Efficient Polynomial Identity Testing Over Nonassociative Algebras

C. Ramya[*]   Partha Mukhopadhyay[†]   Pratik Shastri[‡]

Abstract

We design the first efficient polynomial identity testing algorithms over the *nonassociative* polynomial algebra. In particular, multiplication among the formal variables is commutative but it is not associative. This complements the strong lower bound results obtained over this algebra by Hrubeš, Yehudayoff, and Wigderson [19] and Fijalkow, Lagarde, Ohlmann, and Serre [29] from the identity testing perspective. Our main results are the following:

- We construct nonassociative algebras (both commutative and noncommutative) which have no low degree identities. As a result, we obtain the first Amitsur-Levitzki type theorems [1] over nonassociative polynomial algebras. As a direct consequence, we obtain randomized polynomial-time black-box PIT algorithms for nonassociative polynomials which allow evaluation over such algebras.

- On the derandomization side, we give a deterministic polynomial-time identity testing algorithm for nonassociative polynomials given by arithmetic circuits in the white-box setting. Previously, such an algorithm was known with the additional restriction of noncommutativity [25].

- In the black-box setting, we construct a hitting set of quasipolynomial-size for nonassociative polynomials computed by arithmetic circuits of small depth. Understanding the black-box complexity of identity testing, even in the randomized setting, was open prior to our work.

---

[*]The Institute of Mathematical Sciences (a CI of Homi Bhabha National Institute), Chennai, India, email: ramyac@imsc.res.in.

[†]Chennai Mathematical Institute, Chennai, India email: partham@cmi.ac.in.

[‡]The Institute of Mathematical Sciences (a CI of Homi Bhabha National Institute), Chennai, India, email: pratiks@imsc.res.in.

1 Introduction

The goal of algebraic complexity is to study the complexity of computing multivariate polynomials using basic arithmetic operations, such as addition and multiplication. Arithmetic circuits and formulas are among the well-studied models in this area. In particular, arithmetic circuits are directed acyclic graphs whose leaves are labeled by variables or constants, and whose internal nodes (called gates) are labeled with either $+$ or $\times$. Formulas are circuits whose underlying graph is a tree. Each gate in a circuit computes a polynomial in the natural way and the output of a circuit is said to be the polynomial computed at a distinguished output gate.

There are two central problems studied in algebraic complexity: one is the question of proving size lower bounds for circuits computing explicit polynomials, and the other is the question of derandomizing polynomial identity testing (PIT for short). The PIT problem comes in two variants. First is the black-box setting, where we are given evaluation access to a circuit (given as a black box), and we must decide whether the polynomial it computes is identically zero. The second, seemingly easier is the white-box setting, where we are explicitly given the circuit as a graph to determine whether it computes the zero polynomial.

Baur and Strassen [7] proved that any circuit computing $\sum_{i=0}^{n} x_i^n$ requires size $\Omega(n \log n)$. This is the strongest known lower bound for arithmetic circuits. On the other hand, the PIT problem admits a randomized polynomial-time black-box algorithm over the usual polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$, thanks to the Polynomial Identity Lemma [4–6] (see Lemma 6 for the exact statement). Here, $\mathbb{F}$ can be any field of sufficient size, and the variables $x_1, \ldots, x_n$ commute under multiplication. For the black-box case, the derandomization problem is essentially equivalent to the efficient construction of small *hitting sets*. Despite intense efforts over many years, proving strong lower bounds for circuits and derandomizing PIT for circuits have both remained elusive goals. For more on these problems, the reader is referred to the excellent surveys by Shpilka and Yehudayoff [20] and Saxena [22]. In general, the problems of proving lower bounds and derandomizing PIT are closely related and, in fact, nearly equivalent due to an influential result of Impagliazzo and Kabanets [15]. For example, very recently, a subexponential-size hitting set for PIT of low-depth arithmetic circuits was obtained via a breakthrough lower bound result by Limaye, Srinivasan, and Tavenas [30].

The limitation in our understanding of PIT and lower bounds stems from the difficulty of analyzing how a monomial gets canceled at an intermediate step in an arithmetic circuit computation. In the setting of the usual polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$, multiplication is commutative and associative. In other words, for all x_i, x_j, x_k we have $x_i x_j = x_j x_i$ and $(x_i(x_j x_k)) = ((x_i x_j) x_k)$. A central line of investigation studies arithmetic circuits by restricting the relations satisfied by the multiplication rule. The hope is that, in the absence of such relations, we can better understand *cancellations* of monomials and gain insight into general circuits. For example, we can drop commutativity while preserving associativity.

If we drop commutativity, we obtain the noncommutative polynomial ring $\mathbb{F}\langle x_1, \ldots, x_n \rangle$ (which remains associative), and noncommutative circuits compute polynomials in the ring $\mathbb{F}\langle x_1, \ldots, x_n \rangle$. In his pioneering work [11], Nisan proved that the noncommutative Permanent and Determinant polynomials require exponential-size noncommutative algebraic branching programs (ABPs for short). ABPs are a subclass of circuits. Up to polynomial blowup, they simulate formulas and are simulated by circuits. In the noncommutative setting, exponential

separations are known between ABPs and circuits [11]. On the PIT side, Raz and Shpilka [16] developed a white-box deterministic polynomial-time algorithm for polynomials computed by noncommutative ABPs. In fact, a quasipolynomial time black-box PIT algorithm has also been designed for the same problem by Forbes and Shpilka [21].

However, if we look at *general* noncommutative circuits (instead of ABPs), again we see that progress on the questions of lower bounds and derandomizing PIT has remained elusive. In fact, for general circuits, the best lower bound and PIT results over $\mathbb{F}\langle x_1, \ldots, x_n \rangle$ match those over $\mathbb{F}[x_1 \ldots, x_n]$.

Henceforth, we use X to denote the set $\{x_1, \ldots, x_n\}$. $\mathbb{F}_{A,C}[X]$ and $\mathbb{F}_{A,\bar{C}}[X]$ stand for the rings $\mathbb{F}[X]$ and $\mathbb{F}\langle X \rangle$, respectively. Instead of dropping commutativity, we may choose to drop associativity. This leads us to the algebra $\mathbb{F}_{\bar{A},C}[X]$, the polynomial algebra where multiplication is commutative but nonassociative[1] [2]. If we drop *both* commutativity and associativity, we obtain the polynomial algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$. On the lower bounds side, very impressive progress has been made in the algebra $\mathbb{F}_{\bar{A},C}[X]$. Hrubeš, Yehudayoff, and Wigderson [19] proved the first exponential-size lower bounds for circuits computing explicit polynomials in $\mathbb{F}_{\bar{A},C}[X]$. Subsequently, Fijalkow, Lagarde, Ohlmann, and Serre [29] strengthened this result by providing an *exact* characterization of the size of a minimal circuit computing a polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ in terms of the rank of a certain matrix of coefficients.

On the other hand, in the context of PIT, it is imperative to note that no efficient algorithm is known over the algebra $\mathbb{F}_{\bar{A},C}[X]$, not even a randomized white-box algorithm. This is surprising given the intimate connections between lower bounds and PIT in various settings. Over the algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$, Arvind et al. designed a deterministic *white-box* polynomial-time algorithm for PIT of arithmetic circuits [25]. Furthermore, as mentioned by the authors in [25], a black-box algorithm is not known even over $\mathbb{F}_{\bar{A},\bar{C}}[X]$. Notably, over the algebra $\mathbb{F}\langle X \rangle$, such a randomized PIT algorithm is known due to the Amitsur-Levitzki Theorem [1, 18] (see Theorem 7 for a formal statement).

We note that nonassociative computations are fundamental even beyond algebraic complexity. For example, the composition of operations in computer programs is typically nonassociative. As a concrete algorithmic example, in a seminal work, Valiant designed a sub-cubic algorithm for recognizing Context Free Languages (CFL) [3]. In particular, Valiant developed an algorithm to compute the transitive closure of upper triangular matrices whose entries are elements of a nonassociative monoid [3]. In algebraic complexity, lower bounds for nonassociative circuits have been used prove lower bounds for related *associative* models of computation [29].

The main technical contributions of this paper are as follows. Over the polynomial algebra $\mathbb{F}_{\bar{A},C}[X]$, we design a white-box deterministic polynomial-time algorithm for the PIT of arithmetic circuits. In the black-box setting, we develop the first randomized polynomial-time PIT algorithm for nonassociative arithmetic circuits (over both $\mathbb{F}_{\bar{A},C}[X]$ and $\mathbb{F}_{\bar{A},\bar{C}}[X]$). This is achieved by proving analogues of the Amitsur-Levitzki theorem over the nonassociative polynomial algebras $\mathbb{F}_{\bar{A},\bar{C}}[X]$ and $\mathbb{F}_{\bar{A},C}[X]$. Moreover, for the classes of circuits over the algebras $\mathbb{F}_{\bar{A},C}[X]$ and $\mathbb{F}_{\bar{A},\bar{C}}[X]$ with polylogarithmic depth, we construct quasipolynomial-size hitting sets over nonassociative algebras of small dimension. To the best of our knowl-

---

[1] Formally, an *algebra* $\mathbb{A}$ over a field $\mathbb{F}$ is a vector space equipped with a bilinear product.
[2] In $\mathbb{F}_{\bar{A},C}[X]$, $(x_i(x_j x_k)) \neq ((x_i x_j) x_k)$ even in the case when $x_i = x_j = x_k$.

edge, this is the first black-box derandomization result for a well-studied circuit class over a nonassociative polynomial algebra. We elaborate on our results and techniques in the next section.

1.1 Our Results

In this paper, we complement the strong lower bounds results obtained over the algebra $\mathbb{F}_{\bar{A},C}[X]$ ([19, 29]) by designing efficient PIT algorithms. Our results work over all fields of sufficiently large size.

1.1.1 Black-box randomized nonassociative PIT (Section 3)

Our first result is a black-box randomized polynomial-time identity testing algorithm for polynomials in $\mathbb{F}_{\bar{A},C}[X]$. Of course, to capture nonassociativity, the natural idea is to evaluate the given polynomial over nonassociative algebras.

