Electronic Colloquium on Computational Complexity, Report No. 180 (2018)

LOWER BOUNDS FOR ARITHMETIC CIRCUITS VIA THE HANKEL MATRIX

NATHANAËL FIJALKOW, GUILLAUME LAGARDE, PIERRE OHLMANN, AND OLIVIER SERRE

ABSTRACT. We study the complexity of representing polynomials by arithmetic circuits in both the commutative and the non-commutative settings. Our approach goes through a precise understanding of the more restricted setting where multiplication is not associative, meaning that we distinguish $(xy)z$ from $x(yz)$.

Our first and main conceptual result is a characterization result: we show that the size of the smallest circuit computing a given non-associative polynomial is exactly the rank of a matrix constructed from the polynomial and called the Hankel matrix. This result applies to the class of all circuits in both commutative and non-commutative settings, and can be seen as an extension of the seminal result of Nisan giving a similar characterization for non-commutative algebraic branching programs.

The study of the Hankel matrix provides a unifying approach for proving lower bounds for polynomials in the (classical) associative setting. We demonstrate this by giving alternative proofs of recent results proving superpolynomial and exponential lower bounds for different classes of circuits as corollaries of our characterization result.

Our main technical contribution is to provide generic lower bound theorems based on analyzing and decomposing the Hankel matrix. This yields significant improvements on lower bounds for circuits with many parse trees, in both (associative) commutative and non-commutative settings. In particular in the non-commutative setting we obtain a tight result showing superpolynomial lower bounds for any class of circuits which has a small defect in the exponent of the total number of parse trees.

Keywords. Arithmetic Circuit Complexity, Lower Bounds, Parse Trees, Hankel Matrix

(Nathanaël Fijalkow) CNRS, LaBRI, Bordeaux, France, and The Alan Turing Institute of data science, London, United Kingdom

(Guillaume Lagarde) KTH Royal Institute of Technology, Stockholm, Sweden

(Pierre Ohlmann) IRIF (Université Paris Diderot & CNRS), Paris, France

(Olivier Serre) CNRS, IRIF (Université Paris Diderot & CNRS), Paris, France

E-mail addresses: `Nathanael.Fijalkow@labri.fr`, `glagarde@kth.se`, `Pierre.Ohlmann@irif.fr`, `Olivier.Serre@cnrs.fr`.

Date: November 3, 2018.

ISSN 1433-8092

1. Introduction

The model of arithmetic circuits is the algebraic analogue of Boolean circuits: the latter computes Boolean functions and the former computes polynomials, replacing OR gates by addition and AND gates by multiplication. Computational complexity theory is concerned with understanding the expressive power of such models. A rich theory investigates the algebraic complexity classes **VP** and **VNP** introduced by Valiant [Valiant, 1979]. A widely open problem in this area of research is to explicitly construct hard polynomials, meaning for which we can prove super polynomial lower bounds. To this day the best general lower bounds for arithmetic circuits were given by Baur and Strassen [Baur and Strassen, 1983] for the polynomial $\sum_{i=1}^{n} x_i^d$, which requires $\Omega(n \log d)$ operations.

1.1. Non-Commutative Computations.

The seminal paper of Nisan [Nisan, 1991] initiated the study of non-commutative computation: in this setting variables do not commute, and therefore xy and yx are considered as being two distinct monomials. Non-commutative computations arise in different scenarios, the most common mathematical examples being when working with algebras of matrices, group algebras of non-commutative groups or the quaternion algebra. A second motivation for studying the non-commutative setting is that it makes it easier to prove lower bounds which can then provide powerful ideas for the commutative case. Indeed, commutativity allows a circuit to rely on cancellations and to share calculations across different gates, making them more complicated to analyze.

The main result of Nisan [Nisan, 1991] is to give a characterization of the smallest algebraic branching program (ABP) computing a given polynomial. As a corollary of this characterization Nisan obtains exponential lower bounds for the non-commutative permanent against the subclass of circuits given by ABPs.

We sketch the main ideas behind Nisan's characterization, since our first contribution is to extend these ideas to the class of all non-associative circuits. An ABP is a layered graph with two distinguished vertices, a source and a target. The edges are labelled by affine functions in a given set of variables. An ABP computes a polynomial obtained by summing over all paths from the source to the target, with the value of a path being the multiplication of the affine functions along the traversed edges. Fix a polynomial f, and define following Nisan a matrix N_f whose rows and columns are indexed by monomials: for u, v two monomials, let $N_f(u, v)$ denote the coefficient of the monomial $u \cdot v$ in f.

The beautiful and surprisingly simple characterization of Nisan states that for a homogeneous (i.e., all monomials have the same degree) non-commutative polynomial f the size of the smallest ABP computing f is exactly the rank of N_f. The key idea is that the computation of the polynomial in an ABP can be split into two parts: let r be a vertex in an ABP $\mathcal{C}$ computing the polynomial f, then we can split $\mathcal{C}$ into two ABPs, one with the original source and target r and the other one with source r and the original target. We let L_r and R_r denote the polynomials computed by these two ABPs. For u, v two monomials, we observe that the coefficient of uv in f is equal to $\sum_r L_r(u)R_r(v)$, where r ranges over all vertices of $\mathcal{C}$, $L_r(u)$ is the coefficient of u in L_r, and $R_r(v)$ is the coefficient of v in R_r. We see this as a matrix equality: $N_f = \sum_r L_r \cdot R_r$, where L_r is seen as a column vector, and R_r as a row vector. By subadditivity of the rank and since the product of a column vector by a row vector is a matrix of rank at most 1, this implies that $\operatorname{rank}(N_f)$ is bounded by the size of the ABP, yielding the lower bound in Nisan's result.

The crucial idea of splitting the computation of a monomial into two parts had been independently developed by Fliess when studying so-called *Hankel Matrices* in [Fliess, 1974] to derive a very similar result in the field of *weighted automata*, which are finite state machines recognising *words series*, i.e., functions from finite words into a field. Fliess' theorem [Fliess, 1974, Th. 2.1.1] states that the size of the smallest weighted automaton recognising a word series f is exactly the rank of the Hankel matrix of f. The key insight to relate the two results is to see a non-commutative monomial as a finite word over the alphabet whose letters are the variables. Using this correspondence one can obtain Nisan's theorem from Fliess' theorem, observing that the Hankel matrix coincides with the matrix N_f defined by Nisan and that acyclic weighted automata correspond to ABPs. (We refer to an early technical report of this work for more details on this correspondence [Fijalkow et al., 2018].)

1.2. **Non-Associative Computations.** Hrubeš, Wigderson and Yehudayoff in [Hrubeš et al., 2011] drop another classical rule of computation, namely associativity: $(xy)z$ is no longer equal to $x(yz)$. In this foundational work the authors show how to define the complexity classes **VP** and **VNP** in the absence of either commutativity or associativity (or both) and prove that these definitions are sound in particular by obtaining the completeness of the permanent.

In the same way that a non-commutative monomial can be seen as a word, a non-commutative and non-associative monomial such as $(xy)(x(zy))$ can be seen as a tree, and more precisely as an ordered binary rooted tree whose leaves are labelled by variables. The starting point of our work was to exploit this connection. The work of Bozapalidis and Louscou-Bozapalidou [Bozapalidis and Louscou-Bozapalidou, 1983] extends Fliess' result to trees; although we do not technically rely on their results they serve as a guide, in particular for understanding how to decompose trees.

Let us return to the key idea in Nisan's proof, which is to decompose the computation of an ABP into two parts. The way a monomial, e.g., $x_1x_2x_3\cdots x_d$, is evaluated in an ABP is very constrained, namely from left to right, or if we make the implicit non-associative structure explicit as $w = (\cdots(((x_1x_2)x_3)x_4)\cdots)x_d$. The decompositions of w into two monomials u, v are of the form $u = (\cdots((x_1x_2)x_3)\cdots)x_{i-1})$ and $v = (\cdots((\Box x_i)x_{i+1})\cdots)x_d$. Here $\Box$ is a new fresh variable (the *hole*) to be substituted by u. Moving to non-associative polynomials, a monomial is a tree whose leaves are labelled by variables. A *context* is a monomial over the set of variables extended with a new fresh one denoted $\Box$ and occurring exactly once. For instance the composition of the monomial $t = z((xx)y)$ with the context $c = (xy)((z\Box)y)$ is the monomial $c[t] = (xy)((z(z((xx)y)))y)$.

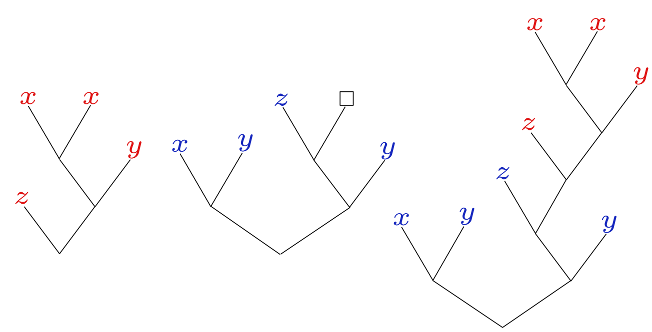

FIGURE 1. On the left hand side the monomial t, in the middle the context c, and on the right hand side the monomial $c[t]$.

Let f be a non-associative (possibly commutative) polynomial f, the *Hankel matrix* H_f of f is defined as follows: the rows of H_f are indexed by contexts and the columns by monomials, the value of $H_f(c,t)$ at row c and column t is the coefficient of the monomial $c[t]$ in f.

Extending Nisan's proof to computations in a *general circuit*, which are done along trees, we obtain a characterization in the non-associative setting.

Theorem 1. *Let f be a non-associative homogeneous polynomial and let H_f be its Hankel matrix. Then, the size of the smallest circuit computing f is exactly $\mathrm{rank}(H_f)$.*

Note that this is a characterization result: the Hankel matrix exactly captures the size of the smallest circuit computing f (upper and lower bounds), exactly as in Nisan's result. Hence, understanding the rank of the Hankel matrix is equivalent to studying circuits for f. We recover and extend Nisan's characterization as a special case of our result. This is better understood with the notion of parse trees at hand, which we present now.

1.3. **Parse Trees.** At an intuitive level, parse trees can be used to explain in what way a circuit uses the associativity rule. Consider the case of a circuit computing the (associative) monomial $2xyz$. Since this monomial corresponds to two non-associative monomials: $(xy)z$ and $x(yz)$, the circuit may sum different computations, for instance $3(xy)z - x(yz)$, which up to associativity is $2xyz$. We say that such a circuit contains two parse trees, corresponding to the two different ways of parenthesizing xyz.

The *shape* of a non-associative monomial is the tree obtained by forgetting the variables, e.g., the shape of $(z((xy)((xx)y)))$ is $(\_\ ((\_\ \_)((\_\ \_)\ \_)))$. The parse trees of a circuit $\mathcal{C}$ are the shapes induced by computations in $\mathcal{C}$.

Unique Parse Trees. Lagarde, Malod and Perifel introduced in [Lagarde et al., 2016] the class of Unique Parse Tree circuits (UPT), which are circuits computing non-commutative homogeneous (but associative) polynomials such that all monomials are evaluated in the same non-associative way. In other words, a circuit is UPT if it has a unique parse tree, which can be understood as a unique evaluation policy. ABPs are UPT circuits with the unique parse tree being the left-comb tree. Indeed, as already mentioned, in an ABP monomials are evaluated left to right, so the parse trees all have the same non-associative structure which is $((\cdots(((\_\ \_)\ \_)\ \_)\cdots)\ \_)$. It intuitively corresponds to a greedy computation; in contrast, a divide and conquer evaluation policy would correspond to using a complete binary tree as unique parse tree.

By restricting a circuit to having a unique parse tree we fix the non-associative polynomial it computes. As a result, we obtain as a Corollary of Theorem 1 a characterization result for UPT circuits. This marginally strengthens the result of [Lagarde et al., 2016] since they require a notion of canonical form for UPT circuits. We apply this characterization and provide for any parse tree s an expression for the size of the smallest UPT circuit with parse tree s computing the non-commutative permanent.

Parse Tree Restrictions. Many interesting classes of circuits can be defined by restricting the set of allowed parse trees, both in the commutative and the non-commutative setting. We already discussed ABP [Nisan, 1991; Dvir et al., 2012; Ramya and Rao, 2018] and UPT circuits [Lagarde et al., 2016]. The class of skew circuits [Toda, 1992; Allender et al., 1998; Malod and Portier, 2008; Limaye et al., 2016] and its extension small non-skew depth circuits [Limaye et al., 2016], together with the class of unambiguous circuits [Arvind and Raja, 2016] are all defined by looking at the parse trees. We propose in our technical developments some related restrictions called slightly balanced and slightly unbalanced circuits. Last but not least, the class of k-PT circuits [Arvind and Raja, 2016; Lagarde et al., 2018; Saptharishi and Tengse, 2017] is simply the class of circuits having at most k parse trees.

1.4. **Contributions and Outline.** In this paper we prove lower bounds for classes of circuits with parse tree restrictions, both in the commutative and non-commutative setting.

Our first and conceptually main contribution is the characterization result stated in Theorem 1 and proved in Section 2, which gives an algebraic approach to understanding circuits in the non-associative setting. All the subsequent results in this paper are based on this approach.

Section 3 is devoted to the definition of parse trees and a classical tool for proving lower bounds, the partial derivative matrices. We can already show at this point how Theorem 1 can be specialized to give a characterization result for UPT circuits, extending Nisan's result. (We note that a characterization result for UPT circuits was already known [Lagarde et al., 2016], we slightly improve on it.) As a corollary we obtain exponential lower bounds on the size of the smallest UPT circuit computing the permanent.

Our most technical developments are discussed in Section 4. We prove generic lower bound results by further analyzing and decomposing the Hankel matrix, with the following proof scheme. We consider a polynomial f in the associative setting. Let $\mathcal{C}$ be a circuit computing f. Forgetting about associativity we can see $\mathcal{C}$ as computing a non-associative polynomial $\tilde{f}$, which projects onto f, meaning is equal to f assuming associativity. This induces a set of linear constraints: for instance if the monomial xyz has coefficient 3 in f, then we know that $\tilde{f}((xy)z) + \tilde{f}(x(yz)) = 3$. We make use of the linear constraints to derive lower bounds on the rank of the Hankel matrix $H_{\tilde{f}}$, yielding a lower bound on the size of $\mathcal{C}$.

The final section is devoted to applications of our results, where we obtain superpolynomial and exponential lower bounds for various classes. In the results mentioned below, n is the number of variables, d is the degree of the polynomial, and k the number of parse trees. We note that the lower bounds hold for any (prime) n, any d, and any field.

