New Lower Bounds against Homogeneous Non-Commutative Circuits

Prerona Chatterjee
Department of Computer Science, Tel Aviv University, Israel

Pavel Hrubeš
Institute of Mathematics, Czech Academy of Sciences, Prague, Czech Republic

Abstract

We give several new lower bounds on size of homogeneous non-commutative circuits. We present an explicit homogeneous bivariate polynomial of degree d which requires homogeneous non-commutative circuit of size $\Omega(d/\log d)$. For an n-variate polynomial with $n > 1$, the result can be improved to $\Omega(nd)$, if $d \leq n$, or $\Omega(nd\frac{\log n}{\log d})$, if $d \geq n$. Under the same assumptions, we also give a quadratic lower bound for the ordered version of the central symmetric polynomial.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic circuit complexity, Non-Commutative Circuits, Homogeneous Computation, Lower bounds against algebraic circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2023.13

Funding *Prerona Chatterjee*: Partially funded by the Harry Bloomfield Postdoctoral Fellowship. This work was done while the author was a postdoctoral researcher at the Czech Academy of Sciences, Prague, and was funded by Czech Science Foundation GAČR grant 19-27871X.
Pavel Hrubeš: Czech Science Foundation GAČR grant 19-27871X.

Acknowledgements Prerona would like to acknowledge Cafedu for being such a nice place to work from. Pavel thanks Amir Yehudayoff for useful ideas on this topic which were exchanged in distant and joyous past.

1 Introduction

Arithmetic Circuit Complexity aims to categorize polynomials according to how hard they are to compute in algebraic models of computation. The most natural model is that of an arithmetic circuit: a directed acyclic graph with constant or variables as the leaf labels and addition or multiplication as labels of the internal nodes. Therefore, starting from variables or constants at the leaves, the every node in the circuit naturally computes new polynomials by means of addition and multiplication operations. The question is how many of these operations are needed.

The most challenging problem is to prove super-polynomial lower bounds against arithmetic circuits computing a low-degree polynomial. This is known as the VP vs VNP problem and is the algebraic analogue of the famed P vs. NP question. The classical result of Baur and Strassen [13, 1] gives an $\Omega(n \log d)$ lower bound for an n variate polynomial of degree d. A variety of lower bounds has since been obtained by imposing various restrictions on the computational model - e.g., arithmetic formulas[1] [8] or monotone circuits[2] [15]. But the result of Baur and Strassen remains the strongest lower bound on unrestricted arithmetic circuits.

[1] Similar to circuits except that the underlying graph is only allowed to be a tree instead of a DAG.
[2] Similar to circuits except that only non-negative constants are allowed.

38th Computational Complexity Conference (CCC 2023).
Editor: Amnon Ta-Shma; Article No. 13; pp. 13:1–13:10
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we are interested in the non-commutative setting where multiplication does not multiplicatively commute. Starting with the seminal works of Hyafil [7] and Nisan [9], non-commutative circuits are a well-studied object. The lack of commutativity is a severe limitation on the computational power which makes the task of proving circuit lower bounds seemingly easier. Nisan gave an exponential lower bound for non-commutative formulas whereas, commutatively, the best bound is only quadratic [8, 4]. Since then, it seemed that exponential non-commutative circuit lower bounds are just around the corner. Recently, Limaye, Srinivasan and Tavenas [14] proved such a lower bound in the *homogeneous, constant depth* setting for a polynomial that can be computed efficiently by non-commutative ABPs[3]. They showed that any constant depth Δ non-commutative homogeneous circuit for the *iterated matrix multiplication polynomial* (a polynomial over n variables of degree d must have size $n^{\Omega(d^{\frac{1}{\Delta}})}$. However for general circuits, even in the non-commutative setting, the strongest lower bound remains $\Omega(n \log d)$.

We improve this lower bound to $\Omega(nd/\log d)$ under the assumption that the non-commutative circuit is additionally homogeneous (see Section 2 for definition). Non-commutatively, this is already interesting if $n = 2$: we obtain a bivariate polynomial of degree d which requires circuit size nearly linear in d. It is well-known that a (commutative or not) circuit computing a homogeneous polynomial of degree d can be converted to an equivalent homogeneous circuit with at most a d^2 increase in size (see, for example, [6]). Hence, homogeneity is not a serious restriction if either d is small or if one proves a super-polynomial lower bound. However, our results fall in neither category and we do not know how to remove the homogeneity restriction. Furthermore, Carmosino et al. [3] have shown that strong enough superlinear lower bounds can be amplified to truly exponential ones. Unfortunately, the parameters of our result are not sufficient to allow amplification. Nevertheless, we strongly believe that it can be removed and that stronger non-commutative circuit lower bounds are just around the corner.