In mathematics, nonassociative algebras are very well-studied. For example, one can see the classic work of Albert [12]. In particular, there are various specific algebras of interest which are nonassociative but commutative. The most important such algebras are, perhaps, the Jordan Algebras. Unfortunately, every Jordan Algebra $\mathbb{J}$ satisfies the *Jordan Identity*: $\forall a, b \in \mathbb{J}$, we have $(ab)(aa) - (a(b(aa))) = 0$. This means that performing PIT using Jordan Algebras is not possible. To see why, suppose we are given a circuit computing a non-zero polynomial in the ideal of $\mathbb{F}_{\bar{A},C}[X]$ generated by $\{(x_i x_j)(x_i x_i) - (x_i(x_j(x_i x_i))) \mid x_i, x_j \in X\}$. On any input from $\mathbb{J}$, the circuit will evaluate to 0, whereas the circuit computes a non-zero polynomial. Even over $\mathbb{F}_{\bar{A},\bar{C}}[X]$, we have a similar difficulty. For example, Matrix Lie algebras are well studied algebras that are nonassociative and noncommutative, but they satisfy the Jacobi Identity [10].

Thus, in order to obtain a fast black-box algorithm, we need to construct a suitable nonassociative algebra that does not satisfy low (in terms of its dimension) degree identities. We should remark that over the noncommutative polynomial ring $\mathbb{F}\langle X \rangle$ (equivalently, $\mathbb{F}_{A,\bar{C}}[X]$), the Amitsur-Levitzki theorem allows us to do black-box randomized PIT for polynomials of degree $\leqslant d$, using random matrices of dimension $(\lceil d/2 \rceil + 1) \times (\lceil d/2 + 1 \rceil)$ [1, 18]. In our setting of polynomials in $\mathbb{F}_{\bar{A},C}[X]$, we construct a unital nonassociative, commutative algebra $\mathbb{C}_d$ of dimension $d(d+1)^2 + 1$ which does not satisfies *any* identity of degree $\leqslant d$.

Lemma 1. *Let $f \in \mathbb{F}_{\bar{A},C}[X]$ be a non-zero polynomial of total degree $\leqslant d$. Then f is not a polynomial identity (PI) for $\mathbb{C}_d$.*

To the best of our knowledge, this is the first Amitsur-Levitzki type theorem in the nonassociative setting. As an immediate consequence, we obtain a nonassociative analogue of the Polynomial Identity Lemma.

Theorem 2. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| > d$, and $S \subset \mathbb{F}$. Let $f \in \mathbb{F}_{\bar{A},C}[X]$ be a non-zero polynomial of degree $\leqslant d$ given as a black-box with query access to evaluations of f on elements of $\mathbb{C}_d$. Let $S \subseteq \mathbb{F}$ with $|S| > d$. Sample $b_1, \ldots, b_n \in \mathbb{C}_d$ as follows: Pick each of the $d(d+1)^2 + 1$ entries of each of the b_i's uniformly and independently from S. Then*

$$\Pr_{b_1, \ldots, b_n \in \mathbb{C}_d} [f(b_1, \ldots, b_n) = 0] \leqslant d/|S|.$$

4

We construct $\mathbb{C}_d$ in two stages. First, we construct a *noncommutative*, nonassociative algebra $\mathbb{A}_d$ (see Section 3.1 for the precise definition) and show that $\mathbb{A}_d$ does not satisfy identities of degree $\leqslant d$. Note that this also gives us a noncommutative, nonassociative analogue of Theorem 2 (see Theorem 12 for exact statement). We prove this by showing that monomials in $\mathbb{F}_{\bar{A},\bar{C}}[X]$ can be *isolated* using substitutions from $(\mathbb{A}_d)^n$. Each monomial $m \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ can be viewed as a rooted, ordered binary tree whose leaves are labeled by variables (see Figure 2 for examples). To each occurrence of a variable x in m, we may associate a level which indicates the depth at which it appears in the monomial m (when viewed as a tree). Noncommutativity of $\mathbb{F}_{\bar{A},\bar{C}}[X]$ also induces a *left to right* order in which the variables appear in m. We show that given the left to right order and the corresponding sequence of levels, we can fully reconstruct m (Lemma 8). Using this lemma and a *three dimensional version* of the set-multilinearization procedure introduced by Forbes and Shpilka [21] in the context of PIT for noncommutative ABPs, we show that $\mathbb{A}_d$ does not satisfy identities of degree $\leqslant d$. The third dimension "keeps track" of the level at which each leaf appears in a monomial. After this, we define the commutative algebra $\mathbb{C}_d$ as follows: the $\mathbb{C}_d$ product of x, y is the *anticommutator*[3] of x, y with respect to the $\mathbb{A}_d$ product. In $\mathbb{C}_d$, there is no unique left to right order of the variables that we can associate with a monomial. But there is a *set* of orders that we can associate with each monomial. This set, together with the corresponding sequence of levels, determines the monomial uniquely. Using this, we show that $\mathbb{C}_d$ also does not satisfy identities of degree $\leqslant d$.

1.1.2 White-box deterministic PIT over $\mathbb{F}_{\bar{A},C}[X]$ (Section 4.1)

Next, we consider the white-box identity testing problem over $\mathbb{F}_{\bar{A},C}[X]$. Raz and Shpilka [16] give a white-box linear algebraic algorithm for identity testing of noncommutative algebraic branching programs. Subsequently, their algorithm has been adapted to obtain PIT algorithms in various settings, for example, for Read-Once Algebraic Branching Programs (ROABPs) [21], noncommutative Unique Parse Tree Circuits [27] and circuits over $\mathbb{F}_{\bar{A},\bar{C}}[X]$ [25]. In this work, we show that an adaption of the Raz-Shpilka algorithm can be used to do PIT for circuits over $\mathbb{F}_{\bar{A},C}[X]$:

Theorem 3. *Let Ψ be a nonassociative arithmetic circuit of size s computing an n variate, degree $\leqslant d$ polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$. Given Ψ as input, we can check whether $f \equiv 0$ deterministically in time $\mathsf{poly}(s, n, d)$.*

The main difference in the application of the Raz-Shpilka algorithm in our setting is that in all previous works (that we are aware of), if a monomial m is generated at a product gate $g = g_1 \times g_2$ in the circuit, then there is a *unique way* it could have been generated: there exist monomials m_1, m_2 such that $\mathsf{coeff}_m(g) = \mathsf{coeff}_{m_1}(g_1) \times \mathsf{coeff}_{m_2}(g_2)$. On the other hand since we are working in the commutative setting of $\mathbb{F}_{\bar{A},C}[X]$, there are two ways of generating m at g. Either m_1 could be contributed by g_1 and m_2 by g_2, or m_2 could be contributed by g_1 and m_1 by g_2. We show (somewhat surprisingly) that the Raz-Shpilka algorithm, suitably modified, works even in this setting.

---

[3]The anticommutator of x, y with respect to a product operation $\cdot$ is defined as $x \cdot y + y \cdot x$.

1.1.3 Black-box deterministic nonassociative PIT (Section 4.2)

We consider next the question of derandomizing black-box PIT for circuits over $\mathbb{F}_{\bar{A},C}[X]$. Towards this, we provide a *hitting set* (consisting of elements of $(\mathbb{C}_d)^n$) for such circuits. A hitting set H for a class $\mathcal{C}$ of circuits is a set of points such that for any non-zero circuit $\Psi \in \mathcal{C}$, there exists an $a \in H$ such that Ψ evaluated at a is not 0.

Theorem 4. *There exists a set $H_{n,s,d,\Delta} \subseteq (\mathbb{C}_d)^n$ of size $(nsd)^{O(\Delta)}$ of points in $(\mathbb{C}_d)^n$ such that for every nonassociative, commutative circuit Ψ of size $\leqslant s$ and product depth $\leqslant \Delta$ computing a non-zero polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ of degree $\leqslant d$, there is a point in $H_{n,s,d,\Delta}$ at which f is non-zero. Furthermore, we can compute $H_{n,s,d,\Delta}$ deterministically in time $(nsd)^{O(\Delta)}$.*

Recall that the product depth of a circuit is the maximum number of product gates encountered on any leaf to root path in the circuit. Theorem 4 gives a non-trivial hitting set when the product depth $\Delta = o(d)$. In particular, when Δ is polylogarithmic, we obtain a quasipolynomial size hitting set.

The result of Kabanets and Impagliazzo [15] shows that explicit lower bound results can give subexponential-time black-box PIT algorithms over the usual commutative polynomial ring $\mathbb{F}[X]$. The main ingredients in their proof are the combinatorial design of Nisan and Wigderson [13] and the factorization algorithm of Kaltofen [9]. Although we have strong and explicit lower bounds over the algebra $\mathbb{F}_{\bar{A},C}[X]$, it is unclear how to use them for PIT algorithms. Note that such a connection is not known even over $\mathbb{F}\langle X \rangle$.

We prove Theorem 4 in two stages. In the first stage, we reduce PIT for circuits over $\mathbb{F}_{\bar{A},C}[X]$ to PIT for *unambiguous* circuits over $\mathbb{F}[Z]$ (where Z is a fresh set of variables and $\mathbb{F}[Z]$ is the usual polynomial ring) via a set-multilinearization argument. We say that a circuit Ψ over is $\mathbb{F}[Z]$ is *unambiguous* if for any monomial $m \in \mathbb{F}[Z]$, there exists a reduced parse tree[45] T_m such that any reduced parse tree computing m at any gate of Ψ is isomorphic to T_m as a labeled, rooted binary tree. These are the natural associative analogues of nonassociative circuits. We also observe that a similar reduction works over $\mathbb{F}_{\bar{A},\bar{C}}[X]$, which gives us an analogue of Theorem 4 over the algebra $\mathbb{A}_d$.