We obtain alternative proofs of some known lower bounds: unambiguous circuits [Arvind and Raja, 2016], skew circuits [Limaye et al., 2016] and small non-skew depth circuits (obtaining a much shorter proof than [Limaye et al., 2016]).

Our main contributions are:

- *Slightly unbalanced circuits.* We extend the exponential lower bound from [Limaye et al., 2016] on $\frac{1}{5}$-unbalanced circuits to $\left(\frac{1}{2} - \varepsilon\right)$-unbalanced circuits.
- *Slightly balanced circuits.* We derive a new exponential lower bound for ε-balanced circuits.
- *Circuits with k parse trees in the non-commutative setting.* We extend the superpolynomial lower bound of [Lagarde et al., 2018] from $k = 2^{d^{1/3-\varepsilon}}$ to $k = 2^{d^{1-\varepsilon}}$.
- *Circuits with k parse trees in the commutative setting.* We extend the superpolynomial lower bound from [Arvind and Raja, 2016] from $k = d^{1/2-\varepsilon}$ to $k = 2^{d^{1/3-\varepsilon}}$, and even to $k = 2^{d^{1-\varepsilon}}$ when d is polylogarithmic in n.

We comment on the last two results. In the non-commutative setting this closes the gap with the upper bound $k = 2^{O(d)}$ on the total number of parse trees. In other words, we obtain superpolynomial lower bound on any class of circuits which has a small defect in the exponent of the total number of parse trees. However, in the commutative setting although we improve from a sublinear to superpolynomial number of parse trees, no gap is closed since the number of commutative parse trees is roughly $d!$.

1.5. **Related Work.** We argued that proving lower bounds in the non-commutative setting is easier, but this has not yet materialized since the best lower bound for general circuits in this setting is the same as in the commutative setting (by Baur and Strassen, already mentionned above). Indeed, recent impressive results suggest that this may be hard: Carmosino, Impagliazzo, Lovett, and Mihajlin [Carmosino et al., 2018] (essentially) proved that a lower bound in the non-commutative setting which would be slightly stronger than superlinear can be amplified to get strong lower bounds (even exponential, in some cases).

Most approaches for proving lower bounds rely on algebraic techniques and the rank of some matrix. A different and beautiful approach was investigated by Hrubeš, Wigderson and Yehudayoff [Hrubeš et al., 2011] in the non-commutative setting through the study of the so-called *sum-of-squares problem.* Roughly speaking, the goal is to decompose $(x_1^2 + \cdots + x_k^2) \cdot (y_1^2 + \cdots + y_k^2)$ into a sum of n squared bilinear forms in the variables x_i and y_j. They show that almost any superlinear bound on n implies non-trivial lower bounds on the size of any non-commutative circuit computing the permanent.

The quest of finding lower bounds is deeply connected to another problem called polynomial identity testing (PIT) for which the goal is to decide whether a given circuit computes the formal zero polynomial. The connection was shown in [Kabanets and Impagliazzo, 2003], in which it is proved that providing an efficient deterministic algorithm to solve the problem implies strong lower bounds either in the arithmetic or boolean setting. PIT was widely investigated in the commutative and non-commutative settings for classes of circuits based on parse trees restrictions, see e.g., [Raz and Shpilka, 2005; Forbes et al., 2014; Agrawal et al., 2015; Gurjar et al., 2017; Saptharishi and Tengse, 2017].

2. Characterizing Non-Associative Circuits

2.1. Basic Definitions.

For an integer $d \in \mathbb{N}$, we let $[d]$ denote the integer interval $\{1, \ldots, d\}$.

Polynomials. Let K be a field and let X be a set of ***variables***. Following [Hrubeš et al., 2011] we consider that unless otherwise stated multiplication is neither commutative nor associative. We assume however that addition is commutative and associative, and that multiplication distributes over addition. A ***monomial*** is a product of variables in X and a polynomial f is a formal finite sum $\sum_i c_i m_i$ where m_i is a monomial and $c_i \in K$ is a non-zero element called the coefficient of m_i in f. We let $f(m_i)$ denote the coefficient of m_i in f, so that $f = \sum_i f(m_i)m_i$.

The ***degree*** of a monomial is defined in the usual way, i.e., $\deg(x) = 1$ when $x \in X$ and $\deg(m_1 m_2) = \deg(m_1) + \deg(m_2)$; the degree of a polynomial f is the maximal degree of a monomial in f. A polynomial is ***homogeneous*** if all its monomial have the same degree. Depending on whether we include the relations $u \cdot v = v \cdot u$ (commutativity) and $u \cdot (v \cdot w) = (u \cdot v) \cdot w$ (associativity) we obtain four classes of polynomials.

Unless otherwise specified, for a polynomial f we use n for the number of variables and d for the degree.

Trees and Contexts. The ***trees*** we consider have a single root and binary branching (every internal node has exactly two children). To account for the commutative and for the non-commutative setting we use either ***unordered trees*** or ***ordered trees***, the only difference being that in the case of ordered trees we distinguish the left child from the right child. We let ***Tree*** denote the set of trees (it will be clear from the context whether they are ordered or not). The size of a tree is defined as its number of leaves.

A non-associative monomial m is a tree with leaves labelled by variables. For instance the monomial $z((xx)y)$ is represented on the left hand side of Figure 1. If m is non-commutative then it is an ordered tree, and if m is commutative then it is an unordered tree. We let $\boldsymbol{Tree(X)}$ denote the set of trees whose leaves are labelled by variables in X and $\boldsymbol{Tree_i(X)}$ denote the subset of such trees with i leaves, which are monomials of degree i.

In this paper we see a non-associative polynomial as a mapping from monomials to K, i.e., an element $f : \mathrm{Tree}(X) \to K$. To avoid possible confusion, let us insist that the notation $f(m)$ refers to the coefficient of the monomial m in the polynomial f, not to be confused with the evaluation of f at a given point. Similarly, a non-commutative associative homogeneous polynomial of degree d is seen as a mapping $f : X^d \to K$.

A (ordered or unordered) ***context*** is a tree with a distinguished leaf labelled by a special symbol called the ***hole*** and written $\square$. We let $\boldsymbol{Context(X)}$ denote the set of contexts whose leaves are labelled by variables in X. Given a context c and a tree t we construct a new tree $c[t]$ by substituting the hole of c by t. This operation is defined in both ordered and unordered settings. An example of a context c and the construction of $c[t]$ is given in Figure 1.

Hankel Matrices. Let f be a non-associative polynomial. The ***Hankel matrix*** H_f of f is the matrix whose rows are indexed by contexts and columns by monomials and such that the value of H_f at row c and column t is the coefficient of the monomial $c[t]$ in f.

Arithmetic Circuits. An (arithmetic) ***circuit*** is a directed acyclic graph such that the vertices are of three types:

- input gates: they have in-degree 0 and are labelled by variables in X,

- addition gates: they have arbitrary in-degree, an output value in K, and a weight $w(a) \in K$ on each incoming arc a,
- multiplication gates: they have in-degree 2, and we distinguish between the left child and the right child.

Each gate v in the circuit computes a polynomial f_v which we define by induction.

- An input gate labelled by a variable $x \in X$ computes the polynomial x.
- An addition gate v with n arcs incoming from gates $v_1, \ldots, v_n$ and with weights $\alpha_1, \ldots, \alpha_n$, computes the polynomial $\alpha_1 f_{v_1} + \cdots + \alpha_n f_{v_n}$.
- A multiplication gate with left child u and right child v computes the polynomial $f_u f_v$.

The circuit itself computes a polynomial given by the sum over all addition gates of the output value times the polynomial computed by the gate. Note that it is slightly unusual that all addition gates contribute to the circuit; one can easily reduce to the classical case where there is a unique output addition gate by adding an extra gate.

To define the size of a circuit we make a syntactic assumption: each arc is either coming from, or going to (but not both), an addition gate. This is a small assumption which can be lifted at the price of a linear blow-up. The ***size*** of a circuit $\mathcal{C}$ is denoted $|\mathcal{C}|$ and defined to be its number of addition gates. Note that this is how the size of ABPs is defined, it will be a convenient definition here since our characterization result captures the exact size of the smallest circuit computing a given polynomial.

Note that the definitions we gave above do not depend on which of the four settings we consider: commutative or non-commutative, associative or non-assocative.

Consider the circuit on the left hand side of Figure 2: it computes the polynomial $7y^2+2xy+yx$, which in the commutative setting is equal to $7y^2 + 3xy$.

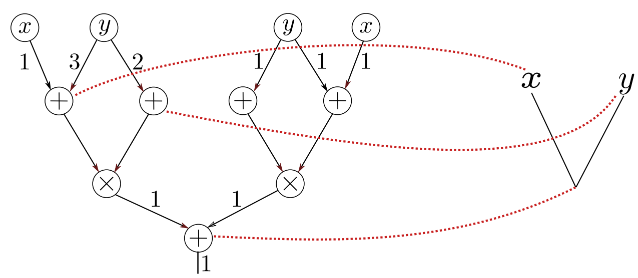

FIGURE 2. On the left hand side a circuit computing the polynomial $7y^2+2xy+yx$, which in the commutative setting is equal to $7y^2+3xy$. The only addition gate with a non-zero output value is at the bottom, its output value is 1. On the right hand side the monomial xy, seen as non-associative. The dashed red arrow show one run of the circuit over this monomial.

2.2. **The Characterization.** We prove the characterization stated in Theorem 1. It extends Nisan's characterization of non-commutative ABPs to general circuits in the non-associative setting. The result holds for both commutative and non-commutative settings, the proof being the same up to cosmetic changes.

The key step to go from ABPs to general circuits is the following: the polynomial computed by an ABP is the sum over the *paths* of the underlying graph, whereas in a general circuit the sum is over *trees*. We formalize this in the next definition by introducing *runs* of a circuit. The definition is given in the non-commutative setting but easily adapts to the commutative setting as explained in Remark 1.

Definition 1. *Let $\mathcal{C}$ be a circuit and $V_\oplus$ denote its set of addition gates. Let $t \in Tree(X)$ be a monomial. A* ***run of $\mathcal{C}$ over t*** *is a map ρ from nodes of t to $V_\oplus$ such that*

(i) *A leaf of t with label $x \in X$ is mapped to a gate with a non-zero edge incoming from an input gate labelled by x.*
(ii) *If n is a node of t with left child n_1 and right child n_2, then $\rho(n)$ has a non-zero edge incoming from a multiplication gate with left child $\rho(n_1)$ and right child $\rho(n_2)$.*
(iii) *The root of t is mapped to a gate with non-zero output value.*

The ***value*** *$val(\rho)$ of ρ is a non-zero element in K defined as the product of the weights of the edges mentioned in items (i) and (ii) together with the output value of $\rho(r)$, r being the root of t.*

We write by a small abuse of notation $\rho : t \to V_\oplus$ for runs of $\mathcal{C}$ over t.

We refer to Figure 2 for an example of a run over the monomial xy. The value of the run is 2.

Remark 1. *In the commutative setting we simply replace item (ii) by: "if n is a node of t with children n_1, n_2, then $\rho(n)$ has a non-zero edge incoming from a multiplication gate with children $\rho(n_1), \rho(n_2)$".*

A run of $\mathcal{C}$ over a monomial t additively contributes to the coefficient of t in the polynomial computed by $\mathcal{C}$, leading to the following lemma.

Lemma 1. *Let $\mathcal{C}$ be a circuit computing the non-associative polynomial $f : Tree(X) \to K$. Then the coefficient $f(t)$ of a monomial $t \in Tree(X)$ in f is equal to*

$$\sum_{\rho : t \to V_\oplus} val(\rho).$$

We may now state and prove our cornerstone result, which holds in both the commutative and non-commutative settings.

Theorem 2. *Let $f : Tree(X) \to K$ be a non-associative polynomial, H_f be its Hankel matrix, and $\mathcal{C}$ be a circuit computing f. Then $|\mathcal{C}| \geq rank(H_f)$. Moreover, if f is homogeneous this bound is tight, meaning there exists a circuit $\mathcal{C}$ computing f of size $rank(H_f)$.*

An interesting feature of this theorem is that the upper bound is effective: given a homogenous polynomial one can construct a circuit computing this polynomial of size rank (H_f).

We only prove the lower bound as the upper bound is not used in the rest of the paper (we refer to Appendix A for the latter). The proof of the lower bound follows the same lines as Nisan's original proof for non-commutative ABPs [Nisan, 1991].

Proof. Let $\mathcal{C}$ be a circuit computing the non-associative polynomial $f : \mathrm{Tree}(X) \to K$. Let $V_\oplus$ denote the set of addition gates of $\mathcal{C}$. To bound the rank of the Hankel matrix H_f by $|\mathcal{C}| = |V_\oplus|$ we show that H_f can be written as the sum of $|V_\oplus|$ matrices each of rank at most 1.

For each $v \in V_\oplus$ we define two circuits which decompose the computations around v. Let $\mathcal{C}_1^v$ be the restriction of $\mathcal{C}$ to descendants of v, and $\mathcal{C}_2^v$ to be a copy of $\mathcal{C}$ with just an extra input gate labelled by a fresh variable $\square \notin X$ with a single outgoing edge with weight 1 going to v.

We let $f^v : \mathrm{Tree}(X) \to K$ denote the polynomial computed by $\mathcal{C}_1^v$ and $g^v : \mathrm{Context}(X) \to K$ denote the restriction of the polynomial computed by $\mathcal{C}_2^v$ to $\mathrm{Context}(X) \subseteq \mathrm{Tree}(X \sqcup \{\square\})$.

