Electronic Colloquium on Computational Complexity, Report No. 77 (2017)

Lower Bounds and PIT for Non-Commutative Arithmetic circuits with Restricted Parse Trees

Guillaume Lagarde* Nutan Limaye† Srikanth Srinivasan‡

Abstract

We investigate the power of *Non-commutative Arithmetic Circuits*, which compute polynomials over the free non-commutative polynomial ring $\mathbb{F}\langle x_1,\ldots,x_N\rangle$, where variables do not commute. We consider circuits that are restricted in the ways in which they can compute monomials: this can be seen as restricting the families of *parse trees* that appear in the circuit. Such restrictions capture essentially all non-commutative circuit models for which lower bounds are known. We prove several results about such circuits.

1. We show explicit exponential lower bounds for circuits with up to an exponential number of parse trees, strengthening the work of Lagarde, Malod, and Perifel (ECCC 2016), who prove such a result for *Unique Parse Tree* (UPT) circuits which have a single parse tree.
2. We show explicit exponential lower bounds for circuits whose parse trees are rotations of a single tree. This simultaneously generalizes recent lower bounds of Limaye, Malod, and Srinivasan (Theory of Computing 2016) and the above lower bounds of Lagarde et al., which are known to be incomparable.
3. We make progress on a question of Nisan (STOC 1991) regarding separating the power of Algebraic Branching Programs (ABPs) and Formulas in the non-commutative setting by showing a tight lower bound of $n^{\Omega(\log d)}$ for any UPT formula computing the product of d $n\times n$ matrices.

   When $d \leq \log n$, we can also prove superpolynomial lower bounds for formulas with up to $2^{o(d)}$ many parse trees (for computing the same polynomial). Improving this bound to allow for $2^{O(d)}$ trees would yield an unconditional separation between ABPs and Formulas.
4. We give deterministic white-box PIT algorithms for UPT circuits over any field (strengthening a result of Lagarde et al. (2016)) and also for sums of a constant number of UPT circuits with different parse trees.

*Univ Paris Diderot, Sorbonne Paris Cité, IRIF, UMR 7089 CNRS, F-75205 Paris, France. Email: guillaume.lagarde@irif.fr.

†Department of Computer Science and Engineering, IIT Bombay, Mumbai, India. Email: nutan@cse.iitb.ac.in.

‡Department of Mathematics, IIT Bombay, Mumbai, India. Email: srikanth@math.iitb.ac.in.

ISSN 1433-8092

1 Introduction

In this paper, we study questions related to Arithmetic Circuits, which are computational devices that use arithmetic operations (such as $+$ and $\times$) to compute multivariate polynomials over a field $\mathbb{F}$. While the more standard work in this area deals with the commutative polynomial ring $\mathbb{F}[x_1,\ldots,x_N]$, there is also a line of research, initiated by Hyafil [14] and Nisan [22], that studies the complexity of computing polynomials from the *non-commutative* polynomial ring $\mathbb{F}\langle x_1,\ldots,x_N\rangle$, where monomials are simply strings over the alphabet $X = \{x_1,\ldots,x_N\}$. The motivation for this is twofold: firstly, the study of polynomial computations over non-commutative algebras (e.g. the algebra of matrices over $\mathbb{F}$) naturally leads to such questions [8, 7], and secondly, computing, say, the Permanent non-commutatively[1] is at least as hard as computing it in the commutative setting and thus, the lower bound question should be easier to tackle in this setting.

In an influential result, Nisan [22] justified this by proving exponential lower bounds for non-commutative formulas, and more generally Algebraic Branching Programs (ABPs), computing the Determinant and Permanent (and also other polynomials). The method used by Nisan to prove this lower bound can also be seen as a precursor to the method of Partial derivatives in Arithmetic circuit complexity (introduced by Nisan and Wigderson [23]), variants of which have been used to prove a large body of lower bound results in the area [23, 26, 11, 16, 18].

While lower bounds for general non-commutative circuits remain elusive, we do have other lower bounds that strengthen Nisan's result. Recently, Malod, along with two of the authors of this paper showed [20] that Nisan's method can be extended to prove lower bounds for skew circuits, which are circuits where every $\times$-gate has at most one non-variable input. Also, the first author, Malod and Perifel [19] proved lower bounds for another variant of non-commutative circuits that they defined to be *unambiguous* circuits (that we describe below). While these two results both strengthen Nisan's result, they are incomparable to each other, as shown by [19].

In this paper, we build on the above work to prove lower bounds that generalize these results significantly and also make progress on other problems related to non-commutative circuits. The circuits we consider are restricted in the ways they are allowed to compute monomials. We do this by restricting the "parse trees" that are allowed to appear in the circuits. Informally, the polynomial computed by any arithmetic circuit C can be written down as an exponentially-large sum of subcircuits, each of which contains only multiplication gates and hence computes a single monomial[2]; each such subcircuit gives rise to a tree, which we call a *parse tree of* C (see [19] and references therein for the background of parse trees), that tells us how the monomial was computed. For example, for the circuit C in Figure 1, the monomial $x_1x_2x_3x_4$ may be computed in C as $(x_1 \cdot x_2) \cdot (x_3 \cdot x_4)$ or as $(x_1 \cdot (x_2 \cdot x_3) \cdot x_4)$, each of which comes from a parse tree of C.

All the non-commutative circuit classes for which we know lower bounds can be defined by restrictions on the parse trees that appear in them. ABPs are circuits where all parse trees are left combs (i.e. a tree where every internal node has two children and the left child is always a leaf); skew circuits are equivalent in power to circuits where the parse trees are *twisted* combs (i.e. a tree where every internal node has two children and at least one of the two children is always a leaf); and unambiguous circuits (that we will call *Unique Parse Tree* (UPT) circuits below) are defined to be circuits that have only one parse tree. It is thus natural to consider other restrictions on the structure of the parse trees that appear in a circuit. We prove several results about such circuits.

Our results. We start by considering circuits that only contain a few different parse trees. The motivation for this is the lower bound of [19] for the case of circuits with a single parse tree and a construction

[1] We can define the Permanent in the non-commutative polynomial ring by ordering the variables in each monomial in the commutative permanent, say, in increasing order of the rows in which they appear.

[2] different subcircuits could compute the same monomial

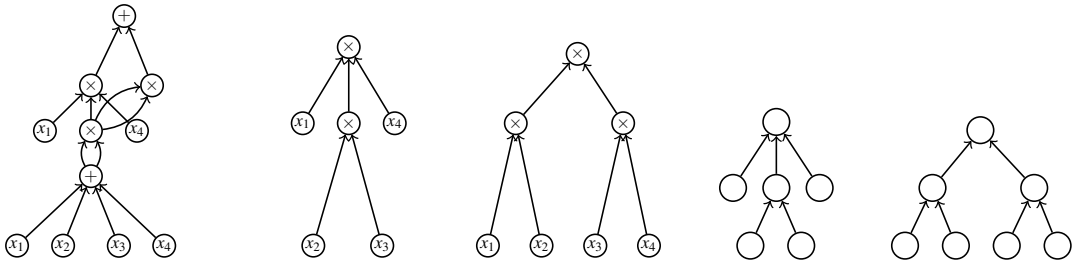

Figure 1: From left to right: a non-commutative arithmetic circuit C; two ways in which the monomial $x_1x_2x_3x_4$ is computed in the circuit; the corresponding parse trees

in [20] that shows that poly(N,d)-sized circuits with $\exp(\Omega(d))$ many parse trees[3] evade all currently known techniques for proving lower bounds for non-commutative circuits. We prove exponential lower bounds for circuits that contain up to an exponential number of parse trees.

Theorem 1 (Informal)**.** *For any $N \geq 2$, there is an explicit polynomial F on N variables of degree d such that any circuit C with at most $2^{d^{1/4}}$ parse trees computing F must have size at least $2^{d^{1/4}}$.*

Next, we consider structural restrictions on the collections of parse trees that appear in our circuits. As mentioned above, skew circuits are circuits where all parse trees are twisted combs, which can be seen as trees obtained by starting with a left comb (which defines an ABP) and successively applying rotations to the internal nodes that swap the children. We say that a circuit C is *rotation Unique Parse tree* (rotUPT) if there is a single tree T such that all the parse trees of C can be obtained as rotations of T.[4] We show the following result that simultaneously generalizes the skew circuit lower bound of Limaye et al. [20] as well as the UPT circuit lower bound of Lagarde et al. [19].

Theorem 2 (Informal)**.** *For any $N \geq 2$, there is an explicit polynomial F on N variables of degree d such that any rotUPT circuit C computing F must have size $N^{\Omega(d)}$.*

We also consider the problem of separating ABPs from formulas, which was posed by Nisan [22] and is the non-commutative arithmetic analogue of separating NL from NC^1. Equivalently, this is the question of whether (an entry of) the product of d $n \times n$ matrices, all of whose entries are distinct variables, can be computed by a poly(n,d)-sized non-commutative formula. The standard divide-and-conquer approach yields, for every even Δ, a non-commutative formula of depth Δ and size $n^{O(\Delta d^{2/\Delta})}$ computing this polynomial and a size $n^{O(\log d)}$ formula in general. Further, these formulas can be seen to have a *unique* parse tree (i.e. they are UPT). We show that this upper bound is nearly tight for UPT formulas and every choice of Δ.

Theorem 3 (Informal)**.** *Any UPT formula of depth Δ for multiplying d $n \times n$ matrices must have size $n^{\Omega(\Delta d^{1/\lfloor \Delta/2 \rfloor})}$.*[5] *In particular, any UPT formula for this polynomial must have size $n^{\Omega(\log d)}$.*

We are also able to extend this to the setting of formulas with 'few' parse trees. However, for this result, we need an upper bound on the number d of matrices (see Remark 36 for more on this).

Theorem 4 (Informal)**.** *Say $d \leq \log n$. Any formula with $k \leq 2^{o(d)}$ parse trees for multiplying d $n \times n$ matrices must have size $n^{\omega(1)}$.*

Finally, we consider the Polynomial Identity Testing (PIT) problem for non-commutative circuits with restricted parse trees. Lagarde et al. [19] show that deterministic PIT algorithms for UPT circuits

[3] A close look at the circuits in [20] indicates that just about all parse trees of fan-in 2 appear in these circuits.

[4] There can be $\exp(\Omega(d))$ of these, as in the case of skew circuits.

[5] Our bounds are actually better stated in terms of the $\times$-*depth* of the formula.

can be obtained by adapting a PIT algorithm for ABPs due to Arvind, Joglekar and Srinivasan [3]. However, this technique only works over fields of characteristic zero. Here, we give a straightforward adaptation of an older PIT algorithm of Raz and Shpilka [25] (also for non-commutative ABPs) to show that PIT for UPT circuits can be solved in deterministic polynomial time over all fields. We also consider circuits that are sums of UPT circuits (with possibly different parse trees). By using ideas from the work of Gurjar, Korwar, Saxena and Thierauf [12], we show that PIT for a sum of constant number of UPT circuits can be solved in deterministic polynomial time (over any field).

Theorem 5 (Informal)**.** *The PIT problem for the sum of k UPT circuits of size s can be solved deterministically in time* $s^{O(2^k)}$.

Related work. Hrubeš, Wigderson and Yehudayoff [13] initiated a study of the asymptotics of the classical *sum-of-squares* problem in mathematics and showed that a suitable result in this direction would yield strong lower bounds against general non-commutative circuits. While this line of work is currently the only feasible attack on the problem of general circuit lower bounds, we do not yet have any lower bounds using this technique.

Nisan and Wigderson [23] prove results that imply[6] some lower bounds for UPT formulas computing iterated matrix product. For depth-3 formulas, they prove an optimal n^d bound on computing the product of d $n \times n$ matrices. For depths $\Delta > 3$ though, the lower bound is only $\exp(\Theta(d^{1/\Delta}))$ and thus does not yield anything non-trivial when Δ approaches $\log d$. Indeed, the proof method of this result in [23] is not sensitive to the value of n and holds for any $n \geq 2$. Such a method cannot yield non-trivial lower bounds for general formulas since we do have $\mathrm{poly}(d)$-sized formulas in the setting when $n = O(1)$.

The results of Kayal, Saha, and Saptharishi [17] and Fournier, Limaye, Malod, and Srinivasan [10] together also prove a superpolynomial lower bound on the size of *regular* formulas (defined by [17]) computing the product of d $n \times n$ matrices in the commutative setting. While these formulas (in the non-commutative setting) are definitely UPT, the converse is not true.

Arvind, Mukhopadhyay and Raja [4] and Arvind, Joglekar, Mukhopadhyay and Raja [2] have some recent work on PIT algorithms for general non-commutative circuits that run in time *polylogarithmic* in the degree of the circuit and polynomial in the size of the circuit. Our results are incomparable with theirs, since our algorithms run in time polynomial in *both* degree and size but are deterministic, whereas the algorithms of [4, 2] are faster (especially in terms of degree) but *randomized*.

An earlier manuscript of Arvind and Raja [6] contains a claim that the PIT problem for non-commutative skew circuits has a deterministic polynomial time algorithm, but the proof is unfortunately flawed.[7]

Techniques. The techniques used to prove the lower bounds in this paper are generalizations of the techniques of Hyafil [14] and Nisan [22]. Given a homogeneous polynomial $f \in \mathbb{F}\langle X \rangle$ of degree d, we associate with it an $N^{d/2} \times N^{d/2}$ matrix whose rows and columns are labelled by monomials (i.e. strings over X) m of degree $d/2$ each. Nisan [22] considers the matrix $M[f]$ where the (m_1, m_2)th entry is the coefficient of the monomial $m_1 m_2$ in f. In [20, 19], along with our co-authors, we considered the more general family of matrices $M_Y[f]$ where $Y \subseteq [d]$ is of size $d/2$ and the (m_1, m_2)th entry of $M_Y[f]$ is the coefficient of the monomial m such that the projection of m to the locations in Y gives m_1 and the locations outside gives m_2.

This is the general technique we use in this paper as well, though choosing the right Y requires some work. In the proof of Theorem 1, it is chosen at random (in a similar spirit to a multilinear lower bound of Raz [24]). In the proof of Theorem 2, it is chosen in a way that depends on the structure of the parse trees in the circuit (combining the approaches of [20, 19]). In the proof of Theorem 3, it is applied (after

[6] The results of [23] in fact hold in the stronger commutative *set-multilinear* setting.

[7] Private communication with the authors.

a suitable restriction) in a way that keeps the iterated matrix product polynomial high rank but reduces the rank of the UPT formula.

For the PIT algorithm for sums of UPT circuits, we use an observation of Gurjar et al. [12] (also see [22, 25]) that any polynomial P that has a small ABP has a small set of *characterizing identitites* such that $Q = P$ iff Q satisfies these identities. We are able to show (using a suitable decomposition lemma of [19]) that a similar fact is also true more generally in the case that P has a small UPT circuit. If Q also has a small UPT circuit, then checking these identities for Q reduces to a PIT circuit for a single UPT circuit, for which we already have algorithms. In this way, given two UPT circuits (with different parse trees) computing P, Q, we can check if $P - Q = 0$. Extending this idea exactly as in [12], we can efficiently check if the sum of any small number of UPT circuits is 0.

2 Preliminaries

We refer the reader to the survey [27] for standard definitions regarding arithmetic circuits.

2.1 Non-commutative polynomials

Throughout, we use $X = \{x_1, \ldots, x_N\}$ to denote the set of variables. We work over the *non-commutative* ring of polynomials $\mathbb{F}\langle X\rangle$ where monomials are *strings* over the alphabet X: for example, x_1x_2 and x_2x_1 are distinct monomials in this ring. For $d \in \mathbb{N}$, we use $\mathscr{M}_d(X)$ to denote the set of monomials (i.e. strings) over the variables in X of degree exactly d.

For $i, j \in \mathbb{N}$, we define $[i, j]$ to be the set $\{i, i+1, \ldots, j\}$ (the set is empty if $i > j$). We also use the standard notation $[i]$ to denote the set $[1, i]$.

Given homogeneous polynomials $g, h \in \mathbb{F}\langle X\rangle$ of degrees d_g and d_h respectively and an integer $j \in [0, d_h]$, we define the *j-product of g and h* — denoted $g \times_j h$ — as follows:

- When g and h are monomials, then we can factor h uniquely as a product of two monomials h_1h_2 such that $\deg(h_1) = j$ and $\deg(h_2) = d_h - j$. In this case, we define $g \times_j h$ to be $h_1 \cdot g \cdot h_2$.
- The map is extended bilinearly to general homogeneous polynomials g, h. Formally, let g, h be general homogeneous polynomials, where $g = \sum_\ell g_\ell$, $h = \sum_i h_i$ and g_ℓ, h_i are monomials of g, h respectively. For $j \in [0, d_h]$, each h_i can be factored uniquely into h_{i_1}, h_{i_2} such that $\deg(h_{i_1}) = j$ and $\deg(h_{i_2}) = d_h - j$. And $g \times_j h$ is defined to be $\sum_i \sum_\ell h_{i_1} g_\ell h_{i_2}$.

Note that $g \times_0 h$ and $g \times_{d_h} h$ are just the products $g \cdot h$ and $h \cdot g$ respectively.

2.2 The partial derivative matrix

Here we recall some definitions from [22] and [20]. Let Π denote a partition of $[d]$ given by an ordered pair (Y, Z), where $Y \subseteq [d]$ and $Z = [d] \setminus Y$. In what follows we only use ordered partitions of sets into two parts. We say that such a Π is *balanced* if $|Y| = |Z| = d/2$.

Given a monomial m of degree d and a set $W \subseteq [d]$, we use m_W to denote the monomial of degree $|W|$ obtained by keeping only the variables in the locations indexed by W and dropping the others.