2 Notation and preliminaries

Let $\mathbb{F}$ be a field. A *non-commutative polynomial* over $\mathbb{F}$ is a formal sum of products of variables and field elements. We assume that the variables do not multiplicatively commute, whereas they commute additively and with elements of $\mathbb{F}$. The ring of non-commutative polynomials in variables $x_1, \ldots, x_n$ is denoted $\mathbb{F}\langle x_1, \ldots, x_n \rangle$. A polynomial is said to be *homogeneous* if all monomials with a non-zero coefficient in f have the same degree.

A *non-commutative arithmetic circuit* $\mathcal{C}$ over the field $\mathbb{F}$ is a directed acyclic graph as follows. Nodes (or gates) of in-degree zero are labelled by either a variable or an element in the field $\mathbb{F}$. All the other nodes have in-degree two and they are labelled by either $+$ or $\times$. The two edges going into a gate labelled by $\times$ are labelled by *left* and *right* to indicate the order of multiplication. Gates of in-degree zero will be called *input* gates; gates of out-degree zero will be called *output* gates.

Every node in $\mathcal{C}$ computes a non-commutative polynomial in the obvious way. We say that $\mathcal{C}$ computes a polynomial f if there is a gate in $\mathcal{C}$ computing f (not necessarily an output gate). $\mathcal{C}$ will be called *homogeneous* if every gate in $\mathcal{C}$ computes a homogeneous polynomial. Given a circuit $\mathcal{C}$, let $\widehat{\mathcal{C}} := (f \;:\; f$ is computed by some gate in $\mathcal{C})$.

A product gate will be called *non-scalar*, if both of its inputs compute a non-constant

[3] An algebraic computational model whose power lies in between that of circuits and formulas.

polynomial. We define the *size* of $\mathcal{C}$ to be the number of non-input gates in it, and the *non-scalar size* of $\mathcal{C}$ to be the number of non-scalar product gates in it.

Given integers n_1, n_2, $[n_1, n_2]$ is the interval $\{n_1, n_1 + 1, \ldots, n_2\}$ and $[n] := [1, n]$.

Note: Unless stated otherwise, circuits and polynomials are assumed to be non-commutative and the underlying field $\mathbb{F}$ is fixed but arbitrary.

3 Main results

For univariate polynomials there is no difference between commutative and non-commutative computations. Already with two variables, non-commutative polynomials display much richer structure. There are 2^d monomials in variables x_0, x_1 of degree d (as opposed to $d + 1$ in the commutative world); so a generic bivariate polynomial requires a circuit of size exponential in d.

Our first result is a lower bound that is almost linear in d. The hard polynomial is a bivariate monomial (a specific product of variables x_0, x_1).

▶ **Theorem 1.** *For every $d > 1$, there exists an explicit bivariate monomial of degree d such that any homogeneous non-commutative circuit computing it has non-scalar size $\Omega(d/\log d)$.*

In Remark 10, we point out a complementary $O(d/\log d)$ upper bound for every bivariate monomial. Note that commutatively every such monomial can be computed in size $O(\log d)$.

For n-variate polynomials, we obtain a stronger result (the hard polynomial is no longer a monomial).

▶ **Theorem 2.** *For every $n, d > 1$ there exists an explicit n-variate homogeneous polynomial of degree d which requires a homogenous non-commutative circuit of non-scalar size $\Omega(nd)$, if $d \leq n$, or $\Omega(nd\frac{\log n}{\log d})$, if $d \geq n$.*

Theorem 1 and Theorem 2 are proved in Sections 4.1 and 4.2 respectively.

Given $0 \leq d, n$, the *ordered symmetric polynomial*, OS_n^d, is the polynomial[4]

$$\mathrm{OS}_n^d(x_1, \ldots, x_n) = \sum_{1 \leq i_1 < \cdots < i_d \leq n} \left(\prod_{j=1}^{d} x_{i_j} \right) .$$

It can be thought of as an ordered version of the commutative elementary symmetric polynomial. In Section 5, we shall prove a lower bound for this polynomial.

▶ **Theorem 3.** *If $2 \leq d \leq n/2$, any homogeneous non-commutative circuit computing $\mathrm{OS}_n^d(x_1, \ldots, x_n)$ must have non-scalar size $\Omega(dn)$.*

For the central ordered symmetric polynomial $\mathrm{OS}_n^{\lfloor n/2 \rfloor}$, the lower bound becomes $\Omega(n^2)$. We also observe that the known commutative upper bounds on elementary symmetric polynomials work non-commutatively as well.