We show the equality

$$H_f(c, t) = \sum_{v \in V_\oplus} f^v(t) g^v(c).$$

Fix a monomial $t \in \mathrm{Tree}(X)$ and a context $c \in \mathrm{Context}(X)$. We let $n_\square$ denote the leaf of c labelled by $\square$, which is also the root of t and the node to which t is substituted with in $c[t]$. Relying on Lemma 1, we calculate the coefficient $f(c[t])$ of $c[t]$ in f.

$$f(c[t]) = \sum_{\rho: c[t]\to V_\oplus} \mathrm{val}(\rho) = \sum_{v\in V_\oplus} \sum_{\substack{\rho: c[t]\to V_\oplus \\ \rho(n_\square)=v}} \mathrm{val}(\rho) = \sum_{v\in V_\oplus} \sum_{\substack{\rho_1^v: t\to V_\oplus \\ \rho_1^v(n_\square)=v}} \sum_{\substack{\rho_2^v: c\to V_\oplus \\ \rho_2^v(n_\square)=v}} \mathrm{val}(\rho_1^v)\mathrm{val}(\rho_2^v)$$

$$= \sum_{v\in V_\oplus} \sum_{\substack{\rho_1^v: t\to V_\oplus \\ \rho_1^v(n_\square)=v}} \mathrm{val}(\rho_1^v) \sum_{\substack{\rho_2^v: c\to V_\oplus \\ \rho_2^v(n_\square)=v}} \mathrm{val}(\rho_2^v) = \sum_{v\in V_\oplus} f^v(t) g^v(c).$$

Let $M_v \in K^{\mathrm{Tree}(X)\times \mathrm{Context}(X)}$ be the matrix given by $M_v(t,c) = f^v(t)g^v(c)$: its rank is at most one as M_v is the product of a column vector by a row vector. The previous equality reads in matrix form $H_f = \sum_{v\in V_\oplus} M_v$. Hence, we obtain the announced lower bound using rank subadditivity:

$$\mathrm{rank}\,(H_f) = \mathrm{rank}\left(\sum_{v\in V_\oplus} M_v\right) \le \sum_{v\in V_\oplus} \mathrm{rank}\,(M_v) \le |V_\oplus| = |\mathcal{C}|.$$

□

The remainder of this paper consists in applying Theorem 1 to obtain lower bounds in various cases. To this end we need a better understanding of the Hankel matrix: in Section 3 we define the notion of parse trees for deconstructing circuits and in Section 4 we develop decomposition theorems for the Hankel matrix.

Before digging deeper we can already give applications of Theorem 1, yielding simple proofs of non-trivial results from the literature.

Our first corollary is an alternative separation argument of the classes **VP** and **VNP** in the commutative non-associative setting. The original proof is due to [Hrubeš et al., 2010, Theorem 6], it exhibits an *explicit* polynomial which requires a superpolynomial circuit to be computed. We give here a different polynomial but our bounds are very similar.

Corollary 1. *Let f be the commutative non-associative polynomial of degree $2d$ and over two variables x_0 and x_1 defined by*

$$f = \sum_{\varepsilon_1,\dots,\varepsilon_d \in \{0,1\}} (((\cdots(x_{\varepsilon_1} x_{\varepsilon_2}) x_{\varepsilon_3})\cdots) x_{\varepsilon_d})^2.$$

Any circuit computing f has size at least 2^{d-1}.

Proof. We give a lower bound on the rank of the Hankel matrix. We look at the submatrix restricted to contexts with $(d+1)$ leaves of the form $(((\cdots(((x_{\varepsilon_1}\cdot x_{\varepsilon_2})\ x_{\varepsilon_3})\ x_{\varepsilon_4})\cdots)\ x_{\varepsilon_d})\square)$ and to rows with d leaves of the form $((\cdots(((x_{\varepsilon'_1}\cdot x_{\varepsilon'_2})\ x_{\varepsilon'_3})\ x_{\varepsilon'_4})\cdots)\ x_{\varepsilon'_d})$. This matrix is (almost) a permutation matrix of size 2^d, the only difference being the symmetry between the two leaves at the top of the comb, hence it has rank 2^{d-1}. □

Our second corollary is an alternative proof of [Arvind and Raja, 2016, Theorem 26], which gives an exponential lower bound for the permanent and the determinant against unambiguous circuits in the associative setting. A circuit is said ***unambiguous***, if for each (associative) monomial m, there is at most one tree t labelled by m such that $\mathcal{C}$ has a run over t. Note that this notion makes sense in both the commutative and the non-commutative settings. Our lower bounds hold in both settings.

Corollary 2. *Any unambiguous circuit computing the determinant or the permanent has size at least $\binom{n}{n/3}$.*

Proof sketch. Consider an unambiguous circuit computing the permanent (the arguments are similar for the determinant) of degree n over the variables $X = \{x_{i,j} \mid i,j \in [n]\}$. For any permutation σ, let $t_\sigma \in \mathrm{Tree}(X)$ be the (non-associative) monomial over which there is a run computing the (associative) monomial $x_{1,\sigma(1)}x_{2,\sigma(2)}\cdots x_{n,\sigma(n)}$. Since the circuit is unambiguous the non-associative polynomial $\tilde{f}$ computed by $\mathcal{C}$ is $\tilde{f} = \sum_\sigma t_\sigma$. We obtain a lower bound on the rank of $H_{\tilde{f}}$ by writing it as a diagonal block matrix and arguing by a simple counting argument that there must be many blocks of rank at least 1. We refer to Appendix B for details. □

This proof goes beyond the case of unambiguous circuits. It is actually sufficient to assume that all non-associative monomials t such that $\tilde{f}(t) \neq 0$ are labelled by a monomial of the form $x_{1,\sigma(1)}x_{2,\sigma(2)}\cdots x_{n,\sigma(n)}$ for some permutation σ.

3. Parse Trees, Partial Derivative Matrices and Unique Parse Tree Circuits

In this section we define two useful tools for analyzing circuits. The first is parse trees, which is the lens through which we analyze circuits. The second is partial derivative matrices, which have been used for proving lower bounds. The principal aim is to use these two tools for studying the Hankel matrix in the next section. Before this we will show how they can be used in combination with Theorem 1 to give a characterization result for unique parse tree circuits.

In this section we restrict ourselves to the *non-commutative setting*. We later explain in Section 4.4 how to extend the study to the commutative case.

3.1. Parse Trees.

With any monomial $t \in \mathrm{Tree}(X)$ we associate its ***shape*** $\mathrm{shape}(t) \in \mathrm{Tree}$ as the tree obtained from t by removing the labels at the leaves.

Definition 2. *Let $\mathcal{C}$ be a circuit computing a non-commutative non-associative polynomial f. A **parse tree** of $\mathcal{C}$ is any shape $s \in$ Tree for which there exists a monomial $t \in \mathrm{Tree}(X)$ whose coefficient in f is non-zero and such that $s = \mathrm{shape}(t)$. We let $PT(\mathcal{C}) = \{\mathrm{shape}(t) \mid f(t) \text{ non-zero}\}$.*

3.2. Partial Derivative Matrices.

For $A \subseteq [d]$ of size i , $u \in X^{d-i}$, and $v \in X^i$, we define the monomial $u \otimes_A v \in X^d$: it is obtained by interleaving u and v with u taking the positions indexed by $[d] \setminus A$ and v the positions indexed by A. For instance $x_1x_2 \otimes_{\{2,4\}} y_1y_2 = x_1y_1x_2y_2$.

Definition 3. *Let f be a homogeneous non-commutative associative polynomial. Let $A \subseteq [d]$ be a set of positions of size i.*

*The **partial derivative matrix** $M_A(f)$ of f with respect to A is defined as follows: the rows are indexed by $u \in X^{d-i}$ and the columns by $v \in X^i$, and the value of $M_A(f)(u,v)$ is the coefficient of the monomial $u \otimes_A v$ in f.*

Example 1. *Let $f = xyxy + 3xxyy + 2xxxy + 5yyyy$ and $A = \{2,4\}$. Then $M_A(f)$ is given below.*

	$\_x\_x$	$\_x\_y$	$\_y\_x$	$\_y\_y$
$x\_x\_$	0	2	0	1
$y\_x\_$	0	0	0	0
$x\_y\_$	0	3	0	0
$y\_y\_$	0	0	0	5

3.3. Application: Unique Parse Tree Circuits.

Recall that UPT is the class of circuits having a unique parse tree. This class was introduced in [Lagarde et al., 2016] and a characterization of the smallest UPT circuit was obtained. We give an alternative simple proof of this result using Theorem 1. We obtain a small improvement since the original result requires a normal form which can lead to an exponential blow-up.

Given a shape $s \in$ Tree of size d, i.e., with d leaves and a node v of s, we let s_v denote the subtree of s rooted in v, and $I_v \subseteq [d]$ denote the interval of positions of the leaves of s_v in s. We say that $s' \in$ Tree is a subshape of s if $s' = s_v$ for some v, and that I is spanned by s if $I = I_v$ for some v.

Let $f : X^d \to K$ be a homogeneous non-commutative associative polynomial of degree d, let $s \in \text{Tree}$ be a shape of size d, and let s' be a subshape of s such that $v_1, \ldots, v_p$ are all the nodes v of s such that $s' = s_v$. We define

$$M_{s'} = \begin{bmatrix} M_{I_{v_1}}(f) \\ M_{I_{v_2}}(f) \\ \vdots \\ M_{I_{v_p}}(f) \end{bmatrix}.$$

Theorem 3. *Let $f : X^d \to K$ be a homogeneous non-commutative associative polynomial and let $s \in$ Tree be a shape of size d. Then the smallest UPT circuit with shape s computing f has size exactly*

$$\sum_{s' \text{ subshape of } s} \mathit{rank}(M_{s'}).$$

Proof. Let $\mathcal{C}$ be a UPT circuit with shape s computing f. We let $\tilde{f}$ denote the non-associative polynomial computed by $\mathcal{C}$. Since $\mathcal{C}$ is UPT with shape s, $\tilde{f}$ is the *unique* non-associative polynomial which is non-zero only on trees with shape s and projects to f, i.e., $\tilde{f}(t) = f(u)$ if $\text{shape}(t) = s$ and t is labelled by u, and $\tilde{f}(t) = 0$ otherwise.

In particular, the size of the smallest UPT circuit with shape s computing f is the same as the size of the smallest circuit computing $\tilde{f}$, which thanks to Theorem 1 is equal to the rank of the Hankel Matrix $H_{\tilde{f}}$.

The Hankel matrix of $\tilde{f}$ may be non-zero only on columns indexed by trees whose shapes s' are subshapes of s, and on such columns, non-zero values are on rows corresponding to a context obtained from s by replacing an occurrence of s' by $\square$. The corresponding blocks are precisely the matrices $M_{s'}$, and are placed in a diagonal fashion, hence the lower bound. $\square$

Theorem 3 can be applied to concrete polynomials, for instance to the permanent of degree d.

Corollary 3. *Let $s \in$ Tree be a shape. The smallest UPT circuit with shape s computing the permanent has size*

$$\sum_{v \text{ node of } s} \binom{d}{|I_v|},$$

where I_v is the set of leaves in the subtree rooted at v in s. In particular, this is always larger than $\binom{d}{d/3}$.

Applied to s being a left-comb, Corollary 3 yields that the smallest ABP computing the permanent has size $2^d + d$. Applied to s being a complete binary tree of depth $k = \log d$, the size of the smallest UPT is $\Theta\left(\frac{2^d}{d}\right)$, showing that this circuit is more efficient than any ABP. We recall here that we count the number of addition gates.

4. Decomposing the Hankel Matrix

We now get to the technical core of the paper where we establish generic lower bounds theorems that we will later instantiate in Section 5 to concrete classes of circuits. We first define a distance which is often used to compare the ranks of partial derivative matrices of a given polynomial with respect to different subsets. We use the same general ideas for both the commutative and the non-commutative settings. However, since technical developments differ in the two settings, and in particular in the commutative setting we need to explain how to adapt the tools defined in Section 3, we treat the two settings one after the other.

4.1. **Distance and Partial Derivative Matrices.** We define a distance $\mathrm{dist} : \mathcal{P}([d]) \times \mathcal{P}([d]) \to \mathbb{N}$ on subsets of $[d]$ by letting $\mathrm{dist}(A, B)$ be the minimal number of additions and deletions of elements of $[d]$ to go from A to B, assuming that complementing is for free. Formally, $\mathrm{dist}(A, B) = \min\{|\Delta(A, B)|, |\Delta(A^{\mathrm{C}}, B)|\}$, where $\Delta(A, B) = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference between A and B.

The following lemma (see e.g., [Limaye et al., 2016]) informally says that, if A and B are close to each other, then the ranks of the corresponding partial derivative matrices are close to each other as well.

Lemma 2. *Let f be a homogeneous non-commutative associative polynomial. Then, for any subsets $A, B \subseteq [d]$, $\mathrm{rank}(M_A(f)) \leq n^{\mathrm{dist}(A,B)} \mathrm{rank}(M_B(f))$.*

4.2. **General Road Map.** We expand here on the proof scheme given in the introduction. Let f be a (commutative or non-commutative) polynomial for which we want to prove lower bounds. Consider a circuit $\mathcal{C}$ which computes f, and let $\tilde{f}$ be the non-associative polynomial computed by $\mathcal{C}$. Our aim is, following Theorem 1, to give lower bounds on the rank of the Hankel matrix $H_{\tilde{f}}$. We know that the $\tilde{f}$ and f are equal up to associativity, which provides linear relations among the coefficients of $H_{\tilde{f}}$.

The bulk of the technical work is to reorganize the rows and columns of $H_{\tilde{f}}$ in order to decompose it into blocks which may be identified as partial derivative matrices with respect to some subsets $A_1, A_2, \dots \subseteq [d]$, of some associative polynomials which depend on $\tilde{f}$ and sum to f. The number and choice of these subsets depend on the parse trees of the circuit $\mathcal{C}$.

Now, assume there exists a subset $A \subseteq [d]$ which is at distance at most δ to each A_i. Losing a factor of n^δ on the rank through the use of Lemma 2 we reduce the aforementioned blocks of $H_{\tilde{f}}$ to partial derivatives with respect to A. Such matrices can then be summed to recover the partial derivative matrix of f with respect to A, yielding in the lower bound a (dominating) factor of $\mathrm{rank}(M_A(f))$.

4.3. **Non-Commutative Setting.** Following the general road map described above, we obtain a first generic lower bound result.

Theorem 4. *Let $f : X^d \to K$ be a non-commutative homogeneous polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span an interval at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}(M_A(f))\, n^{-\delta} |PT(\mathcal{C})|^{-1}$.*

We now explain how to improve on this first result. Our main result, stated in Theorem 5, replaces the multiplicative factor $|\mathrm{PT}(\mathcal{C})|^{-1}$ by d^{-2}. This is an important improvement since the number of parse trees can be up to about 2^{2d}.

The crux to prove Theorem 4 is to identify for each parse tree s of $\mathcal{C}$ a block in $H_{\tilde{f}}$ containing the partial derivative matrix $M_{I(s)}(f_s)$ where f_s is the polynomial corresponding to the contribution of the parse tree s in the computation of f and $I(s)$ is an interval spanned by s.

However, we do not consider in this analysis how these blocks are located relative to each other. A more careful analysis of $H_{\tilde{f}}$ consists in grouping together all parse trees that lead to the same spanned interval. Aligning and then summing these blocks we remove the dependence in $|\mathrm{PT}(\mathcal{C})|$ and instead use d^2 which is the total number of possibly spanned intervals of $[d]$. This yields Theorem 5.