In the second stage, we suitably adapt the machinery of *basis isolating weight assignments* developed by Agrawal et al. [23] to construct hitting sets for unambiguous circuits.

Theorem 5. *There exists a set $H_{s,n,d,\Delta} \subseteq \mathbb{F}^n$ such that for any unambiguous circuit Ψ of size s and product depth Δ computing a non-zero polynomial $f \in \mathbb{F}[z_1,\ldots,z_n]$ of degree $\leqslant d$, f is non-zero on some point of $H_{s,n,d,\Delta}$. Furthermore, $|H_{s,n,d,\Delta}| = (nds)^{O(\Delta)}$ and $H_{s,n,d,\Delta}$ can be constructed in time $(nds)^{O(\Delta)}$.*

Informally, given a polynomial f with coefficients coming from a vector space, a *basis isolating weight assignment* for f is a function from the underlying set of variables to $\mathbb{N}$ that isolates a minimum weight basis (among the coefficients) for the space spanned by the coefficients of f. Basis isolating weight assignments were used in [23] to construct quasipolynomial size hitting-sets for ROABPs (these are commutative analogues of noncommutative ABPs).

---

[4] See 5 for a definition.

[5] The term *unambiguous circuit* has been used in different contexts in earlier works [24, 28].

Subsequently, they were also used to construct hitting sets for set-multilinear Unique Parse Tree circuits (UPT circuits for short) [26]. UPT set-multilinear circuits generalize ROABPs. In a UPT set-multilinear circuit, every parse tree (see 4 for the definition) at the output has exactly the same shape as a rooted, ordered binary tree. In particular this implies that there is a universal parse tree shape at the root that induces a unique parse tree for each monomial. In unambiguous circuits, this is no longer true. In particular, there is a unique reduced parse tree *for each monomial*, but two different monomials could have different parse tree shapes. Also, note that unambiguous circuits need not be set-multilinear. On the other hand, we require that for any monomial m, there is a unique parse tree computing it independent of the gate at which m is being computed.

Suppose we have an unambiguous circuit of product depth Δ. We construct a basis isolating weight assignment w for it in multiple stages (as in [23]). At each stage we handle monomials of increasing depths. The proof that w is a basis isolating weight assignment involves isolation of a set M_i of monomials for each depth $i \in [\Delta]$ such that the coefficient of every other monomial of depth i is spanned by coefficients of M_i. The construction of M_i and the proof that its coefficients indeed span coefficients of other monomials is the main technical content of the proof and uses the fact that the circuit is unambiguous. Combining these M_i's we get the isolated set M of monomials. Identity testing follows from the construction of a basis isolating weight assignment.

The result of Valiant, Skyum, Berkowitz and Rackoff [8] shows that arithmetic circuits can be depth reduced to depth polylogarithmic in the size and degree of the original circuit while incurring only a polynomial blowup in size. Unfortunately, we do not know if such a depth reduction is possible while preserving unambiguousness.

Organization

The paper is organized as follows. In Section 2, we provide the necessary background. Section 3 contains the randomized polynomial-time PIT algorithms over nonassociative algebras. The deterministic PIT algorithms (white-box and black-box) are presented in Section 4. We state a few questions for further research in Section 5.

2   Preliminaries

Definition 1 (Algebra over a field)**.** Let $\mathbb{F}$ be a field. An *algebra* $\mathbb{A}$ over $\mathbb{F}$ is an $\mathbb{F}$-vector space together with a product operation on the elements of the vector space that is *bilinear*. The *dimension* of $\mathbb{A}$ is defined to be the dimension of the underlying vector space. In particular, if the underlying vector space is finite (say n) dimensional and identified with $\mathbb{F}^n$ after choice of a basis, an algebra $\mathbb{A}$ is uniquely defined by n matrices $L_1, \ldots, L_n \in \mathbb{F}^{n \times n}$ as follows: for $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$, their $\mathbb{A}$-product is precisely

$$\mathbf{x} \cdot \mathbf{y} = (x_1, \ldots, x_n) \cdot (y_1, \ldots, y_n) = \left(\mathbf{x}^T L_1 \mathbf{y}, \ldots, \mathbf{x}^T L_n \mathbf{y} \right).$$

An algebra $\mathbb{A}$ is called *unital* if it contains a multiplicative identity i such that $\forall x \in \mathbb{A}, x \cdot i = i \cdot x = x$. We define the following four different polynomial algebras, depending on the relations the variables satisfy:

(1) $\mathbb{F}_{A,C}[X]$: This is the polynomial ring $\mathbb{F}[X]$. The product operation is both commutative and associative.

(2) $\mathbb{F}_{A,\bar{C}}[X]$ is the noncommutative polynomial ring $\mathbb{F}\langle X\rangle$. The product operation is noncommutative but associative.

(3) $\mathbb{F}_{\bar{A},C}[X]$ is the $\mathbb{F}$-vector space generated by commutative, nonassociative monomials in the variables X. This vector space becomes an $\mathbb{F}$-algebra with the commutative, nonassociative product of monomials extended to all of $\mathbb{F}_{\bar{A},C}[X]$ by bilinearity.

(4) $\mathbb{F}_{\bar{A},\bar{C}}[X]$ is the $\mathbb{F}$-vector space generated by noncommutative, nonassociative monomials in the variables X. This vector space becomes an $\mathbb{F}$-algebra with the noncommutative, nonassociative product of monomials extended to all elements of $\mathbb{F}_{\bar{A},\bar{C}}[X]$ by bilinearity.

Definition 2 (Polynomial Identities). A *Polynomial Identity* (PI for short) for an algebra $\mathbb{A}$ is a polynomial $f(x_1,\ldots,x_n)$ in a set of variables $\{x_1,\ldots,x_n\}$ such that for all $A_1,\ldots,A_n \in \mathbb{A}$, $f(A_1,\ldots,A_n) = 0$ where the multiplication is according to the product operation in $\mathbb{A}$. An algebra that satisfies nontrivial identities is called a *PI-algebra*.

The study of polynomial identities is a classical and very rich subject in mathematics. For a comprehensive details, see [17].

Definition 3 (Arithmetic Circuit). An *Arithmetic Circuit* Ψ over a field $\mathbb{F}$ is a directed acyclic graph whose leaves (called input gates) are labeled by either variables (say $X = \{x_1,\ldots,x_n\}$) or field elements and whose internal vertices (called gates) are labeled by either a sum $(+)$ or a product $(\times)$. In our case the product operation will often be *nonassociative*, and we will assume that the fan-in of each product gate is 2. If in addition we are working in $\mathbb{F}_{\bar{A},\bar{C}}[X]$, the product will also be noncommutative: each product gate will have designated left and and right child. Each gate in a circuit naturally computes a polynomial. The circuit Ψ has a designated output gate and Ψ is said to compute the polynomial computed at the output gate. The *size* of a circuit is the number of gates in it and the *depth* of a circuit is the length of the longest leaf-to-root path.

Next we define the concept of a *parse tree*, that depicts the generation of a particular monomial in the circuit.

Definition 4 (Parse trees). Let $X = \{x_1,\ldots,x_n\}$ be a set of variables, $\mathbb{F}$ be a field and Ψ be an arithmetic circuit computing a polynomial $f \in \mathbb{F}[X]$. The set of *parse trees* for Ψ will be defined by induction on the size of Ψ as follows:

- If Ψ is just a leaf labeled by either a variable or a constant, then it has only one parse tree, itself.

- If the root g of Ψ is a sum gate with subcircuits Ψ_1 and Ψ_2 as children, the set of parse trees for Ψ is the set of all trees obtained by taking the root g, and attaching to it a parse tree of either Ψ_1 or Ψ_2.

- If the root g of Ψ is a product gate with subcircuits Ψ_1 and Ψ_2, we define the set of parse trees for Ψ to be the set of all trees T obtained by taking a parse tree T_1 for Ψ_1, a parse tree T_2 for a disjoint copy of Ψ_2 and making T_1, T_2 the children of g.

(a) A circuit Ψ (b) A parse tree T for Ψ (c) Reduced parse tree T'
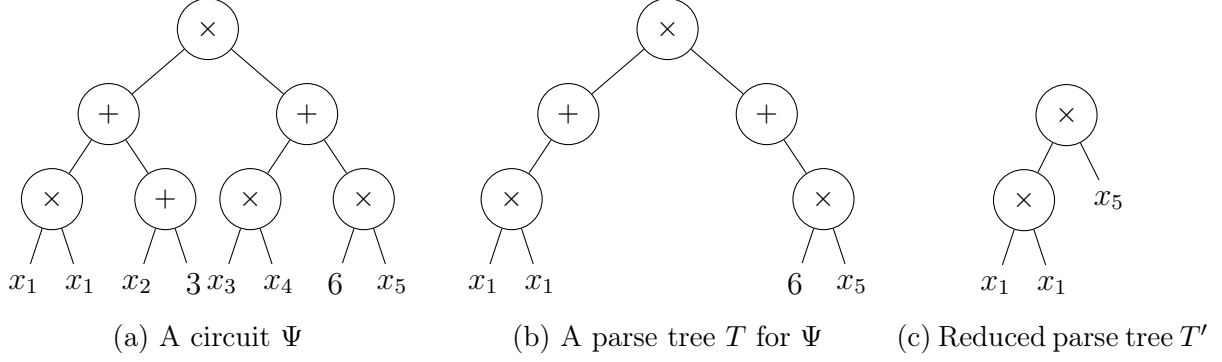
Note that each parse tree T for Ψ computes a monomial (with coefficient).

Definition 5 (Reduced Parse Trees)**.** From each parse tree T of a circuit Ψ, we may obtain a *reduced parse tree* T' by short-circuiting the sum gates, removing leaves labeled by constants and restructuring the tree in the natural way. T' is a full binary tree all of whose leaves are labeled by a variable and all of whose gates are product gates. T' captures the multiplicative structure of T. The *set of reduced parse trees* for Ψ is defined as $\{T' \mid T \text{ is a parse tree for } \Psi\}$. See Figure 1c for an illustrative example.