Definition 6 (Partial Derivative matrix)**.** *Let $f \in \mathbb{F}\langle X\rangle$ be a homogeneous polynomial of degree d. Given a partition $\Pi = (Y, Z)$ of $[d]$, we define an $N^{|Y|} \times N^{|Z|}$ matrix $M[f, \Pi]$ with entries from $\mathbb{F}$ as follows: the rows of $M[f, \Pi]$ are labelled by monomials from $\mathscr{M}_{|Y|}(X)$ and the columns by elements of $\mathscr{M}_{|Z|}(X)$. Let $m' \in \mathscr{M}_{|Y|}(X)$ and $m'' \in \mathscr{M}_{|Z|}(X)$; the (m', m'')th entry of $M[f, \Pi]$ is the coefficient in the polynomial f of the unique monomial m such that $m_Y = m'$ and $m_Z = m''$.*

We will use the rank of the matrix $M[f,\Pi]$ — denoted $\text{rank}(f,\Pi)$ — as a measure of the complexity of f . Note that since the rank of the matrix is at most the number of rows, we have for any $f \in \mathbb{F}\langle X\rangle$ $\text{rank}(f,\Pi) \leq N^{|Y|}$.

Definition 7 (Relative Rank)**.** *Let $f \in \mathbb{F}\langle X\rangle$ be a homogeneous polynomial of degree d. For any $Y \subseteq [d]$, we define the* relative rank of f w.r.t. $\Pi = (Y,Z)$ *— denoted* rel-rank(f,Π) *— to be*

$$\text{rel-rank}(f,\Pi) := \frac{\text{rank}(M[f,\Pi])}{N^{|Y|}}.$$

Fix a partition $\Pi = (Y,Z)$ of $[d]$ and two homogeneous polynomials g,h of degrees d_g and d_h respectively. Let $f = g \times_j h$ for some $j \in [0,d_h]$. This induces naturally defined partitions Π_g of $[d_g]$ and Π_h of $[d_h]$ respectively in the following way. Let $I_g = [j+1, j+d_g]$ and $I_h = [d] \setminus I_g$. We define $\Pi_g = (Y_g, Z_g)$ such that $Y_g = \{j \in [d_g] \mid Y \text{ contains the } j\text{th smallest element of } I_g\}$; $\Pi_h = (Y_h, Z_h)$ is defined similarly with respect to I_h. Let $|Y_g|, |Z_g|, |Y_h|, |Z_h|$ be denoted d'_g, d''_g, d'_h, d''_h respectively.

In the above setting, we have a simple description of the matrix $M[f,\Pi]$ in terms of $M[g,\Pi_g]$ and $M[h,\Pi_h]$. We use the observation that monomials of degree $|Y| = d'_g + d'_h$ are in one-to-one correspondence with pairs (m'_g, m'_h) of degrees d'_g and d'_h respectively (and similarly for monomials of degree $|Z|$). The following appears in [20].

Lemma 8 (Tensor Lemma)**.** *Say $f = g \times_j h$ as above. Then, $M[f,\Pi] = M[g,\Pi_g] \otimes M[h,\Pi_h]$.*

Corollary 9. *Say $f = g \times_j h$ as above. We have* $\text{rank}(f,\Pi) = \text{rank}(g,\Pi_g) \cdot \text{rank}(h,\Pi_h)$. *In the special case that one of Y_g, Z_g, Y_h, or Z_h is empty, the tensor product is an outer product of two vectors and hence* $\text{rank}(f,\Pi) \leq 1$.

We associate any partition $\Pi = (Y,Z)$ with the string in $\{-1,1\}^d$ that contains a -1 in exactly the locations indexed by Y. Given partitions $\Pi_1, \Pi_2 \in \{-1,1\}^d$, we now define $\Delta(\Pi_1,\Pi_2)$ to be the Hamming distance between the two strings or equivalently as $|Y_1 \Delta Y_2|$ where $\Pi_1 = (Y_1, Z_1)$ and $\Pi_2 = (Y_2, Z_2)$.

Proposition 10. *Let $f \in \mathbb{F}\langle X\rangle$ be homogeneous of degree d and say $\Pi \in \{-1,1\}^d$. Then,* $\text{rank}(f,\Pi) = \text{rank}(f,-\Pi)$.

Proof. Follows from the fact that $M[f,-\Pi]$ is the transpose of $M[f,\Pi]$. □

Lemma 11 (Distance lemma)**.** *Let $f \in \mathbb{F}\langle X\rangle$ be homogeneous of degree d and say $\Pi_1, \Pi_2 \in \{-1,1\}^d$. Then,* $\text{rank}(f,\Pi_2) \leq \text{rank}(f,\Pi_1) \cdot N^{\Delta(\Pi_1,\Pi_2)}$.

Proof. See Appendix A. □

2.3 Standard definitions related to non-commutative circuits

We consider noncommutative arithmetic circuits that compute polynomials over the ring $\mathbb{F}\langle X\rangle$. These are arithmetic circuits where the children of each $\times$ gate are ordered and the polynomial computed by a $\times$ gate is the product of the polynomials computed by its children, where the product is computed in the given order. Further, unless mentioned otherwise, we allow both $+$ and $\times$ gates to have unbounded fan-in and the $+$ gates to compute arbitrary linear combinations of its inputs (the input wires to the $+$ gate are labelled by the coefficients of the linear combination). A noncommutative formula is a circuit where the underlying directed graph is a rooted tree. The size of an arithmetic circuit or formula will the number of edges or wires in the circuit (which can be assumed to be at least the number of gates in the circuit).

We always assume that the output gate of the circuit is a $+$ gate (possibly of fan-in 1) and that input gates feed into $+$ gates. We also assume that $+$ and $\times$ gates alternate on any path from the output gate

to an input gate (some of these gates can have fan-in 1). Any circuit can be converted to one of this form with at most a constant blow-up in size.

Throughout, our circuits and formulas will be *homogeneous* in the following sense. Define the formal degree of a gate in the circuit as follows: the formal degree of an input gate is 1, the formal degree of a $+$ gate is the maximum of the formal degrees of its children, and the formal degree of a $\times$ gate is the sum of the formal degrees of its children. We say that a circuit is homogeneous if each gate computes a homogeneous polynomial and any gate computing a non-zero polynomial computes one of degree equal to the formal degree of the gate. Note, in particular, that every input node is labelled by a variable only (and not by constants from $\mathbb{F}$).

Homogeneity is *not* a strong assumption on the circuit: it is a standard fact that any homogeneous polynomial of degree d computed by a non-commutative circuit of size s can be computed by a homogeneous circuit of size $O(sd^2)$ [13].

We also consider homogeneous Algebraic Branching Programs (ABPs), defined by Nisan [22] in the non-commutative context. We give here a slightly different definition that is equivalent up to polynomial factors.

Assume that $N = n^2 \cdot d$ for positive $n, d \in \mathbb{N}$ and let $\mathrm{IMM}_{n,d}(X)$ denote the following polynomial in N variables (see, e.g. [23]). Assume X is partitioned into d sets of variables $X_1, \ldots, X_d$ of size n^2 each and let $M_1, \ldots, M_d$ be $n \times n$ matrices such that the entries of M_i ($i \in [d]$) are distinct variables in X_i. Let $M = M_1 \cdot M_2 \cdots M_d$; each entry of M is a homogeneous polynomial of degree d from $\mathbb{F}\langle X \rangle$. We define the polynomial $\mathrm{IMM}_{n,d}$ to be the sum of the diagonal entries of M.

A homogeneous ABP for a homogenous polynomial $f \in \mathbb{F}\langle X \rangle$ of degree d is a pair (n_1, ρ) where $n_1 \in \mathbb{N}$ and ρ is a map from $X' = \{x'_1, \ldots, x'_{n_1^2 d}\}$ to homogeneous linear functions from $\mathbb{F}\langle X \rangle$ such that f can be obtained by substituting $\rho(x'_i)$ for each x'_i in the polynomial $\mathrm{IMM}_{n_1,d}(X')$. The parameter n_1 is called the *width* of the ABP.

2.4 Non-commutative circuits with restricted parse trees

In this paper, we study restricted forms of non-commutative arithmetic circuits. The restrictions are defined by the way the circuits are allowed to multiply variables to compute a monomial. To make this precise we need the notion of a parse tree of a circuit, which has been considered in many previous works [15, 1, 21, 19].

Fix a homogeneous non-commutative circuit C. A *parse formula* of C is a *formula* C' obtained by making copies of gates in C as follows:

- Corresponding to the output $+$ gate of C, we add an output $+$ gate to C',
- For every $+$ gate Φ' added to C' corresponding to a $+$ gate Φ in C, we choose exactly one child Ψ of Φ in C and add a copy Ψ' to C' as a child of Φ'. The constant along the wire from Ψ' to Φ' remains the same as in C.
- For every $\times$ gate Φ' added to C' corresponding to a $\times$ gate Φ in C and *every wire* from a child Ψ to Φ in C, we make a copy of Ψ' to C' and make it a child of Φ.

Any such parse formula C' computes a *monomial* (with a suitable coefficient) and the polynomial computed by C is the sum of all monomials computed by parse formulas C' of C. We define $val(C')$ to be the monomial computed by C'.

A parse tree of C is a rooted, ordered tree obtained by taking a parse formula C' of C, "short circuiting" the $+$ nodes (i.e. we remove the $+$ nodes and connect the edges that were connected to it directly), and deleting all labels of the nodes and the edges of the tree. See Figure 2 for an example. Note that in a homogeneous circuit C, each such tree has exactly d leaves. We say that the tree T is the *shape* of the parse formula C'.

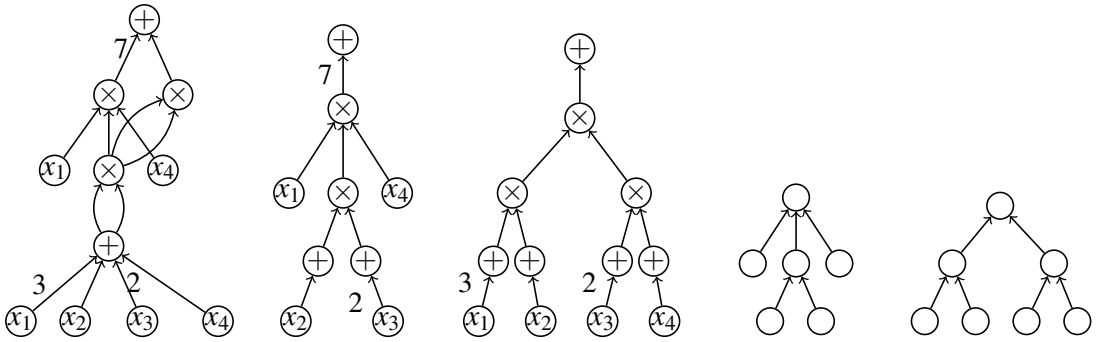

Figure 2: From left to right: a non-commutative arithmetic circuit; two parse formulas in the circuit; the corresponding parse trees.(To simplify the picture, we have not depicted the edges that carry the constant 1. Also we have not introduced $+$ gates between the two layers of $\times$ gates; the reader should assume that the edges between the two layers carry $+$ gates of fan-in 1.)

The process that converts the parse formula C' to T associates each internal node of T with a multiplication gate of C' and each leaf of T with an input gate of C'.

Let T be a parse tree of a homogeneous circuit C with d leaves. Given a node $v \in V(T)$, we define the $\deg(v)$ to be the number of leaves in the subtree rooted at v and $\mathrm{pos}(v) := (1 + \text{the number of leaves preceding } v \text{ in an in-order traversal of } T)$. The type of v is defined to be $\mathrm{type}(v) := (\deg(v), \mathrm{pos}(v))$. (The reason for this definition is that in any parse formula C' of shape T, the monomial computed by the multiplication gate or input gate corresponding to v in C' computes a monomial of degree $\deg(v)$ which sits at position $\mathrm{pos}(v)$ w.r.t. the monomial computed by the circuit C'.) We also use $\mathscr{I}(T)$ to denote the set of internal nodes of T and $\mathscr{L}(T)$ to denote the set of leaves of T.

We use $\mathscr{T}(C)$ to denote the set of parse trees that can be obtained from parse formulas of C. We say that a homogeneous non-commutative arithmetic circuit is a *Unique Parse Tree circuit* (or *UPT circuit*) if $|\mathscr{T}(C)| = 1$. More generally if $|\mathscr{T}(C)| \leq k$, we say that C is *k-PT*. Finally, if $\mathscr{T}(C) \subseteq \mathscr{T}$ for some family $\mathscr{T}$ of trees, we say that C is *$\mathscr{T}$-PT*. Similarly, we also define UPT formulas, k-PT formulas and $\mathscr{T}$-PT formulas. If C be a UPT circuit with $\mathscr{T}(C) = \{T\}$, we say that T is the *shape* of the circuit C.

We say that a UPT circuit C is in *normal form* if we can associate with each gate Φ of the circuit a node $v(\Phi) \in V(T)$ such that the following holds: if Φ is an input gate, then $v(\Phi)$ is a leaf; if Φ is a $\times$ gate with children $\Psi_1, \ldots, \Psi_t$ (in that order), then the nodes $v(\Psi_1), \ldots, v(\Psi_t)$ are the children of $v(\Phi)$ (in that order); and finally, if Φ is a $+$ gate with children $\Psi_1, \ldots, \Psi_t$ (which are all $\times$ or input gates since we assume that $+$ and $\times$ gates are alternating along each input to output path), then $v(\Phi) = v(\Psi_1) = \cdots = v(\Psi_t)$. (Intuitively, what this means is that in any unravelling of a parse formula containing a (multiplication or input) gate Φ to get the parse tree T, the gate Φ always takes the position of node $v(\Phi)$.)

We state below some simple structural facts about UPT circuits.

Proposition 12. *1. Let C be a UPT* formula*. Then C is in normal form.*

2. For any UPT circuit C of size s and shape T, there is another UPT circuit C' of size $O(s^2)$ and shape T in normal form computing the same polynomial as C. Further, given C and T, such a C' can be constructed in time $\mathrm{poly}(s)$.

Proof. See Appendix B. □

Let C be either a UPT formula or a UPT circuit of shape T in normal form. We say that a $+$ gate Φ in C is a $(v,+)$ gate if $v(\Phi) = v$. Similarly, we refer to a $\times$ gate Φ in C as a $(v,\times)$ gate if $v(\Phi) = v$. For

simplicity of notation, we also refer to an *input* gate Φ as a $(v,\times)$ gate if $v(\Phi) = v$. Note that the output gate is a $(v_0,+)$ gate where v_0 is the root of T.

We now observe that any UPT formula or circuit in normal form can be converted to another (of a possibly different shape) where each multiplication gate has fan-in at most 2.

Lemma 13. *Let C be a normal form UPT circuit (resp. formula) of size s and shape T. Then there is a tree T' and normal form UPT circuit (resp. formula) C' of size $O(s)$ and shape T' such that C' computes the same polynomial as C and every multiplication gate in C' has fan-in at most 2. (This implies that every internal node of T' also has fan-in at most 2.) Further, there is a deterministic polynomial-time algorithm, which when given C, computes C' as above.*

Proof. See Appendix C. □

Let C be a UPT circuit of shape T computing a homogeneous polynomial f of degree d. Given any node $u \in V(T)$, we define partition Π_u of $[d]$ so that $\Pi_u = (Y_u, Z_u)$ where

$$Y_u = \{\text{pos}(v) \mid v \text{ a leaf and descendant of } u\}.$$

We will need the following lemma of Lagarde et al. [19].

Lemma 14 ([19])**.** *Let C be a normal form UPT circuit of size s computing a homogeneous polynomial $f \in \mathbb{F}\langle X\rangle$ of degree d. Assume that the fan-in of each multiplication gate is bounded by 2. Then, for any $u \in V(T)$, $\text{rank}(f, \Pi_u) \leq s$, where Π_u is as defined above.*

2.5 A polynomial that is full rank w.r.t. all partitions

The following was shown in [20].

Theorem 15. *For any even d and any positive $N \in \mathbb{N}$, there is a $q_0(N,d)$ such that the following holds over any field of size at least $q_0(N,d)$. There is an explicit homogeneous polynomial $F_{N,d} \in \mathbb{F}\langle X\rangle$ of degree d such that for any balanced partition $\Pi = (Y,Z)$ of $[d]$, $\text{rank}(f,\Pi) = N^{d/2}$ (equivalently, $\text{rel-rank}(f,\Pi) = 1$). Further, $F_{N,d}$ can be computed by an explicit homogeneous non-commutative arithmetic circuit of size $\text{poly}(N,d)$.*

3 Lower bounds for k-PT circuits

In this section, we show that any k-PT circuit computing a polynomial of degree d where k is subexponential in d cannot compute the polynomial $F_{N,d}$ from Theorem 15. We will show that if both k and the size of the circuit are subexponential in d, then there is a Π such that $\text{rel-rank}(f,\Pi) < 1$.

Our proof is based on the following lemmas.

Lemma 16. *Let C be a k-PT circuit (resp. formula) of size s with $\mathscr{T}(C) = \{T_1,\ldots,T_k\}$ computing $f \in \mathbb{F}\langle X\rangle$. Then there exist normal form UPT circuits (resp. formulas) $C_1,\ldots,C_k$ of size at most s^2 each such that $\mathscr{T}(C_i) = \{T_i\}$ and $f = \sum_{i=1}^{k} f_i$, where f_i the polynomial computed by C_i.*

Proof. See Appendix D. □

Lemma 17. *Let C be a UPT circuit in normal form over $\mathbb{F}\langle X\rangle$ of size $s = N^c$ and f a homogeneous polynomial of degree d computed by C. Let Π be a uniformly random partition of the variables of $[d]$ into two sets. Then for any parameter $b \in \mathbb{N}$,*

$$\Pr_{\Pi}\left[\text{rank}(f,\Pi) \geq N^{d/2-b}\right] \leq \exp(-\Omega(d/(b+c)^2)).$$

The above lemmas imply the following lower bound for homogeneous non-commutative circuits with few parse trees. Note that when the field $\mathbb{F}$ is large enough, this proves a lower bound for $F_{N,d}$ from Theorem 15.