Theorem 5. *Let f be a non-commutative homogeneous polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span an interval at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}(M_A(f))\, n^{-\delta} d^{-2}$.*

As we shall see in Section 5 the lower bounds we obtain using Theorem 4 match known results, while using Theorem 5 yields substantial improvements.

4.4. **Commutative Setting.** We explain how to extend the notions of parse trees and the generic lower bound theorems to the commutative setting.

Let $X = X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d$ be a partition of the variable set X. A monomial is ***set-multilinear*** with respect to the partition if it is the product of exactly one variable from each set X_i, and a polynomial is set-multilinear if all its monomials are.

The permanent and the determinant of degree d are set-multilinear with respect to the partition $X = X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d$ where $X_i = \{x_{i,j}, j \in [d]\}$. The iterated matrix multiplication polynomial is another example of important and well-studied set-multilinear polynomial.

The notion of shape was defined by [Arvind and Raja, 2016], and it slightly differs from the non-commutative case because we need to keep track of the indices of the variable sets given by the partition from which the variables belong. More precisely, given a partition of $X = X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d$, we associate to any monomial $t \in \mathrm{Tree}(X)$ of degree d its ***shape*** $\mathrm{shape}(t) \in \mathrm{Tree}([d])$ defined as the tree obtained from t by replacing each label by its index in the partition. We let $\mathcal{T}_d \subseteq \mathrm{Tree}([d])$ denote the set of trees for which the leaves cover all $[d]$.

Let $\mathcal{C}$ be a circuit computing a non-commutative non-associative polynomial f. A ***parse tree*** of $\mathcal{C}$ is any shape $s \in \mathcal{T}_d$ for which there exists a monomial $t \in \mathrm{Tree}(X)$ whose coefficient in f is non-zero and such that $s = \mathrm{shape}(t)$. We let $\mathrm{PT}(\mathcal{C}) = \{\mathrm{shape}(t) \mid f(t) \text{ non-zero}\} \cap \mathcal{T}_d$.

Given a shape $s \in \mathrm{Tree}([d])$ with d leaves and a node v of s, we let s_v denote the subtree rooted at v and $A_v \subseteq [d]$ denote the set of labels appearing on the leaves of s_v.

Definition 4. *Let $X = X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d$, f be a set-multilinear polynomial of degree d, and $A \subseteq [d]$ be a set of indices. The* ***partial derivative matrix*** $\boldsymbol{M_A(f)}$ *of f with respect to A is defined as follows: the rows are indexed by set-multilinear monomials g with respect to the partition $\bigsqcup_{i \notin A} X_i$ and the columns are indexed by set-multilinear monomials h with respect to the partition $\bigsqcup_{i \in A} X_i$. The value of $M_A(f)(g,h)$ is the coefficient of the monomial $g \cdot h$ in f.*

Following the same road map as in the non-commutative setting we obtain the following counterpart of Theorem 4. We assume that the set of variables is partitioned into d parts of equal size n (this is a natural setting for polynomials such as the determinant, the permanent or the iterated matrix multiplication). In particular, it means that the polynomials we consider are of degree d and over nd variables.

Theorem 6. *Let f be a set-multilinear polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span a subset at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}(M_A(f))\, n^{-\delta} |PT(\mathcal{C})|^{-1}$.*

A notable difference with the non-commutative setting is that now parse trees no longer span intervals of $[d]$ but subsets of $[d]$. As a consequence, the technique used to prove Theorem 5 groups together blocks corresponding to the same *subset* of $[d]$ and therefore the multiplicative factor is now 2^{-d} as there are 2^d such subsets.

Theorem 7. *Let f be a set-multilinear polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span a subset at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}(M_A(f))\, n^{-\delta} 2^{-d}$.*

While in the non-commutative setting, Theorem 5 strengthens Theorem 4 (when d^2 is small), this is no longer the case in the commutative setting. Indeed, the maximal number of commutative parse trees being roughly $d!$ (the exact asymptotic is $\frac{\sqrt{2-\sqrt{2}}\, d^{d-1}}{e^d(\sqrt{2}-1)^{d+1}}$, see e.g., `https://oeis.org/A036774`), Theorem 6 and Theorem 7 are incomparable.

5. Applications

In this section we instantiate our generic lower bounds theorems on concrete classes of circuits. We first show how the weaker version (Theorem 4) yields the best lower bounds to dates for skew and small non-skew depth circuits. Extending these ideas we obtain exponential lower bounds for $\left(\frac{1}{2}-\varepsilon\right)$-unbalanced circuits, an extension of skew circuits which are just slightly unbalanced. We also adapt the proof to ε-balanced circuits, which are slightly balanced.

Our main results concern circuits with many parse trees. In this context we see the benefits of Theorem 5 over Theorem 4, considerably improving previously known lower bounds.

High-ranked polynomials. The lower bounds we state below hold for any polynomial whose partial derivative matrices with respect to either a fixed subset A or all subsets have large rank. Such polynomials exist for all fields in both the commutative and non-commutative settings, and can be explicitly constructed. For instance the so-called Nisan-Wigderson polynomial given in [Kayal et al., 2014] (inspired by the notion of designs by Nisan and Wigderson [Nisan and Wigderson, 1994]) has this property. They are given by

$$NW_{n,d} = \sum_{\substack{h \in \mathbb{F}_n[z] \\ \deg(h) \leq d/2}} \prod_{i=1}^{d} x_{i,h(i)},$$

where $\mathbb{F}_n[z]$ denotes univariate polynomials with coefficients in the finite field of prime order n. The fact that there exists a unique polynomial $h \in \mathbb{F}_n[z]$ of degree at most $d/2$ which takes $d/2$ given values at $d/2$ given positions exactly implies that the partial derivative matrix of $NW_{n,d}$ with respect to any $A \subseteq [d]$ of size $d/2$ is a permutation matrix. This is then easily extended to any $A \subseteq [d]$.

5.1. **Skew, Slightly Unbalanced, Slightly Balanced and Small Non-Skew Depth Circuits.**

We show how using Theorem 4 yields exponential lower bounds for four classes of circuits in the non-commutative setting.

Skew Circuits. A circuit $\mathcal{C}$ is ***skew*** if all its parse trees are skew, meaning that each node has at least one of its children which is a leaf. As a direct application of Theorem 4, we obtain the following result.

Theorem 8. *Let f be a homogeneous non-commutative polynomial such that $M_{[d/4+1,3d/4]}(f)$ has full rank $n^{d/2}$. Then any skew circuit computing f has size at least $2^{-d}n^{d/4}$.*

Slightly unbalanced circuits. A circuit $\mathcal{C}$ computing a homogeneous non-commutative polynomial of degree d is said to be ***α-unbalanced*** if every multiplication gate has at least one of its children which computes a polynomial of degree at most αd.

Theorem 9. *Let f be a homogeneous non-commutative polynomial such that $M_{[d/4+1,3d/4]}(f)$ has full rank $n^{d/2}$. Then any $\left(\frac{1}{2}-\varepsilon\right)$-unbalanced circuit computing f has size at least $4^{-d}n^{\varepsilon d}$.*

This result improves over a previously known exponential lower bound on $\left(\frac{1}{5}\right)$-unbalanced circuits [Limaye et al., 2016].

Slightly balanced circuits. A circuit $\mathcal{C}$ computing a homogeneous non-commutative polynomial of degree d is said to be ***α-balanced*** if every multiplication gate which computes a polynomial of degree k has both of its children which compute a polynomial of degree at least αk.

Theorem 10. *Let f be a homogeneous non-commutative polynomial such that $M_{[1,d/2]}(f)$ has full rank $n^{d/2}$. Then any ε-balanced circuit computing f has size at least $4^{-d}n^{\varepsilon d}$.*

Small Non-skew Depth Circuits. A circuit $\mathcal{C}$ has ***non-skew depth k*** if all its parse trees are such that each path from the root to a leaf goes through at most k non-skew nodes, i.e., nodes for which the two children are inner nodes.

We obtain an alternative proof of the exponential lower bound of [Limaye et al., 2016] on non-skew depth k circuits as an application of Theorem 4. In the statement below A refers to an explicit subset of $[d]$ that we do not define here (see Appendix K for more details).

Theorem 11. *Let f be a homogeneous non-commutative polynomial of degree $d = 12kp$ such that $M_A(f)$ has full rank $n^{d/2}$. Then any circuit of non-skew depth k computing f has size at least $4^{-d}n^{p/3} = 4^{-d}n^{d/36k}$.*

5.2. **Circuits with Many Parse Trees.** We focus on ***k-PT circuits*** which are circuits with at most k different parse trees.

The Non-commutative Setting. Lagarde, Limaye, and Srinivasan [Lagarde et al., 2018] obtained a superpolynomial lower bound for superpolynomial k (up to $k = 2^{d^{\frac{1}{3}-\varepsilon}}$). We first show how to obtain the same result using Theorem 4.

For $s \in \mathrm{Tree}_d$ and $A \subseteq [d]$, we define $\mathrm{dist}(A, s) = \min\{\mathrm{dist}(A, I) \mid I \text{ spanned by } s\}$. The following lemma is a subtle probabilistic analysis ensuring the existence of a subset which is close enough to all k parse trees.

Lemma 3 (adapted from Claim 15 in [Lagarde et al., 2018])**.** *Let $s \in Tree_d$ be a shape with d leaves, and $\delta \leq \sqrt{d}$. Then*

$$\Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[dist(A, s) > d/2 - \delta \right] \leq 2^{-\alpha d/\delta^2},$$

where α is some positive constant and $\mathcal{U}\left(\binom{[d]}{d/2}\right)$ the uniform distribution of subsets of d of size $d/2$.

Proof sketch. Following [Lagarde et al., 2018], we find a sequence of $r = \Omega(d/\delta^2)$ nodes of s which all span distant enough subtrees. We then obtain the bound by splitting the previous event into r essentially independent events. □

From there, the lower bound is obtained using Theorem 4 and a fine tuning of the parameters.

Theorem 12. *Let f be a homogeneous non-commutative polynomial such that for all $A \subseteq [d]$ $M_A(f)$ has full rank. Let $k = 2^{d^{1/3-\varepsilon}}$ and $\varepsilon > 0$. Then for large enough d any k-PT circuit computing f has size at least $2^{d^{1/3}(\log n - d^{-\varepsilon})}$.*

Proof. Let $\mathcal{C}$ be a k-PT circuit computing f, and $\delta = d^{1/3} \leq \sqrt{d}$. We first show that there exists a subset $A \subseteq [d]$ which is close to all parse trees in $\mathcal{C}$. Indeed, a union bound and Lemma 3,

$$\begin{aligned}\Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\exists s \in \mathrm{PT}(\mathcal{C}), \mathrm{dist}(A, s) > d/2 - \delta \right] &\leq \sum_{s \in \mathrm{PT}(\mathcal{C})} \Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\mathrm{dist}(A, s) > d/2 - \delta \right] \\ &\leq k 2^{-\alpha d/\delta^2} = 2^{d^{1/3-\varepsilon} - \alpha d^{1/3}} < 1,\end{aligned}$$

for large enough d. We now pick a subset $A \subseteq [d]$ of size $d/2$ such that for all $s \in \mathrm{PT}(\mathcal{C}), \mathrm{dist}(A, s) \leq d/2 - \delta$, that is, any $s \in \mathrm{PT}(\mathcal{C})$ spans an interval $I(s)$ at distance at most $d/2 - \delta$ from A. Finally, we apply Theorem 4 to obtain

$$|\mathcal{C}| \geq \mathrm{rank}(M_A(f))\, n^{-(d/2-\delta)} k^{-1} = n^{d/2} n^{-(d/2 - d^{1/3})} 2^{-d^{1/3-\varepsilon}} = 2^{d^{1/3}(\log n - d^{-\varepsilon})}.$$

□

We may improve the previous bound by applying Theorem 5 instead of Theorem 4. Indeed, since Theorem 5 gets rid of the factor k^{-1} in the lower bound, picking a much smaller δ ($\delta = d^{\varepsilon/3}$ instead of $d^{1/3}$) still leads to a superpolynomial lower bound, while allowing for more parse trees.

Theorem 13. *Let f be a homogeneous non-commutative polynomial such that for all $A \subseteq [d]$ $M_A(f)$ has full rank. Let $k = 2^{d^{1-\varepsilon}}$ and $\varepsilon > 0$. Then for large enough d any k-PT circuit computing f has size at least $n^{d^{\varepsilon/4}} d^{-2}$.*

The bound $2^{d^{1-\varepsilon}}$ on the number of parse trees is to be compared to the total number of shapes of size d which is $\frac{1}{d}\binom{2(d-1)}{d-1} \sim \frac{4^d}{d^{3/2}\sqrt{\pi}} \leq 2^{2d}$. As explained in the introduction this means that we obtain superpolynomial lower bounds for any class of circuits which has a small defect in the exponent of the total number of parse trees.

The Commutative Setting. Arvind and Raja [Arvind and Raja, 2016] showed a superpolynomial lower bound for sublinear k (up to $k = d^{1/2-\varepsilon}$). We improve this to superpolynomial k (up to $k = 2^{d^{1-\varepsilon}}$).

Indeed, in the commutative setting, Lemma 3 holds as such (with a shape being an element of $\mathcal{T}_d$, that is, a commutative parse tree of size d). However, the generic lower bound theorems, namely Theorem 6 and Theorem 7, are not exactly the same, so we obtain slightly different results. In particular, the two results we obtain are incomparable. Applying Theorem 6 leads to Theorem 14, whereas Theorem 7 leads to Theorem 15

Theorem 14. *Let f be a set-multilinear commutative polynomial such that for all $A \subseteq [d]$ $M_A(f)$ has full rank. Let $k = 2^{d^{1/3-\varepsilon}}$ and $\varepsilon > 0$. Then for large enough d any k-PT circuit computing f has size at least $2^{d^{1/3}(\log n - d^{-\varepsilon})}$.*

Theorem 15. *Let f be a set-multilinear commutative polynomial such that for all $A \subseteq [d]$ $M_A(f)$ has full rank. Let $k = 2^{d^{1-\varepsilon}}$ and $\varepsilon > 0$. Then for large enough d any k-PT circuit computing f has size at least $n^{d^{\varepsilon/4}} 2^{-d}$.*

6. Discussion

We presented a new tool for proving lower bounds for arithmetic circuits in the form of the Hankel matrix. We obtained strong lower bounds both in the commutative and non-commutative settings using generic decompositions of the Hankel matrix. A natural question is how far this approach can be pushed. The first remark is that the rank of the Hankel matrix is exactly the size of the smallest circuit computing a given (non-associative) polynomial, hence the potential loss can only be in analyzing the Hankel matrix. Indeed, our generic theorems find inside the Hankel matrix blocks corresponding to partial derivate matrices. Limaye, Malod and Srinivasan defined in [Limaye et al., 2016] a polynomial computed by a circuit of polynomial size but such that all partial derivative matrices have full rank: this shows that one cannot use our decomposition of the Hankel matrix to obtain strong lower bounds for the class of all circuits. This limitation is an invitation to get a deeper understanding of the Hankel matrix and to find other ways of decomposing it.

