } j \neq i \\ \left(\lambda(j), \omega(i-1)_{[1:p_{\leq i-1}]} \parallel \omega(i)_{[p_{\leq i-1}+1:k]}\right) \text{ if } j = i \end{cases}$$

where again $w_1 \parallel w_2$ is the concatenation of w_1 with w_2.

Since furthermore σ_{m_+} is non-decreasing, we know that $\omega(i-1)_{[1:p_{\leq i-1}]} < \omega(i)_{[1:p_{\leq i-1}]}$ (indeed, either $p_{i-1} < 0$ and $\lambda(i) = \lambda(i-1)$, or $p_{i-1} < 0$ and the first common ancestor in T_p of $\lambda(i)$ and $\lambda(i-1)$ is $(i-1, p_{\leq i-1})$). Consequently, $\sigma \prec \sigma_{m_+}$.

Similarly, σ is also non-decreasing. Indeed, if $p_{i-1} > 0$, then by minimality of i, we know that $\omega(i-1) = \omega(i-1)_{[1:p_{\leq i-1}]} \parallel \{1\}^{k-p_{\leq i-1}} \preccurlyeq \omega(i-1)_{[1:p_{\leq i-1}]} \parallel \omega(i)_{[p_{\leq i-1}+1:k]}$. Otherwise $p_{i-1} < 0$, $\lambda(i-1) \prec \lambda(i)$ and these two leaves have $(i-1, p_{\leq i-1})$ for first common ancestor. In this case $\sigma(i-1) \prec \sigma(i)$ since their length-$p_{\leq i-1}$ prefixes of their second projection are identical. We again conclude that $\sigma \in \mathfrak{S}_{m_+}$ which is a contradiction. □

Finally this last lemma combines previous results to show that the data of the negative part is exactly determined by the positive part and the path p.

Lemma 31. *Let $m_+ \in \mathcal{M}_{>0}$ and $\sigma_{m_+} = (\lambda, \omega) = \min\left(\mathfrak{S}_{m_+}\right)$. For all i such that $p_i < 0$ and for all $1 \leq j \leq p_{\leq i-1}$, we have $\omega(i)_j = \omega(i_1)_j$ where i_1 is the maximal $i' \leq i$ such that $p_{\leq i'-1} < j$. In particular $p_{i_1} > 0$.*

Proof. It immediately follows the fact (implied by Lemmas 30 and 29) that if $\min(p_{\leq i-1}, p_{\leq i}) \geq j$, then $\omega(i-1)_j = \omega(i)_j$. The last step $\omega(i)_j = \omega(i-1)_j$ is ensured since $p_i < 0$. □

Proof of Lemma 28. As $|\mathcal{M}_{>0}| = |\mathcal{M}_{<0}|$, it is enough to prove that θ is an injection. As the path p is fixed, the injectivity should follows from Lemma 31. Indeed, assume that m_{+1} and m_{+2} are two distinct monomials. Let i such that the $\mathcal{X}_i$-variables x_{i,w_1} of m_{+1} and x_{i,w_2} of m_{+2} differ. Let j be an index such that $(w_1)_{j-p_{\leq i-1}} \neq (w_2)_{j-p_{\leq i-1}}$. In particular $1 \leq j \leq p_{\leq i}$. Let $i_2 > i$ minimal such that $p_{\leq i_2} < j$. So, if x_{i_2,w_3} (resp. x_{i_2,w_4}) is the $\mathcal{X}_{i_2}$-variable of $\theta(m_{+1})$ (resp. $\theta(m_{+2})$), then by Lemma 31, $(w_3)_{j-p_{\leq i_2}} = (w_1)_{j-p_{\leq i-1}} \neq (w_2)_{j-p_{\leq i-1}} = (w_4)_{j-p_{\leq i_2}}$, and so $\theta(m_{+1}) \neq \theta(m_{+2})$. □

8 No Girard-Newton identities in positive characteristic

The proof is a consequence of Lucas' theorem (see, e.g. [Mes14]), 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 32 (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*

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

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. [Lu01, Proposition 1] for a similar statement when $p = 2$).

Corollary 33. *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 32), this is the $(k+1)$th least significant digit of w written in base p.

On the other hand, again by Theorem 32, 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 10. Assume that $d = p^k$ and $n \geq d$. For the sake of contradiction, assume that

$$\mathrm{E}_n^d = Q_d(P_1, \ldots, P_m) \tag{11}$$

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 (11) we get the functional equality

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

However, Corollary 33 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. A part of this work was done when the second and third authors were visiting the Simons Institute for the Theory of Computing in spring 2023. The third author is grateful for a research visit sponsored by the Guest researchers faculty program at Université Paris Cité in summer 2022, where the work was initiated.

References

[AGK+23] Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. Low-depth arithmetic circuit lower bounds: Bypassing set-multilinearization. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*, volume 261 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[BKM73] Richard Brent, Daniel Kuck, and Kiyoshi Maruyama. The parallel evaluation of arithmetic expressions without division. *IEEE Transactions on Computers*, 22(05):532–534, 1973.

[DGI+23] Pranjal Dutta, Fulvio Gesmundo, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. De-bordering and geometric complexity theory for waring rank and related models, 2023.

[DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012.

[FLM+23] Hervé Fournier, Nutan Limaye, Guillaume Malod, Srikanth Srinivasan, and Sébastien Tavenas. Towards optimal depth-reductions for algebraic formulas. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 28:1–28:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[FLMS13] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. *Electron. Colloquium Comput. Complex.*, TR13-100, 2013.

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

[HY11] Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011.

[Hya79] Laurent Hyafil. On the parallel evaluation of multivariate polynomials. *SIAM J. Comput.*, 8(2):120–123, 1979.