Theorem 18. *Assume that $N \geq 2$ is any constant and d an even integer parameter that is growing. Let $F \in \mathbb{F}\langle X\rangle$ be any polynomial such that for each balanced partition Π, $\operatorname{rank}(F,\Pi) = N^{d/2}$. Then, for any constant $\varepsilon \in (0,1)$, any circuit that computes F and satisfies $|\mathscr{T}(C)| = k \leq 2^{d^{\frac{1}{3}-\varepsilon}}$ must have size at least $2^{d^{\frac{1}{3}-\frac{\varepsilon}{2}}}$.*

Proof. Let C be any circuit of size $s \leq N^c$ for $c = d^{1/3-\varepsilon/2}$ with $|\mathscr{T}(C)| = k \leq 2^{d^{1/3-\varepsilon}}$ and computing $f \in \mathbb{F}\langle X\rangle$. We show that there is a balanced partition Π such that $\operatorname{rank}(f,\Pi) < N^{d/2}$. This will prove the theorem.

To show this, we proceed as follows. Using Lemma 16, we can write $f = \sum_{i\in[k]} f_i$ where each $f_i \in \mathbb{F}\langle X\rangle$ is computed by a normal form UPT circuit C_i of size at most $s^2 \leq N^{2c}$.

Fix any $i \in [k]$. By Lemma 17, the number of partitions Π for which $\operatorname{rank}(f_i,\Pi) \geq N^{\frac{d}{2}-c}$ is at most $2^d \cdot \exp(-\Omega(d/c^2))$. In particular, since the number of balanced partitions is $\binom{d}{d/2} = \Theta(\frac{2^d}{\sqrt{d}})$, we see that for a random *balanced* partition Π,

$$\Pr_{\Pi \text{ balanced}}\left[\operatorname{rank}(f_i,\Pi) \geq N^{d/2-c}\right] \leq \sqrt{d}\cdot\exp(-\Omega(d/c^2)) \leq \exp(-d^{1/3}).$$

Say f_i is good for Π if $\operatorname{rank}(f_i,\Pi) \geq N^{d/2-c}$. By the above, we have

$$\Pr_{\Pi \text{ balanced}}[\exists i \in [k] \text{ s.t. } f_i \text{ good for } \Pi] \leq k\cdot\exp(-d^{1/3}) \leq 2^{d^{1/3-\varepsilon}}\cdot\exp(-d^{1/3}) < 1.$$

In particular, there is a balanced Π such that no f_i is good for Π. Fix such a balanced partition Π. By the subadditivity of rank, we have

$$\begin{aligned}\operatorname{rank}(f,\Pi) &\leq \sum_{i\in[k]} \operatorname{rank}(f_i,\Pi) \leq k\cdot N^{d/2-c} \leq 2^{d^{1/3-\varepsilon}}\cdot N^{d/2-c}\\ &= N^{d/2}\cdot\exp(O(d^{1/3-\varepsilon}) - \Omega(d^{1/3-\varepsilon/2})) < N^{d/2}.\end{aligned}$$

This proves the theorem. □

3.1 Proof of Lemma 17

Notation. Recall from Section 2.2 that we identify each partition Π with an element of $\{-1,1\}^d$. Given partitions $\Pi_1,\Pi_2 \in \{-1,1\}^d$ we use $\langle\Pi_1,\Pi_2\rangle$ to denote their inner product: i.e., $\langle\Pi_1,\Pi_2\rangle := \sum_{i\in[d]}\Pi_1(i)\Pi_2(i)$. Note that the Hamming distance $\Delta(\Pi_1,\Pi_2)$ is

$$\Delta(\Pi_1,\Pi_2) = \frac{d}{2} - \frac{1}{2}\langle\Pi_1,\Pi_2\rangle. \tag{1}$$

Let $\mathscr{T}(C) = \{T\}$. Recall that $|\mathscr{L}(T)| = d$ and by Lemma 13, we can assume that the fan-in of each internal node of T is bounded by 2. For any $u \in \mathscr{I}(T)$ (recall $\mathscr{I}(T)$ is the set of internal nodes of T), let $\mathscr{L}(u)$ denote the set of leaves of the subtree rooted at u. We identify each leaf $\ell \in V(T)$ with $\operatorname{pos}(\ell) \in [d]$. For each $u \in \mathscr{I}(T)$, we can define the partition Π_u from Section 2.4 by $\Pi_u(\ell) = -1$ iff $\ell \in \mathscr{L}(u)$.

For $\gamma > 0$, define a partition Π to be *γ-correlated to T* if for each $u \in \mathscr{I}(T)$, we have $\left|\sum_{\ell\in\mathscr{L}(u)}\Pi(\ell)\right| \leq \gamma$.

Lemma 17 immediately follows from Claims 19 and 20, stated below.

Claim 19. *Let Π be any partition of $[d]$ such that $\mathrm{rank}(f,\Pi) \geq N^{d/2-b}$. Then Π is $O(b+c)$-correlated to T.*

Proof. We know from Lemma 14 and Proposition 10 that for each $u \in \mathscr{I}(T)$, $\mathrm{rank}(f,\Pi_u), \mathrm{rank}(f,-\Pi_u) \leq N^c$. If Π is a partition such that either $\Delta(\Pi,\Pi_u)$ or $\Delta(\Pi,-\Pi_u)$ is strictly smaller than $\frac{d}{2}-(b+c)$ for *some* $u \in \mathscr{I}(T)$, then by Lemma 11 we would have $\mathrm{rank}(f,\Pi) < N^{d/2-b}$.

Thus, if $\mathrm{rank}(f,\Pi) \geq N^{d/2-b}$, we must have $\min\{\Delta(\Pi,\Pi_u),\Delta(\Pi,-\Pi_u)\} \geq \frac{d}{2}-(b+c)$ for each $u \in \mathscr{I}(T)$. By (1), this means that for each $u \in \mathscr{I}(T)$, $|\langle \Pi,\Pi_u\rangle| \leq \gamma$ for some $\gamma = O(b+c)$.

Let v be the root of T. Note that $\Pi_v \in \{-1,1\}^d$ is the vector with all its entries being -1. Hence, we have for any $u \in \mathscr{I}(T)$,

$$\left|\sum_{\ell \in \mathscr{L}(u)} \Pi(\ell)\right| = \left|\langle \Pi, \frac{-(\Pi_u+\Pi_v)}{2}\rangle\right| \leq \frac{1}{2}(|\langle \Pi,\Pi_u\rangle| + |\langle \Pi,\Pi_v\rangle|) \leq O(\gamma).$$

This proves the claim. □

Claim 20. *Say $\Pi \in \{-1,1\}^d$ is chosen uniformly at random and $\gamma \leq \sqrt{d}$. Then*

$$\Pr_{\Pi}[\Pi \text{ is } \gamma\text{-correlated to } T] \leq \exp(-\Omega(\frac{d}{\gamma^2})).$$

The following subclaim is useful for proving Claim 20.

Subclaim 21. *Assume that $r,t \in \mathbb{N}$ such that $rt \leq d/4$. Then we can find a sequence $u_1,\ldots,u_r \in \mathscr{I}(T)$ such that for each $i \in [r]$ we have $|\mathscr{L}(u_i) \setminus \bigcup_{j=1}^{i-1} \mathscr{L}(u_j)| \geq t$.*

Proof. Consider the following 'greedy' procedure for choosing the u_i. Order the nodes of $\mathscr{I}(T)$ in topological order (recall that the edges of T are directed toward the root). We choose u_1 to be the least node in this order so that $|\mathscr{L}(u_1)| \geq t$ (such a node must exist since there are $d \geq t$ leaves in T). Further, having chosen $u_1,\ldots,u_i$ we choose u_{i+1} to be the least node greater than or equal to $u_1,\ldots,u_i$ in the topological order such that $|\mathscr{L}(u_{i+1}) \setminus \bigcup_{j=1}^{i} \mathscr{L}(u_j)| \geq t$.

To argue that this process produces a sequence of size at least r, note that for each $i \geq 0$, $|\mathscr{L}(u_{i+1}) \setminus \bigcup_{j=1}^{i} \mathscr{L}(u_j)| \leq 2t$. This is because the fan-in of u_{i+1} in T is at most 2 and by assumption, for each child u' of u_{i+1}, we have $|\mathscr{L}(u') \setminus \bigcup_{j=1}^{i} \mathscr{L}(u_j)| < t$. Thus for each $i \geq 0$, we have $|\bigcup_{j=1}^{i+1} \mathscr{L}(u_j)| \leq 2t(i+1)$.

In particular, if $i+1 < r$, we have $|\bigcup_{j=1}^{i+1} \mathscr{L}(u_j)| < 2tr \leq d/2$. Thus, for v being the root of the tree, we have $|\mathscr{L}(v) \setminus \bigcup_{j=1}^{i+1} \mathscr{L}(u_j)| > d/2 \geq t$. In particular, there is at least one node u of the tree such that $|\mathscr{L}(u) \setminus \bigcup_{j=1}^{i+1} \mathscr{L}(u_j)| \geq t$. This allows us to extend the sequence further. □

Proof of Claim 20. We apply Subclaim 21 with $t = \Theta(\gamma^2)$ and $r = \Theta(d/\gamma^2)$ to get a sequence $u_1,\ldots,u_r \in \mathscr{I}(T)$ such that for each $i \in [r]$, we have $|\mathscr{L}(u_i) \setminus \bigcup_{j=1}^{i-1} \mathscr{L}(u_j)| \geq t$.

By the definition of γ-correlation, we have

$$\begin{aligned}\Pr_{\Pi}[\Pi \text{ } \gamma\text{-correlated to } T] &\leq \Pr_{\Pi}\left[\forall i \in [r],\ \left|\sum_{\ell \in \mathscr{L}(u_i)} \Pi(\ell)\right| \leq \gamma\right] \\ &\leq \prod_{i \in [r]} \Pr_{\Pi}\left[\left|\sum_{\ell \in \mathscr{L}(u_i)} \Pi(\ell)\right| \leq \gamma \;\middle|\; \{\Pi(\ell) \mid \ell \in \bigcup_{j<i} \mathscr{L}(u_j)\}\right] \qquad (2)\end{aligned}$$

Fix any $i \in [r]$ and $\Pi(\ell)$ for each $\ell \in \mathscr{L}_{<i} := \bigcup_{j<i} \mathscr{L}(u_j)$. Note that the event $|\sum_{\ell \in \mathscr{L}(u_i)} \Pi(\ell)| \leq \gamma$ is equivalent to $\sum_{\ell \in \mathscr{L}(u_i) \setminus \mathscr{L}_{<i}} \Pi(\ell) \in I$ for some interval I of length $2\gamma = O(\sqrt{t})$. This is the probability that the sum of at least t independent uniformly chosen $\{-1,1\}$-valued random variables lies in an interval of length $O(\sqrt{t})$. By the Central Limit theorem, this can be bounded by $1-\Omega(1)$.

By (2), we get $\Pr_{\Pi}[\Pi \text{ } \gamma\text{-correlated to } T] \leq \exp\{-\Omega(r)\}$, which gives the statement of the claim. □

4 Lower bounds for circuits with rotations of one parse tree

Given two parse trees T_1 and T_2 with the same number of leaves, we say that T_1 is a *rotation* of T_2, denoted $T_1 \sim T_2$, if T_1 can be obtained from T_2 by repeatedly reordering the children of various nodes in T_2. Clearly, $\sim$ is an equivalence relation. We use $[\![T]\!]$ to denote the equivalence class of tree T. We say that a homogeneous circuit C is *rotation UPT or rotUPT* if there is a tree T such that $\mathscr{T}(C) \subseteq [\![T]\!]$. The tree T is said to be a *template* for C.

Our main result in this section is the following.

Theorem 22. *Let C be a rotUPT circuit of size s computing a polynomial $f \in \mathbb{F}\langle X\rangle$ of degree d over n variables, then there exists a partition $\Pi = \Pi_C$ such that* rel-rank(f,Π) *is at most* $\mathrm{poly}(s) \cdot N^{-\Omega(d)}$.

In particular, we get the following corollary.

Corollary 23. *Let $N, d \in \mathbb{N}$ be parameters with d even. Let $|\mathbb{F}| > q_0(N,d)$ where $q_0(N,d)$ is as in Theorem 15. Then, any rotUPT circuit for $F_{N,d}$ has size $N^{\Omega(d)}$.*

We will need a decomposition lemma for non-commutative circuits in the proof of Theorem 22. The following is a variant of lemmas that are proved in [13, 20, 19].

Lemma 24 (A decomposition lemma for homogeneous circuits)**.** *Let C be any homogeneous arithmetic circuit of size s computing $f \in \mathbb{F}\langle X\rangle$ of degree d. Assume that there is some $d' \in [d/2+1, d]$ such that every parse formula C' of C contains a gate computing a (homogeneous) polynomial of degree d'. Let $\Phi_1, \ldots, \Phi_r$ ($r \leq s$) be the set of $\times$ gates computing polynomials of degree d' in C and let $g_1, \ldots, g_r$ be the polynomials they compute (respectively). Then, for homogeneous polynomials $h_{i,j}$ of degree $d - d'$ ($i \in [r], j \in [0, d-d']$) we have*

$$f = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} g_i \times_j h_{i,j}.$$

Proof. See Appendix E. □

Proof of Theorem 22. Let C be rotUPT of size s computing a polynomial f of degree d over n variables, and let T be a template for C. Let $\Pi_i = (Y_i, Z_i)$ for $i \in [2]$ be two partitions of $[d]$ with $Y_1 = [d/2]$ and $Y_2 = [d/4+1, 3d/4]$. We will show that rel-rank$(f,\Pi) \leq \mathrm{poly}(s) \cdot N^{-\Omega(d)}$ for some $\Pi \in \{\Pi_1, \Pi_2\}$.

We consider two cases: **Case 1: there is a node of degree $d_0 \in [\frac{3}{4}d, \frac{11}{12}d]$ in the template T.** Note that every rotation $T' \in [\![T]\!]$ also has a node of degree d_0. Since every parse tree of C is a member of $[\![T]\!]$, this implies that each parse formula C' of C contains a gate of degree d_0. Applying Lemma 24 with $d' = d_0$, we see that there are $k \leq s$ homogeneous polynomials $g_1, \ldots, g_k$ of degree d_0 and $k \cdot (d - d_0)$ homogeneous polynomials $h_{1,1}, \ldots h_{1,d-d_0}, h_{2,1}, \ldots h_{k,d-d_0}$ of degree $d - d_0$ such that:

$$f = \sum_{i=1}^{k} \sum_{j=0}^{d-d_0} g_i \times_j h_{i,j} \tag{3}$$

We show that each term of the above decomposition has low relative rank w.r.t. the partition Π_2 defined above. Fix a term $g_i \times_j h_{i,j}$ from the decomposition above. Let $\Pi'_j = (Y'_j, Z'_j)$ where $Y'_j = [j+1, j+d_0]$. By Corollary 9, we see that rank$(g_i \times_j h_{i,j}, \Pi'_j) = 1$.

A straightforward calculation shows that $\Delta(\Pi'_j, \Pi_2) = d_0 - \frac{d}{2} = \frac{d}{2} - \Omega(d)$ for all j. Hence, by Lemma 11, we see that rank$(g_i \times_j h_{i,j}, \Pi_2) \leq N^{(d/2)-\Omega(d)}$ and hence rel-rank$(g_i \times_j h_{i,j}, \Pi_2) \leq N^{-\Omega(d)}$ for each i, j.

Using (3) and the subadditivity of rank, we see that rel-rank$(f, \Pi_2) \leq (sd) \cdot N^{-\Omega(d)} \leq s^2 \cdot N^{-\Omega(d)}$.

Case 2: there is no gate of degree $d_0 \in [\frac{3}{4}d, \frac{11}{12}d]$ in the template T. In this case, we show that rel-rank(f, Π_1) is small, where Π_1 is as defined above.

Let v be the node in T such that $\deg(v) > \frac{11}{12}d$ and all its children have degree $\leq \frac{11}{12}d$. Note that such a v is uniquely defined (if there were another such node v', it cannot be an ancestor or descendant of v; hence, we see that the number of leaves in T is at least $\deg(v)+\deg(v') > d$ which is a contradiction). Let $d'_0 = \deg(v)$. Let $v_1,\ldots,v_t$ be the children of v in T and assume that $\deg(v_i) = d'_i$. Note that $d'_i < \frac{3d}{4}$ for each i.

As in the previous case, we see that every parse formula contains a gate of degree d'_0 and hence applying the lemma with $d' = d'_0$ we get

$$f = \sum_{i=1}^{\ell} \sum_{j=0}^{d-d'_0} g'_i \times_j h'_{i,j} \tag{4}$$

where $g'_1,\ldots,g'_\ell$ ($\ell \leq s$) are the polynomials of degree d'_0 computed by multiplication gates in C. We show that for each i,j, rel-rank$(g'_i \times_j h'_{i,j},\Pi_1) \leq N^{-\Omega(d)}$. As in the previous case, this will imply rel-rank$(f,\Pi) \leq s^2 \cdot N^{-\Omega(d)}$.