▶ **Proposition 4.** *$\mathrm{OS}_n^1, \ldots, \mathrm{OS}_n^n$ can be simultaneously computed by a non-commutative circuit of size $O(n \log^2 n \log\log n)$, and by a homogeneous non-commutative circuit of size $O(n^2)$.*

[4] Hence $\mathrm{OS}_n^0 = 1$ and $\mathrm{OS}_n^d = 0$ whenever $d > n$.

The polylog factor in the proposition depends on the underlying field and can be improved for some $\mathbb{F}$s. Moreover, when measuring non-scalar size, one can obtain an $O(n \log n)$ upper bound if $\mathbb{F}$ is infinite – this is tight by [1].

The ordered symmetric polynomial can be contrasted with the truly symmetric polynomial

$$S_n^k = \sum_{1 i_1, \dots, i_k \in [n] \text{ distinct}} x_{i_1} \cdots x_{i_k} \,,$$

Non-commutatively, already S_n^n is as hard as the permanent [6] and is expected to require exponential circuits.

▶ Remark 5. A polynomial of degree d can be uniquely written as $f = \sum_{k=0}^{d} f^{(k)}$ where $f^{(k)}$ is homogeneous of degree k. It is well-known that if f has a circuit of size s, the homogeneous parts $f^{(0)}, \dots, f^{(d)}$ can be simultaneously computed by a homogeneous circuit of size $O(sd^2)$ (this holds non-commutatively as well [6]). Note that $\mathrm{OS}_n^0, \dots, \mathrm{OS}_n^n$ are the homogeneous parts of $\prod_{i=1}^{n}(1 + x_i)$ which has a circuit of a linear size. Theorem 3 shows that in this case, homogenization provably costs a factor of the degree.

4 Lower bounds against homogeneous non-commutative circuits

Let us define the measure we use to prove our lower bounds. Suppose $f \in \mathbb{F}\langle x_1, \dots, x_n \rangle$ is a homogeneous polynomial of degree d. Given an interval $J = [a, b] \subseteq [d]$, the polynomial f^J is obtained be setting variables in position *outside* of J to one. More precisely, if $\alpha = \prod_{i=1}^{d} x_{j_i}$ is a monomial then $\alpha^J := \prod_{i=a}^{b} x_{j_i}$, and the map is extended linearly so that $f^J = \sum_k c_k \alpha_k^J$ whenever $f = \sum_k c_k \alpha_k$. Given a non-negative integer ℓ, let

$$\mathcal{F}_\ell(f) = \left(f^J \;:\; J \subseteq [d] \text{ is an interval of length } \ell \right).$$

Given homogeneous polynomials $f_1, \dots, f_m$, our hardness measure is defined as

$$\mu_\ell(f_1, \dots, f_m) := \dim(\mathsf{span}(\bigcup_{i=1}^{m} \mathcal{F}_\ell(f_i)))\,.$$

Here, $\mathsf{span}(\mathcal{F})$ denotes the vector space of $\mathbb{F}$-linear combinations of polynomials in $\mathcal{F}$ and dim is its dimension.

The following lemma bounds the measure in terms of circuit size.

▶ **Lemma 6.** *Let $\mathcal{C}$ be a homogeneous circuit with s non-scalar multiplication gates. Then for every $\ell \geq 2$, $\mu_\ell(\widehat{\mathcal{C}}) \leq (\ell - 1)s$.*

Proof. This is by induction on the size of $\mathcal{C}$. If $\mathcal{C}$ consists of input gates only then $\mathcal{F}_\ell(\widehat{\mathcal{C}}) = \emptyset$, as we assumed $\ell \geq 2$ and $\widehat{\mathcal{C}}$ consists of linear polynomials.

Otherwise, assume that u is some output gate of $\mathcal{C}$ and let $\mathcal{C}'$ be the circuit obtained by removing that gate. If u is a sum gate or a scalar product gate then

$$\mu_\ell(\widehat{\mathcal{C}}) \leq \mu_\ell(\widehat{\mathcal{C}'})\,.$$

For if u computes f then $f = a_1 f_1 + a_2 f_2$ for some constants a_1, a_2 and $f_1, f_2 \in \widehat{\mathcal{C}'}$. If f has degree d then for every interval $J \subseteq [d]$ of length ℓ, $f^J = (a_1 f_1 + a_2 f_2)^J = a_1 f_1^J + a_2 f_2^J \in \mathsf{span}(\mathcal{F}_\ell(\widehat{\mathcal{C}'}))$.