[JS82] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *Journal of the ACM (JACM)*, 29(3):874–897, 1982.

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

[KS15] Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: It's all about the top fan-in. *SIAM J. Comput.*, 44(6):1601–1625, 2015.

[KS17a] Neeraj Kayal and Chandan Saha. Multi-k-ic depth three circuit lower bound. *Theory Comput. Syst.*, 61(4):1237–1251, 2017.

[KS17b] Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 31:1–31:30. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[KS17c] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.*, 46(1):336–387, 2017.

[KS23] Deepanshu Kush and Shubhangi Saraf. Near-optimal set-multilinear formula lower bounds. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference*, CCC '23, Dagstuhl, DEU, 2023. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[KSS14] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153. ACM, 2014.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 626–632. ACM, 2016.

[LST21a] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 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*, pages 804–814. IEEE, 2021.

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

[LST22] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. On the partial derivative method applied to lopsided set-multilinear polynomials. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Lu01] Chi-Jen Lu. An exact characterization of symmetric functions in $\mathrm{qac}^0[2]$. *Theor. Comput. Sci.*, 261(2):297–303, 2001.

[Mac98] Ian Grant Macdonald. *Symmetric functions and Hall polynomials*, volume 2nd edition. Oxford university press, 1998.

[Mac15] I. G. Macdonald. *Symmetric functions and Hall polynomials*. Oxford Classic Texts in the Physical Sciences. The Clarendon Press, Oxford University Press, New York, second edition, 2015. With contribution by A. V. Zelevinsky and a foreword by Richard Stanley, Reprint of the 2008 paperback edition [MR1354144].

[MB19] Sajal Kumar Mukherjee and Sudip Bera. Combinatorial proofs of the newton–girard and chapman–costas-santos identities. *Discrete Mathematics*, 342(6):1577–1580, 2019.

[Mes14] Romeo Mestrovic. Lucas' theorem: its generalizations, extensions and applications (1878–2014), 2014.

[MP08] Guillaume Malod and Natacha Portier. Characterizing valiant's algebraic complexity classes. *J. Complex.*, 24(1):16–38, 2008.

[MS77] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*, volume 16. Elsevier, 1977.

[MV99] Meena Mahajan and V Vinay. Determinant: Old algorithms, new insights. *SIAM journal on Discrete Mathematics*, 12(4):474–490, 1999.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991.

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

[Raz06] Ran Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(6):121–135, 2006.

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

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

[RY08] Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Comput. Complex.*, 17(4):515–535, 2008.

[Sam42] Paul A Samuelson. A method of determining explicitly the coefficients of the characteristic equation. *The Annals of Mathematical Statistics*, 13(4):424–429, 1942.

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

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010.

[TLS22] Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 416–425. ACM, 2022.

[Val79] Leslie G Valiant. Negation can be exponentially powerful. In *Proceedings of the eleventh annual ACM symposium on theory of computing*, pages 189–196, 1979.

A The connection between non-commutative and commutative homogenization

In this section, we make precise a connection between homogenizing formulas in the non-commutative and commutative settings, hinted at in Section 4.3. This follows easily from a recent result of [DGI+23] along with other standard machinery, so we only sketch the details here.

The following is an immediate consequence of [DGI+23, Proposition 6.4].

Lemma 34. *Let n, d be growing parameters with $d = d(n) \leq n$. Let $A_1, \ldots, A_n$ be 3×3 matrices with distinct variable entries, and define the 3×3 matrix A by*

$$A = \mathrm{E}_n^d(A_1, A_2, \ldots, A_n).$$

where E_n^d is as defined in (1). Let $P_{n,d}$ denote the $(1,2)$th entry of A. Then, $P_{n,d}$ has a homogeneous algebraic formula of size $\mathrm{poly}(n)$ if and only if any degree-d polynomial computed by an algebraic formula of size $\mathrm{poly}(n)$ has a homogeneous algebraic formula of size $\mathrm{poly}(n)$.

The following lemma now makes precise the connection between the complexity of E_n^d in the non-commutative setting and homogenization.

Lemma 35. *Let n, d be growing parameters with $d = d(n) \leq n$. The polynomial E_n^d has a depth-3 non-commutative formula of size $\mathrm{poly}(n)$. Moreover, if E_n^d has a non-commutative homogeneous formula of size $\mathrm{poly}(n)$, then so does $P_{n,d}$ as defined above.*

Proof. The upper bound for E_n^d follows directly from Ben-Or's interpolation argument (see, e.g. [SW01, Theorem 5.1]) which also works in the non-commutative setting.

Now, assume that E_n^d has a non-commutative homogeneous formula F of size $\mathrm{poly}(n)$. By standard depth-reduction results [BKM73] for formulas which also hold in the non-commutative setting, we can assume without loss of generality that F has depth $O(\log n)$ and fan-in at most 2. If we replace each input x_i by the matrix A_i from Lemma 34 above and treat the sum and product gates of F as matrix sums and products respectively, we see that the output of F is the matrix A from Lemma 34.

To get a commutative formula for $P_{n,d}$, we operate directly on the entries of the underlying matrices. In particular, we start with the 9 variable entries of each matrix and compute the matrix sum and product at each sum and product gate of F respectively. This requires replacing each gate of F by a constant-sized circuit (note that each gate of F has fan-in at most 2), resulting in a circuit of depth $O(\log n)$ with gates of fan-in 2 computing each of the entries of the output matrix A. The $(1,2)$th entry of A then gives the polynomial $P_{n,d}$.

As $P_{n,d}$ is computed by a circuit C of depth $O(\log n)$ with gates of fan-in 2, it also has formulas of size $\mathrm{poly}(n)$ (obtained by converting C to a formula via the standard approach of replicating gates). □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il