Fix any i,j. We use g and h instead of g'_i and $h'_{i,j}$. We know that g is a polynomial computed by some $\times$ gate Φ of degree d'_0 in the circuit. Consider the $+$ gates feeding into Φ. Since every parse tree T' of C is a rotation of T, it must be the case that there are exactly t such $+$ gates $\Psi_1,\ldots,\Psi_t$ computing polynomials $\tilde{g}_1,\ldots,\tilde{g}_t$ such that $g = \tilde{g}_1 \cdots \tilde{g}_t$. Assume that $\deg(\tilde{g}_a) = d''_a$ for each $a \in [t]$. Then $\{d''_1,\ldots,d''_t\} = \{d'_1,\ldots,d'_t\}$ as multisets, where $d'_a = \deg(v_a)$ as defined above; in particular, $d''_a < \frac{3}{4}d$ for each a.

Thus, $g \times_j h = (\tilde{g}_1 \cdots \tilde{g}_t) \times_j h$. For any $a \in [t]$, we note that we can also write $g \times_j h = (\tilde{g}_1 \cdots \tilde{g}_a) \times_j h_a$ where $h_a := (\tilde{g}_{a+1} \cdots \tilde{g}_t) \times_j h$. Let $\Pi''_a = (Y''_a, Z''_a)$ be the partition of $[d]$ such that $Y''_a = [j+1, j+d''_1 + \cdots + d''_a]$. By Corollary 9, we know that $\text{rank}(g \times_j h, \Pi''_a) = \text{rank}((\tilde{g}_1 \cdots \tilde{g}_a) \times_j h_a, \Pi''_a) \leq 1$ for *each* $a \in [t]$. Therefore, by Lemma 11, to prove that rel-rank$(g \times_j h, \Pi_1) \leq N^{-\Omega(d)}$, it suffices to show that $\Delta(\Pi''_a, \Pi_1) \leq \frac{d}{2} - \Omega(d)$ for *some* $a \in [t]$. We do this now.

Consider the least $b \in [t]$ so that $d''_1 + d''_2 + \cdots + d''_b \geq \frac{1}{20}d$. Let $\delta = d''_1 + \cdots + d''_b$. Since each $d''_a < \frac{3}{4}d$, we know that $\delta \in [\frac{1}{20}d, d - \frac{1}{5}d]$. Note that for partition Π''_b, we have $\Delta(\Pi''_b, \Pi_1) = |Y''_b \Delta Y_1|$ where $Y''_b = [j+1, j+\delta]$ and $Y_1 = [\frac{1}{2}d]$. We thus get

$$\begin{aligned} |Y''_b \Delta Y_1| &= j + |\frac{d}{2} - (j+\delta)| = j + \max\{\frac{d}{2} - (j+\delta), j+\delta - \frac{d}{2}\} \\ &= \max\{\frac{d}{2} - \delta, 2j + \delta - \frac{d}{2}\} = \frac{d}{2} - \min\{\delta, d - (2j+\delta)\}. \end{aligned}$$

Since $\delta \geq \frac{1}{10}d$ and $(2j+\delta) \leq 2 \cdot \frac{1}{12}d + d - \frac{1}{5}d < d - \Omega(d)$, we see that $\Delta(\Pi''_b, \Pi_1) = |Y''_b \Delta Y_1| = \frac{d}{2} - \Omega(d)$. This completes the proof. □

Remark 25. *We note that the proof of the theorem yields the stronger statement that f is low-rank w.r.t. one of the two partitions Π_1 and Π_2. It is not hard to use this to prove a lower bound for an even simpler polynomial than the polynomial $F_{N,d}$ (from Theorem 15).*

5 Separation between Few PT formulas and ABPs

In this section, we prove several lower bounds for formulas (with some restrictions on the parse trees) against $\text{IMM}_{n,d}$, yelding separations with ABPs. More specifically, we prove in Section 5.2 a tight superpolynomial lower bound on the size of any UPT formula that computes $\text{IMM}_{n,d}$. In Section5.3, we prove a superpolynomial lower bound for any formulas computing $\text{IMM}_{n,d}$ as long as the number of distinct parse trees is significantly smaller than 2^d (assuming $d \leq \log n$).

5.1 Notations and decomposition lemma for labelled UPT formulas

The main purpose of this section is to fix some notations that will be used in the whole section, and to prove that any polynomial computed by a labelled UPT formula (defined below) admits a very specific decomposition given by Lemma30.

We start with some notation. For $I = \{i_1 < \cdots < i_t\} \subseteq [d]$, we define the set of *I-monomials* to be the set of monomials of the form $x_1 \cdots x_t$ where $x_j \in X_{i_j}$. Also, we define $\mathscr{P}_I$ to be the set of those polynomials P over the variables $\bigcup_{j\in[t]} X_{i_j}$ that can be written as a linear combination of I-monomials. We define IMM_I to be $\mathrm{Tr}(M_{i_1} \cdots M_{i_t})$. Note that $\mathrm{IMM}_I \in \mathscr{P}_I$.

Given I as above and $f \in \mathscr{P}_I$, we define $M_I(f)$ to be a $n^{2\lceil \frac{t}{2} \rceil} \times n^{2\lfloor \frac{t}{2} \rfloor}$ matrix whose rows and columns are labelled by I_{odd}-monomials and I_{even}-monomials respectively, where $I_{\text{odd}} = \{i_1, i_3, i_5, \ldots\}$ and $I_{\text{even}} = \{i_2, i_4, i_6, \ldots\}$. The (m', m'')th entry of $M_I(f)$ is the coefficient in f of the I-monomial which is equal to m' when restricted to its odd locations and m'' when restricted to its even locations. Note that $\mathrm{rank}(M_I(f)) \leq n^{2\lfloor \frac{t}{2} \rfloor}$. We define $\text{rel-rank}_I(f) = \mathrm{rank}(M_I(f))/n^{2\lfloor \frac{t}{2} \rfloor}$.

We will need the following standard fact.

Fact 26. *For any $I \subseteq [d]$ of size t and any $f \in \mathscr{P}_I$, we have* $\mathrm{rank}(M_I(f)) \leq n^{2\lfloor t/2 \rfloor}$. *Also, for any $I \subseteq [d]$,* $\mathrm{rank}(M_I(\mathrm{IMM}_I)) = n^{2\lfloor t/2 \rfloor}$ *and hence* $\text{rel-rank}_I(\mathrm{IMM}_I) = 1$.

Let T be a parse tree with t leaves and $I = \{i_1 < i_2 < \cdots < i_t\}$. The *I-labelling of T* is the function $\mathrm{lab} : V(T) \to 2^I \setminus \{\emptyset\}$ defined as follows. For each $u \in \mathscr{L}(T)$ (recall that $\mathscr{L}(T)$ is the set of leaves of T and $\mathscr{I}(T)$ is the set of internal nodes of T), we define $\mathrm{lab}(u)$ to be $\{i_j\}$ if u is the jth leaf in the in-order traversal of T. We will sometimes abuse notation and assume $\mathrm{lab}(u) = i_j$. For each $v \in \mathscr{I}(T)$, we define $\mathrm{lab}(v)$ to be the set of labels of the leaves in the subtree rooted at v.

We say that a UPT formula F of shape T with t leaves is I-labelled if for each input gate Φ that is a $(u, \times)$-gate with $u \in \mathscr{L}(T)$, the variable labelling the input gate from the set $X_{\mathrm{lab}(u)}$. The following is an easy observation.

Lemma 27. *If F is an I-labelled UPT formula with shape T, then it computes a polynomial from $\mathscr{P}_I$. Further, for any F that is a UPT formula of size at most s computing polynomial f, there is an I-labelled UPT formula F' of shape T and size at most s that computes the polynomial $f' \in \mathscr{P}_I$ that is obtained from f by zeroing out the coefficients of all monomials that are not I-monomials.*

Proof. Let F be an I-labelled UPT formula. If we take a I-labelled parse tree T, then it computes a monomial which is the product of the leaves of T given by the in-order traversal of T (in that order): by the definition of the labelling, this monomial is a I-monomial. Now, observe that every parse tree of F is I-labelled, therefore F computes a polynomial which is a sum of I-monomials, so that computes a polynomial in $\mathscr{P}_I$.
Let F be a UPT formula of shape T that computes a polynomial f and I be a labelling. We construct F' in the following way: we delete every leaf Φ that is a $(u, \times)$-gate and not a variable in $X_{\mathrm{lab}(u)}$. By doing this, F' is I-labelled. Moreover, the polynomial f' computed by F' is simply the polynomial f where the monomials that are not I-monomials have been associated to a zero coefficient. □

The main idea in the proofs, following Nisan and Wigderson [23], is to apply a *restriction* (defined below) to the polynomial $\mathrm{IMM}_{n,d}$ by choosing an $I \subseteq [d]$ and setting each M_j ($j \notin I$) to the identity matrix. Under such a restriction, $\mathrm{IMM}_{n,d}$ becomes the polynomial IMM_I which, by Fact 26, has high relative rank. On the other hand, the restriction will be choosen such that given a small formula (with some restriction on the parse trees), there is a suitable choice of the restriction that makes its relative rank quite small.

Let us now carry out the above strategy. First we define a *restriction*, which is formally just a subset $I \subseteq [d]$. The set I defines a substitution ρ_I of the set of variables in $X = \bigcup_{i\in[d]} X_i$ as follows:

$$\rho_I(x) = \begin{cases} x & \text{if } x \in \bigcup_{i\in I} X_i, \\ 0 & \text{if } x \text{ is an offdiagonal entry of } M_j \text{ for } j \notin I, \\ 1 & \text{if } x \text{ is a diagonal entry of } M_j \text{ for } j \notin I. \end{cases}$$

In other words, we substitute all the variables in $\bigcup_{j\notin I} X_j$ such that each M_j ($j \notin I$) becomes the identity matrix. All variables from the set $\bigcup_{i\in I} X_i$ are left as is.

Every polynomial $P \in \mathbb{F}\langle X\rangle$ is transformed in the natural way by such a substitution. We call this new polynomial a *restriction of* P and denote it by $P|_I$.

For any restriction I, let $T|_I$ denote the tree obtained by removing all nodes $u \in V(T)$ such that $\mathrm{lab}(u) \cap I = \emptyset$ (in particular only leaves with labels from I survive in $T|_I$). The I-labelling of the tree $T|_I$ is given by the labelling function lab_I where $\mathrm{lab}_I(u) = \mathrm{lab}(u) \cap I$.

We make the following simple observations.

Observation 28. *1. If $P \in \mathscr{P}_{[d]}$, then $P|_I \in \mathscr{P}_I$.*

2. $\mathrm{IMM}_{n,d}|_I = \mathrm{IMM}_I$.

Lemma 29. *For any $[d]$-labelled UPT formula F of size s and shape T computing some f (note that $f \in \mathscr{P}_{[d]}$ by Lemma 27), there is a UPT formula $F|_I$ of size at most s and shape $T|_I$ computing $f|_I$.*

Proof. (Sketch) Let F be as in the statement and I be any restriction. If we replace every variable in the formula F by $\rho_I(x)$, we obtain by definition a formula F' which computes $f|_I$. F' is not a UPT formula since the leaves which were labeled by $j \notin I$ have been replaced by some constants – whereas leaves in a UPT formula have to be variables. We now transform F' to a UPT formula $F|_I$ in the following way: each addition gate Ψ in F' which was j-labelled in F for $j \notin I$ computes a constant (say α) and is wired to some multiplication gates $\Phi_0, \ldots, \Phi_l$. We delete Ψ from F' and multiply by α each edge outgoing Φ_k for any k. By doing this, the formula F' becomes a new formula $F|_I$ which is UPT with shape $T|_I$, and still computes $f|_I$.

□

Let T be a parse tree and $\pi = (v_r, \ldots, v_0)$ a path of length r in it[8]. We say that u is an *off-path node* of π if there is an $i < r$ such that u is a child of v_i and $u \neq v_{i+1}$. The set of off-path nodes of π is denoted $\mathrm{off}(\pi)$.

Lemma 30. *Let F be an I-labelled UPT formula of size s with shape T computing a polynomial $f \in \mathscr{P}_I$, and let $\pi = (v_r, \ldots, v_0)$ be a path in T. If we define u_j to be the jth node of* $\mathrm{off}(\pi) \cup \{v_r\}$ *that appears in the in-order traversal of T, then we can decompose f as:*

$$f = \sum_{i=1}^{k} \prod_{j=1}^{t} f_{i,j}$$

where:

- $k \leq s$
- $t = |\mathrm{off}(\pi) \cup \{v_r\}|$
- $f_{i,j} \in \mathscr{P}_{lab(u_j)}$.

[8]Recall that our trees are oriented towards the root.

Proof. Let F be a I-labelled UPT formula as in the statement and $\pi = (v_r, \ldots, v_0)$ be a path in T. Let $(u_1, \ldots, u_t)$ be the ordering of the set of nodes $(\text{off}(\pi) \cup \{v_r\})$ given by an in-order traversal of T. By Proposition 12, F is in normal form.

We say that a path in the circuit F is of *signature* π if the $+$-gates along this path are successively a $(v_r, +)$ gate, a $(v_{r-1}, +)$ gate, and so on until we get a $(v_0, +)$ gate. Let k be the number of paths in F with signature π, and $p_1, p_2, ..., p_k$ be these paths. As F is a formula, the number of paths from a leaf to the root is upper bounded by s. Therefore $k \leq s$.

Each parse formula F' (which is a subformula of F) of F has shape T and further each $+$-gate in F' has fan-in 1; thus, each parse formula contains one and only one path of signature π. The set S of parse formula of F is therefore naturally partitioned as $S = S_1 \cup \cdots \cup S_k$, where S_i is the set of parse formulas that contain the path p_i. Recall that if F' is a parse formula of F, we denote by $val(F')$ the monomial (along with its coefficient) computed by it. We have:

$$f = \sum_{F' \in S} val(F') = \sum_{i=1}^{k} \sum_{F' \in S_i} val(F')$$

From now, what remains to prove is that for each $a \in [k]$, $\sum_{F' \in S_a} val(F')$ is of the form $\prod_{j=1}^{t} f_{i,j}$ with the additional property that each $f_{i,j}$ is a polynomial in $\mathscr{P}_{lab(u_j)}$.

We fix a particular $a \in [k]$. The polynomial $\sum_{F' \in S_a} val(F')$ is nothing else than the polynomial computed by the I-labelled UPT formula G where for all j, each $(v_j, +)$ gate that is not present on the path p_a has been deleted (together with the entire subformula at that gate). Observe that for all m, the $(v_m, +)$ gate in G is of in-degree and out-degree 1, except for the output gate, which is a $(v_0, +)$ gate of in-degree 1 and is of out-degree 0.

Let $p_a = \Phi_r, \Psi_{r-1}, \Phi_{r-1}, \Psi_{r-2}, \Phi_{r-2}, \ldots, \Phi_1, \Psi_1, \Phi_0$, where Φ_0 is the root and for each m, Φ_m is the $(v_m, +)$-gate and Ψ_m is the $(v_m, \times)$-gates in p_a. By construction, the path p_a is present in G. We prove by induction that the following statement $H(m)$ holds for each $m \leq r$,

$H(m)$: If we denote by $(w_1, \ldots, w_{t_m})$ the ordering of the set of nodes $\text{off}((v_r, \ldots, v_m)) \cup \{v_r\}$ given by an in-order traversal of T, then the polynomial computed by Φ_m in G is of the form $\prod_{j=1}^{t_m} g_j$ where:

- $t_m = |\text{off}((v_r, \ldots, v_m)) \cup \{v_r\}|$
- $g_j \in \mathscr{P}_{lab(w_j)}$

We now prove $H(m)$ by downward induction on m. It is clearly true when $m = r$ since $t_r = 1$, as F is an I-labelled UPT formula and hence the polynomial computed by Φ_r is an element of $\mathscr{P}_{\text{lab}(v_r)}$.

Assume the statement $H(m+1)$ and let $\prod_{j \in [t_{m-1}]} h_j$ be the decomposition of the polynomial computed by the gate Φ_{m-1} given by the induction hypothesis. The polynomial computed by Φ_m (a $+$-gate of fan-in 1 in the formula G) is the product of the inputs of Ψ_m: assume that these inputs are (in left-to-right order) $\Phi'_1, \ldots, \Phi'_r$ with each Φ'_ℓ being a $(w''_\ell, +)$-gate for some $w''_\ell \in V(T)$. Let P_ℓ be the polynomial computed by Φ'_ℓ ($\ell \in [r]$). The gate Φ_{m+1} is one among $\Phi'_1, \ldots, \Phi'_r$: let us say it is Φ'_c. The polynomial computed by Φ_m is equal to $\prod_{\ell=1}^{r} P_\ell = (\prod_{\ell=1}^{c-1} P_\ell).(\prod_{j=1}^{t_{m-1}} h_j).(\prod_{\ell=c+1}^{r} P_\ell)$ by the induction hypothesis. Each P_ℓ is computed by a $(w''_\ell, +)$-gate and is thus a polynomial in $\mathscr{P}_{lab(w''_\ell)}$, and by induction, the h_j are polynomials in $\mathscr{P}_{\text{lab}(w'_j)}$, where $w'_1, \ldots, w'_{t_{m-1}}$ is the ordering of $\text{off}((v_r, \ldots, v_{m+1})) \cup \{v_r\}$ given by the in-order traversal of T. But it is not hard to see that $w''_1, \ldots, w''_{c-1}, w'_1, \ldots, w'_{t_{m-1}}, w''_{c+1}, \ldots, w''_{t_m}$ is exactly the ordering of $\text{off}((v_r, \ldots, v_m)) \cup \{v_r\}$ given by the in-order traversal of T, so that the induction holds, and the lemma is proved. □

5.2 Lower bound for a single UPT formula of $\times$-depth Δ

We define the *$\times$-depth* of a formula to be the maximum number of $\times$-gates that one can meet on a path from the root to a leaf. Note that if a formula has alternating $+$ and $\times$ gates on each path and has depth Δ' and $\times$-depth Δ, then $\Delta' \geq \Delta \geq \lceil \frac{\Delta'}{2} \rceil$. We will state our lower bounds bounds in terms of $\times$-depth.