On a different perspective, the Hankel matrix has been successfully used as a data structure for learning algorithms (in both supervised and unsupervised settings). It is tempting, using the characterization that we present in this paper, to construct algorithms for learning polynomials relying on the Hankel matrix as algorithmic representation.

TECHNICAL APPENDIX

The ***permanent*** and ***determinant*** are the two most studied polynomials in this area, they are homogeneous polynomials of degree d over the d^2 variables $\{x_{i,j} \mid 1 \leq i, j \leq d\}$ defined by

$$\mathrm{Per} = \sum_{\sigma \mathfrak{S}_d} \prod_{i=1}^{d} x_{i,\sigma(i)} \qquad \mathrm{Det} = \sum_{\sigma \mathfrak{S}_d} (-1)^{\mathrm{sig}(\sigma)} \prod_{i=1}^{d} x_{i,\sigma(i)}$$

where σ ranges over permutations of $[d]$.

Appendix A. Proof of the Upper Bound in Theorem 1

We prove the upper bound in Theorem 1 that we recall below.

Theorem 1 (Upper bound). *Let f be a non-associative homogeneous polynomial and let H_f be its Hankel matrix. Then, the size of the smallest circuit computing f is exactly rank(H_f).*

We first give a construction of a circuit, then provide and prove by induction a strong invariant which implies that the circuit does indeed compute f. For every $t \in \mathrm{Tree}(X)$, we let H_t denote the corresponding column in the Hankel matrix, *i.e.* $H_t : c \mapsto c[t]$.

Let $T \subseteq \mathrm{Tree}(X)$ be such that $(H_t)_{t\in T}$ is a basis of $\{H_t \mid t \in \mathrm{Tree}(X)\}$. In particular T has size rank(H_f). For any $t' \in \mathrm{Tree}(X)$, we let $\alpha_t^{t'}$ denote the coefficient of H_t in the decomposition of $H_{t'}$ on $(H_t)_{t\in T}$, that is,

$$\sum_{t\in T} \alpha_t^{t'} H_t = H_{t'}. \tag{1}$$

We may now explicitly define circuit $\mathcal{C}$:

- The addition gates are (identified with) elements of T. The output value of $t \in T$ is $f(t)$.
- The input gates are given by elements of X (and the matching label). The input gate $x \in X$ has an outgoing arc to the addition gate $t \in T$ with weight α_t^x.
- The multiplication gates are given by elements $(t_0, t_1, t) \in T^3$. Such a multiplication gate has an incoming arc from t_0 on the left, an incoming arc from t_1 on the right, and an outgoing arc to t, with weight $\alpha_t^{t_1 \cdot t_2}$.

Note that the size of $\mathcal{C}$ is $|T| = \mathrm{rank}(H_f)$.

For $\mathcal{C}$ to be well-defined as a circuit, it remains to show that its underlying graph is acyclic. This is implied by the fact that $\alpha_t^{t_1 \cdot t_2}$ may only be non-zero if $\deg(t) = \deg(t_1) + \deg(t_2)$, which we now prove. Since f is homogeneous of degree d, H_t may be non-zero only on contexts c such that $\deg(c[t]) = d$, that is, $\deg(c) = d - \deg(t) + 1$. Hence, the set $\{H_t, t \in T\}$ may be partitioned according to the degree of t into parts with disjoint support, so for the decomposition (1) to hold, it must be that $\alpha_t^{t'} \neq 0$ implies $\deg(t) = \deg(t')$.

For $t \in T$, we let $g_t : \mathrm{Tree}(X) \to K$ denote the polynomial computed at gate t in $\mathcal{C}$. We will now show, by induction on the size of $t' \in \mathrm{Tree}(X)$, that

$$g_t(t') = \alpha_t^{t'}.$$

If $t' = x \in X$, then $g_t(t') = \alpha_t^x$, so the base case is clear. We now assume that $t' = t_1' \cdot t_2' \in \mathrm{Tree}(X)$, and show that $\sum_{t\in T} g_t(t') H_t = H_{t'}$, which is enough to conclude by uniqueness of the decomposition in (1). For that we will show that the previous equality holds for any context $c \in \mathrm{Context}(X)$.

We first remark the following

$$\begin{aligned}
\sum_{t\in T} g_t(t')H_t &= \sum_{t\in T}\left(\sum_{t_1,t_2\in T}\alpha_t^{t_1\cdot t_2}g_{t_1}(t'_1)g_{t_2}(t'_2)\right)H_t\\
&= \sum_{t\in T}\left(\sum_{t_1,t_2\in T}\alpha_t^{t_1\cdot t_2}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}\right)H_t\\
&= \sum_{t_1,t_2\in T}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}\left(\sum_{t\in T}\alpha_t^{t_1\cdot t_2}H_t\right)\\
&= \sum_{t_1,t_2\in T}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}H_{t_1\cdot t_2}.
\end{aligned}$$

Now, let $c\in \mathrm{Context}(X)$. For any tree $t\in \mathrm{Tree}(X)$, we define $c_t^1 = c[\square\cdot t]\in \mathrm{Context}(X)$, and $c_t^2 = c[t\cdot\square]\in \mathrm{Context}(X)$ (see Figure 3).Then for any $t_1,t_2, c[t_1\cdot t_2] = c^1_{t_2}[t_1] = c^2_{t_1}[t_2]$.

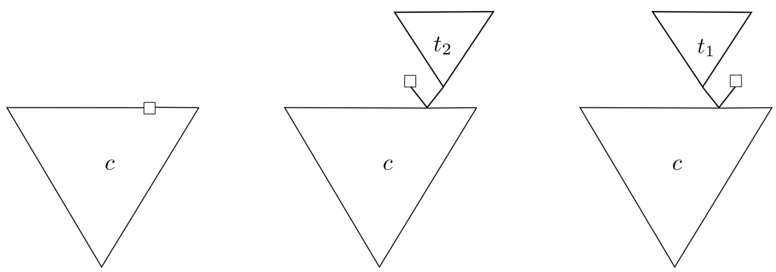

FIGURE 3. A context c, and the contexts $c^1_{t_2}$ and $c^2_{t_1}$.

Evaluating at c, we now obtain

$$\begin{aligned}
\sum_{t\in T} g_t(t')H_t(c) &= \sum_{t_1,t_2\in T}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}H_{t_1\cdot t_2}(c) = \sum_{t_1,t_2\in T}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}f(c[t_1\cdot t_2])\\
&= \sum_{t_1,t_2\in T}\alpha_{t_1}^{t'_1}\alpha_{t_2}^{t'_2}f(c^1_{t_2}[t_1]) = \sum_{t_1,t_2\in T}\alpha_{t_2}^{t'_2}H_{t_1}(c^1_{t_2})\\
&= \sum_{t_2\in T}\alpha_{t_2}^{t'_2}H_{t'_1}(c^1_{t_2}) = \sum_{t_2\in T}\alpha_{t_2}^{t'_2}H_{t'_1\cdot t_2}(c)\\
&= \sum_{t_2\in T}\alpha_{t_2}^{t'_2}f(c^2_{t'_1}[t_2]) = \sum_{t_2\in T}\alpha_{t_2}^{t'_2}H_{t_2}(c^2_{t'_1}) = H_{t'_2}(c^2_{t'_1})\\
&= H_t(c),
\end{aligned}$$

which proves the wanted invariant, namely $g_t(t') = \alpha_t^{t'}$. Hence, the value computed by the circuit for monomial t' is precisely

$$\sum_{t\in T} g_t(t')f(t) = \sum_{t\in T}\alpha_t^{t'}H_t(\square) = H_{t'}(\square) = f(t'),$$

which concludes the proof.

Appendix B. Proof of Corollary 2

We give a detailed proof of Corollary 2 which is a lower bound for unambiguous circuits computing the associative permanent or determinant. The proof holds in either the commutative or the non-commutative setting.

Corollary 2. *Any unambiguous circuit computing the determinant or the permanent has size at least* $\binom{n}{n/3}$.

Proof. Consider an unambiguous circuit computing the permanent (the proof is easily adapted to a circuit computing the determinant) of degree n on variables $X = \{x_{i,j} \mid i, j \in [n]\}$. For any permutation σ, let $t_\sigma \in \mathrm{Tree}(X)$ be the (non-associative) monomial along which there is a run computing the (associative) monomial $x_{1,\sigma(1)}x_{2,\sigma(2)} \cdots x_{n,\sigma(n)}$. Then, the non-associative polynomial $\tilde{f}$ computed by $\mathcal{C}$ when it is seen as a non-associative circuit is precisely $\tilde{f} = \sum_\sigma t_\sigma$. According to Theorem 1, it suffices to lower bound the rank of $H_{\tilde{f}}$.

Let $(A, S) \subseteq [n]^2$ be a pair of subsets. We let $T_{A\to S} \subseteq \mathrm{Tree}(X)$ be the subset of trees t such that the set of first (resp. second) indices of the labels of t is precisely A (resp. S). Symmetrically, let $C_{A\to S} \subseteq \mathrm{Context}(X)$ be the subset of contexts c such that the set of first (resp. second) indices of the labels (except for the $\square$) of c is precisely $[n] \setminus A$ (resp. $[n] \setminus S$). If $(A, S) \neq (A', S')$, then $T_{A\to S}$ and $T_{A'\to S'}$ are disjoint, as is the case for $C_{A\to S}$ and $C_{A'\to S'}$. Moreover, if $t \in T_{A\to S}$ and $c \in C_{A'\to S'}$, it must be that $\tilde{f}(c[t]) = 0$. Hence, $H_{\tilde{f}}$ is a block-diagonal matrix, with blocks $H_{A,S}$ being given by restricting the columns to some $T_{A\to S}$ and the rows to $C_{A\to S}$. Note that if $|A| \neq |S|$ then $H_{A,S} = 0$. In particular, $\mathrm{rank}\left(H_{\tilde{f}}\right) = \sum_{\substack{A,S\subseteq[n]\\|A|=|S|}} \mathrm{rank}\,(H_{A,S})$. We now show using a counting argument that an exponential number of such blocks are non-zero and hence, have rank at least 1.

For all permutations σ, we choose a subtree t'_σ of t_σ which has size in $[n/3, 2n/3]$, and let (A_σ, S_σ) be such that $t'_\sigma \in T_{A_\sigma\to S_\sigma}$. Note that $n/3 \leq |A_\sigma| = |S_\sigma| = |t'_\sigma| \leq 2n/3$, and that $H_{A_\sigma,S_\sigma} \neq 0$. Moreover, it must be that $\sigma(A_\sigma) = S_\sigma$. Hence, if $A, S \subseteq [n]$ are fixed such that $n/3 \leq |A| = |S| \leq 2n/3$,

$$|\{\sigma \mid A_\sigma = A \text{ and } S_\sigma = S\}| \leq |\{\sigma \mid \sigma(A) = S\}| \leq \left(\frac{n}{3}\right)!\left(\frac{2n}{3}\right)!$$

Hence, the number of non-zero blocks $H_{A,S}$ is at least

$$\frac{n!}{\left(\frac{n}{3}\right)!\left(\frac{2n}{3}\right)!} = \binom{n}{n/3}$$

which concludes the proof. $\square$

Appendix C. Proof of Corollary 3

We now prove Corollary 3, which characterizes the size of the smallest UPT circuit with given shape T computing the permanent.

Corollary 3. *Let $s \in$ Tree be a shape. The smallest UPT circuit with shape s computing the permanent has size*

$$\sum_{v \text{ node of } s} \binom{d}{|I_v|},$$

where I_v is the set of leaves in the subtree rooted at v in s. In particular, this is always larger than $\binom{d}{d/3}$.

Let s' be a sub-shape of s, and $v_1, ..., v_p$ be all the nodes of s such that $s_{v_i} = s'$. Let $\ell = |I_{v_i}|$ which does not depend on i. There are no $i \neq j$ such that v_i is a descendant of v_j, so the I_{v_i} are pairwise disjoint. Let $I_{v_i} = [a_i, a_i + \ell - 1]$. The coefficient of $M_{I_{v_i}}(\mathrm{Per})$ in $(u, w) \in X^{d-\ell} \times X^{\ell}$, namely, $\mathrm{Per}(u \otimes_{I_{v_i}} w)$, may be non-zero only if w is of the form $x_{a_i,b_1} x_{a_i+1,b_2} \cdots x_{a_i+\ell-1,b_\ell}$ for some $b_1, \ldots, b_\ell \in [d]$. In particular, the $M_{I_{v_i}}(\mathrm{Per})$ have non-zero columns with disjoint supports, so $\mathrm{rank}\,(M_{s'}) = \sum_i \mathrm{rank}\left(M_{I_{v_i}}(\mathrm{Per})\right)$.

We claim now that $\mathrm{rank}\left(M_{I_{v_i}}(\mathrm{Per})\right) = \binom{d}{\ell}$, which leads to the announced formula. Indeed, any subset A of $[d]$ of size ℓ corresponds to a block full of 1 in the matrix $M_{I_{v_i}}(\mathrm{Per})$ in the following way: $\mathrm{Per}(u \otimes_{I_{v_i}} w) = 1$ whenever u is a monomial whose first indices are $[d] \setminus I_{v_i}$ and the second indices are any permutation of $[d] \setminus A$, and w is a monomial whose first indices are I_{v_i} and the second indices are any permutation of A. Two such blocks have disjoint rows and columns, and these are the only 1's in $M_{I_{v_i}}(\mathrm{Per})$. Moreover, there are $\binom{d}{\ell}$ such sets A.

Appendix D. Proof of Theorem 4

This appendix is devoted to the proof of Theorem 4 that we recall below.

Theorem 4. *Let $f : X^d \to K$ be a non-commutative homogeneous polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span an interval at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}\,(M_A(f))\, n^{-\delta} |PT(\mathcal{C})|^{-1}$.*

The proof relies on a better understanding of the structure of the Hankel matrix $H = H_{\tilde{f}}$ of a general non-associative polynomial $\tilde{f} : \mathrm{Tree}(X) \to K$.