We also recall the following standard result over $\mathbb{F}_{A,C}[X]$:

Lemma 6 (Polynomial Identity Lemma [4–6])**.** *Suppose $f(x) \in \mathbb{F}[X]$ is an n-variate polynomial of degree d, and let $S \subseteq F$ be a finite set of size strictly larger than d. Then $f(\bar{a}) \neq 0$ for at least $\left(1 - \frac{d}{|S|}\right)$ fraction of $\bar{a}$'s in S^n.*

Over the algebra $\mathbb{F}_{A,\bar{C}}[X]$, the following is a well-known result of Amitsur and Levitzki [1].

Theorem 7 (Amitsur-Levitzki Theorem [1])**.** *Over any field $\mathbb{F}$, the matrix algebra $\mathbb{F}^{k \times k}$ satisfies no PI of (total) degree less than $2k$, and satisfies exactly one (up to constant factor) PI of degree $2k$.*

2.1 A structural lemma about nonassociative monomials

2.1.1 Over the algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$

Let $\mathbb{F}$ be a field and $X = \{x_1, \ldots, x_n\}$ be a set of variables. There is a natural correspondence between rooted binary trees with leaves labeled by elements of X and monomials in $\mathbb{F}_{\bar{A},\bar{C}}[X]$. The nonassociativity introduces a unique product structure which may be interpreted as a binary tree: Let $m \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ be a monomial. Then there is a unique rooted binary tree T_m whose the leaves (labeled by elements of X) represent the variables, and whose internal nodes compute the product their two children. The root computes the monomial m. For instance, the binary trees in Figure 2(a) and 2(b) compute monomials $m = ((x_{i_1} x_{i_2}) x_{i_3})$ and $m' = (x_{i_1}(x_{i_2} x_{i_3}))$ respectively. Noncommutativity implies that each internal node has designated left and right children, and swapping this order changes the monomial.
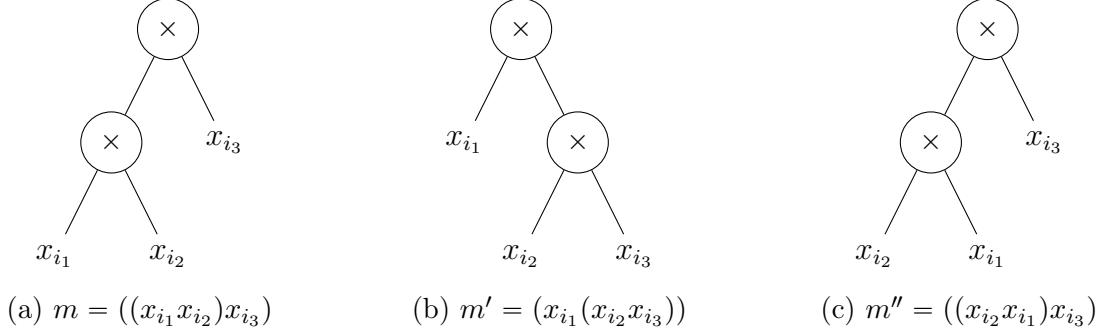
(a) $m = ((x_{i_1}x_{i_2})x_{i_3})$ (b) $m' = (x_{i_1}(x_{i_2}x_{i_3}))$ (c) $m'' = ((x_{i_2}x_{i_1})x_{i_3})$

Figure 2: Examples of nonassociative, noncommutative monomials

Suppose m has degree d. Noncommutativity gives us a unique string $\sigma_m = (i_1, \ldots, i_d) \in [n]^d$ which is the unique *left to right order* $x_{i_1}, x_{i_2} \ldots x_{i_d}$ in which the variables appear in m. We will think of $\sigma_m : [d] \to [n]$ as a function defined as $\sigma_m(j) = i_j$ for all $j \in [d]$. Note that σ_m does not uniquely define a monomial. For instance, for the monomials $m, m' \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ shown in Figure 2, $\sigma_m = \sigma_{m'}$ but $m \neq m'$. Given a monomial $m \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ as a binary tree T_m, we assign *level numbers* to the nodes of T_m. Define the level of a node v in T_m to be $1 + d(\text{root}, v)$, where $d(\cdot, \cdot)$ is the *distance* function. That is, the root is at level 1, the children of the root are at level 2 and so on. For each $j \in [d]$, let l_j^m denote the level at which the jth variable (in the left to right order) appears in m.

For monomials $m = ((x_{i_1}x_{i_2})x_{i_3})$ and $m' = (x_{i_1}(x_{i_2}x_{i_3}))$ shown in Figure 2, $l_1^m = 3, l_2^m = 3, l_3^m = 2$ and $l_1^{m'} = 2, l_2^{m'} = 3, l_3^{m'} = 3$. In general, the variable order σ_m and the level numbers $(l_1^m, \ldots, l_d^m)$ *together* uniquely determine the monomial m.

We state this formally in the following lemma:

Lemma 8. *Let m, m' be distinct monomials in $\mathbb{F}_{\bar{A},\bar{C}}[X]$. Let $\deg(m) = d$ and $\deg(m') = d'$. Then the tuples $(\sigma_m, l_1^m, \ldots, l_d^m)$ and $(\sigma_{m'}, l_1^{m'}, \ldots, l_{d'}^{m'})$ are distinct.*

Proof. We assume that $\sigma_m = \sigma_{m'}$, for otherwise the statement is evidently true. In particular, assume that the degrees of m and m' are both equal to d. Next, we simply observe that it is possible to iteratively reconstruct m uniquely, given the sequence $l_1^m, \ldots, l_d^m$ of levels. Start with just the root. To it, add a path of length $l_1^m - 1$ consisting entirely of left edges (every node is a left child of it's parent). Label the ultimate node (leaf) on the path by $x_{\sigma_m(1)}$. Now suppose we already have a leaf for the i-th variable, $x_{\sigma_m(i)}$. To generate the leaf labeled by $x_{\sigma_m(i+1)}$, find the ancestor v of $x_{\sigma_m(i)}$ closest to $x_{\sigma_m(i)}$ that only has a left child. Let the *level* of v be l. Add a right v' to child to v. If v' is at level $l_{\sigma_m(i+1)}$, label it by $x_{\sigma_m(i+1)}$. Otherwise, add to it a path of length $l_{\sigma_m(i+1)} - l - 1$ consisting only of left edges and label the ultimate node on this path by $x_{\sigma_m(i)}$. This process recovers m. The key property of trees representing nonassociative monomials that is used in the procedure above is that they are *full* binary trees, that is, each node either has exactly two children or none at all. This justifies our choice of going back to the the closest ancestor v of $x_{\sigma_m(i)}$ that only has a left child and no right child. $\square$

2.1.2   Over the algebra $\mathbb{F}_{\bar{A},C}[X]$

Next, we consider the case of monomials in $\mathbb{F}_{\bar{A},C}[X]$. One may partition the set of all monomials in $\mathbb{F}_{\bar{A},\bar{C}}[X]$ into equivalence classes under commutativity: For all monomials $m, m' \in \mathbb{F}_{\bar{A},\bar{C}}[X]$, $m \sim m'$ if and only if m and m' are equal up commutativity. For example the monomials shown in Figure 2(a) and Figure 2(b) belong to different equivalence classes. On the other hand, monomials in Figure 2(a) and Figure 2(c) belong to the same equivalence class.

Each equivalence class may be identified with a monomial in $\mathbb{F}_{\bar{A},C}[X]$. Looking at it from the other direction, suppose m, m' are monomials in $\mathbb{F}_{\bar{A},C}[X]$ and suppose M_m, $M_{m'}$ are the equivalence classes of monomials in $\mathbb{F}_{\bar{A},\bar{C}}[X]$ they represent. Since $\sim$ as defined above is an equivalence relation, we have the following simple observation:

Observation 9. *For any two distinct monomials* $m, m' \in \mathbb{F}_{\bar{A},C}[X]$ *we have* $M_m \cap M_{m'} = \varnothing$.

3   Randomized Black-box PIT for Nonassociative circuits

Let $\mathbb{F}$ be a field and let $X = \{x_1, \ldots, x_n\}$. In this section, we work with both the nonassociative, noncommutative polynomial algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$ as well as the commutative $\mathbb{F}_{\bar{A},C}[X]$. In subsection 3.1, we give a randomized, polynomial time black-box algorithms to test whether a nonassociative, noncommutative circuit Φ over $\mathbb{F}$ computes the identically zero polynomial. The algorithm assumes that we have access to evaluations of Φ on a suitable k dimensional $\mathbb{F}$-algebra A, and that the cost of one A-query is $\mathsf{poly}(\mathsf{size}(\Phi), k)$. That is, the cost is polynomial in the size of the circuit and the dimension of the algebra. To the best of our knowledge, this gives the first Amitsur-Levitzki type theorem [1] over nonassociative polynomial algebras.

3.1   Nonassociative, Noncommutative Randomized Black-box PIT

The key idea is to construct a noncommutative, nonassociative $\mathbb{F}$-algebra and show that it does not satisfy low degree polynomial identities. This will imply that a random non-zero substitution from this algebra will make a non-zero circuit Φ evaluate to something non-zero, with high probability.