Throughout this section, we assume that all the UPT formulas we consider don't have any multiplication gate of fan-in 1, or equivalently, the shape of any UPT formula we consider does not have any internal node of fan-in 1. This assumption is w.l.o.g. as shown below.

Lemma 31. *Given any UPT formula F of shape T and size s computing a polynomial f, there is another UPT formula F' of shape T' and size at most s computing f where T' has no internal nodes of fan-in 1 (and consequently F' has no $\times$-gates of fan-in 1). Further, if all internal nodes of T have fan-in at most $k \in \mathbb{N}$, then the same holds for T'.*

Proof. The transformation process is the following: a multiplication gate of fan-in 1 does not change its input and therefore can be deleted without changing the polynomial computed. Merging the two layers of $+$-gates above and below the deleted gate ensures the formula still alternates between $+$-gate and $\times$-gate. The shape T' of the new formula is simply the shape T where the internal nodes of fan-in 1 have been removed and replaced by an edge. Clearly, the new shape T' has the required property. □

Before attacking the main theorem, we will need one more lemma.

Lemma 32. *Let T be a tree with d leaves and depth Δ, such that all internal nodes are of in-degree strictly greater than 1. Then there is a path $\pi = (v_\ell, \ldots, v_0)$ in T such that $|\mathrm{off}(\pi) \cup \{v_\ell\}| \geq \Omega(\Delta d^{1/\Delta})$.*

Proof. See Appendix F. □

Theorem 33. *Let F be a UPT formula of $\times$-depth Δ, size s, computing $\mathrm{IMM}_{n,d} \in \mathbb{F}\langle X \rangle$. Then, $s \geq n^{\Omega(\Delta d^{1/\Delta})}$. In particular, any UPT formula for $\mathrm{IMM}_{n,d}$ must have size $n^{\Omega(\log d)}$.*

By our earlier observation relating the $\times$-depth with depth, we get the lower bound stated in the introduction.

Proof. Let F be a UPT formula as in the statement. By Lemma 27, we assume w.l.o.g that the formula is $[d]$-labelled and that the variables that appear in an input gate Φ of F, corresponding to a node $v(\Phi) \in T$, are all included in $X_{\mathrm{lab}(v(\Phi))}$ where lab is the $[d]$-labelling of T. By Lemma32, there is a path $\pi = (v_\ell, \ldots, v_0)$ in the shape T of F, such that $|\mathrm{off}(\pi) \cup \{v_\ell\}| \geq \Omega(\Delta d^{1/\Delta})$. Let us denote by t the size of $\mathrm{off}(\pi) \cup \{v_\ell\}$. We decompose $\mathrm{IMM}_{n,d}$ along this path by Lemma 30, as:

$$\mathrm{IMM}_{n,d}(X_1, X_2, ..., X_d) = \sum_{i=1}^{k} \prod_{j=1}^{t} f_{i,j}$$

with $k \leq s$. Each $f_{i,j}$ is a polynomial in $\mathscr{P}_{\mathrm{lab}(u_j)}$ where $(u_1, \ldots, u_t)$ is the ordering of $\mathrm{off}(\pi) \cup \{v_\ell\}$ given by an in-order traversal of T.
We now apply a restriction to this equality given by the subset I choosen in the following way: for each j, we select one element from $\mathrm{lab}(u_j)$ — we call it p_j — and add it to I. The set I is of size t. Under this restriction, each $f_{i,j}$ becomes a homogeneous linear polynomial in the variables X_{p_j}. We call these homogeneous linear polynomials $l_{i,j}$. We thus get

$$\mathrm{IMM}_I = \sum_{i=1}^{k} \prod_{j=1}^{t} l_{i,j}.$$

It is not hard to see that for each i, $\text{rank}(M_I(\prod_{j=1}^{t} l_{i,j})) \leq 1$. By Fact26 and subadditivity of the rank, we get:

$$n^{2\lfloor t/2 \rfloor} \leq k$$

Therefore, we get

$$s \geq k \geq n^{\Omega(\Delta t^{1/\Delta})}$$

as wanted. □

Remark 34. *Notice that this lower bound is tight for every $\times$-depth Δ, since the standard divide and conquer approach to computing $\text{IMM}_{n,d}$ gives in fact a UPT formula of size $n^{O(\Delta d^{1/\Delta})}$ and $\times$-depth Δ, for any $\Delta \leq \log d$.*

5.3 Separation between k-PT formulas and ABPs

In this section, we will prove a lower bound on the size of k-PT formulas computing $\text{IMM}_{n,d}$ as long as k is significantly smaller than 2^d. Recall that the total number of parse trees with d leaves is $2^{O(d)}$ (see for example [9]) and hence the results of this section intuitively imply that under any non-trivial upper bound on the number of parse trees appearing in the formula, we can obtain a separation between non-commutative formulas and ABPs.

The main theorem of this section is the following.

Theorem 35. *Let n,d be growing parameters with $d \leq \log n$. Then, any k-PT formula F computing $\text{IMM}_{n,d}$ has size at least n^{ℓ} where $\ell = \Omega(\lg d - \lg\lg k)$. In particular, if $k = 2^{o(d)}$, the $\text{size}(F) \geq n^{\omega(1)}$ and if $k = 2^{d^{1-\Omega(1)}}$, then $\text{size}(F) \geq n^{\Omega(\log d)}$.*

Remark 36. *We say a few words about the assumption $d \leq \log n$. The standard divide-and-conquer approach for computing $\text{IMM}_{n,d}$ yields a (UPT) formula of size $n^{O(\log d)}$. It would be quite surprising if this standard algorithm were not optimal in terms of formula size.*

Intuitively, improving on the standard divide-and-conquer algorithm gets harder as d gets smaller: this is because any formula of size $n^{o(\log d)}$ for computing $\text{IMM}_{n,d}$ can be straightforwardly used to recursively obtain formulas for $\text{IMM}_{n,D}$ of size $n^{o(\log D)}$ for any $D > d$. Thus, the case of smaller d, which seems harder algorithmically, is natural first candidate for lower bounds.

Let T be a parse tree. We say that a node $u \in V(T)$ is *odd* if the number of leaves in the subtree rooted at u is odd. Given a path π, let $\text{odd}(\pi)$ denote the set of odd off-path nodes in it.

Lemma 37. *Let F be an I-labelled UPT formula of size s with shape T computing polynomial f. If T has a path $\pi = (v_r, \ldots, v_0)$ with $|\text{odd}(\pi)| \geq \ell$, then $\text{rel-rank}_I(f) \leq \frac{s}{n^{\ell-1}}$.*

Proof. Let $(u_1, \ldots, u_t)$ be the ordering of the set of nodes $(\text{off}(\pi) \cup \{v_0\})$ given by an in-order traversal of T, and $f = \sum_{i=1}^{k} \prod_{j=1}^{t} f_{i,j}$ be a decomposition given by Lemma30, where each $f_{i,j}$ is in $\mathscr{P}_{lab(u_j)}$. By Fact 26, we know that $\text{rank}(M_I(f_{i,j})) \leq n^{2\lfloor |\text{lab}(u_j)|/2 \rfloor}$. Hence, by the subadditivity of rank and Lemma8,

we have:

$$\begin{aligned}\operatorname{rank}(M_I(f)) &\leq \sum_{i=1}^{k}\prod_{j=1}^{t}\operatorname{rank}(M_I(f_{i,j}))\\ &\leq \sum_{i=1}^{k}\prod_{j=1}^{t} n^{2\lfloor\frac{|lab(u_j)|}{2}\rfloor}\\ &\leq \sum_{i=1}^{k} n^{2\sum_{j=1}^{t}\lfloor\frac{|lab(u_j)|}{2}\rfloor}\\ &\leq \sum_{i=1}^{k} n^{2(\lfloor\frac{|I|}{2}\rfloor-\lfloor\frac{|\mathrm{odd}(\pi)|}{2}\rfloor)}\end{aligned}$$

As $k \leq s$, this implies that $\text{rel-rank}_I(f) \leq \frac{s}{n^{2\lfloor\frac{|\mathrm{odd}(\pi)|}{2}\rfloor}} \leq \frac{s}{n^{\ell-1}}$. □

We now try to show that, given a small k-PT formula, there is a suitable choice of the restriction that makes its relative rank quite small. To do this, we will use Lemma 37, which translates the statement to a combinatorial statement about some trees. On the other hand, IMM remains high rank under arbitrary restrictions by Fact 26. This will prove Theorem 35.

The main technical lemma in the proof of Theorem 35 is the following.

Lemma 38. *Let T be any tree with d leaves such that every internal node has fan-in exactly 2. Assume we choose $I \subseteq [d]$ by adding each $i \in [d]$ to I independently with probability $1/2$. Then for any $\ell \in \mathbb{N}$*

$$\Pr_I[T|_I \text{ has no path } \pi \text{ such that } |\mathrm{odd}(\pi)| \geq \ell] \leq \exp(-\Omega(\frac{d}{2^{8\ell}})).$$

Assuming the above lemma, we can prove Theorem 35 as follows.

Proof of Theorem 35. We assume throughout that $\lg d - \lg\lg k$ is larger than a large enough constant (to be chosen later), since otherwise the theorem is trivial. Let $\ell = \lfloor\frac{1}{10}(\lg d - \lg\lg k)\rfloor$, which is also assumed to be large enough.

Assume that F is a k-PT formula of size s computing $\mathrm{IMM}_{n,d}$. If $s \geq n^{\ell/4}$, then we are done.

Otherwise, we argue as follows. By Lemma 16, there exist UPT formulas $F_1, \ldots, F_k$ of size at most s^2 each such that

$$\mathrm{IMM}_{n,d} = \sum_{i=1}^{k} f_i$$

where f_i is the polynomial computed by F_i. Let T_i denote the shape of F_i. By Lemma 13, we can assume that each internal node of T_i has fan-in exactly 2.

By Lemma 27 and Lemma 31, for each F_i, there is a $[d]$-labelled UPT formula F_i' of shape T_i and size at most s^2 that computes the polynomial f_i' that is obtained from f_i by removing moonomials that are not $[d]$-monomials. Since $\mathrm{IMM}_{n,d} \in \mathscr{P}_{[d]}$, we see that

$$\mathrm{IMM}_{n,d} = \sum_{i=1}^{k} f_i'. \tag{5}$$

Now, choose a random restriction I by adding each $j \in [d]$ to I independently with probability $1/2$. Consider the relative rank of the polynomials on both sides of (5) after the restriction. For the left hand side, we know using Fact 26 that for any I,

$$\text{rel-rank}_I(\mathrm{IMM}_{n,d}|_I) = \text{rel-rank}_I(\mathrm{IMM}_I) = 1. \tag{6}$$

We now consider the right hand side of (5). By Lemma 29, for any choice of restriction I and $i \in [k]$, the restricted polynomial $f_i'|_I$ has a UPT formula $F_i'|_I$ of size at most s^2 and shape $T_i|_I$ computing $f_i'|_I$. For each $i \in [k]$, let $\mathscr{E}_i$ denote the event that $T_i|_I$ has no path π such that $|\mathrm{odd}(\pi)| \geq \ell$. By Lemma 38, we know that the probability of $\mathscr{E}_i$ is at most $\exp(-\Omega(\frac{d}{2^{2\ell}}))$. Let $\mathscr{E} = \bigvee_{i=1}^{k} \mathscr{E}_i$. By a union bound we have

$$\Pr[\mathscr{E}] \leq k \cdot \exp\left(-\Omega\left(\frac{d}{2^{2\ell}}\right)\right) < 1 \tag{7}$$

if $(\lg d - \lg\lg k)$ is large enough.

If I is such that the event $\mathscr{E}$ does not occur, then for this choice of I and any $i \in [k]$, by Lemma 37, $\text{rel-rank}_I(f_i'|_I) \leq \frac{s^2}{n^{\ell-1}} \leq \frac{1}{n^{(\ell/2)-1}}$, where the final inequality follows from our assumption that $s < n^{\ell/4}$. Now, since $\text{rel-rank}_I(\cdot)$ is subadditive, we have

$$\text{rel-rank}_I\Big(\sum_{i\in[k]} f_i'\Big) \leq \frac{k}{n^{(\ell/2)-1}} \leq \frac{2^d}{n^{(\ell/2)-1}} \leq \frac{1}{n^{(\ell/2)-2}} < \frac{1}{n}$$

where the final two inequalities follow from the fact that $d \leq \lg n$ and the assumption that ℓ is greater than some fixed constant. This contradicts (5) and (6) and hence concludes the proof of the theorem. □

5.3.1 Proof of Lemma 38

We impose a natural partial order on the vertices in $V(T)$ by saying that $u \preceq v$ for $u, v \in V(T)$. Given a set of paths $P = \{\pi_1, \ldots, \pi_r\}$ in the tree T, we say that P is *independent* if the sets $\mathrm{off}(\pi_i)$ ($i \in [r]$) are pairwise disjoint and moreover, the set $\mathrm{off}(P) := \bigcup_i \mathrm{off}(\pi_i)$ forms an antichain w.r.t. the partial order $\preceq$ (informally, no node in $\mathrm{off}(P)$ is an ancestor of another).

We show the following claim.

Claim 39. *Assume T is as in the statement of the lemma. Then for any $\ell \geq 1$, there is an independent set P of paths in T of length ℓ such that $|P| = \Omega(d/2^{2\ell})$.*

Proof. Given a tree T as in Lemma 38, let us define T' to be the subtree of T which contains every node of T that has height ℓ or more (here the height of a node u is the length of the longest path from a leaf from $\mathscr{L}(u)$ to u). Though every internal node in T has degree two, T' may have internal nodes of degree two as well as one. The leaves of T' are those internal nodes of T that have height exactly ℓ.

The main idea is as follows: Suppose T' has 'many' leaves, then it is easy to find many independent paths in T. This is because each leaf v of T' is a node in T with at least one path of length ℓ rooted at v. Also, two leaves u, v of T' are nodes in T at height exactly ℓ. This gives us as many independent paths as the number of leaves in T'. On the other hand, if T' does not have many leaves, then it also does not have many degree two nodes. In this case, by throwing away all degree two nodes of T', we get many components. Each component is a path and not all can be of length less than ℓ. Subdividing the long paths into paths of length ℓ then gives us the set of independent paths[9]. We now work out the details.

As every internal node of T has degree two, the number of nodes at height h, for any parameter $h \geq 1$, is at least half of the number of nodes at height $h-1$. Using this, we can inductively prove that the number of leaves in T' and therefore $|V(T')|$ is $\geq \frac{d}{2^\ell}$. We use s to denote $\frac{d}{2^\ell}$.

Case I, the number of leaves in T' is $\geq s/100\ell$: Each leaf v in T' has a subtree rooted at it in T, say T_v. For each leaf v in T', T_v has at least one path of length ℓ from v to a leaf of T. Let us call this path π_v. As the leaves of T' are all the nodes at height ℓ, for two leaves of T', say $u \neq v$, and for any vertex x

[9]Note that we only need the off-path nodes to form an antichain and not the nodes on the path itself.

in off(π_v) and any vertex y in off(π_u), neither $x \preceq y$ nor $y \preceq x$. (If one of the conditions holds then it will contradict that both u, v have height ℓ.)

Case II, the number of leaves in T' is $< s/100\ell$: It is easy to see that the number of degree two nodes in any tree is upper bounded by the number of leaves in the tree. Therefore, T' has at most $s/100\ell$ degree two nodes. Let F' be the forest obtained by deleting all degree two nodes of T'. F' is a collection of paths. As we deleted degree two nodes, the total number of components created in F' is at most twice the number of degree two nodes, i.e at most $s/50\ell$.

We call a component *small* if it has at most ℓ nodes; *large* otherwise. The total number of nodes in small components is at most $(s/50\ell) \cdot \ell = s/50$. We will not consider such components. We know that $|V(T')| = s$. Therefore, we are left with at least $s - s/50 \geq s/2$ nodes even if we discard all the small components.

We only consider the large components. Let C be a component with r vertices, where $r > \ell$. It can be broken down into $\lfloor \frac{r}{\ell+1} \rfloor$ paths, each of length ℓ. This will give us $\lfloor s/2(\ell+1) \rfloor$ many paths in total. As $\lfloor s/2(\ell+1) \rfloor > d/2^{2\ell}$, if we argue that all these paths are independent, we will be done.

It is not very hard to see why these paths are all pairwise independent. Suppose two paths π, π' belonged to the same large component, then consider a vertex $x \in \text{off}(\pi)$ and $y \in \text{off}(\pi')$. We observe that neither $x \preceq y$ nor $y \preceq x$. This is because a common ancestor of x, y is a vertex of either π or π'. Therefore, any such two paths are independent. Now say π, π' are two paths which come from two different large components. Then for $x \in \text{off}(\pi)$ and $y \in \text{off}(\pi')$, the common ancestor of x, y is a degree two node, which we deleted. Again, we can see that neither $x \preceq y$ nor $y \preceq x$.

□

Given Claim 39, we proceed as follows. Applying Claim 39 with 4ℓ in place of ℓ, we obtain a set $P = \{\pi_1, \ldots, \pi_r\}$ of independent paths in T with $r = \Omega(d/2^{8\ell})$. For each π_i, let off(π_i) $= \{u_{i,1}, \ldots, u_{i,4\ell}\}$. Note that these off-path nodes all exist since each internal node of T is assumed to have fan-in 2.