More precisely, we organize the columns androws of H in order to write it as a block matrix in which we can identify and understand the blocks in terms of partial derivative matrices of some non-commutative (but associative) polynomials which will eventually correspond to parse trees. In the following we refer to Figure 4 for illustration of the decompositions.

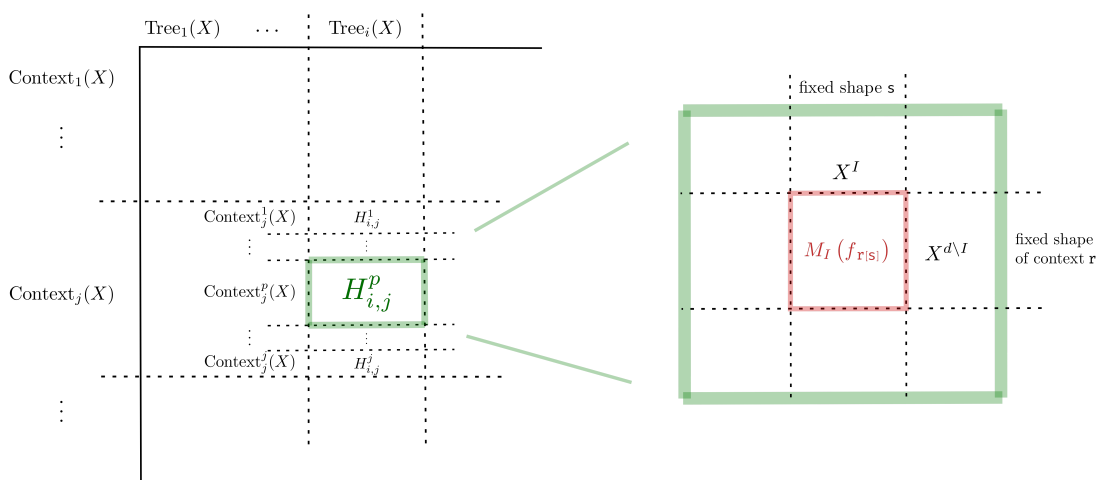

Figure 4. Decomposing H as blocks $H_{i,j}^p$, which further decompose into partial derivative matrices. Here, I denotes the interval $[p, p+i-1]$.

Recall that $\mathrm{Tree}_k(X) \subseteq \mathrm{Tree}(X)$ denotes the set of trees with k leaves, and let $\mathrm{Context}_k(X) \subseteq \mathrm{Context}(X)$ denote the set of contexts with k leaves (among which one is labelled by $\square$). We further partition $\mathrm{Context}_k(X) = \bigcup_{p=1}^{k} \mathrm{Context}_k^p(X)$, with $\mathrm{Context}_k^p(X)$ being the set of contexts where $\square$ is on the p-th leaf. Note that any tree $t \in \mathrm{Tree}_d(X)$ decomposes into $2d-1$ different couples $(t', c) \in \mathrm{Tree}_k(X) \times \mathrm{Context}_{d-k+1}(X)$ for some k, such that $c[t'] = t$, which correspond to the $2d-1$ nodes in t.

Using these partitions for trees and contexts, we may write H as a block matrix with blocks $H_{i,j} = H_{|\mathrm{Tree}_i(X)\times \mathrm{Context}_j(X)}$. Using the finer refinement of contexts, we write block $H_{i,j}$ as a tower[1] of sub-blocks $H^p_{i,j}$, for $p \in j$, where $H^p_{i,j} = H_{|\mathrm{Tree}_i(X)\times \mathrm{Context}^p_j(X)}$. We now focus on $H^p_{i,j}$, which we will decompose into blocks that are partial derivative matrices of some homogeneous non-commutative polynomials on the interval $[p, p+i-1]$.

As $\mathrm{Tree}_i(X)$ is the set of trees with i leaves, it can be seen as all possible labeling of shapes with i leaves by variables in X. Hence, $\mathrm{Tree}_i(X) \simeq \mathrm{Tree}_i \times X^i \simeq \mathrm{Tree}_i \times X^{[p,p+i-1]}$. Likewise, $\mathrm{Context}^p_j(X)$ is the set of contexts with j leaves and $\square$ on the p-th leave, which can be seen as $\mathrm{Context}^p_j(X) \simeq \mathrm{Context}^p_j \times X^{j-1} \simeq \mathrm{Context}^p_j \times X^{[1,i+j-1]\setminus[p,p+i-1]}$, where $\mathrm{Context}^p_j$ is the set of contexts of size j with no labels, except for a unique $\square$ on the p-th leaf. We now let, for any shape $s \in \mathrm{Tree}_{i+j-1}$, the non-commutative (but associative) homogeneous polynomial f_s of degree $i+j-1$ be defined by

$$
\begin{aligned}
f_s : X^{i+j-1} &\to K \\
u &\mapsto \tilde{f}(s \text{ labelled by } u)
\end{aligned}
$$

Now, grouping the columns $t \in \mathrm{Tree}_i(X)$ of $H^p_{i,j}$ which correspond to the same shape $s \in \mathrm{Tree}_i$, and the rows $c \in \mathrm{Context}^p_j(X)$ which correspond to the same shape (of context) $r \in \mathrm{Context}^p_j$, we obtain a block matrix, in which the block indexed by (s, r) is precisely the partial derivative matrix $M_{[p,p+i-1]}\left(f_{r[s]}\right)$.

In the following, we will be interested in non-associative polynomials $\tilde{f} : \mathrm{Tree}(X) \to K$ which project to a given associative $f : X^* \to K$, meaning that for each $u \in X^*$,

$$
\sum_{\substack{t\in \mathrm{Tree}(X)\\ \mathrm{label}(t)=u}} \tilde{f}(t) = f(u).
$$

In this setting, one can see the decomposition $f = \sum_{s\in\mathrm{Tree}} f_s$ as a decomposition over parse trees of a circuit computing f, f_s being the contribution of the parse tree s in the computation of f. We have seen that if $I = [p, p+i-1]$ is an interval such that s decomposes into $s = r[s']$ for $(s', r) \in \mathrm{Tree}_i \times \mathrm{Context}^p_j$, which means that I is spanned by s, then $M_I(f_s)$ appears as a sub-matrix of H. Hence,

$$
\operatorname{rank}(H) \geq \max_{\substack{s\in\mathrm{Tree}\\ I \text{ spanned by } s}} M_I(f_s). \tag{2}
$$

Now, we have all the necessary tools to prove Theorem 4. Let $\tilde{f} : \mathrm{Tree}(X) \to K$ be the non-associative polynomial computed by $\mathcal{C}$ when it is seen as a non-associative circuit. For any shape $s \in \mathrm{Tree}_d$, let $f_s : X^d \to K$ be defined as previously. In particular, $\sum_{s\in \mathrm{PT}(\mathcal{C})} f_s = f$.

With a shape $s \in \mathrm{PT}(\mathcal{C})$, we associate an interval $I(s)$ spanned by s and such that $\mathrm{dist}(A, I(s)) \leq \delta$. Then we have

[1] Recall that contexts label the *rows* of H.

$$
\begin{aligned}
\operatorname{rank}\left(M_A\left(f\right)\right) &= \operatorname{rank}\left(\sum_{s\in \mathrm{PT}(\mathcal{C})} M_A\left(f_s\right)\right) & \\
&\leq \sum_{s\in \mathrm{PT}(\mathcal{C})} \operatorname{rank}\left(M_A\left(f_s\right)\right) & \text{by rank subadditivity} \\
&\leq \sum_{s\in \mathrm{PT}(\mathcal{C})} n^{\delta} \operatorname{rank}\left(M_{I(s)}\left(f_s\right)\right) & \text{by Lemma 2} \\
&\leq \left|\mathrm{PT}\left(\mathcal{C}\right)\right| n^{\delta} \operatorname{rank}\left(H\right) & \text{by equation (2)}
\end{aligned}
$$

Since, by Theorem 1, $\operatorname{rank}\left(H\right) \geq \operatorname{rank}\left(M_A\left(f\right)\right) n^{-\delta} \left|\mathrm{PT}\left(\mathcal{C}\right)\right|^{-1}$ is a lower bound on $|\mathcal{C}|$, we obtain the announced result.

Appendix E. Proof of Theorem 5

This appendix is devoted to the proof of Theorem 5, which is a refinement of the proof of Theorem 4, given in Appendix D. In particular, we will use, without re-introducing them, some notations used in Appendix D.

Theorem 5. *Let f be a non-commutative homogeneous polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span an interval at distance at most δ from A. Then $\mathcal{C}$ has size at least $\operatorname{rank}\left(M_A\left(f\right)\right) n^{-\delta} d^{-2}$.*

Before going on to the formal proof, we start by giving a high-level interpretation of the techniques used to go from Theorem 4 to Theorem 5. Our aim is still to lower bound the rank of the Hankel matrix $H = H_{\tilde{f}}$ of some (unknown) non-associative polynomial $\tilde{f}$, under the constraints that, for each $u \in X^*$,

$$\sum_{\substack{t\in \mathrm{Tree}(X)\\ \mathrm{label}(t)=u}} \tilde{f}(t) = f(u),$$

for some non-commutative (but associative) polynomial $f : X^* \to K$ that we control. Given the form of our constraints, a natural strategy would be to sum some well chosen sub-matrices of H in order to obtain a matrix that depends only on f, which we could choose to have high rank.

As exposed earlier when proving Theorem 4, it is possible to decompose f as the sum of some f_s's, where s ranges over the shapes used by $\tilde{f}$, and then obtain partial derivative matrices of the f_s's with respect to interval spanned by s, as sub-matrices of H. If one can find a subset $A \subseteq [d]$ such that each s spans an interval $I(s)$ that is δ-close to A for some small δ, then one obtains a lower bound for polynomials f with high rank with respect to A.

This first method leads to Theorem 4 as exposed in Appendix D, and it is already strong enough to prove several lower bounds. We believe that in many occurrences in the literature, when obtaining lower bounds involving a circuit decomposition and a partial derivative matrix with respect to a given partition of the set of positions $[d]$, this is somehow the underlying method.

However, this method poorly makes use of the structure of H, since it may happen that some of the chosen sub-blocks are face to face with one another. A short illustration of this phenomenon is the following. Let

$$M = \left(\begin{array}{cc|cc} A_{1,1} & A_{1,2} & \multicolumn{2}{c}{\multirow{2}{*}{C_1}} \\ A_{2,1} & A_{2,2} & & \\ \hline \multicolumn{2}{c|}{\multirow{2}{*}{C_2}} & B_{1,1} & B_{1,2} \\ & & B_{2,1} & B_{2,2} \end{array}\right)$$

be a block matrix, for which one wants to obtain a lower bound on the rank, knowing a lower bound on $\operatorname{rank}\left(\sum_{i,j} A_{i,j} + B_{i,j}\right)$, and with no assumption on the C_i's.

The previous method would go as follows:

$$\begin{aligned}\operatorname{rank}(M) &\geq \max\left[\max_{i,j}\operatorname{rank}(A_{i,j}), \max_{i,j}\operatorname{rank}(B_{i,j})\right] \geq \frac{1}{8}\sum_{i,j}\operatorname{rank}(A_{i,j}) + \operatorname{rank}(B_{i,j}) \\ &\geq \frac{1}{8}\operatorname{rank}\left(\sum_{i,j} A_{i,j} + B_{i,j}\right).\end{aligned}$$

Note that we have lost a factor of 8, which is the number of small blocks that we wish to sum.

A more efficient method would consist in first summing rows and columns of M in order to put together the A's and the B's. This would go as follows, for some matrices C'_1 and C'_2,

$$\begin{aligned}\operatorname{rank}(M) \geq \operatorname{rank}\left(\begin{bmatrix}\sum_{i,j} A_{i,j} & C'_1 \\ C'_2 & \sum_{i,j} B_{i,j}\end{bmatrix}\right) &\geq \max\left[\operatorname{rank}\left(\sum_{i,j} A_{i,j}\right), \operatorname{rank}\left(\sum_{i,j} B_{i,j}\right)\right] \\ &\geq \frac{1}{2}\operatorname{rank}\left(\sum_{i,j} A_{i,j} + B_{i,j}\right).\end{aligned}$$

By doing so, we have decreased the factor 8 to 2, which is the number of larger blocks.

Going back to the matrix H, this corresponds to putting together the polynomials f_s for which we have chosen the same spanned interval (this corresponds to d^2 larger blocks) instead of considering them separately (which corresponds to $|\mathrm{PT}(\mathcal{C})|$ smaller blocks). We now formalize this idea, using a total order to model the choice of intervals for convenience.

Lemma 4. *Let $\tilde{f} : \mathit{Tree}(X) \to K$ be a non-associative non-commutative polynomial and let $\leq_{int}$ be a total order on intervals of $[d]$. For any shape s, we let $I(s)$ be the smallest (with respect to $\leq_{int}$) interval spanned by s. For any interval I, we define a non-commutative associative polynomial by*

$$\begin{aligned} f_I : X^* &\to K \\ u &\mapsto \sum_{\substack{t \in \mathit{Tree}(X) \\ \mathit{label}(t) = u \\ I(\mathit{shape}(t)) = I}} \tilde{f}(t).\end{aligned}$$

Then,

$$\mathit{rank}\left(H_{\tilde{f}}\right) \geq \max_I \mathit{rank}\left(M_I\left(f_I\right)\right).$$

Proof. Our aim is to obtain $M_I(f_I)$ from $H_{\tilde{f}}$, by first taking a sub-matrix, then adequately summing its rows and columns. The proof is summarized in Figure 5.

Let $I = [p, p+i-1]$ be some fixed interval and $j = d - i + 1$. The proof relies on the fact that for any shape $s \in \mathrm{Tree}_d$, $I = I(s)$ if and only if $s = r[s']$ for some $(s', r) \in \mathrm{Tree}_i \times \mathrm{Context}_j^p$ such that I is the smallest interval spanned by r (when it is assumed that $\square$ spans the positions I), and also the smallest interval spanned by s' (when it is assumed that all intervals are shifted by p), these two conditions being somehow independent.