If u is a non-scalar product gate computing $f = f_1 \cdot f_2$ then

$$\mu_\ell(\widehat{\mathcal{C}}) \leq \mu_\ell(\widehat{\mathcal{C}'}) + (\ell - 1)\,.$$

To see this assume f_1, f_2 have degrees d_1 and d_2 respectively, and let $J \subseteq [d_1 + d_2]$ be an interval of length ℓ. If J is contained in $[d_1]$, $f^J = (f_1 f_2)^J = f_1^J f_2^{\emptyset}$ is a scalar multiple of f_1^J and hence f^J is contained in $\mathsf{span}(\mathcal{F}_\ell(\widehat{\mathcal{C}'}))$; similarly if J is contained in $[d_1 + 1, d_2]$. Otherwise, both d_1 and $d_1 + 1$ are contained in J. But there are only $\ell - 1$ such intervals. Hence $\mathcal{F}_\ell(\widehat{\mathcal{C}})$ contains at most $\ell - 1$ polynomials outside of $\mathsf{span}(\mathcal{F}_\ell(\widehat{\mathcal{C}'}))$.

This means that μ_ℓ increases only at product gates, and that it increases only by $\ell - 1$ at such gates. Hence $\mu_\ell(\widehat{\mathcal{C}}) \leq (\ell - 1)s$. ◀

▶ Remark 7. If f has n variables and degree d, the measure $\mu_\ell(f)$ can be at most the minimum of $d - (\ell - 1)$ and n^ℓ. Hence, Lemma 6 can by itself give a lower of at most the order of $d \log n / \log d$.

4.1 Lower bounds for a single monomial

Interestingly, Lemma 6 gives non-trivial lower bounds for f being merely a product of variables (that is, monomials), namely lower bounds of the form $\tilde{\Omega}(d)$ for a monomial of degree d. The simplest example is for an n-variate monomial of degree n^2.

▶ **Proposition 8.** *Every homogeneous circuit computing $f = \prod_{i=1}^n \prod_{j=1}^n (x_i x_j)$ contains at least n^2 non-scalar product gates.*

Proof. This is an application of Lemma 6 with $\ell = 2$. The family $\mathcal{F}_2(f)$ consists of all monomials $x_i x_j$. Hence, $\mu_2(f) = n^2$. If $\mathcal{C}$ computes f, we have $\mu_2(\widehat{\mathcal{C}}) \geq \mu_2(f)$ and hence $\mathcal{C}$ contains at least n^2 product gates. ◀

Another case of interest is a monomial in two variables, x_0, x_1, of degree d. Suppose $f = \prod_{i=1}^d x_{\sigma_i}$ where $\sigma = (\sigma_1, \ldots, \sigma_d) \in \{0,1\}^d$. Then $\mu_\ell(f)$ equals the number of distinct substrings of σ of length ℓ. Hence we want to find a σ which contains as many substrings as possible. One construction of such an object is provided by the *de Bruijn sequence* [5].

de Bruijn sequences

For a given k, a de Bruijn sequence of order k over alphabet A is a cyclic sequence σ in which every k-length string from A^k occurs exactly once as a substring. Note that σ must have length $|A|^k$. Furthermore, precisely $k - 1$ of the substrings overlap the beginning and the end of the sequence and σ contains $|A|^k - (k - 1)$ substrings when viewed as an ordinary sequence. de Bruijn sequences are widely studied and, in particular, they exist. Moreover, efficient algorithms are known for constructing de Bruijn sequences (see, for example, [11] and its references). In the case of binary alphabet $A = \{0, 1\}$, this is especially so. We can start with a string of k zeros. At each stage, extend the sequence by 1, unless this results in a k-string already encounters, otherwise extend by 0.

Given $d \geq 2$, let σ be a binary de Bruijn sequence of order $\lceil \log_2 d \rceil$. It has length $2^{\lceil \log_2 d \rceil} \geq d$. Define the polynomial

$$\mathsf{B}_d(x_0, x_1) := \prod_{i=1}^d x_{\sigma_i} \,.$$

The following implies the result of Theorem 1.