We now consider the effect of the random restriction I, chosen as in the lemma statement, on the tree T. For any $i \in [r]$ and $j \in [4\ell]$, let $Z_{i,j} \in \{0,1\}$ be the random variable that is 1 if $u_{i,j}$ is present in $T|_I$ and is an odd node, and 0 otherwise; equivalently, if lab is the $[d]$-labelling of T, then $Z_{i,j} = 1$ iff the number of leaves v such that lab$(v) \in I$ is odd. Note that $\mathbf{E}[Z_{i,j}] = (1/2)$ for each $i \in [r]$ and $j \in [4\ell]$. Moreover, since P is an independent set of paths, the sets of leaves in the subtrees of $u_{i,j}$ (for different i, j) are pairwise disjoint and consequently, the random variables $Z_{i,j}$ (for various i, j) are mutually independent. In particular, by a Chernoff bound applied to $Z := \sum_{i \in [r], j \in [4\ell]} Z_{i,j}$, we get

$$\Pr[Z \leq r\ell] = \Pr\left[Z \leq \frac{1}{2}\mathbf{E}[Z]\right] \leq \exp(-\Omega(\mathbf{E}[Z])) \leq \exp(-\Omega(r\ell)) \leq \exp(-\Omega(\frac{d}{2^{8\ell}})).$$

Note that when Z is the total number of nodes in off(P) that end up as odd nodes in $T|_I$. Hence, if $Z > r\ell$, then the number of odd nodes per (surviving) path of P in $T|_I$ is at least $r\ell/r = \ell$. In particular, there must be some path π in $T|_I$ such that $|\text{odd}(\pi)| \geq \ell$. This concludes the proof of Lemma 38.

6 Deterministic PIT

6.1 PIT for UPT circuits

In this section, we give a deterministic PIT algorithm for UPT circuits. A previous algorithm for this problem by Lagarde et al. [19], based on the ideas of Arvind et al. [3], only works over fields of characteristic 0. Our algorithm, which is an adaptation of the algorithm of Raz and Shpilka [25], is field independent. The algorithm is *whitebox* in the sense that it needs access to the circuit itself and not simply an oracle that evaluates the polynomial computed by the circuit at chosen points.

Theorem 40. *Let $N, s \in \mathbb{N}$ be parameters. There is a deterministic algorithm running in time* $\mathrm{poly}(s)$ *which, on input a UPT circuit C of size at most s over N variables, checks if C computes the zero polynomial or not.*

Proof. Let C be the input UPT circuit. Let T be the unique parse tree of the circuit C (it is easy to determine T from the circuit C by constructing an arbitrary parse formula of C and obtaining the parse tree corresponding to it). By Proposition 12 and Lemma 13, we can assume without loss of generality that C is in normal form and that T has fan-in bounded by 2.

For each node $v \in V(T)$, let r_v denote the number of $(v,\times)$-gates and t_v the number of $(v,+)$-gates. We also identify the $(v,\times)$-gates with $[r_v]$ and $(v,+)$-gates with $[t_v]$ in an arbitrary way. For any $v \in V(T)$ and any monomial $m \in \mathscr{M}_{\deg(v)}(X)$, let $\xi^v_m \in \mathbb{F}^{r_v}$ be defined so that for any $i \in [r_v]$, the ith entry $\xi^v_m(i)$ of the vector ξ^v_m is the coefficient of the monomial m in the polynomial computed at the ith $(v,\times)$ gate. Similarly, let $\chi^v_m \in \mathbb{F}^{t_v}$ be the coefficient vector of the monomial m at the $(v,+)$-gates.

The idea of the algorithm is to compute, for each $v \in V(T)$, a set $B_{v,+} \subseteq \mathscr{M}_{\deg(v)}(X)$ of size at most t_v such that the set of vectors $\tilde{B}_{v,+} = \{\chi^v_m \mid m \in B_{v,+}\}$ is a linearly independent set of vectors that generates all the vectors in $\tilde{C}_{v,+} := \{\chi^v_m \mid m \in \mathscr{M}_{\deg(v)}(X)\} \subseteq \mathbb{F}^{t_v}$. In particular, the polynomial computed by the circuit C is non-zero iff for u being the root of T, there is a $\chi \in \tilde{B}_{u,+}$ such that the entry of χ corresponding to the output gate of the circuit is non-zero.[10]

Thus, it suffices to compute the sets $B_{v,+}$ and $\tilde{B}_{v,+}$ for each $v \in V(T)$. In order to do so, it will also help to compute $B_{v,\times} \subseteq \mathscr{M}_{\deg(v)}(X)$ and $\tilde{B}_{v,\times} = \{\xi^v_m \mid m \in B_{v,\times}\}$ of size at most r_v each so that the set of vectors $\tilde{B}_{v,\times}$ is a linearly independent set of vectors that generates all the vectors in $\tilde{C}_{v,\times} := \{\xi^v_m \mid m \in \mathscr{M}_{\deg(v)}(X)\} \subseteq \mathbb{F}^{r_v}$.

The algorithm begins by choosing the sets $B_{v,\times}$ for each leaf node $v \in V(T)$. This may be done efficiently since $\deg(v) = 1$ for each leaf node and hence the number of monomials $m \in \mathscr{M}_{\deg(v)}(X)$ is exactly $|X| = N$. By computing the coefficient vectors for each such monomial and performing Gaussian elimination, we can find a suitable set $B_{v,\times}$ as required in time $\mathrm{poly}(N,s) = \mathrm{poly}(s)$.

To compute these bases for nodes higher up in T we proceed inductively as follows. **Sum gates.** We first describe how to construct $B_{v,+}$ and $\tilde{B}_{v,+}$ given $B_{v,\times}$ and $\tilde{B}_{v,\times}$. Since each $(v,+)$-gate computes a linear combination of the $(v,\times)$-gates, we see that there is a matrix $M_v \in \mathbb{F}^{t_v \times r_v}$ such that $\chi^v_m = M_v \xi^v_m$ for every $m \in \mathscr{M}_{\deg(v)}(X)$. In particular, given sets $B_{v,\times}$ and $\tilde{B}_{v,\times}$ as above, the set $\{\chi^v_m \mid m \in B_{v,\times}\}$ is a *spanning set* for the set $\tilde{C}_{v,+}$. By Gaussian elimination, we can choose a basis $\tilde{B}_{v,+} \subseteq \tilde{B}_{v,\times}$ in time $\mathrm{poly}(N, t_v, r_v) = \mathrm{poly}(s)$ and choose $B_{v,+}$ to be the corresponding set of monomials. **Multiplication gates.** Now let $v \in V(T)$ be an internal node with children u and w. We show how to compute $B_{v,\times}$ and $\tilde{B}_{v,\times}$ given $B_{u,+}, B_{w,+}, \tilde{B}_{u,+}$ and $\tilde{B}_{w,+}$.

Let $r = r_v$ and let Φ_i be the ith $(v,\times)$-gate in C for each $i \in [r]$. Let Φ'_i and Φ''_i be the left and right children respectively of Φ_i; note that Φ'_i is a $(u,+)$-gate and Φ''_i a $(w,+)$-gate. For monomials $m' \in \mathscr{M}_{\deg(u)}(X)$ and $m'' \in \mathscr{M}_{\deg(w)}(X)$, let $\lambda^u_{m'}$ and $\lambda^w_{m''} \in \mathbb{F}^r$ denote the coefficient vectors of m' and m'' at the gates Φ'_i ($i \in [r]$) and Φ''_i ($i \in [r]$) respectively.[11] For any monomial m', each entry of the vector $\lambda^u_{m'}$ is the coefficient of the monomial m' at some $(u,+)$-gate and hence an entry of the vector $\chi^u_{m'}$. In particular, $\lambda^u_{m'} = P_u \chi^u_{m'}$ for some linear projection P_u; a similar fact is true for the $\lambda^w_{m''}$ as well. Thus, the vectors $\{\lambda^u_{m'} \mid m' \in B_{u,+}\}$ span all the vectors in $\{\lambda^u_{m'} \mid m' \in \mathscr{M}_{\deg(u)}(X)\}$ and similarly, $\{\lambda^w_{m''} \mid m'' \in B_{w,+}\}$ spans all the vectors in $\{\lambda^w_{m''} \mid m'' \in \mathscr{M}_{\deg(w)}(X)\}$.

Now, note that for any monomial $m \in \mathscr{M}_{\deg(v)}(X)$, there is a unique pair of monomials $m' \in \mathscr{M}_{\deg(u)}(X)$ and $m'' \in \mathscr{M}_{\deg(w)}(X)$ such that $m = m'm''$. Further, the coefficient of monomial m in the polynomial computed at Φ_i is the product of the coefficients of m' at Φ'_i and m'' at Φ''_i. In other words, we have $\lambda^v_m = \lambda^u_{m'} \cdot \lambda^w_{m''}$, the pointwise product of the vectors $\lambda^u_{m'}$ and $\lambda^w_{m''}$. By linearity, it follows that the coefficient vectors corresponding to the monomials in $B_{u,w} := B_{u,+} \cdot B_{w,+} = \{m'm'' \mid m' \in B_{u,+}, m'' \in B_{w,+}\}$

[10] Recall that the output gate of the circuit C is always assumed to be a $+$ gate, possibly of fan-in 1.

[11] Note that the gates Φ'_i and Φ'_j may coincide even if $i \neq j$. This does not matter for our argument.

span $\tilde{C}_{v,\times}$. Since $|B_{u,w}|$ has size at most s^2, both $B_{u,w}$ and the corresponding coefficient vectors can be computed in time $\text{poly}(s)$. By Gaussian elimination, we can find in time $\text{poly}(s)$ the sets $B_{v,\times}$ and $\tilde{B}_{v,\times}$ as required.

This completes the description of the algorithm and its analysis. From the analysis above, it is clear that the algorithm runs in time $\text{poly}(s)$. We have shown Theorem 40. □

6.2 PIT for sum of UPT circuits

In this section we will give a deterministic polynomial time algorithm for the PIT problem for the sum of k UPT circuits. Recently a deterministic algorithm was designed by Gurjar et al. [12] for polynomial identity testing of sum of ROABPs. Our algorithm uses a similar idea for the PIT of sum of UPT circuits. Our PIT algorithm is white box, i.e. it uses the structure of the underlying UPT circuits.

Theorem 41. *Let $N, s, k \in \mathbb{N}$ be parameters. There is a deterministic algorithm running in time $s^{O(2^k)}$ which, on input $k+1$ UPT circuits $C_0, C_1, \ldots, C_k$ (of possibly differing shapes) each of of size at most s over N variables, checks if $\sum_{i=0}^{k} C_i$ computes the zero polynomial or not.*

Proof idea:

Say that circuit C_i has shape T_i for $i \in [0,k]$ (it is easy to compute T_i given each C_i as observed in Section 6.1). By Proposition 12 and Lemma 13, we can assume without loss of generality that each C_i is in normal form and that T_i has fan-in bounded by 2.

Let P_i be the polynomial computed by the UPT circuit C_i for each $0 \leq i \leq k$. Let $P = -P_0$ and let $Q = \sum_{i=1}^{k} P_i$. Note that in this notation, checking whether $\sum_{i=0}^{k} P_i \equiv 0$ or not is equivalent to checking whether $P \equiv Q$ or not. We will present an algorithm to do this in four steps.

> **Step 1:** We show how to build efficiently a small set of *characterizing identities* for the polynomial P. We will ensure that this set of identities is of size $\text{poly}(N,s,d) = \text{poly}(s)$.
>
> **Step 2:** We will then check whether all the identities hold for the polynomial Q as well. This is done by a call to the PIT algorithm for the sum of k UPT circuits. We will analyze the complexity of this step and bound it by $s^{O(2^k)}$.
>
> **Step 3:** We will then show that if Q satisfies all the characterizing identities, and moreover P and Q agree on a small set of coefficents, then the two polynomials are in fact identical.
>
> **Step 4:** We will show that testing the equality of the above set of coefficients of P and Q can also be performed in time $\text{poly}(s)$.

We now give a more detailed outline of the above steps with the statements of many formal claims. For the sake of exposition, we postpone the proofs of these intermediate claims to the end of this section.

Step 1: We now introduce some notation to formally define the characterizing identities for a polynomial defined by a UPT circuit. Let $I = [i,j]$ be an interval in $[d]$, i.e. $1 \leq i \leq j \leq d$. If the interval is of size 1, i.e. $I = [i,i]$ then we simply use i to denote it. Recall that $\mathscr{M}_{|I|}(X)$ stands for all monomials of degree exactly $|I|$. For any r, let $\mathbb{F}\langle X\rangle_r$ be the set of homogeneous polynomials of degree r.

For any interval I in $[d]$ and any monomial $m \in \mathscr{M}_{|I|}(X)$, we define a map $\partial_{I,m} : \mathbb{F}\langle X\rangle_d \to \mathbb{F}\langle X\rangle_{d-|I|}$, which is defined as follows:

$$\partial_{I,m}(P) = \sum_{m_1,m_2 : deg(m_1)=i-1, deg(m_2)=d-j} \alpha_{m_1,m,m_2} \cdot m_1 \cdot m_2,$$

where α_{m_1,m,m_2} is the coefficient of the monomial $m_1 \cdot m \cdot m_2$ in P. Informally, $\partial_{I,m}$ is an operator, which when applied to a polynomial P of degree d, retains only those monomials of P (along with their

coefficients) which have the monomial m at exactly the positions in the interval I, while substituting the constant 1 for all the variables in positions indexed by I.

Let T be T_0, the shape of the parse tree corresponding to P. For each $v \in V(T)$, we use I_v to denote the interval $[\mathrm{pos}(v), \mathrm{pos}(v) + \deg(v) - 1]$ where $\mathrm{type}(v) = (\mathrm{pos}(v), \deg(v))$ is as defined in Section 2.4.

Starting from the leaves, we start building identities corresponding to each of the nodes in the tree T. Formally, we show the following inductive claim.

Claim 42. *There is an algorithm that runs in time* $\mathrm{poly}(s)$ *that, for every node* v *in* T*, computes a set* $B_v \subseteq \mathscr{M}_{\deg(v)}(X)$ *such that* $|B_v| \leq s$ *and also*

- *if* v *is a leaf node and* $I_v = i$ *then, for each* $x \in X$ *and for each* $m \in B_v$*, it computes coefficients* $c^v_{x,m}$ *such that* $\partial_{i,x}(P) = \sum_{m \in B_v} c^v_{x,m} \cdot \partial_{i,m}(P)$.
- *if* v *is an internal node with children* u, w*, then for all* $m' \in B_{u,w} := B_u \cdot B_w$ *and for all* $m \in B_v$*, it computes coefficients* $c^v_{m',m}$ *such that* $\partial_{I_v,m'}(P) = \sum_{m \in B_v} c^v_{m',m} \cdot \partial_{I_v,m}(P)$.

The algorithm for the above is almost identical to the PIT algorithm in Section 6. Note that the size of the output of the algorithm is $\mathrm{poly}(N, s, d) = \mathrm{poly}(s)$.

We will prove this claim later.

Step 2: Let us assume that the above claim holds. Now if $P = Q$, then the same set of identities must also hold for the polynomial Q. We now describe how one can check that Q satisfies these identities (the algorithm can safely reject if some identity is not satisfied by Q). Suppose we have all the identities for P along with the sets B_v for all nodes v in T and all the coefficients $\left(c^v_{m',m}\right)_{m' \in B_{u,w}, m \in B_v}$ again for every node v in T.

In general, we need to check identities of the following form when v is an internal node with children u, w: $\partial_{I_v,\tilde{m}}(Q) = \sum_{m \in B_v} c_{\tilde{m},m} \cdot \partial_{I_v,m}(Q)$ for each $\tilde{m} \in B_{u,w}$. (A similar check has to be made when v is a leaf node.)

Recall that $Q = \sum_{i=1}^{k} P_i$. Therefore, we can rewrite the above identity as follows.

$$\sum_{i=1}^{k} \partial_{I_v,\tilde{m}}(P_i) = \sum_{m \in B_v} c_{\tilde{m},m} \cdot \sum_{i=1}^{k} \partial_{I_v,m}(P_i).$$

Rearranging this we get

$$\sum_{i=1}^{k} \left[\sum_{m \in B_v} c_{\tilde{m},m} \cdot \partial_{I_v,m}(P_i) - \partial_{I_v,\tilde{m}}(P_i) \right] \equiv 0. \tag{8}$$

We first show that each of the k terms in the above sum has a small UPT circuit.

Claim 43. *For each* $i \in [k]$, $\sum_{m \in B_v} c_{\tilde{m},m} \cdot \partial_{I_v,m}(P_i) - \partial_{I_v,\tilde{m}}(P_i)$ *can be computed by a UPT circuit of size at most* $O(s^2)$*. Further, these circuits can be constructed in time* $\mathrm{poly}(k, s)$.

By the above claim, Equation 8 reduced to an identity testing question for the sum of at most k UPT circuits, and hence can be solved recursively. Finally, when we get to the case that $k = 1$, we simply appeal to our result from Section 6.1. Using Claim 43 (which we will prove later) and the algorithm from Section 6.1 for a single UPT circuit, we see that this step can be performed in time $(s^2)^{O(2^{k-1})} = s^{O(2^k)}$.

Step 3: Now suppose all the above checks succeed. That is, we have been able to ensure that the following statements hold:

- For every leaf node v, $x \in X, m \in B_v$ and i such that $I_v = i$:

$$\partial_{i,x}(P) = \sum_{m \in B_v} c^v_{x,m} \cdot \partial_{i,m}(P) \text{ and } \partial_{i,x}(Q) = \sum_{m \in B_v} c^v_{x,m} \cdot \partial_{i,m}(Q). \tag{9}$$

- For every internal node v with children u, w, for every $m \in B_v$ and $m' \in B_{u,w}$ we have:

$$\partial_{I_v,m'}(P) = \sum_{m \in B_v} c^v_{m',m} \cdot \partial_{I_v,m}(P) \text{ and } \partial_{I_v,m'}(Q) = \sum_{m \in B_v} c^v_{m',m} \cdot \partial_{I_v,m}(Q) \tag{10}$$

Claim 44. *Equations 9, 10 imply that for any node $v \in V(T)$ and any $m' \in \mathscr{M}_{|I_v|}(X)$ and $m \in B_v$, there exist $c^v_{m',m} \in \mathbb{F}$ such that*

$$\partial_{I_v,m'}(P) = \sum_{m \in B_v} c^v_{m',m} \cdot \partial_{I_v,m}(P) \text{ and } \partial_{I_v,m'}(Q) = \sum_{m \in B_v} c^v_{m',m} \cdot \partial_{I_v,m}(Q).$$

Note that (9) and (10) only give us a polynomially large set of common identities satisfied by P and Q. The content of Claim 44 is that we can use these to infer an *exponentially* (since the size of $\mathscr{M}_{|I|}(X)$ is exponential) large set of common identitites for P and Q.