We will query noncommutative, nonassociative circuits computing a polynomial $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ of degree $\leqslant d$ on a particular $d(d+1)^2 + 1$ dimensional algebra $\mathbb{A}_d$ that we now describe. First, we construct an algebra $\mathbb{A}'_d$ of dimension $d(d+1)^2$ and then construct the desired algebra $\mathbb{A}_d$ by adjoining an identity element to $\mathbb{A}'_d$. Additively, $\mathbb{A}'_d$ is an $\mathbb{F}$-vector space of dimension $d(d+1)^2$. We will think of an element of $\mathbb{A}'_d$ as a set of d matrices each of dimension $(d+1) \times (d+1)$:

Let $x, y \in \mathbb{F}^{d(d+1)^2}$. We index x, y as $x[i, j, k]$ and $y[i, j, k]$ for $1 \leqslant i, j \leqslant d+1$ and $1 \leqslant k \leqslant d$. Here, k can be thought of as indexing the set of d matrices and for a fixed k, i and j index respectively the rows and columns of the k-th matrix. Next, we define the bilinear $\mathbb{A}'_d$-product of x, y as follows: $z \triangleq x \circ y$ such that
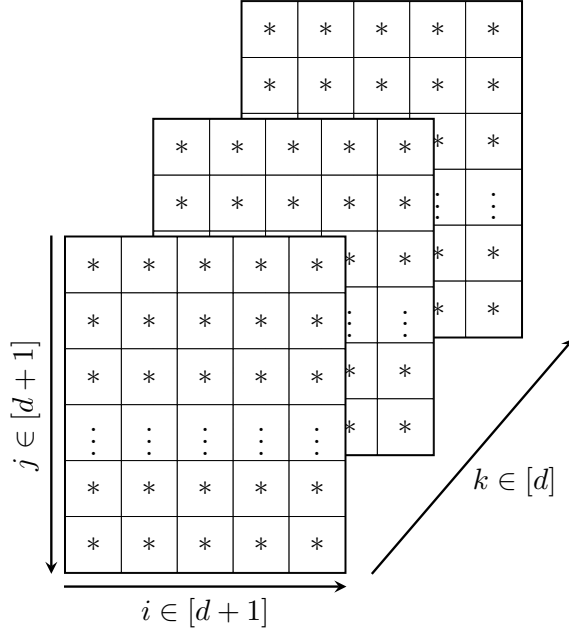
Figure 3: 3-dimensional view of an element of $\mathbb{A}'_d$

$$
z[i,j,k] = \begin{cases} 0 & k = d \\ \displaystyle\sum_{l=1}^{d+1} x[i,l,k+1]y[l,j,k+1] & 1 \leqslant k \leqslant d-1 \end{cases}
$$

Clearly, $\mathbb{A}'_d$ is an $\mathbb{F}$-algebra. From the proof of Lemma 10, it will be evident that $\mathbb{A}'_d$ is a nonassociative algebra.

We require the algebra $\mathbb{A}'_d$ to be unital to make sense of $\mathbb{A}'_d$-evaluations of polynomials with a non-zero constant term: For any $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$, $f(0,\ldots,0) = c \cdot \mathbf{1}$ where f has constant term $c \in \mathbb{F}$ and $\mathbf{1}$ is the identity element of $\mathbb{A}'_d$. However, $\mathbb{A}'_d$ is non-unital. To construct the unital algebra $\mathbb{A}_d$ (from $\mathbb{A}'_d$), we use a standard procedure of adjoining an identity to an algebra:

The Algebra $\mathbb{A_d}$: Define the algebra $\mathbb{A}_d$ to be the vector space $\mathbb{F}^{d(d+1)^2+1} = \{(a,\alpha) \mid a \in \mathbb{A}'_d, \alpha \in \mathbb{F}\}$ together with the bilinear product $\cdot$ defined as follows:

$$
(a_1,\alpha_1) \cdot (a_2,\alpha_2) = (a_1 \circ a_2 + \alpha_1 a_2 + \alpha_2 a_1, \alpha_1\alpha_2)
$$

We will refer to this newly added index as the last index. It is easy to verify that $\mathbb{A}_d$ is indeed a nonassociative algebra with $(\mathbf{0},1)$ being the multiplicative identity, and that $\mathbb{A}'_d$ is isomorphic to the sub-algebra $\{(a,0) \mid a \in \mathbb{A}'_d\}$ of $\mathbb{A}_d$.

Now, we show that $\mathbb{A}_d$ cannot have polynomial identities of degree $\leqslant d$.

Lemma 10. *Let $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ be a non-zero polynomial of total degree $\leqslant d$. Then f is not a PI for $\mathbb{A}_d$.*
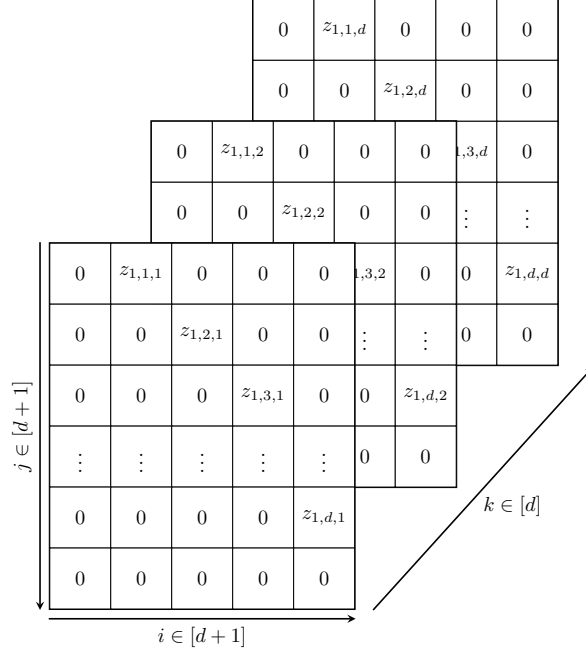
Figure 4: $Z_1 = \Phi(x_1)$ visualized as an element of $\mathbb{A}'_d$

Proof. It suffices to consider polynomials that do not have a constant term, since a polynomial with a non zero constant term evaluates to a non-zero value at the all zeroes input. Also, it suffices to prove Lemma 10 for the algebra $\mathbb{A}'_d$ instead of $\mathbb{A}_d$, since $\mathbb{A}'_d$ is a subalgebra of A_d. In order to prove the lemma, we reduce the problem to the associative, commutative setting.

For each x_i, we introduce an *associative, commutative* set of variables $\{z_{i,j,k} \mid 1 \leq j, k \leq d\}$. For convenience, denote the vector $(z_{i,j,k})_{j,k\in[d]}$ by $\mathbf{z_i}$. Extend the ground field $\mathbb{F}$ to the function field $\mathbb{F}' = \mathbb{F}(\mathbf{z}_1, \ldots, \mathbf{z}_n)$. We define the algebra $\mathbb{A}'_d$ over the field $\mathbb{F}'$ as described earlier. Eventually, the $\mathbf{z}$ variables will be fixed suitably from the base field $\mathbb{F}$.

Next, consider the evaluation map $\Phi : \mathbb{F}_{\bar{A},\bar{C}}[X] \to \mathbb{A}'_d$ that sends x_i to Z_i where for each $1 \leq j, k \leq d$, the $(j, j+1, k)$-th entry of Z_i is $z_{i,j,k}$ and the rest of the entries are zero. For an illustration, we describe Z_1 explicitly in Figure 4.

Let us look at the image of a monomial $m \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ (of degree at most d) under Φ. Let $d' \leq d$ be the degree of m. We interpret m as a binary tree with leaves labeled by variables. Since we are in the noncommutative setting, there is a unique function $\sigma_m : [d'] \to [n]$ that describes the left-to-right order in which the variables appear in m. We will *level* the nodes (including leaves) of m as described in Section 2.1. Let l^m_t denote the *level* at which the t-th variable in σ_m (i.e., variable $x_{\sigma_m(t)}$) appears in m. Let the *depth* of m (i.e., $\max_i\{l_i\}$) be l.

Claim 11. *Let m be as above. For each $k_1 \in [d - d' + 1]$ and each $k_2 \in [d - l + 1]$, the $(k_1, k_1 + d', k_2)$-th entry of $\Phi(m)$ is the monomial $\prod_{t=1}^{d'} z_{\sigma_m(t),t+k_1-1,l^m_t+k_2-1}$, and every other entry is zero.*

Proof. We prove this by induction on the degree d' of m. For $\mathsf{deg}(m) = 1$, the claim follows from the definition of Z_i's. Now suppose $d' > 1$ and m is uniquely written as $m_1 m_2$ such

13

that $\deg(m_1) = d_1$, $\deg(m_2) = d_2$ and $d_1 + d_2 = d'$. Then,

$$\Phi(m)[k_1, k_1 + d', k_2] = \sum_{i=1}^{d+1} \Phi(m_1)[k_1, i, k_2 + 1]\Phi(m_2)[i, k_1 + d', k_2 + 1]$$

By induction, we see that exactly one term in this sum is non zero, the one corresponding to $i = k_1 + d_1$ and the sum is therefore equal to

$$\left(\prod_{t=1}^{d_1} z_{\sigma_{m_1}(t), t+k_1-1, l_t^{m_1}+k_2}\right) \left(\prod_{t=1}^{d_2} z_{\sigma_{m_2}(t), t+d_1+k_1-1, l_t^{m_2}+k_2}\right) \tag{1}$$

Notice that

$$\sigma_m(t) = \begin{cases} \sigma_{m_1}(t) & 1 \leqslant t \leqslant d_1 \\ \sigma_{m_2}(t - d_1) & d_1 + 1 \leqslant t \leqslant d' \end{cases}$$

and that

$$l_t^m = \begin{cases} l_t^{m_1} + 1 & 1 \leqslant t \leqslant d_1 \\ l_{t-d_1}^{m_2} + 1 & d_1 + 1 \leqslant t \leqslant d' \end{cases}$$

Using these observations, we find that (1) is exactly equal to $\prod_{t=1}^{d'} z_{\sigma_m(t), t+k_1-1, l_{\sigma_m(t)}^m+k_2-1}$.

Now let us look at $\Phi(m)[i_1, i_2, i_3]$ such that $i_2 \neq i_1 + d'$ and $i_3 \in [d-1]$ (if $i_3 = d$, $\Phi(m)[i_1, i_2, i_3] = 0$ by definition of the $\mathbb{A}_d$ product).

$$\Phi(m)[i_1, i_2, i_3] = \sum_{i=1}^{d+1} \Phi(m_1)[i_1, i, i_3 + 1]\Phi(m_2)[i, i_2, i_3 + 1]$$

Using the induction hypothesis, we see that each summand in the sum above is actually zero, and therefore so is $\Phi(m)[i_1, i_2, i_3]$.