▶ **Proposition 9.** *Every homogeneous circuit computing B_d contains $\Omega(d / \log d)$ non-scalar product gates.*

Proof. This is an application of Lemma 6 with $\ell = \lceil \log_2 d \rceil$. $[d]$ contains $d - \ell - 1$ intervals of length ℓ, all of which give rise to different substrings of σ. The family $\mathcal{F}_\ell(\mathsf{B}_d)$ consists of $d - (\ell - 1)$ different monomials and hence $\mu_\ell(\mathsf{B}_d) = d - (\ell - 1)$. By the lemma, assuming $\ell > 1$, a homogenous circuit for B_d must contain $(d - (\ell - 1))/(\ell - 1) = \Omega(d/\log d)$ product gates. ◀

▶ Remark 10. Using de Bruijn sequences over alphabet of size n, one can give an explicit monomial in $n > 1$ variables and degree $d \geq n$ which requires homogeneous circuit of non-scalar size $\Omega(d \log n / \log d)$. This can also be deduced from Proposition 9 by viewing degree k bivariate monomials as a single variable.

Conversely, every such monomial α can be computed in size $O(d \log n / \log d)$ using multiplication gates only (such a computation is automatically homogeneous). Indeed, we can first compute all monomials of degree at most k by a circuit of size $O(n^{k+1})$ and then compute α using $\lceil d/k \rceil$ additional multiplication gates. Choosing k around $0.5 \log_2 d \log_2^{-1} n$ is sufficient. This also means that the bound in Theorem 2 is tight.

4.2 Computing partial derivatives simultaneously

In order to obtain stronger lower bounds, we will translate the classical theorem of Baur and Strassen [1] on computing partial derivatives to the non-commutative setting.

We define partial derivative with respect to first position only, as follows. Given a polynomial f and a variable x, f can be uniquely written as $f = xf_0 + f_1$ where no monomial in f_1 contains x in the first position. We set $\partial_x f := f_0$.

The proof of the following lemma is almost the same as the one due to Baur and Strassen. The only additional subtlety is that we need the derivatives to be computed by a homogeneous circuit. This requires the generalization of homogeneity to allow arbitrary variable weights. We emphasize that taking derivatives with respect to the first position is essential in the non-commutative setting.

▶ **Lemma 11.** *Assume that $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ can be computed by a homogeneous circuit of size s and non-scalar size $s_\times$. Then $\partial_{x_1} f, \ldots, \partial_{x_n} f$ can be simultaneously computed by a homogeneous circuit of size $O(s)$ and non-scalar size $O(s_\times)$.*

Proof. Given $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$, let w_i be the *weight* of x_i and let the weight of a monomial $\alpha = \prod_{j=1}^d x_{i_j}$ be defined as $\mathsf{wt}(\alpha) = \sum_{j=1}^d w_{i_j}$. A polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is said to be $\mathbf{w}$-homogeneous if every monomial in it has the same weight. We call this the weight of f, denoted by $\mathsf{wt}(f)$. Furthermore we say that a circuit $\mathcal{C}$ is $\mathbf{w}$-homogeneous if every gate in it computes a $\mathbf{w}$-homogeneous polynomial. The weight of any node, v, in a $\mathbf{w}$-homogeneous circuit is defined to be the weight of the polynomial being computed by it.

Note that if $(w_1, \ldots, w_n) = (1, \ldots, 1)$, then $\mathbf{w}$-homogeneity coincides with the usual notion of homogeneity. Therefore Lemma 11 follows from the following claim.

▷ Claim 12. For any $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$, if there is a $\mathbf{w}$-homogenous circuit that computes $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ of size s and non-scalar size $s_\times$, then there is a $\mathbf{w}$-homogeneous circuit that computes $\mathbb{D}(f) = \{\partial_{x_1} f, \ldots, \partial_{x_n} f\}$ of size at most $5s$ and non-scalar size at most $2s_\times$.

We prove this claim by induction on s. Recall that circuit size is measured by the number of non-input gates. For the base case, $s = 0$, the circuit only consists of leaves. The derivatives are then either 0 or 1 and can again be computed in zero size.

Assume $s > 0$. Let $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$ be arbitrarily fixed. Furthermore, suppose there is a $\mathbf{w}$-homogenous circuit $\mathcal{C}$ that computes $f \in \mathbb{F}\langle x_1, \ldots, x_n\rangle$ of size s. Choose a vertex v in $\mathcal{C}$ such that both its children are leaves, and let $\widehat{v}$ be the polynomial it computes. $\widehat{v}$ is a homogeneous polynomial in at most two variables and degree at most two; w.l.o.g., we can also assume that $\widehat{v}$ is at least linear (otherwise v could be replaced by a leaf).