We will present the proof of Claim 44 later. For now let us assume this claim.

Now, let v_0 be the root of T. We check that for each $m \in B_{v_0}$, $\partial_{[d],m}(P) = \partial_{[d],m}(Q)$ (as described in Step 4). Note that for any m of degree d, $\partial_{[d],m}(P)$ and $\partial_{[d],m}(Q)$ are simply coefficients of the monomial m in P and Q respectively. Again, if any of these coefficients are not equal, we can safely reject. However, if these checks succeed, using Claim 44, we can see that *all* the coefficients of polynomials P and Q are equal and hence they are the same polynomial. In this case, we accept.

Step 4: As noted above, $\partial_{[d],m}(P)$ and $\partial_{[d],m}(Q)$ are simply coefficients of the monomial m in the polynomials P and Q respectively. We use the following lemma proved in [5] to compute these coefficients.

Lemma 45 ([5])**.** *Given access to a non-commutative circuit C of size s which is computing the polynomial f of degree d and given a monomial m, the coefficient of m in f can be computed in time polynomial in s, d.*

This finishes the description of the four main steps. We now prove the claims used in these steps.

Proof sketch of Claim 42. We follow exactly the procedure in the PIT algorithm for UPT circuits in Section 6.1 and compute sets $B_{v,+}, \tilde{B}_{v,+}, B_{v,\times}$, and $\tilde{B}_{v,\times}$ exactly as in that algorithm. We will take our sets B_v to be the sets $B_{v,\times}$ for each $v \in V(T)$. Clearly $|B_v| \leq s$ for each v.

To compute the coefficients $c^v_{m',m} \in \mathbb{F}$, we proceed as follows. For any leaf node $v \in V(T)$, $y \in X$ and $x \in B_v$, we choose $c^v_{y,x}$ such that we have $\xi^v_y = \sum_{x \in B_v} c^v_{y,x} \xi^v_x$.

For an internal node $v \in V(T)$ with children u and w, and any $m' \in B_u \cdot B_w$, we note that by the definition of $B_{v,\times}$ in the proof of Theorem 40, each $m' \in B_u \cdot B_w$, $\xi^v_{m'}$ lies in $\mathrm{Span}(\tilde{B}_{v,\times})$ and hence we can find $c^v_{m',m}$ such that $\xi^v_{m'} = \sum_{m \in B_v} c^v_{m',m} \xi^v_m$.

This concludes the description of the algorithm. To show that this works as intended, it suffices to prove the following claim.

Claim 46. *Let $v \in V(T)$ and $t := \deg(v)$. For any m' such that $\deg(m') = t$ and for any set $B \subseteq \mathscr{M}_t(X)$, if $\xi^v_{m'} = \sum_{m \in B} c^v_{m',m} \cdot \xi^v_m$ then $\partial_{I,m'}(P) = \sum_{m \in B} c^v_{m',m} \cdot \partial_{I,m}(P)$.*

Proof. Let $v \in V(T)$ be such that $\mathrm{type}(v) = (t, p)$, where t is $\deg(v)$ and p is $\mathrm{pos}(v)$. Let K_v be the number of nodes in C_0 corresponding to v. For any polynomial computed by a UPT circuit, [19] proved the following decomposition lemma, which we will recall and use below.

Lemma 47 (Proposition 1 [19])**.** *Let P be a polynomial of degree d computed by a UPT circuit of size s with a parse tree T. Let (t,p) be the type of a node $v \in V(T)$ and let K_v be the number of gates in C of that type. Let $f_1, f_2, \ldots, f_{K_v}$ be the polynomials computed by these gates each of degree t. Then P can be written as*

$$P = \sum_{j=1}^{K_v} f_j \times_p h_j,$$

where $\forall j, 1 \leq j \leq K_v \deg(h_j) = d - t$.

Using the above lemma our claim follows. Given below is the detailed proof of the claim.

$$\begin{aligned}
\partial_{I,m'}(P) &= \partial_{I,m'}\left(\sum_{j=1}^{K_v} f_j \times_p h_j\right) && (a)\\
&= \partial_{I,m'}\left(\sum_{j=1}^{K_v} \xi^v_{m'}(j)\cdot m' \times_p h_j + \sum_{\tilde{m}\neq m'} \xi^v_{\tilde{m}}(j)\cdot \tilde{m} \times_p h_j\right)\\
&= \partial_{I,m'}\left(\sum_{j=1}^{K_v} \xi^v_{m'}(j)\cdot m' \times_p h_j\right) + \partial_{I,m'}\left(\sum_{\tilde{m}\neq m'} \xi^v_{\tilde{m}}(j)\cdot \tilde{m} \times_p h_j\right)\\
&= \sum_{j=1}^{K_v} \xi^v_{m'}(j)\cdot 1 \times_p h_j + \partial_{I,m'}\left(\sum_{\tilde{m}\neq m'} \xi^v_{\tilde{m}}(j)\cdot \tilde{m} \times_p h_j\right) \quad (\to 0)\\
&= \sum_{j=1}^{K_v}\sum_{m\in B} c^v_{m',m}\cdot \xi^v_m(j) \times_p h_j && (b)\\
&= \sum_{m\in B} c^v_{m',m} \sum_{j=1}^{K_v} \xi^v_m(j) \times_p h_j\\
&= \sum_{m\in B} c^v_{m',m}\cdot \partial_{I,m}\left(\sum_{j=1}^{K_v} \xi^v_m(j)\cdot m \times_p h_j\right)\\
&= \sum_{m\in B} c^v_{m',m}\cdot \left(\partial_{I,m}\left(\sum_{j=1}^{K_v} \xi^v_m(j)\cdot m \times_p h_j\right) + \partial_{I,m}\left(\sum_{\tilde{m}\neq m} \xi^v_{\tilde{m}}(j)\cdot \tilde{m} \times_p h_j\right)\right)\\
&= \sum_{m\in B} c^v_{m',m}\cdot \partial_{I,m}\left(\sum_{j=1}^{K_v} \xi^v_{m'}(j)\cdot m \times_p h_j + \sum_{\tilde{m}\neq m} \xi^v_{\tilde{m}}(j)\cdot \tilde{m} \times_p h_j\right)\\
&= \sum_{m\in B} c^v_{m',m}\cdot \partial_{I,m}(P)
\end{aligned}$$

The identity (a) holds due to Lemma 47. The identity (b) follows due to our assumption in the statement of the claim. The other identities follow due to the definition and/or by the linearity of $\partial_{I,m}$. □

□

Proof of Claim 43. We know that P_i has a UPT circuit C_i with shape T_i. Say we fix a monomial m and an interval $I = [i_1, i_2]$ such that $\deg(m) = |I|$. Let T_i' be the tree obtained from T_i by deleting all nodes u such that $I_u \subseteq I$. We claim that $\partial_{I,m}(P_i)$ is computed by a UPT circuit of size at most s and shape T_i'.

Consider any leaf node $w \in V(T_i)$ such that $\text{pos}(w) = i_1 + \ell - 1 \in I$. We consider each $(w,\times)$ gate Φ of C (note that these are input gates) and replace the gate by 0 if the variable x labelling Φ is the ℓth variable in m and 0 otherwise.

This gives us a non-commutative arithmetic circuit where some leaves are labelled by constants. However, these constants are easily eliminated inductively as follows. For any $\times$ gate Φ' which has a child labelled by a constant α, we can remove the child and multiply the label of each wire leaving Φ' by α; for any $+$ gate Φ' which has a child labelled by a constant, it must be the case that *all* its children are labelled by constants (this follows from the UPT restriction) and hence the $+$ gate can now be labelled by a constant as well. Continuing this way, all the gates with constant labels are eliminated.

It can be checked that the circuit thus obtained is a UPT circuit of size at most s and shape T_i' computing $\partial_{I,m}(P_i)$. Returning to the statement of the claim, we have therefore shown that each of $\partial_{I_v,\tilde{m}}(P_i)$ and $\partial_{I_v,m}(P_i)$ can be computed by a UPT circuit of size s and shape T_i'.

Therefore, we can compute $\sum_{m\in B_v} c_{\tilde{m},m}\cdot\partial_{I_v,m}(P_i)-\partial_{I_v,\tilde{m}}(P_i)$ by a linear combination of the $O(s)$ UPT circuits computing $\partial_{I_v,m}(P)$ for $m\in B_v\cup\{\tilde{m}\}$. Overall, this gives a UPT circuit of size $O(s^2)$. Since the above proof is constructive, we can actually find this circuit in time $\mathrm{poly}(s)$. □

Proof of Claim 44. We will prove this claim by induction on $|I_v|$.

The base case is $|I_v|=1$ which follows directly from Equation 9.

Suppose $|I_v|=t>1$. Then v is an internal node in T. Let u,w be its two children. This implies that $I_v=I_u\cup I_w$. Let $|I_u|=t_1,|I_w|=t_2$. Note that $t_1,t_2<t$.

We wish to prove that any $m'\in\mathscr{M}_{|I_v|}(X)$, $\partial_{I_v,m'}(P)=\sum_{m\in B_v}c^v_{m',m}\cdot\partial_{I_v,m}(P)$ and $\partial_{I_v,m'}(Q)=\sum_{m\in B_v}c^v_{m',m}\cdot\partial_{I_v,m}(Q)$ for a suitable choice of $c^v_{m',m}$. Note that this already follows for $m'\in B_u\cdot B_w$ from (10). So we assume that $m'\notin B_u\cdot B_w$.

Let $m'=m'_1\cdot m'_2$, where $\deg(m'_1)=t_1$ and $\deg(m'_2)=t_2$. Let R be either of P or Q.

$$
\begin{aligned}
\partial_{I_v,m'}(R) &= \partial_{I_u\cup I_w,m'_1m'_2}(R)\\
&= \partial_{I_u,m'_1}\circ\partial_{I_w,m'_2}(R) && (a)\\
&= \partial_{I_u,m'_1}\circ\left[\sum_{\overline{m}\in B_w} c^w_{m'_2,\overline{m}}\cdot\partial_{I_w,\overline{m}}(R)\right] && (b)\\
&= \sum_{\overline{m}\in B_w} c^w_{m'_2,\overline{m}}\cdot\partial_{I_u,m'_1}\circ\partial_{I_w,\overline{m}}(R)\\
&= \sum_{\overline{m}\in B_w} c^w_{m'_2,\overline{m}}\cdot\partial_{I_w-|I_u|,\overline{m}}\circ\partial_{I_u,m'_1}(R) && (a)\\
&= \sum_{\overline{m}\in B_w} c^w_{m'_2,\overline{m}}\sum_{\underline{m}\in B_u} c^u_{m'_1,\underline{m}}\cdot\partial_{I_w-|I_u|,\overline{m}}\circ\partial_{I_u,\underline{m}}(R) && (b)\\
&= \sum_{\overline{m}\in B_w,\underline{m}\in B_u} c^u_{m'_1,\underline{m}}\cdot c^w_{m'_2,\overline{m}}\cdot\partial_{I_u\cup I_w,\underline{m}\overline{m}}(R) && (a)\\
&= \sum_{\underline{m}\cdot\overline{m}\in B_{u,w}} c^u_{m'_1,\underline{m}}\cdot c^w_{m'_2,\overline{m}}\cdot\partial_{I_v,\underline{m}\overline{m}}(R)\\
&= \sum_{m\in B_v}\left(\sum_{\underline{m},\overline{m}} c^u_{m'_1,\underline{m}}\cdot c^w_{m'_2,\overline{m}}\cdot c^v_{\underline{m}\overline{m},m}\right)\cdot\partial_{I_v,m}(R). && (c)
\end{aligned}
$$

The equalities marked by (a) follow due to Observation 48 given below. The equalities marked (b) follow due to the induction hypothesis. Finally, the equality marked (c) follows due to Equation 10.

The above implies the inductive claim with $c^v_{m',m}$ defined to be $\left(\sum_{\underline{m}\in B_u,\overline{m}\in B_w} c^u_{m'_1,\underline{m}}\cdot c^w_{m'_2,\overline{m}}\cdot c^v_{\underline{m}\overline{m},m}\right)$. Since the choice of $c^v_{m',m}$ is the same for both P and Q, we are done. □

Observation 48. *Let I, J be two contiguous intervals in $[d]$ such that I precedes J, i.e. if $I=[i_1,i_2]$ and $J=[j_1,j_2]$ then $1\le i_1,i_2+1=j_1$ and $j_2\le d$. Then $\partial_{I\cup J,m_1m_2}=\partial_{J-|I|,m_2}\circ\partial_{I,m_1}=\partial_{I,m_1}\circ\partial_{J,m_2}$, where for any two intervals I,J, $J-|I|$ denotes the interval $\{j-|I|\mid j\in J\}\cap[d]$.*

References

[1] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theor. Comput. Sci.*, 209(1-2):47–86, 1998.

[2] Vikraman Arvind, Pushkar S. Joglekar, Partha Mukhopadhyay, and S Raja. Identity testing for +-regular noncommutative arithmetic circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:193, 2016.

[3] Vikraman Arvind, Pushkar S. Joglekar, and Srikanth Srinivasan. Arithmetic circuits and the hadamard product of polynomials. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, pages 25–36, 2009.

[4] Vikraman Arvind, Partha Mukhopadhyay, and S Raja. Randomized polynomial time identity testing for noncommutative circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:89, 2016.

[5] Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. *Computational Complexity*, 19(4):521–558, 2010.

[6] Vikraman Arvind and S. Raja. The complexity of two register and skew arithmetic computation. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:28, 2014.

[7] Steve Chien, Lars Eilstrup Rasmussen, and Alistair Sinclair. Clifford algebras and approximating the permanent. *J. Comput. Syst. Sci.*, 67(2):263–290, 2003.

[8] Steve Chien and Alistair Sinclair. Algebras with polynomial identities and computing the determinant. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 352–361, 2004.

[9] Kenneth Church and Ramesh Patil. Coping with syntactic ambiguity or how to put the block in the box on the table. *Comput. Linguist.*, 8(3-4):139–149, July 1982.

[10] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015.

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

[12] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 323–346, 2015.

[13] Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. Non-commutative circuits and the sum-of-squares problem. *Journal of the American Mathematical Society*, 24(3):871–898, 2011.

[14] Laurent Hyafil. The power of commutativity. In *FOCS*, pages 171–174, 1977.

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

[16] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 61–70, 2014.

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

[18] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 364–373, 2014.

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

[20] Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for non-commutative skew circuits. *Theory of Computing*, 12(1):1–38, 2016.

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

[22] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *STOC*, pages 410–418, 1991.

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

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

[25] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005.

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

[27] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

A Proof of Lemma 11

Lemma 11 (restated)**.** *Let $f \in \mathbb{F}\langle X\rangle$ be homogeneous of degree d and say Π_1, Π_2 are partitions of $[d]$. Then,* $\mathrm{rank}(f, \Pi_2) \leq \mathrm{rank}(f, \Pi_1) \cdot N^{\Delta(\Pi_1, \Pi_2)}$.

Proof. We prove this by induction on $\Delta(\Pi_1, \Pi_2)$.

The base case of the induction is the case that $\Delta(\Pi_1, \Pi_2) = 0$ i.e. $\Pi_1 = \Pi_2$. In this case, the statement is trivial.

Now consider when $\Delta(\Pi_1, \Pi_2) = \Delta \geq 1$. We can find a partition Π such that $\Delta(\Pi_1, \Pi) = \Delta - 1$ and $\Delta(\Pi, \Pi_2) = 1$. By the induction hypothesis, we know that $\mathrm{rank}(f, \Pi) \leq \mathrm{rank}(f, \Pi_1) \cdot N^{\Delta - 1}$ and so it suffices to show that $\mathrm{rank}(f, \Pi_2) \leq \mathrm{rank}(f, \Pi) \cdot N$.

Assume that $\Pi = (Y, Z)$ and $\Pi_2 = (Y_2, Z_2)$. We know that $\Delta(\Pi, \Pi_2) = |Y \Delta Y_2| = 1$. W.l.o.g. assume that $Y = Y_2 \setminus i$ for some $i \in [d]$ (the other case, when $Y = Y_2 \cup \{i\}$ is similar). Note that $Z = Z_2 \cup \{i\}$.

Consider the matrix $M_2 := M[f, \Pi_2]$. We divide M_2 into N blocks as follows. For each $x \in X$, let M_2^x be the submatrix where we only keep the rows corresponding to monomials of degree $|Y_2|$ that contain the variable x in the location "corresponding" to i (i.e. in the jth position where j is the rank of i in Y_2). Clearly, we have $\mathrm{rank}(M_2) \leq \sum_{x \in X} \mathrm{rank}(M_2^x)$.

On the other hand, we also see that each M_2^x is a submatrix of $M := M[f, \Pi]$: namely, the submatrix obtained by only keeping the columns corresponding to those monomials that contain the variable x in the location corresponding to i (as above but w.r.t. Z). Hence, $\mathrm{rank}(M_2^x) \leq \mathrm{rank}(M)$ for each x.