Let us also look at $\Phi(m)[i_1, i_2, i_3]$ with $i_3 > d - l + 1$. If $d' = 2$ then all such entries of $\Phi(m)$ are easily seen to be zero. Now suppose $d' > 2$ and assume, without loss of generality, that the depth of m_1 is $l - 1$. Then by induction, all entries $\Phi(m_1)[i_1, i_2, i_3]$ of $\Phi(m_1)$ with $i_3 > d-l$ are zero and therefore so are all the entries $\Phi(m)[i_1, i_2, i_3]$ of $\Phi(m)$ with $i_3 > d-l+1$.

This concludes the proof of Claim 11. $\qquad\square$

Now, by setting $k_1, k_2 = 1$ in the statement of Claim 11 we see that for a monomial m of degree d', the $(1, d' + 1, 1)$-th entry of $\Phi(m)$ is $\prod_{t=1}^{d'} z_{\sigma_m(t), t, l_t}$. For any monomial of degree $\neq d'$, this entry is zero. This observation combined with Lemma 8 gives us Lemma 10. $\qquad\square$

Using Lemma 10, we exhibit a randomized black-box identity testing algorithm for nonassociative, noncommutative circuits.

Theorem 12. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| > d$, and $S \subset \mathbb{F}$. Let $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ be a nonzero polynomial of degree $\leqslant d$ given as a black-box with query access to evaluations of f on*

elements of $\mathbb{A}_d$. *Let* $S \subseteq \mathbb{F}$ *with* $|S| > d$. *Sample* $b_1, \ldots, b_n \in \mathbb{A}_d$ *as follows: Pick each of the* $d(d+1)^2 + 1$ *entries of each of the* b_i'*s uniformly and independently from* S. *Then*

$$\Pr_{b_1, \ldots, b_n \in \mathbb{A}_d} [f(b_1, \ldots, b_n) = 0] \leqslant d/|S|.$$

Proof. From Lemma 10, it follows that f is not a PI for $\mathbb{A}_d$. By slight abuse of notation, define $\mathbb{A}_d$ over the field $\mathbb{F}(Y)$ where Y is a commutative, associative set of variables and $|Y| = nd(d+1)^2 + 1$. Replace each x_i by $Y_i \in \mathbb{A}_d$ defined as follows: Y_i has dimension $d(d+1)^2 + 1$ and each entry of each Y_i is a fresh variable from Y. Since f is not a PI for $\mathbb{A}_d$, $f(Y_1, \ldots, Y_n) \not\equiv 0$ and so at least one entry of $f(Y_1, \ldots, Y_n)$ is a non-zero polynomial in $\mathbb{F}[Y]$. By the Polynomial Identity Lemma (Lemma 6), under a random S-substitution, the probability that that entry of $f(Y_1, \ldots, Y_n)$ evaluates to zero is $\leqslant d/|S|$. $\square$

Theorem 12 can be thought of as a version of Theorem 7 over $\mathbb{F}_{\bar{A}, \bar{C}}$. It immediately gives us the desired black-box PIT algorithm.

3.2 Nonassociative, Commutative Randomized Black-box PIT

Next, we construct a nonassociative, *commutative* algebra $\mathbb{C}_d$ that does not satisfy low degree identities in $\mathbb{F}_{\bar{A}, C}[X]$. $\mathbb{C}_d$ is constructed using the algebra $\mathbb{A}_d$ from the previous section (Section 3.1). Recall that $\cdot$ denotes the $\mathbb{A}_d$-product.

The Algebra $\mathbb{C}_d$: $\mathbb{C}_d$ is isomorphic to $\mathbb{A}_d$ as a vector space, and the $\mathbb{C}_d$ product of a, b (denoted by $\odot$) is simply the *anticommutator* of a, b with respect to the $\mathbb{A}_d$ product "$\cdot$". That is, $a \odot b = a \cdot b + b \cdot a$. Let $\mathbb{C}'_d$ denote the sub-algebra of $\mathbb{C}_d$ obtained by setting the last index to 0. It is easily verified that $\mathbb{C}'_d$ is isomorphic to the algebra whose product is the anticommutator with respect to the $\mathbb{A}'_d$ product $\circ$. For convenience, in the sequel, we will drop the last index of elements of $\mathbb{C}'_d$.

Lemma 1. *Let* $f \in \mathbb{F}_{\bar{A}, C}[X]$ *be a non-zero polynomial of total degree* $\leqslant d$. *Then* f *is not a polynomial identity (PI) for* $\mathbb{C}_d$.

Proof. The proof of Lemma 1 is similar to the proof of Lemma 10. As before, we note that it suffices to prove the lemma for constant free polynomials and that it suffices to prove the claim for $\mathbb{C}'_d$ instead of $\mathbb{C}_d$, because any identity of $\mathbb{C}_d$ is also an identity of $\mathbb{C}'_d$.

For each x_i, we introduce the same *associative, commutative* set of variables $\{z_{i,j,k} \mid 1 \leqslant j, k \leqslant d\}$ as before. Denote the vector $(z_{i,j,k})_{j,k \in [d]}$ by $\mathbf{z_i}$. As in the proof of Lemma 10, we consider the extended field $\mathbb{F}' = \mathbb{F}(\mathbf{z_1}, \ldots, \mathbf{z_n})$ and define $\mathbb{C}'_d$ over $\mathbb{F}'$. We consider the evaluation map $\Phi : \mathbb{F}_{\bar{A}, C}[X] \to \mathbb{C}'_d$ that sends x_i to Z_i where for each $1 \leqslant j, k \leqslant d$, the $(j, j+1, k)$-th entry of Z_i is $z_{i,j,k}$ and the rest of the entries are zero. Again, we would like to inspect the image of a monomial $m \in \mathbb{F}_{\bar{A}, C}[X]$ of degree $d' \leqslant d$ under Φ. As before, we interpret m as a binary tree. Without loss of generality, let the variables appearing in m be $x_1, \ldots, x_{d'}$. Unlike in the noncommutative case, there is no unique left to right order of variables that one can associate with the monomial m.

There is, however, a *set* of orders that one can associate with m: For each internal node in the tree representing m, arbitrarily designate one of the children to be the left child and

the other to be the right child. This procedure induces an order σ on the variables of m. Furthermore, every distinct way of designating left and right children at the internal nodes of m induces a unique order. Consider the union of all these orders and denote this set by Σ_m. Each $\sigma \in \Sigma_m$ corresponds to a unique monomial from $\mathbb{F}_{\bar{A},\bar{C}}[X]$ in the equivalence class M_m corresponding to the monomial m (see Section 2.1, the commutative case). Also, for a fixed $\sigma \in \Sigma_m$, let $l_t^{m,\sigma}$ denote the *level* at which the t-th variable in the order σ (i.e., variable $x_{\sigma(t)}$) appears in m. Let the *depth* of m (i.e., $\max_i\{l_i\}$) be l.

Claim 13. *Let m be as above. For each $k_1 \in [d - d' + 1]$ and each $k_2 \in [d - l + 1]$, the $(k_1, k_1 + d', k_2)$-th entry of $\Phi(m)$ is the polynomial $\sum\limits_{\sigma \in \Sigma_m} \prod\limits_{t=1}^{d'} z_{\sigma(t), t+k_1-1, l_t^{m,\sigma}+k_2-1}$, and every other entry is zero.*

Proof. We prove this by induction on the degree d' of m. For $\deg(m) = 1$, $|\Sigma_m| = 1$ and the claim follows from the definition of Z_i's. Now suppose $d' > 1$ and m is uniquely written as $m_1 m_2$ such that $\deg(m_1) = d_1$, $\deg(m_2) = d_2$ and $d_1 + d_2 = d'$. Then,

$$\Phi(m)[k_1, k_1 + d', k_2] = \sum_{i=1}^{d+1} \Phi(m_1)[k_1, i, k_2 + 1]\Phi(m_2)[i, k_1 + d', k_2 + 1]$$

$$+ \sum_{i=1}^{d+1} \Phi(m_2)[k_1, i, k_2 + 1]\Phi(m_1)[i, k_1 + d', k_2 + 1]$$

By induction, we see that the first sum $\sum\limits_{i=1}^{d+1} \Phi(m_1)[k_1, i, k_2+1]\Phi(m_2)[i, k_1+d', k_2+1]$ is equal to

$$\left(\sum_{\pi \in \Sigma_{m_1}} \prod_{t=1}^{d_1} z_{\pi(t), t+k_1-1, l_t^{m_1,\pi}+k_2} \right) \left(\sum_{\tau \in \Sigma_{m_2}} \prod_{t=1}^{d_2} z_{\tau(t), t+d_1+k_1-1, l_t^{m_2,\tau}+k_2} \right)$$

while the second sum $\sum_{i=1}^{d+1} \Phi(m_2)[k_1, i, k_2 + 1]\Phi(m_1)[i, k_1 + d', k_2 + 1]$ is equal to

$$\left(\sum_{\tau \in \Sigma_{m_2}} \prod_{t=1}^{d_2} z_{\tau(t), t+k_1-1, l_t^{m_2,\tau}+k_2} \right) \left(\sum_{\pi \in \Sigma_{m_1}} \prod_{t=1}^{d_1} z_{\pi(t), t+d_2+k_1-1, l_t^{m_1,\pi}+k_2} \right)$$

Expanding the products, we see that the first sum generates the monomials in the polynomial $\sum\limits_{\sigma \in \Sigma_m} \prod\limits_{t=1}^{d'} z_{\sigma(t), t+k_1-1, l_t^{m,\sigma}+k_2-1}$ corresponding to the $\sigma \in \Sigma_m$ that make m_1 the left child of the root of m and m_2 the right, and the second term generates the rest of the monomials.