Let $\mathcal{C}'$ be the circuit obtained from $\mathcal{C}$ by removing the incoming edges to v and labelling the vertex v with a new variable, say x_0. Let us assign it weight $w_0 := \mathsf{wt}(\widehat{v})$.

Let f' be the polynomial computed by $\mathcal{C}'$. Then, $\mathbb{D}(f) = \{\partial_{x_1} f, \ldots, \partial_{x_n} f\}$ can be recovered from $\mathbb{D}(f') = \{\partial_{x_0} f', \partial_{x_1} f', \ldots, \partial_{x_n} f'\}$ using the following version of chain rule:

$$\partial_{x_k} f = (\partial_{x_k} f' + \partial_{x_k}\widehat{v} \cdot \partial_{x_0} f')|_{x_0 := \widehat{v}} .$$

Note that $\partial_{x_k}\widehat{v}$ is a variable or a constant, and that it is zero except for at most two of the x_k's.

Let us set $\mathbf{w}' = (w_0', w_1, \ldots, w_n)$. Note that the weight of every vertex in $\mathcal{C}'$ is the same as the corresponding vertex in $\mathcal{C}$. Therefore, since $\mathcal{C}$ is $\mathbf{w}$-homogeneous, $\mathcal{C}'$ is $\mathbf{w}'$-homogeneous. Furthermore, $\mathcal{C}'$ has $s-1$ non-input gates and, by the inductive assumption, there is a $\mathbf{w}'$-homogeneous circuit $\mathcal{D}'$ of size $5(s-1)$ which computes $\mathbb{D}(f')$. Using $\mathcal{D}'$ and the chain rule above, we can construct a circuit with 5 additional gates which computes $\mathbb{D}(f)$. The size of this circuit is at most $5(s-1)+5 = 5s$ and is easily seen to be $\mathbf{w}$-homogeneous.

When counting non-scalar complexity, note that in the construction, only non-scalar product gates introduce non-scalar gates, and we always introduce at most two such gates. ◀

We can now prove Theorem 2.

Proof of Theorem 2. Let n, d be given with[5] $n > 1$, $d > 2$. Let k be the smallest integer such that $n^k \geq n(d-1)$. Take a de Bruijn sequence σ of order k in alphabet $[n]$. Take sequences $\sigma^1, \ldots, \sigma^n \in [n]^{d-1}$ so that their concatenation $\sigma^1 \ldots \sigma^n$ is the initial segment of σ. Define the polynomial

$$f = x_1\alpha_1 + \cdots + x_n\alpha_n \,, \text{ where } \alpha_i = \prod_{j=1}^{d-1} x_{\sigma^i_j} \,.$$

Assume f has a homogeneous circuit of non-scalar size s. Then, by Lemma 11, $\alpha_1, \ldots, \alpha_n$ can be simultaneously computed by a homogeneous circuit of size $s' = O(s)$. We now apply Lemma 6 with $\ell = k$. By construction, $\mu_k(\alpha_1, \ldots, \alpha_n) = n(d-1-(k-1)) = n(d-k)$. This is because α_i^J are distinct monomials for different i's and intervals of length k. The lemma then gives $s' \geq n(d-k)/(k-1)$. If $d \leq n$, we have $k = 2$ and so $s' \geq n(d-2)$. If $d > n$, we have $k \leq c_1 \log_2 d / \log_2 n$ and $d - k \geq c_2 d$, for some constants $c_1, c_2 > 0$. Hence indeed $s' \geq \Omega(nd\frac{\log n}{\log d})$. ◀

4.3 Lower bound for ordered symmetric polynomials

We now prove Theorem 3. Firstly, we note the following.

▶ Remark 13. OS_n^2 requires $\Omega(n)$ non-scalar product gates (even in the commutative setting). This can be proved by a standard partial derivatives argument as in [10].

Hence we can focus on degree $d > 2$, in which case we give the following strengthening of Theorem 3:

[5] If $d = 2$, OS_n^2 satisfies the theorem; see Remark 13.