Now, for any node v of shape of a context $r \in \mathrm{Context}_j^p$, we define the interval I'_v by

$$I'_v = \begin{cases} [a, b] \text{ if } b < p \\ [a, b+i-1] \text{ if } a \leq p \leq b \\ [a+i-1, b+i-1] \text{ if } a > p, \end{cases}$$

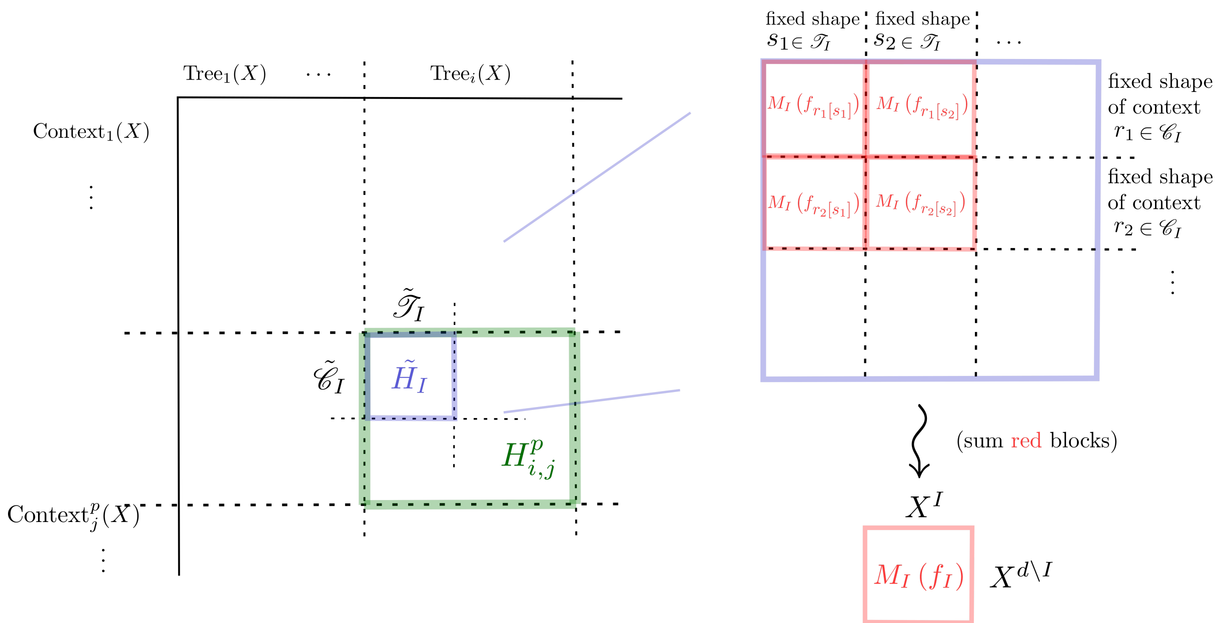

FIGURE 5. Decomposition of the Hankel matrix used in the proof of Lemma 4. Here, $I = [p, p+i-1]$.

where $[a,b]$ is the interval of positions in r of the leaves that are descendants of v in r. The interval I'_v is to be seen as the interval of positions of the leaves that are descendants of v in some $r[s']$ where s' is any element of Tree_i. In particular, if v is the leaf labelled by $\square$ in r, then $I'_v = I$.

Likewise, for a node v of a (sub)shape $s' \in \mathrm{Tree}_i$, we define I'_v by $I'_v = [a+p-1, b+p-1]$, where $[a,b]$ is the interval of positions of descendants of v in s'. Note that if v is the root of s' then $I_v = I$. We may now define

$$\mathscr{C}_I = \{r \in \mathrm{Context}^p_j \mid I = \min_{v \text{ node in } r} I'_v\},$$

and

$$\mathscr{T}_I = \{s' \in \mathrm{Tree}_i \mid I = \min_{v \text{ node in } s'} I'_v\}.$$

We extend these subsets to labelled trees and context in a straightforward fashion by defining $\tilde{\mathscr{C}}_I = \{c \in \mathrm{Context}^p_j(X) \mid \mathrm{shape}(c) \in \mathscr{C}_I\}$ and $\tilde{\mathscr{T}}_I = \{t \in \mathrm{Tree}_i(X) \mid \mathrm{shape}(t) \in \mathscr{T}_I\}$. We now consider the submatrix $\tilde{H}_I$ of $H^p_{i,j}$ where the rows are restricted to $\tilde{\mathscr{C}}_I$ and the columns to $\tilde{\mathscr{T}}_I$. In this matrix, we now sum the rows which have the same label, and the columns which have the same label, to obtain matrix H_I. Clearly, $\mathrm{rank}\left(H_I\right) \leq \mathrm{rank}\left(H_{\tilde{f}}\right)$. We finally prove that $H_I = M_I\left(f_I\right)$. Indeed, let $g \in X^I \simeq X^i$ and $h \in X^{d\setminus A} \simeq X^j$. Then

$$M_I\left(f_I\right)(g,h) = \sum_{\substack{t \in \mathrm{Tree}(X) \\ \mathrm{label}(t) = g \otimes_I h \\ I(\mathrm{shape}(t)) = I}} \tilde{f}(t) = \sum_{\substack{s \in \mathscr{T}_I \\ c \in \mathscr{C}_I \\ \mathrm{label}(s) = g \\ \mathrm{label}(c) = h}} \tilde{f}(c[s]) = H_I(g,h),$$

which concludes the proof of Lemma 4. $\square$

With Lemma 4 in hands, we may now prove Theorem 5. Let $\tilde{f} : \mathrm{Tree}(X) \to K$ be the non-associative polynomial computed by $\mathcal{C}$ when seen as non-associative. Let $\leq_{int}$ be a total order on intervals of d such that $I \mapsto \mathrm{dist}(I, A)$ is non-decreasing. In other words, $I_1 <_{int} I_2$ if and only if $d(I_1, A) < d(I_2, A)$. Let $f_I : X^d \to K$ be given by

$$f_I(u) = \sum_{\substack{t \in \mathrm{Tree}(X) \\ \mathrm{label}(t)=u \\ I(\mathrm{shape}(t))=I}} \tilde{f}(t).$$

Then any interval I such that $d(I, A) > \delta$ is such that for every parse tree $s \in \mathrm{PT}(\mathcal{C})$, one has $I \neq I(s)$, so $f_I = 0$. Hence, we obtain

$$\begin{aligned}
\mathrm{rank}\left(M_A\left(f\right)\right) &= \mathrm{rank}\left(M_A\left(\sum_{I \text{ interval of } [d]} f_I\right)\right) \\
&= \mathrm{rank}\left(M_A\left(\sum_{\substack{I \text{ interval of } [d] \\ \mathrm{dist}(A,I) \le \delta}} f_I\right)\right) \\
&\le \sum_{\substack{I \text{ interval of } [d] \\ \mathrm{dist}(A,I) \le \delta}} \mathrm{rank}\left(M_A\left(f_I\right)\right) && \text{by rank subadditivity} \\
&\le \sum_{\substack{I \text{ interval of } [d] \\ \mathrm{dist}(A,I) \le \delta}} n^\delta \mathrm{rank}\left(M_I\left(f_I\right)\right) && \text{by Lemma 2} \\
&\le d^2 n^\delta \mathrm{rank}\left(H_{\tilde{f}}\right) && \text{by Lemma 4}
\end{aligned}$$

which leads the announced lower bound.

Appendix F. Proof of Theorem 6

We now give the proof of Theorem 6 which is the following. As this proof is an adaptation to the commutative setting of the proof of Theorem 4 given in Appendix D, we only highlight the changes.

Theorem 6. *Let f be a set-multilinear polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span a subset at distance at most δ from A. Then $\mathcal{C}$ has size at least $\mathrm{rank}\left(M_A\left(f\right)\right) n^{-\delta} |PT(\mathcal{C})|^{-1}$.*

Let $X_1 \sqcup X_2 \sqcup \cdots \sqcup X_d = X$ denote the underlying partition. Previously, we grouped together (sub-)trees and (sub-)contexts which correspond to a given interval of positions. In the commutative setting, we instead group together the (sub-)trees and (sub-)contexts which correspond to a given *subset* of positions, where a position is now being given by its index in the partition. Formally, for $A \subseteq [d]$, we let

$$\mathrm{Tree}_A(X) = \{t \in \mathrm{Tree}(X) \mid \text{ the set of indices of variables labeling } t \text{ is } A\},$$

and likewise,

$$\mathrm{Context}_A(X) = \{c \in \mathrm{Context}(X) \mid \text{ the set of indices of variables (different from } \square) \text{ labeling } c \text{ is } A\},$$

and finally $H_A = H_{|_{\mathrm{Tree}_A(X) \times \mathrm{Context}_{A^C}(X)}}$.

Now, grouping together the columns of H_A which correspond to trees which have a given fixed shape s' (recall that a commutative shape contains the index in the partition of each leaf), and the rows which correspond to contexts which have a given fixed shape of context r yields the partial derivative matrix $M_A\left(f_{r[s']}\right)$, where the (commutative, associative) polynomial f_s is defined, for any commutative shape s, by

$$f_s(u) = \tilde{f}(s \text{ labelled by } u),$$

where the labeling respects the partition of X. Hence, $\operatorname{rank}(H) \geq \operatorname{rank}(M_A(f_s))$ whenever A is spanned by s. The remainder of the proof exactly follows Appendix D.

Appendix G. Proof of Theorem 7

We now give the proof of Theorem 7 which is the following.

Theorem 7. *Let f be a set-multilinear polynomial computed by a circuit $\mathcal{C}$. Let $A \subseteq [d]$ and $\delta \in \mathbb{N}$ such that all parse trees of $\mathcal{C}$ span a subset at distance at most δ from A. Then $\mathcal{C}$ has size at least rank$(M_A(f))\, n^{-\delta} 2^{-d}$.*

Again, we extend the ideas for the non-commutative setting (see Appendix E) to the commutative setting, and we reuse the notations of Appendix F. As for proving Theorem 5, we start with a Lemma.

Lemma 5. *Let $\tilde{f} : Tree(X) \to K$ be a non-associative commutative polynomial and let $\leq_{int}$ be a total order on subsets of $[d]$. For any commutative shape s, we let $A(s)$ be the smallest (with respect to $\leq_{int}$) subset spanned by s. For any subset A, we define a commutative associative polynomial by*

$$f_A(u) = \sum_{\substack{t \in Tree(X) \\ label(t)=u \\ A(shape(t))=A}} \tilde{f}(t).$$

Then,

$$rank\left(H_{\tilde{f}}\right) \geq \max_A rank\left(M_A\left(f_A\right)\right).$$

The proof of Lemma 5 is very similar, yet (surprisingly!) a bit more pleasant than that of Lemma 4, since we no longer need to shift any interval. Formally, for $A \subseteq [d]$ we define

$$\mathscr{T}_A = \{t \in \text{Tree}_A(X) \mid A \text{ is the smallest interval spanned by shape}(t)\},$$

and likewise,

$$\mathscr{C}_A = \{c \in \text{Context}_A(X) \mid A \text{ is the smallest interval spanned by shape}(c)\}.$$

Now, the lemma follows from the fact that $M_A(f_A)$ is obtained by summing rows from $\mathscr{T}_A$ and columns from $\mathscr{C}_A$ in H.

The remainder of the proof is a very straightforward adaptation of the end of the proof of Theorem 5 from Appendix E.

Appendix H. Proof of Theorem 8

We now give the proof of Theorem 8 which is the following.

Theorem 8. *Let f be a homogeneous non-commutative polynomial such that $M_{[d/4+1,3d/4]}(f)$ has full rank $n^{d/2}$. Then any skew circuit computing f has size at least $2^{-d}n^{d/4}$.*

The proof relies on the following easy observations.

- Any skew tree spans intervals of each possible size, and in particular, an interval of size $3d/4$.
- Any interval of size $3d/4$ is at distance at most (in fact, equal to) $d/4$ from $I_{mid} = [d/4 + 1, 3d/4]$ (see Figure 6).

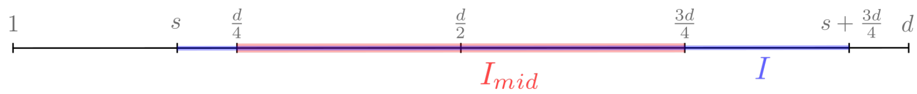

FIGURE 6. Any interval I of size $\frac{3d}{d}$ is at distance $\frac{d}{4}$ from I_{mid}.

A skew circuit has only skew parse trees, which all span an interval of size $3d/4$. Such an interval is at distance $d/4$ from I_{mid}, so the announced lower bound follows directly from Theorem 4, together with the fact that there are 2^d skew trees.

Remark 2. *Note that the factor 2^{-d} is easily replaced by d^{-2} by applying Theorem 5 instead, but we find it remarkable that simply using a decomposition of H into blocks is enough to obtain such an exponential lower bound.*

Appendix I. Proof of Theorem 9

We now give the details for the exponential lower bound on $\left(\frac{1}{2}-\varepsilon\right)$-unbalanced circuits. This is really the same idea as for skew circuits. Note that we use Theorem 4 with δ being really close to $d/2$, which will also be the case for k-PT circuits.

Theorem 9. *Let f be a homogeneous non-commutative polynomial such that $M_{[d/4+1,3d/4]}(f)$ has full rank $n^{d/2}$. Then any $\left(\frac{1}{2}-\varepsilon\right)$-unbalanced circuit computing f has size at least $4^{-d}n^{\varepsilon d}$.*

We now rely on these two observations:

- Any $(\frac{1}{2}-\varepsilon)$-unbalanced shape spans an interval of size between $3d/4-(\frac{1}{2}-\varepsilon)d/2$ and $3d/4+(\frac{1}{2}-\varepsilon)d/2$, that is, between $d/2+d\varepsilon/2$ and $d-d\varepsilon/2$.
- Any such interval is at distance at most $d/2-\varepsilon/2$ from $[d/4,3d/4]$.

We finally conclude by applying Theorem 4, just as for skew circuits.

Appendix J. Proof of Theorem 10

We now give the details for the exponential lower bound on ε-balanced circuits.

Theorem 10. *Let f be a homogeneous non-commutative polynomial such that $M_{[1,d/2]}(f)$ has full rank $n^{d/2}$. Then any ε-balanced circuit computing f has size at least $4^{-d}n^{\varepsilon d}$.*

Let s be an ε-balanced shape, and r be the root of s. Let $I=[1,b]$ be the interval spanned by the left child of r. Since s is ε-balanced, $\varepsilon d\leq|I|=b\leq(1-\varepsilon)d$. Hence, I is at a distance of atmost $d/2-\varepsilon$ from $[1,d/2]$, which allows us to conclude using Theorem 4. Note that it is sufficient to just restrict the *last* multiplication in the circuit to be ε-balanced.