Also, note that the same argument as in the proof of Lemma 10 tells us that if we have i_1, i_2, i_3 such that $i_2 \neq i_1 + d'$ and $i_3 \in [d]$ then $\Phi(m)[i_1, i_2, i_3] = 0$ and that if $i_3 > d - l + 1$ then again $\Phi(m)[i_1, i_2, i_3] = 0$. $\qquad\square$

In particular, by setting $k_1, k_2 = 1$ in the statement of Claim 13 we see that for a monomial m of degree d', the $(1, d'+1, 1)$-th entry of $\Phi(m)$ is $\sum_{\sigma \in \Sigma_m} \prod_{t=1}^{d'} z_{\sigma(t),t,l_t^{m,\sigma}}$. For a monomial of degree $\neq d'$, this entry is zero.

Therefore, Combining Claim 13, Lemma 8 and Observation 9, we see that f cannot be an a PI for $\mathbb{C}'_d$ (and therefore for $\mathbb{C}_d$). $\qquad\square$

The proof of Theorem 2 follows from Lemma 1, in a way similar to the proof of Theorem 12. We record the statement here for completeness

Theorem 2. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| > d$, and $S \subset \mathbb{F}$. Let $f \in \mathbb{F}_{\bar{A},C}[X]$ be a non-zero polynomial of degree $\leqslant d$ given as a black-box with query access to evaluations of f on elements of $\mathbb{C}_d$. Let $S \subseteq \mathbb{F}$ with $|S| > d$. Sample $b_1, \ldots, b_n \in \mathbb{C}_d$ as follows: Pick each of the $d(d+1)^2 + 1$ entries of each of the b_i's uniformly and independently from S. Then*

$$\Pr_{b_1,\ldots,b_n \in \mathbb{C}_d} [f(b_1, \ldots, b_n) = 0] \leqslant d/|S|.$$

4 Deterministic PIT Algorithms Over $\mathbb{F}_{\bar{A},C}[X]$

In this section, we develop efficient deterministic PIT algorithms over the algebra $\mathbb{F}_{\bar{A},C}[X]$. First, we present the white-box algorithm and then the black-box algorithm.

4.1 White-box Deterministic PIT over $\mathbb{F}_{\bar{A},C}[X]$

We give a polynomial time white-box PIT algorithm for commutative, nonassociative circuits. We use linear algebraic ideas from the PIT algorithm by Raz and Shpilka [16] for noncommutative algebraic branching programs. These ideas have later been used to give polynomial time white-box PIT algorithms for various models, such as nonassociative, *noncommutative* circuits [25] and noncommutative unique parse tree circuits [27].

Theorem 3. *Let Ψ be a nonassociative arithmetic circuit of size s computing an n variate, degree $\leqslant d$ polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$. Given Ψ as input, we can check whether $f \equiv 0$ deterministically in time $\mathsf{poly}(s, n, d)$.*

Proof. For each monomial m of degree $\leqslant d$, we may associate a vector $v_m \in \mathbb{F}^s$ where s is the number of gates in Ψ. The vector v_m is indexed by the gates of Ψ such that $v_m(g) = \mathsf{coeff}_m(g)$. For each $i \in \{0\} \cup [d]$, we wish to maintain a polynomially bounded set M_i of monomials of degree i and a corresponding set $B_i = \{v_m \mid m \in M_i\}$ of vectors such that $\mathsf{span}\{B_i\} = \mathsf{span}\{v_m \mid \deg(m) = i\}$, and we build these sets inductively, starting from $i = 0, 1$. For $i = 0, 1$, we set M_i to be the set of all monomials of degree i and populate the vectors in B_i in a brute force manner.

Next, suppose we have the sets M_i, B_i for $0 \leqslant i < j$ and we want to construct M_j and B_j ($j \geqslant 2$). We set

$$M'_j = \bigcup_{\substack{i+k=j \\ i,k \geqslant 1}} \{m \times m' \mid m \in M_i \text{ and } m' \in M_k\}$$

17

For all $m = m_1 m_2 \in M_j'$ we do the following: we sort the gates of Ψ in topological order and fill the entries of v_m in that order as follows:

- If g is a leaf, we set $v_m(g) = 0$ (since $j = \deg(m) \geqslant 2$).

- If $g = g_1 \times g_2$ is a product gate, set $v_m(g) = v_{m_1}(g_1)v_{m_2}(g_2) + v_{m_2}(g_1)v_{m_1}(g_2) + v_m(g_1)v_1(g_2) + v_1(g_1)v_m(g_2)$. We can do this since we know the vectors v_{m_1}, v_{m_2} by induction on the degree, and we know $v_m(g_1), v_m(g_2)$ as g_1, g_2 appear before g in the topological order.

- If $g = g_1 + g_2$ is a sum gate, set $v_m(g) = v_m(g_1) + v_m(g_2)$. Again, we know $v_m(g_1), v_m(g_2)$ as g_1, g_2 appear before g.

Finally, we select a maximal linearly independent subset B_j (using Gaussian elimination) from the set of vectors $\{v_m \mid m \in M_j'\}$ and call the corresponding set of monomials M_j. Clearly, $|M_j| \leqslant s$.

Claim 14. *For any monomial m such that $\deg(m) = j$, $v_m \in \mathsf{span}\{B_j\}$.*

Proof. The proof is by induction on the degree j. For $j = 0, 1$, the claim is trivially true, so assume $j \geqslant 2$. Now suppose m is uniquely decomposed (up to commutativity) as $m = m_1 m_2$ such that $\deg(m_1) = i$ and $\deg(m_2) = k$ (with $i, k \geqslant 1$). By induction on degree, we assume that

$$v_{m_1} = \sum_{m' \in M_i} \alpha_{m'} v_{m'} \text{ and } v_{m_2} = \sum_{m'' \in M_k} \beta_{m''} v_{m''} \tag{2}$$

for constants $\{\alpha_{m'} \mid m' \in M_i\}$ and $\{\beta_{m''} \mid m'' \in M_k\}$. We will prove that for each gate g of Ψ

$$v_m(g) = \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} v_{m'm''}(g) \tag{3}$$

This puts v_m in $\mathsf{span}\{B_j\}$ by construction. We prove (3) gate by gate, by induction on the depth of the gate.

- For a leaf g, g is labeled by a variable or a constant, so $v_m(g) = 0$ (recall that $j \geqslant 2$) and $v_{m'm''}(g)$ for each $m' \in B_i, m'' \in B_k$ is also all zero by construction, so (3) is true in this case.

- For a product gate $g = g_1 \times g_2$,

$$v_m(g) = v_{m_1}(g_1)v_{m_2}(g_2) + v_{m_2}(g_1)v_{m_1}(g_2) + v_m(g_1)v_1(g_2) + v_1(g_1)v_m(g_2).$$

After substituting (2) for v_{m_1} and v_{m_2} using the (degree) induction hypothesis, substituting (3) for $v_m(g_1), v_m(g_2)$ using induction on the depth of g_1, g_2 and simplifying, we get that

18

$$v_m(g) = \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} \left(\sum_{(r,t) \in \{(1,2),(2,1)\}} v_{m'}(g_r) v_{m''}(g_t) + v_{m'm''}(g_r) v_1(g_t) \right)$$

Note that the inner expression is nothing but $v_{m'm''}(g)$ (by construction), and therefore (3) is true when g is a product gate.

- For a sum gate $g = g_1 + g_2$, we have

$$
\begin{aligned}
v_m(g) &= v_m(g_1) + v_m(g_2) \\
&= \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} v_{m'm''}(g_1) + \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} v_{m'm''}(g_2) \\
&= \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} \left(v_{m'm''}(g_1) + v_{m'm''}(g_2) \right) \\
&= \sum_{\substack{m' \in M_i \\ m'' \in M_k}} \alpha_{m'} \beta_{m''} v_{m'm''}(g)
\end{aligned}
$$

where the second step follows by induction on depth and the fourth by construction of $v_{m'm''}$. This verifies (3) when g is a sum gate.

These three cases together prove the claim. □

Clearly, this procedure of constructing B_j's and M_j's takes polynomial time since $|B_j| \leqslant \max\{s, n, 1\}$ for each $0 \leqslant j \leqslant d$. To check whether $\Psi \equiv 0$, we simply check whether there exists an $i \in \{0, \ldots, d\}$ such that there exists a monomial $m \in M_i$ such that $v_m(g_s) \neq 0$, where g_s is the output gate of Ψ. If yes, we say $\Psi \not\equiv 0$, otherwise we say $\Psi \equiv 0$. □

4.2 Deterministic Nonassociative Black-box PIT

In this section, we design black-box PIT algorithms for polynomials in $\mathbb{F}_{\bar{A},C}[X]$ and $\mathbb{F}_{\bar{A},\bar{C}}[X]$ computed by low depth circuits. Our algorithms run in quasipolynomial time as long as the input circuit Ψ has depth *polylogarithmic* in the size of Ψ and the degree of the polynomial computed by it. The algorithms query Ψ on elements of the algebra $\mathbb{A}_d$ in the noncommutative case and on elements of $\mathbb{C}_d$ in the commutative case, defined in sections 3.1 and 3.2 respectively. In particular, we construct *hitting sets* of quasipolynomial size in both the commutative and noncommutative setting, for circuits that have depth polylogarithmic in its size and degree.

Theorem 4. *There exists a set $H_{n,s,d,\Delta} \subseteq (\mathbb{C}_d)^n$ of size $(nsd)^{O(\Delta)}$ of points in $(\mathbb{C}_d)^n$ such that for every nonassociative, commutative circuit Ψ of size $\leqslant s$ and product depth $\leqslant \Delta$ computing a non-zero polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ of degree $\leqslant d$, there is a point in $H_{n,s,d,\Delta}$ at which f is non-zero. Furthermore, we can compute $H_{n,s,d,\Delta}$ deterministically in time $(nsd)^{O(\Delta)}$.*

Over the algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$, we have the following analogous result.