▶ **Theorem 14.** *If $1 < k < n$, any homogeneous circuit computing $\mathrm{OS}_n^{k+1}(x_1, \ldots, x_n)$ requires non-scalar size $\Omega(k(n-k))$.*

Proof. Assume that a homogeneous circuit computes $f = \mathrm{OS}_n^{k+1}(x_1, \ldots, x_n)$ using s non-scalar product gates. Then by Lemma 11 there is a homogeneous circuit of non-scalar size $O(s)$ which simultaneously computes $\{\partial_{x_1} f, \ldots, \partial_{x_n} f\}$. Let this circuit be $\mathcal{C}$. Then, by Lemma 6, $\mu_2(\widehat{\mathcal{C}}) \leq O(s)$. Note that

$$\partial_{x_i} f = \mathrm{OS}_{n-i}^k(x_{i+1}, \ldots, x_n)\,.$$

Let $f_{i,j} := (\partial_{x_i} f)^{[j,j+1]}$. We claim that the polynomials in $F := (f_{i,j} \;:\; i \in [n-k], j \in [k-1])$ are linearly independent. This implies that $\mu_2(\widehat{\mathcal{C}}) \geq (n-k)(k-1)$ and gives a lower bound of $\Omega(k(n-k))$ as required.

We now prove that F is indeed linearly independent. Consider the lexicographic ordering on $S := [n-k] \times [k-1]$ defined by:

$$(i_0, j_0) < (i, j) \text{ iff } (j_0 > j) \text{ or } (j_0 = j \text{ and } i_0 < i)\,.$$

Let $(i_0, j_0) \in S$ be given. Denote $\delta_{i_0,j_0}(g)$ the coefficient of the monomial $x_{i_0+j_0} x_{n+j_0-k+1}$ in g. Then for every $(i,j) \in S$,

$$\delta_{i_0,j_0}(f_{i,j}) = \begin{cases} 1 & \text{if } (i_0, j_0) = (i,j) \\ 0 & \text{if } (i_0, j_0) < (i,j)\,. \end{cases} \tag{1}$$

To see (1), assume that $\partial_{x_i} f$ contains x_{n+j_0-k+1} in position $j+1$ in some monomial α with a non-zero coefficient. The degree of α is k, and the positions $j+1, \ldots, k$ need to be filled with variables from $x_{n+j_0-k+1}, \ldots, x_n$ in an ascending order. There are $k-j$ such positions and $k - j_0$ such variables. Therefore $j \geq j_0$. Furthermore, if $j = j_0$, the last $k - j_0$ positions in α are uniquely determined as the variables $x_{n+j_0-k+1}, \ldots, x_n$ in that order. Similarly, if $\partial_{x_i} f$ contains $x_{i_0+j_0}$ in position j_0 in some α, the first j_0 positions must be filled with variables from $x_{i+1}, \ldots, x_{i_0+j_0}$. Hence $i \leq i_0$, and in case of equality, the first j_0 positions are uniquely determined. This means that $\delta_{i_0,j_0}(f_{i,j}) = 0$ whenever $(i_0, j_0) < (i,j)$. Furthermore, $\alpha := \prod_{p=i_0+1}^{i_0+j_0} x_p \prod_{p=n+j_0-k+1}^{n} x_p$ is the unique monomial in f_{i_0,j_0} with $\delta_{i_0,j_0}(\alpha) = 1$, concluding (1).

Finally, assume for the sake of contradiction that there exists a non-trivial linear combination

$$\sum_{(i,j) \in S} \gamma_{i,j} f_{i,j} = 0\,.$$

Let (i_0, j_0) be the first pair in the lexicographic ordering with $\gamma_{i_0,j_0} \neq 0$. Then we have

$$0 = \sum_{(i,j) \in S} \gamma_{i,j} \delta_{i_0,j_0}(f_{i,j}) = \gamma_{i_0,j_0} \delta_{i_0,j_0}(f_{i_0,j_0}) + \sum_{(i,j) > (i_0,j_0)} \gamma_{i,j} \delta_{i_0,j_0}(f_{i,j})\,.$$

Using (1), the last sum is zero and $\gamma_{i_0,j_0} \delta_{i_0,j_0}(f_{i_0,j_0}) = \gamma_{i_0,j_0} = 0$, contrary to the assumption $\gamma_{i_0,j_0} \neq 0$. ◀

5 Upper bounds for ordered symmetric polynomials

In Proposition 4, we promised upper bounds on the complexity of elementary symmetric polynomials. The promise we now fulfil.

A quadratic upper bound in the homogeneous setting

We want to show that for $d \in \{0, \dots, n\}$, OS_n^d can be simultaneously computed by a homogeneous circuit of size $O(n^2)$.

Note that

$$\mathrm{OS}_n^d(x_1, \dots, x_n) = \mathrm{OS}_{n-1}^{d-1}(x_1, \dots, x_{n-1}) \cdot x_n + \mathrm{OS}_{n-1}^d(x_1, \dots, x_{n-1}).$$

Hence, once we have computed OS_{n-1}^d, $d \in \{0, \dots, n-1\}$, we can compute OS_n^d, $d \in \{0, \dots, n\}$ using $O(n)$ extra gates. The overall complexity is quadratic.