Appendix K. Proof of Theorem 11

This appendix is devoted to the proof of Theorem 11 that we recall below. We will make extended use of the subset $A\subseteq[d]$ introduced in [Limaye et al., 2016],

$$A=[1,3kp]\cup\bigcup_{i=1}^{3k}[3(k+i)p+2p+1,3(k+i+1)p]\subseteq[d],$$

of size $d/2$ which is better understood in Figure 7.

Theorem 11. *Let f be a homogeneous non-commutative polynomial of degree $d=12kp$ such that $M_A(f)$ has full rank $n^{d/2}$. Then any circuit of non-skew depth k computing f has size at least $4^{-d}n^{p/3}=4^{-d}n^{d/36k}$.*

We shall prove that any $s\in\mathrm{Tree}_d$ with non-skew depth k spans an interval $I(s)$ at distance $\leq d/2-p/3$ from A. Then, the result follows by applying Theorem 4.

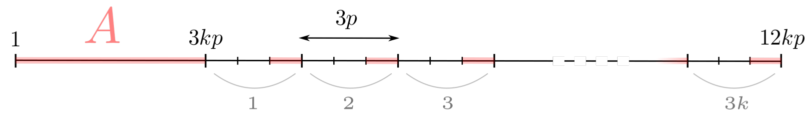

FIGURE 7. Subset $A \subseteq [d]$.

Assume towards contradiction that a non-skew depth k shape $s \in \mathrm{Tree}_d$ spans only interval at distance $> d/2 - p/3$ from A. We consider (see Figure 8) the path $v_1 \cdots v_r$ in s from its root to the leaf with position $3kp$, and write u_i for $i \in r-1$, to refer to the child node of v_i which is not v_{i+1} (see Figure 8). Since s has non-skew depth k, at least $r-k$ nodes among $v_1, \dots, v_{r-1}$ are leaves.

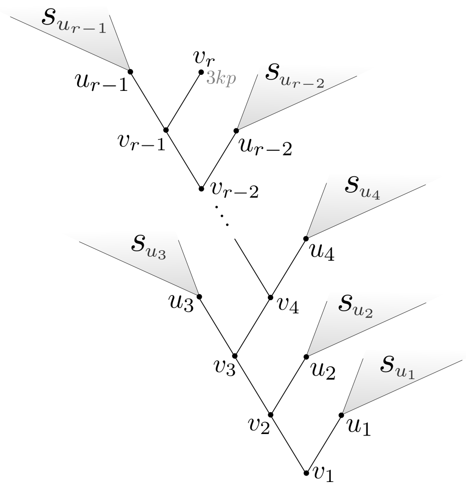

FIGURE 8. The path from the root v_1 to v_r, the leaf with position $3kp$.

We now state and prove some facts which then lead to a contradiction:

Fact 1. *For every $i \in [r]$, if v_i is the left child of u_i then $|I_{v_i}| < p/3$.*

Indeed, v_i being at the left of the path to the leaf at position $3kp$, $I_{v_i} \subseteq [1, 3kp] \subseteq A$. But $\mathrm{dist}(I_{v_i}, A) > d/2 - p/3$, so it must be that $|I_{v_i}| < p/3$.

Fact 2. *For every $i \in [r]$, if v_i is the right child of u_i then $|I_{v_i}| < 5p$.*

Likewise, we now have $I_{v_i} \subseteq [3kp + 1, d]$. Intuitively, a large interval in this zone must contain roughly twice as much elements from A^{C} than from A, so they cannot be at distance close to the maximum $d/2$. Formally, each block of the form $[3(k+i)p+2p+1, 3(k+i+1)p] \subseteq A$ which intersects I_{v_i}, apart possibly from the rightmost one, is such that $[3(k+i+1)p, 3(k+i+1)p+2p] \subseteq A^{\mathrm{C}}$ is contained in I_{v_i}. Now, if l is the number of such blocks, it follows that $|I_{v_i} \cap A| \leq lp + p$ and $|I_{v_i} \cap A^{\mathrm{C}}| \geq 2lp$. If $|I_{v_i}| > 5p$, then either $l \geq 2$ which implies $d(A, I_{v_i}) = d/2 - (|A^{\mathrm{C}} \cap I_{v_i}| - |A \cap I_{v_i}|) \leq d/2 - 2lp + lp - p \leq d/2 - p$, a contradiction, or $l = 1$, in which case $|I_{v_i} \cap A^{\mathrm{C}}| = |I_{v_i}| - |I_{v_i} \cap A| \geq 5p - 2p = 3p$ which leads to the same contradiction.

Fact 3. *It must be that $r \geq 7kp$.*

Indeed, since $[1, d] \setminus \{3kp\} = [1, 12kp] \setminus \{3kp\}$ is covered by the I_{v_i}, which have size bounded by $5p$ and among which all but k may have size > 1, there must be at least $12kp - 5kp = 7kp$ of them.

Fact 4. *There is some index i_0 such that $v_{i_0}, v_{i_0+1}, \dots, v_{i_0+7p-1}$ are all leaves in s.*

Indeed, only k among the $7kp$ v_i's may not be leaves, so there must be $7p$ successive indexes i such that v_i is a leaf.

We now consider the decreasing sequence $I_{v_{i_0}} \supseteq I_{v_{i_0+1}} \supseteq \cdots \supseteq I_{v_{i_0+7p-1}}$ of intervals (where the nodes $v_{i_0}, v_{i_0+1}, \ldots, v_{i_0+7p-1}$ are those given by Fact 4), which we simply denote $I_1 \supseteq I_2 \supseteq \cdots \supseteq I_{7p}$. Each $I_i = [a_i, b_i]$ contains $3kp$, and $|I_{i+1}| = |I_i| + 1$. We put $n_i = |I_i \cap A|$ and $m_i = |I_i \cap A^{\mathrm{C}}|$. Fact 1 stating that $d(A, I_i) > d/2 - p/3$ can be rewritten as $|n_i - m_i| \leq p/3$. We now prove that there must be $i_1 \leq 6p$ such that b_{i_1} is a multiple of $3p$.

Indeed, otherwise $b_1 - b_{6p} \leq 3p - 1$ so $a_{6p} - a_1 \geq 3p + 1$, hence $n_{6p} - n_1 \geq a_{6p} - a_1 \geq 3p + 1$, but since $m_{6p} - m_1 \leq 2p$,

$$p/3 \geq n_{6p} - m_{6p} \geq 3p + 1 + n_1 - m_{6p} \geq 3p + 1 + n_1 - m_1 - 2p \geq p + n_1 - m_1,$$

so $n_1 - m_1 \leq -2p/3$, a contradiction.

Since b_{i_1} is a multiple of $3p$, both intervals $[a_{i_1}, a_{i_1} + p - 1]$ and $[b_{i_1} - p + 1, b_{i_1}]$ are contained in A. Hence, $n_{i_1+p} = n_{i_1} + p$, whereas $m_{i_1+p} = m_{i_1}$, which contradicts the fact that $|n_{i_1} - m_{i_1}| \leq p/3$ and $|n_{i_1+p} - m_{i_1+p}| \leq p/3$.

Appendix L. Proof of Lemma 3

We now prove the main technical result to obtain lower bound on k-PT, which is adapted from [Lagarde et al., 2018] in our vocabulary. It holds in both the commutative and the non-commutative settings (even though it was originally proved only in the non-commutative setting).

Lemma 3 (adapted from Claim 15 in [Lagarde et al., 2018]]). *Let $s \in \mathit{Tree}_d$ be a shape with d leaves, and $\delta \leq \sqrt{d}$. Then*

$$\Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\mathit{dist}(A, s) > d/2 - \delta\right] \leq 2^{-\alpha d/\delta^2},$$

where α is some positive constant and $\mathcal{U}\left(\binom{[d]}{d/2}\right)$ the uniform distribution of subsets of d of size $d/2$.

We shall use an intermediate result from the aforementioned paper. Their proof can be read just as such in the commutative setting.

Lemma 6 (Subclaim 21 in [Lagarde et al., 2018]). *Let $s \in \mathit{Tree}_d$, and r, t be integers such that $rt \leq d/4$. Then there exists a sequence $v_1, ..., v_r$ of nodes of s such that for all $i \in [r]$,*

$$\left| I_{v_i} \setminus \left(\bigcup_{j=1}^{i-1} I_{u_j} \right) \right| \geq t.$$

In the commutative setting, replace the spanned intervals of the form I_v by spanned subsets of the form A_v in the statement above as well as in the proof below. We now prove Lemma 3. We pick $t = \delta^2$ and $r = \frac{d}{4\delta^2}$, and apply Lemma 6 to obtain sequence $v_1, ..., v_r$ of nodes of s. We first note that if X and Y are two sets and X has size $d/2$ then $\mathrm{dist}(X, Y)$ rewrites as $\mathrm{dist}(X, Y) = d/2 - ||X \cap Y| - |X^{\mathrm{C}} \cap Y||$. As $\mathrm{dist}(A, s) = \min\{\mathrm{dist}(A, I) \mid I \text{ spanned by } s\}$, the previous remark leads the first equality below.

$$\Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\mathrm{dist}(A, s) > d/2 - \delta\right] = \Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\text{for all node } v \text{ of } s, \left||A \cap I_v| - |A^{\mathrm{C}} \cap I_v|\right| \leq \delta\right]$$

$$\leq d \Pr_{A \sim \mathcal{U}\left(2^{[d]}\right)} \left[\text{for all node } v \text{ of } s, \big||A \cap I_v| - |A^{\mathrm{C}} \cap I_v|\big| \leq \delta\right] \qquad \text{as } \tbinom{d}{d/2}/2^d \leq d$$

$$\leq d \Pr_{A \sim \mathcal{U}\left(2^{[d]}\right)} \left[\forall i \in [r], \big||A \cap I_{v_i}| - |A^{\mathrm{C}} \cap I_{v_i}|\big| \leq \delta\right]$$

$$\leq d \prod_{i=1}^{r} \Pr_{A \sim \mathcal{U}\left(2^{[d]}\right)} \left[\big||A \cap I_{v_i}| - |A^{\mathrm{C}} \cap I_{v_i}|\big| \leq \delta \,\middle|\, A \cap \left(\bigcup_{j<i} I_{u_j}\right)\right]$$

If A is sampled uniformly among $[d]$ and $A \cap \left(\bigcup_{j<i} I_{u_j}\right)$ is fixed, realizing the event $\big||A \cap I_{v_i}| - |A^{\mathrm{C}} \cap I_{v_i}|\big| \leq \delta$ amounts to having a random variable following an unbiased binomial law of size at least $t = \delta^2$ sit in a certain interval of size at most δ, which is bounded by a constant $\beta < 1$. Hence,

$$\Pr_{A \sim \mathcal{U}\left(\binom{[d]}{d/2}\right)} \left[\operatorname{dist}(A, s) > d/2 - \delta\right] \leq d\beta^r = d\beta^{\frac{d}{4\delta^2}} \leq 2^{-\alpha d/\delta^2}$$

for some positive constant α.

References

Agrawal, M., Gurjar, R., Korwar, A., and Saxena, N. (2015). Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697.

Allender, E., Jiao, J., Mahajan, M., and Vinay, V. (1998). Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theoretical Computer Science*, 209(1-2):47–86.

Arvind, V. and Raja, S. (2016). Some lower bound results for set-multilinear arithmetic computations. *Chicago Journal of Theoretical Computer Science*, 2016.

Baur, W. and Strassen, V. (1983). The complexity of partial derivatives. *Theoretical Computer Science*, 22:317–330.

Bozapalidis, S. and Louscou-Bozapalidou, O. (1983). The rank of a formal tree power series. *Theoretical Computer Science*, 27:211–215.

Carmosino, M. L., Impagliazzo, R., Lovett, S., and Mihajlin, I. (2018). Hardness amplification for non-commutative arithmetic circuits. In *Proceedings of the 33rd Computational Complexity Conference (CCC 2018)*, volume 102 of *LIPIcs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Dvir, Z., Malod, G., Perifel, S., and Yehudayoff, A. (2012). Separating multilinear branching programs and formulas. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC 2012)*, pages 615–624. ACM.

Fijalkow, N., Lagarde, G., and Ohlmann, P. (2018). Tight bounds using hankel matrix for arithmetic circuits with unique parse trees. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:38.

Fliess, M. (1974). Matrices de Hankel. *Journal de Mathématiques Pures et Appliquées*, 53:197–222.

Forbes, M. A., Saptharishi, R., and Shpilka, A. (2014). Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Symposium on Theory of Computing, (STOC 2014)*, pages 867–875. ACM.

Gurjar, R., Korwar, A., Saxena, N., and Thierauf, T. (2017). Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Computational Complexity*, 26(4):835–880.

Hrubeš, P., Wigderson, A., and Yehudayoff, A. (2010). Relationless completeness and separations. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC 2010)*, pages 280–290. IEEE Computer Society.

Hrubeš, P., Wigderson, A., and Yehudayoff, A. (2011). Non-commutative circuits and the sum-of-squares problem. *Journal of the American Mathematical Society*, 24(3):871–898.

Kabanets, V. and Impagliazzo, R. (2003). Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 355–364. ACM.

Kayal, N., Saha, C., and Saptharishi, R. (2014). A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the 46th Symposium on Theory of Computing, (STOC 2014)*, pages 146–153. ACM.

Lagarde, G., Limaye, N., and Srinivasan, S. (2018). Lower bounds and PIT for non-commutative arithmetic circuits with restricted parse trees. *Computational Complexity*, pages 1–72. https://doi.org/10.1007/s00037-018-0171-9.

Lagarde, G., Malod, G., and Perifel, S. (2016). Non-commutative computations: lower bounds and polynomial identity testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:94.

Limaye, N., Malod, G., and Srinivasan, S. (2016). Lower bounds for non-commutative skew circuits. *Theory of Computing*, 12(1):1–38.

Malod, G. and Portier, N. (2008). Characterizing valiant's algebraic complexity classes. *Journal of Complexity*, 24(1):16–38.

Nisan, N. (1991). Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd Symposium on Theory of Computing (STOC 1991)*, pages 410–418. ACM.

Nisan, N. and Wigderson, A. (1994). Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167.

Ramya, C. and Rao, B. V. R. (2018). Lower bounds for special cases of syntactic multilinear abps. In *Proceedings of the 24th International Computing and Combinatorics Conference (COCOON 2018)*, volume 10976 of *Lecture Notes in Computer Science*, pages 701–712. Springer.

Raz, R. and Shpilka, A. (2005). Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19.

Saptharishi, R. and Tengse, A. (2017). Quasi-polynomial hitting sets for circuits with restricted parse trees. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:135.

Toda, S. (1992). Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Trans. Inf. Systems*, E75-D(1):116–124.

Valiant, L. G. (1979). The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il