Hence, we see that $\mathrm{rank}(M_2) \leq \sum_{x \in X} \mathrm{rank}(M_2^x) \leq N \cdot \mathrm{rank}(M)$ and this completes the induction. □

B Proof of Proposition 12

Proposition 12 (restated)**.** *1. Let C be a UPT* formula*. Then C is in normal form.*

2. For any UPT circuit C of size s and shape T, there is another UPT circuit C' of size $O(s^2)$ and shape T in normal form computing the same polynomial as C. Further, given C and T, such a C' can be constructed in time $\mathrm{poly}(s)$.

Proof.

- **Proof of 1.** Let C be a UPT *formula* with shape T. We want to prove that C is in *normal form*; this is equivalent to proving that for any multiplication gate $\Phi \in C$ and for any parse formula containing the gate Φ, the gate always takes the same position in T. Let D, D' be any two parse formulas containing Φ. D (resp. D') is a formula, therefore there is a unique path p (resp. p') from the root to Φ in D (resp. D'). The crucial point is the following: as C is also a formula with D and D' as subformulas, these two paths must be equal. By definition, the position of Φ in T with respect to D is characterized by $(\deg(\Phi), \mathrm{pos}(\Phi))$ where $\deg(\Phi)$ is the degree of the monomial computed at the gate Φ in D and $\mathrm{pos}(\Phi)$ equals $1 +$ the sum of the degrees of the monomials computed at the children of the multiplication gates along the path p which are on the left side of the path. As the formula is UPT, the monomials computed in a gate are all of same degree for any parse formulas containing the gate; moreover $p = p'$ so in both cases we consider the same gates in the definition of $(\deg(\Phi), \mathrm{pos}(\Phi))$ in D or D' so that the positions of Φ in T according to D or D' are equal.

- **Proof of 2.** We refer the reader to [19, Lemma 1]. It can be checked that the proof of this result in [20] also yields the algorithmic conclusion.

□

C Proof of Lemma 13

Lemma 13 (restated)**.** *Let C be a normal form UPT circuit (resp. formula) of size s and shape T. Then there is a tree T' and normal form UPT circuit (resp. formula) C' of size $O(s)$ and shape T' such that C' computes the same polynomial as C and every multiplication gate in C' has fan-in at most 2. (This implies that every internal node of T' also has fan-in at most 2.) Further, there is a deterministic polynomial-time algorithm, which when given C computes C'.*

Proof. We give the proof only for UPT circuits, since the transformation is the same in both cases. Let C a UPT circuit as in the statement. For any $\times$-gate Φ with $k > 2$ children $\Psi_0, \ldots, \Psi_{k-1}$, we replace Φ by the following gadget of $2.(k-1) - 1$ gates $\Phi_0, \ldots, \Phi_{2.(k-2)}$. For any $i \in [0, k-3]$, Φ_{2i} is a multiplication gate with inputs Ψ_i and Φ_{2i+1} and Φ_{2i+1} is an addition gate with input $\Phi_{2(i+1)}$. Finally, $\Phi_{2(k-1)}$ is a multiplication gate with inputs Ψ_{k-2} and Ψ_{k-1}. The new circuit is still in alternating layer form, and is

clearly UPT because we apply the same process to any multiplication gate of fan-in strictly greater than 2. For any such gate, we add $k-2$ edges which is less than the fan-in of the previous multiplication gate. Therefore, the number of edges in the final circuit increases by at most two times the number of edges in the original circuit, so that the size of the circuit obtained by this process is $O(s)$.

The shape T' of the new formula is simply the modified version of the shape T obtained by replacing the internal nodes of fan-in $k > 2$ by right combs with k leaves.

This completes the construction of C' from C. The construction can easily be seen to be implementable by a deterministic polynomial-time algorithm. □

D Proof of Lemma 16

Lemma 16 (restated)**.** *Let C be a k-PT circuit (resp. formula) of size s with $\mathscr{T}(C) = \{T_1, \ldots, T_k\}$ computing $f \in \mathbb{F}\langle X \rangle$. Then there exist normal form UPT circuits (resp. formulas) $C_1, \ldots, C_k$ of size at most s^2 each such that $\mathscr{T}(C_i) = \{T_i\}$ and $f = \sum_{i=1}^{k} f_i$, where f_i the polynomial computed by C_i.*

Proof. Let C be as in the statement. We show how to construct k UPT circuits (resp. formulas) $C_1, \ldots, C_k$ of size at most s^2, of shapes $T_1, \ldots, T_k$ respectively, computing $f_1, \ldots, f_k$ respectively such that each f_i is equal to the sum of the monomials computed by all the parse formulas of C of shape T_i. Given this, the polynomial f, which is equal to the sum of all monomials computed by all parse formulas of C, will be equal to $\sum_{i=1}^{k} f_i$ and the lemma will be proved.

Construction of C_i.

- **The gates** of C_i are denoted by pairs of the form (Φ, v). For each gate $\Phi \in C$ and for each node $v \in V(T_i)$ such that $\deg(v) = \deg(\Phi)$, we initially add a gate (Φ, v) to the circuit C_i.
- **Edges:**
  - If $\Phi \in C$ is an addition gate with children $\Psi_1, \ldots, \Psi_t$, then (Φ, v) is an addition gate in C_i with children $(\Psi_1, v), \ldots, (\Psi_t, v)$.
  - If $\Phi \in C$ is a multiplication gate with children $\Psi_1, \ldots, \Psi_t$ (in this order), then (Φ, v) is a multiplication gate with children $(\Psi_1, v_1), \ldots, (\Psi_t, v_t)$, *as long as* the children of v in T_i are exactly $v_1, \ldots, v_t$ (in this order) with $\deg(v_j) = \deg(\Psi_j)$ for each j. Otherwise, we label (Ψ, v) with 0.
  - If Φ is an input gate labelled by $x \in X$ and v a leaf node, then the gate (Φ, v) is also an input gate with the same label.

Notice that the size of C_i is upper bounded by s^2. Further, any parse formula that does not contain any of the nodes labelled 0 has shape T_i. Finally, if C is actually a formula, then $C_1, \ldots, C_k$ are formulas as well.

We prove by induction (on any toplogical orderings of T_i and C) that for any $v \in V(T)$ and Φ in C such that $\deg(\Phi) = \deg(v)$, the gate (Φ, v) in C_i computes the sum of all parse formulas C' of C starting at Φ with the shape $T_i[v]$, where $T_i[v]$ is the subtree of i rooted at v. This will prove that the output gate of C_i computes the sum of the monomials computed by all the parse formulas of C of shape T_i. This is clearly true for the leaves.

Take now any (Φ, v) in C_i. We assume it is a multiplication gate (the other case is similar). Assume that the children of $\Phi \in C$ are $\Psi_1, \ldots, \Psi_t$ and that the children of $v \in T_i$ are $v_1, \ldots, v_r$. If either $r \neq t$ or there is an $a \in [t]$ such that $\deg(v_a) \neq \deg(\Psi_a)$, then there are no parse formulas starting at Φ of shape $T_i[v]$ and hence the gate (Φ, v) which is labelled with 0 computes the correct polynomial. So we now assume that $r = t$ and $\deg(v_a) \neq \deg(\Psi_a)$ for each $a \in [t]$.

Let us denote by S' the set of parse formulas C' of C starting at Φ with a shape $T_i[v]$, and S'_1 (respectively $S'_2, \ldots, S'_t$) the set of parse formulas starting at the gates Ψ_1 (resp. $\Psi_2, \ldots, \Psi_t$) with a shape $T_i[v_1]$ (resp. $T_i[v_2], \ldots, T_i[v_t]$).

The set S' is obtained by taking all possible combinations of parse formulas coming from $S'_1, \ldots, S'_t$. In symbols

$$\sum_{C' \in S'} val(C') = \prod_{j=1}^{t} \sum_{C'' \in S'_j} val(C'')$$

If we denote by $P(\Psi_j, v_j)$ each polynomial computed by a gate (Ψ_j, v_j) in C_i, we get by induction hypothesis that

$$\sum_{C' \in S'} val(S) = \prod_{j=1}^{t} P(\Psi_j, v_j)$$

and hence

$$\sum_{C' \in S'} val(S) = P(\Phi, v)$$

as wanted.

Finally, note that some of the leaves of the circuit are labelled by the constant 0. To eliminate this, we can repeatedly apply the following procedure. If Φ is labelled with 0 and feeds into a $\times$ gate Ψ, then remove Φ and all wires feeding into Ψ, and relabel Ψ with 0. If Φ is labelled with 0 and feeds into a $+$ gate Ψ, then simply remove Φ and if Φ has no inputs left, then relabel it with 0. This process produces a UPT circuit with shape T_i and size at most s^2. Further, since each gate is already associate with a node of T in a natural way, the circuit C_i is already in normal form. □

E Proof of Lemma 24

Lemma 24 (restated)**.** *Let C be any homogeneous arithmetic circuit of size s computing $f \in \mathbb{F}\langle X \rangle$ of degree d. Assume that there is some $d' > d/2$ such that every parse formula C' of C contains a gate computing a (homogeneous) polynomial of degree d'. Let $\Phi_1, \ldots, \Phi_r$ ($r \leq s$) be the set of $\times$ gates computing polynomials of degree d' in C and let $g_1, \ldots, g_r$ be the polynomials they compute (respectively). Then, we have*

$$f = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} g_i \times_j h_{i,j}$$

for some homogeneous polynomials $h_{i,j}$ of degree $d - d'$ ($i \in [r], j \in [0, d-d']$).

Proof. We will first simplify the circuit C so that each gate Φ appears in *some* parse formula of C. If Φ appears in no parse formula of C, then we can remove it from the circuit without changing the polynomial computed by the circuit.

We consider a topological ordering of the gates of the circuit C so that if the gate Φ computes a polynomial of degree at most the degree of the gate Ψ, then Φ appears before Ψ in the ordering. This can be done since C is a homogeneous circuit.

Let $\Psi_1, \ldots, \Psi_p$ ($p \leq s$) be this topological ordering of the gates and let f_k be the polynomial computed at Ψ_k ($k \in [p]$). Let $d_k = \deg(f_k)$. We prove by induction on $k \in [p]$ that if $d_k \geq d'$, then

$$f_k = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} g_i \times_j h_{i,j}^{(k)}. \tag{11}$$

for some homogeneous polynomials $h_{i,j}^{(k)}$ of degree $d - d'$ each. Note that this is vacuously true for k such that $\deg(f_k) < d'$.

If the gate Ψ_k is a $\times$ gate of degree $d_k \geq d'$, then we have the following possibilities:

- $d_k = d'$: In this case $f_k = g_i$ for some $i \in [r]$ and hence we can take $h^{(k)}_{i,0} = 1$ and $h^{(k)}_{i',j'} = 0$ for all other i', j' pairs.

- $d_k > d'$: In this case $f_k = f_{k_1} \cdots f_{k_t}$ for some t and $k_1, \ldots, k_t < k$. We observe that one of $d_{k_1}, \ldots, d_{k_t}$ must be at least d'.

  To see this, assume that $d_{k_a} < d'$ for each $a \in [t]$ and consider any parse formula C' containing Ψ_k (such a formula must exist since otherwise Ψ_k would have been removed in the first simplification step). By our assumption on the circuit, C' must also contain some gate Ψ' computing a polynomial of degree exactly d'. Note that Ψ' does not lie in the subcircuit of C' induced by the gate Ψ_k since all the non-output gates of this subcircuit compute polynomials of degree $< d'$ and the output gate computes a polynomial of degree $> d'$. Also, as $d_k > d'$, Ψ_k does not appear in the subcircuit of C' induced by Ψ'. Consider the parse tree T obtained by unraveling the circuit C'. By the observations above, Ψ_k gives rise to (at least) one node u in T of degree $d_k > d'$ and Ψ' gives rise to a node v in T of degree d'. Thus, the degree of the root is at least $\deg(u) + \deg(v) > 2d' > d$, which is a contradiction since in a homogeneous circuit all parse trees have exactly d leaves.

  So we can assume that $\deg(f_{k_a}) \geq d'$ for some $a \in [t]$. Applying the induction hypothesis to f_{k_a} we have

$$f_{k_a} = \sum_{i=1}^{r} \sum_{j=0}^{d'} g_i \times_j h^{(k_a)}_{i,j}.$$

  for suitable $h^{(k_a)}_{i,j}$ of degree $d_{k_a} - d'$ each. Thus, we have

$$\begin{aligned} f = f_{k_1} \cdots f_{k_t} &= \sum_{i=1}^{r} \sum_{j=0}^{d_{k_a}-d'} f_1 \cdots f_{k_a-1} \cdot (g_i \times_j h^{(k_a)}_{i,j}) \cdot f_{k_a+1} \cdots f_{k_t} \\ &= \sum_{i=1}^{r} \sum_{j=0}^{d_{k_a}-d'} g_i \times_{j+d_{k_1}+\cdots+d_{k_{a-1}}} (f_1 \cdots f_{k_a-1} h^{(k_a)}_{i,j} f_{k_a+1} \cdots f_{k_t}) \end{aligned}$$

  where for the final equality we have used the observation that $(g \times_j h) \times_{j'} h' = g \times_{j+j'} (h \times_{j'} h')$ for any homogeneous polynomials g, h, h' and any relevant j, j'.

  For any $j \in [0, d_{k_a} - d']$, we have $j' := j + d_{k_1} + \cdots + d_{k_{a-1}} \in [d_{k_1} + \cdots + d_{k_{a-1}}, d_k - d']$. Hence, setting $h^{(k)}_{i,j'} = f_1 \cdots f_{k_a-1} h^{(k_a)}_{i,j} f_{k_a+1} \cdots f_{k_t}$ for each $j' \in\in [d_{k_1} + \cdots + d_{k_{a-1}}, d_k - d']$, and 0 for all $j' < d_{k_1} + \cdots + d_{k_{a-1}}$, the above yields (11) in this case.

If the gate Ψ_k is a $+$ gate of degree $d_k \geq d'$, then it is a linear combination of gates $\Psi_{k_1}, \ldots, \Psi_{k_t}$ of degree d_k each. By induction, for each $a \in [t]$, we have

$$f_{k_a} = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} g_i \times_j h^{(k_a)}_{i,j}.$$

Say $f_k = \sum_a \alpha_a f_{k_a}$ where $\alpha_a \in \mathbb{F}$. Then using the fact that $\times_j$ is bilinear, we get

$$f_k = \sum_{a \in [t]} \alpha_a f_{k_a} = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} \sum_a \alpha_a g_i \times_j h^{(k_a)}_{i,j}) = \sum_{i=1}^{r} \sum_{j=0}^{d-d'} g_i \times_j (\sum_a \alpha_a h^{(k_a)}_{i,j}))$$

which is of the form required in (11). This completes the induction. □

F Proof of Lemma 32

Lemma 32. *Let T be a tree with d leaves and depth Δ, such that all internal nodes are of in-degree strictly greater than* 1. *Then there is a path $\pi = (v_\ell, \ldots, v_0)$ in T such that $|\mathrm{off}(\pi) \cup \{v_\ell\}| \geq \Omega(\Delta d^{1/\Delta})$.*

Proof. Let T be a tree as in the statement. We denote by $wt(v)$ the fan-in of the node v. We will prove the following equivalent conclusion: there is a path $\pi = (v_\ell, \ldots, v_0)$ from a leaf to the root such that $1 + \sum_{0 \leq i < \ell} (wt(v_i) - 1) \geq \Omega(\Delta d^{1/\Delta})$.

We consider two distinct cases:

- **Case 1:** $\Delta > \log(d)$. In this case, any path $p = (v_0, \ldots, v_\Delta)$ of depth Δ respects $1 + \sum_{0 \leq i < \Delta} (wt(v_i) - 1) \geq \Delta + 1 = \Omega(\Delta d^{1/\Delta})$.

- **Case 2:** $\Delta \leq \log(d)$. We consider the following greedy procedure to choose the path of internal nodes: starting from the root, repeatedly choose a child such that the number of leaves in the resulting subtree is maximized. Let $p = v_0, \ldots, v_\ell$ be the sequence of nodes thus obtained. Note that $\ell \leq \Delta \leq \log(d)$.

  We prove by induction on the tree T that the number of leaves of the tree is at most the product of the fan-ins of $v_0, \ldots, v_{\ell-1}$ (this fact is true for any tree T and not just trees T such of depth at most $\log d$). If $\ell = 0$, the entire tree consists of just the root: hence the number of leaves is 1 and the (empty) product also evaluates to 1. Assume now that the root v_0 has k children corresponding to subtrees $T_1, T_2, \ldots T_k$. We assume the number of leaves in the subtree T_i is t_i. Assume the greedy algorithm chooses v_1 corresponding to the subtree rooted in T_i (thus, we must have $t_i \geq t_j$ for any $j \in [k]$). By the induction hypothesis, $\prod_{i=1}^{\ell} wt(v_i) \geq t_i$. Therefore

  $$\prod_{i=0}^{\ell-1} wt(v_i) \geq k \cdot t_i \geq \sum_{i=1}^{k} t_j = d$$

  which concludes the induction.

  By the inequality of arithmetic and geometric means:

  $$\frac{\sum_{0 \leq i < l} wt(v)}{\ell} \geq \Big(\prod_{0 \leq i < \ell} wt(v)\Big)^{1/\ell} \geq d^{1/\ell}.$$

  So, we have

  $$\sum_{0 \leq i < \ell} (wt(v) - 1) \geq \ell(d^{1/\ell} - 1).$$

  Notice that the right part of this inequality is a decreasing function of ℓ in the regime $\ell \leq \log(d)$, so that:

  $$\sum_{0 \leq i < \ell} (wt(v) - 1) \geq \Delta(d^{1/\Delta} - 1) = \Delta d^{1/\Delta}(1 - \frac{1}{d^{1/\Delta}}) \geq \Delta d^{1/\Delta}(1 - \frac{1}{e^1}) = \Omega(\Delta d^{1/\Delta}).$$

□

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il