An almost linear upper bound in the non-homogeneous setting

We want to show that OS_n^d, $d \in \{0, \dots, n\}$, can be simultaneously computed by a non-commutative circuit of size $n \cdot \mathrm{poly}(\log n)$.

The proof is the same as its commutative analog for elementary symmetric polynomials, see [1] or the monograph by Burgisser et al. [2, Chapters 2.1-2.3].

The main observation is that polynomial multiplication can be done efficiently. Let

$$f = \sum_{i=0}^{n} y_i t^i, \qquad g = \sum_{i=0}^{n} z_i t^i,$$

where $f, g \in \mathbb{F}\langle y_0, \dots, y_n, z_0, \dots, z_n \rangle [t]$. In other words, we assume that t commutes with otherwise non-commuting variables $y_0, \dots, y_n, z_0, \dots, z_n$. We view f, g as univariate polynomials in the variable t with non-commutative coefficients. Then $fg = \sum_{i=0}^{2n} c_i t^i$ with $c_i = \sum_{j=0}^{i} y_j z_{i-j}$. Commutatively, the polynomials $c_0, \dots, c_{2n}$ can be simultaneously computed by a small circuit. Indeed, if $\mathbb{F}$ contains sufficiently many roots of unity, one can obtain an $O(n \log n)$ circuit using Fast Fourier Transform; in other fields there are modification giving a circuit of size $O(n \log n \log \log n)$ see [12, 2]. When counting only non-scalar product gates, this can be improved to $O(n)$ if $\mathbb{F}$ is sufficiently large. We observe that the same holds if the coefficients of f, g do not commute. This is because the polynomials c_k are bilinear in $y_0, \dots, y_n, z_0, \dots, z_n$. Commutativity does not make a difference in this case (an exercise).

Now consider the polynomial $h_n(t) = \prod_{i=1}^{n} (x_i + t) \in \mathbb{F}\langle x_1, \dots, x_n \rangle [t]$. Then one can see that $\mathrm{OS}_n^d(x_1, \dots, x_n)$ is the coefficient of t^{n-d} in $h(t)$. The coefficients can be be recursively computed by first computing $\prod_{i=1}^{\lceil n/2 \rceil} (x_i + t)$, $\prod_{i=\lceil n/2 \rceil + 1}^{n} (x_i + t)$, and then combining the two by means of the fast polynomial multiplication above. This gives the claimed complexity.

6 Open problems

We end with two open problems.

▶ **Open Problem 1.** *Find an explicit bivariate polynomial of degree d which requires non-commutative homogeneous circuit of size superlinear in d*

▶ **Open Problem 2.** *Given a non-commutative monomial α, can addition gates help to compute α?*

Observe that the bounds obtained in this paper are barely linear in d. Problem 1 simply asks for a quantitative improvement. A circuit with no addition gates is automatically homogeneous – hence a negative answer to Problem 2 would allow to remove the homogeneity assumption in Theorem 1.

References

1. Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.
2. Peter Bürgisser, Michael Clausen, and M Amin Shokrollahi. Algebraic complexity theory, with the collaboration of thomas lickteig. *Grundlehren der Mathematischen Wissenschaften*, 315, 1997.
3. Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin. Hardness amplification for non-commutative arithmetic circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
4. Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. Quadratic lower bounds for algebraic branching programs and formulas. *Comput. Complex.*, 31(2):8, 2022.
5. N.G. de Bruijn. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, 49(7):758–764, 1946.
6. Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. Non-commutative circuits and the sum-of-squares problem. *J. Amer. Math. Soc.*, 24(3):871–898, 2011.
7. Laurent Hyafil. The power of commutativity. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 171–174. IEEE Computer Society, 1977.
8. K. Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985.
9. 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.
10. Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complex.*, 6(3):217–234, 1997.
11. Joe Sawada, Aaron Williams, and Dennis Wong. Generalizing the classic greedy and necklace constructions of de bruijn sequences and universal cycles. *Electron. J. Comb.*, 23(1):1, 2016.
12. Arnold Schönhage and Volker Strassen. Schnelle multiplikation großer zahlen. *Computing*, 7(3-4):281–292, 1971.
13. Volker Strassen. Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973.
14. 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.
15. Leslie G. Valiant. Negation can be exponentially powerful. *Theor. Comput. Sci.*, 12:303–314, 1980.