Theorem 15. *There exists a set $H_{n,s,d,\Delta} \subseteq (\mathbb{A}_d)^n$ of size $(nsd)^{O(\Delta)}$ of points in $(\mathbb{A}_d)^n$ such that for every nonassociative, noncommutative circuit Ψ of size $\leqslant s$ and product depth $\leqslant \Delta$ computing a non-zero polynomial $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ of degree $\leqslant d$, there is a point in $H_{n,s,d,\Delta}$ at which f is non-zero. Furthermore, we can compute $H_{n,s,d,\Delta}$ in time $(nsd)^{O(\Delta)}$.*

We prove Theorem 4 and Theorem 15 in the next two subsections. The idea is to reduce PIT for nonassociative circuits to PIT for associative, unambiguous circuits (see below for a formal definition) and then give hitting sets for unambiguous circuits.

Definition 6 (Unambiguous Circuits)**.** Let $Z = \{z_1, \ldots, z_n\}$ be a commutative, associative set of variables and let $\mathrm{mons}(Z)$ denote the set of monomials in the variables in Z. We say that a circuit Ψ computing a polynomial $f \in \mathbb{F}[Z]$ is *unambiguous* if for each monomial m in f there is a *unique* reduced parse tree T_m in Ψ generating m. That is, if another reduced parse tree T generates m at any gate in Ψ, then the trees T and T_m are identical as labeled rooted binary trees.

4.2.1 Reduction from non-associative to associative, unambiguous circuits

The first, fairly straightforward step is to reduce PIT for nonassociative circuits to PIT for associative, unambiguous circuits. We do this via a set-multilinearization argument. This type of reduction is commonplace in noncommutative PIT literature (see for example [21], [26])

We first describe the reduction in the nonassociative, commutative setting. The proof is along exactly the same lines in the noncommutative setting as well (in fact it is simpler).

Let $\{x_1, \ldots, x_n\}$ be a set of variables. Let Ψ be a circuit computing a polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ of degree $d' \leqslant d$. For each variable $x_i, i \in [n]$, we introduce a set $\{z_{i,j,k} \mid 1 \leqslant j, k \leqslant d\}$. Let $Z = \bigcup_{i=1}^{n}\{z_{i,j,k} \mid 1 \leqslant j, k \leqslant d\}$ and define the algebras $\mathbb{C}_d$ and $\mathbb{C}'_d$ over the field $\mathbb{F}(Z)$. We consider the evaluation map $\Phi : \mathbb{F}_{\bar{A},C}[X] \to \mathbb{C}'_d$ defined in Section 3.2. Let us recall the definition of Φ: Φ maps x_i to Z_i where for each $1 \leqslant j, k \leqslant d$, the entry $(j, j + 1, k)$ of Z_i is $z_{i,j,k}$ and the rest of the entries are zero.

We will examine the image of the circuit Ψ under Φ. To this end, we define another map $\phi : \mathbb{F}_{\bar{A},C}[X] \to \mathbb{F}[Z]$. Let m be a monomial in $\{x_1, \ldots, x_n\}$ of degree d'. Recall that in Section 3.2, we associated to m a set Σ_m of "orders" and for each $\sigma \in \Sigma_m$, we had $l_t^{m,\sigma}$ (for each $1 \leqslant t \leqslant d'$), the level at which the t-th variable in the order σ appears in m. For a monomial m in $\mathbb{F}_{\bar{A},C}[X]$ define

$$\phi(m) \triangleq \sum_{\sigma \in \Sigma_m} \prod_{t=1}^{d'} z_{\sigma(t),t,l_t^{m,\sigma}}$$

and extend ϕ linearly to all of $\mathbb{F}_{\bar{A},C}[X]$.

Lemma 16. *Let $f \in \mathbb{F}_{\bar{A},C}[X]$ be a homogeneous polynomial of degree $d' \leqslant d$. The $(1, d'+1, 1)$-th entry of $\Phi(f)$ is equal to $\phi(f_{d'})$ where $f_{d'}$ is the homogeneous degree d' component of f.*

Proof. This is a simple corollary of Claim 13. Setting $k_1, k_2 = 1$ in the statement of Claim 13, we see that for a monomial m of degree d', the $(1, d' + 1, 1)$-th entry of $\Phi(m)$ is $\phi(m)$. On the other hand, for a monomial m of degree $\neq d'$, we get that the $(1, d' + 1, 1)$-th entry of $\Phi(m)$ is 0. Therefore, Lemma 16 follows by summing over monomials of f. $\qquad\square$

Lemma 17. *Let Ψ be a nonassociative, commutative arithmetic circuit of size s and product depth Δ computing a polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ of degree $d' \leqslant d$. Then, there exists an unambiguous circuit Ψ' computing $\phi(f_{d'})$ where $f_{d'}$ is the homogeneous degree d' component of f. Furthermore, the size of Ψ' is at most $3d^4 s$ and product depth at most Δ.*

Proof. We assume without loss of generality that we have a homogeneous circuit $\Psi^{d'}$ for computing $f_{d'}$, with size at most $d^2 s$ and product depth at most Δ. We can do this since nonassociative circuits can be homogenized using a standard technique [20] with a multiplicative blowup of d^2 in size. This has also been observed by the authors in [19]. Furthermore, we may also assume (without loss of generality) that $\Psi^{d'}$ does not have gates (including leaves) computing constants from $\mathbb{F}$.

We will build Ψ' using $\Psi^{d'}$, in a bottom up fashion such that Ψ' computes the non-zero entries of $\Phi(f_{d'})$. First, for each leaf labeled by x_i in $\Psi^{d'}$, introduce d^2 leaves in Ψ', each labeled by one of the d^2 nonzero entries ($\{z_{i,j,k} \mid j, k \in [d]\}$) of Z_i. Recall that $z_{i,j,k}$ appears as the $(j, j + 1, k)$-th entry of $\Phi(x_i) = Z_i$. Next, suppose we are at an internal gate g of $\Psi^{d'}$ with children g_1, g_2. Let f^g denote the polynomial computed at gate g in $\Psi^{d'}$ and let the degree of f^g be $d'' \leqslant d'$.

- g *is a sum gate in* $\Psi^{d'}$. In this case, $\mathsf{deg}(f^{g_1}) = \mathsf{deg}(f^{g_2}) = d''$ since $\Psi^{d''}$ is homogeneous. For each $k_1 \in [d - d'' + 1]$ and $k_2 \in [d]$, we compute in Ψ' the $(k_1, k_1 + d'', k_2)$-th entry of $\Phi(f^g)$ by summing the corresponding entries of $\Phi(f^{g_1})$ and $\Phi(f^{g_2})$, since sum in $\mathbb{C}'$ is pointwise. Note that these are the only entries of $\Phi(f^g)$ that can be nonzero, by Claim 13.

- g *is a product gate in* $\Psi^{d'}$ *with children* g_1, g_2: Let the degrees of f^{g_1}, f^{g_2} be d_1, d_2 respectively with $d_1 + d_2 = d''$. For each $k_1 \in [d - d'' + 1]$ and $k_2 \in [d]$, we compute in Ψ' the $(k_1, k_1 + d'', k_2)$-th entry of $\Phi(f^g)$ as follows:

$$\Phi(f^g)[k_1, k_1 + d'', k_2] = \Phi(f^{g_1})[k_1, k_1 + d_1, k_2 + 1]\Phi(f^{g_2})[k_1 + d_1, k_1 + d'', k_2 + 1] + \\ \Phi(f^{g_2})[k_1, k_1 + d_2, k_2 + 1]\Phi(f^{g_1})[k_1 + d_2, k_1 + d'', k_2 + 1]$$

  Again, this is justified by Claim 13. The entries we have computed are the only entries of $\Phi(f^g)$ that can be nonzero.

By Lemma 16, the gate in Ψ' computing the $(1, d' + 1, 1)$-th entry of $\Phi(f_{d'})$ computes $\phi(f_{d'})$. By construction, Ψ' has size $3d^4 s$ and product depth at most Δ.

Claim 18. *Ψ' is unambiguous.*

Proof. This follows from the fact that $\Psi^{d'}$ is nonassociative. In particular, owing to Claim 13, we have that for every monomial $m \in \mathbb{F}[Z]$ (with degree $d'' \leqslant d'$) computed by Ψ', there

21

exists a unique monomial $\hat{m} \in \mathbb{F}_{\bar{A},C}[X]$ with degree d'' and depth l, an order $\sigma \in \Sigma_{\hat{m}}$, a $k_1 \in [d - d'' + 1]$ and a $k_2 \in [d - l + 1]$ such that

$$m = \prod_{t=1}^{d'} z_{\sigma(t), t+k_1-1, l_t^{\sigma,\hat{m}}+k_2-1}$$

Furthermore, if the reduced parse tree for m decomposes m as $m = m_1 m_2$ then $\hat{m} = \hat{m_1}\hat{m_2}$. This property ensures that Ψ' is indeed unambiguous $\qquad\square$

This finishes the proof of Lemma 17. $\qquad\square$

In the non-commutative setting, let $\{x_1, \ldots, x_n\}$ be a set of variables and let the set Z of variables be as before. We consider the map $\Phi : \mathbb{F}_{\bar{A},\bar{C}}[X] \to \mathbb{A}'_d$ from Section 3.1 (where $\mathbb{A}'_d$ is an algebra over the field $\mathbb{F}(Z)$). In this case, Φ sends x_i to $Z_i \in \mathbb{A}'_d$ where for each $1 \leqslant j, k \leqslant d$ the $(j, j+1, k)$-th entry of Z_i is $z_{i,j,k}$ and every other entry is zero.

Recall that for any monomial $m \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ of degree d' there is a unique left to right order $\sigma_m : [d'] \to [n]$ of variables associated to m. To study the image of a polynomial under Φ, we need the auxiliary map $\phi : \mathbb{F}_{\bar{A},\bar{C}}[X] \to \mathbb{F}[Z]$ defined as follows: for a monomial m as above,

$$\phi(m) = \prod_{t=1}^{d'} z_{\sigma_m(t), t, l_t^m}$$

Using these definitions of Φ and ϕ, and along the same lines as the proofs of Lemmas 16 and 17, we obtain the following:

Lemma 19. *Let $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ be a homogeneous polynomial of degree $d' \leqslant d$. The $(1, d'+1, 1)$-th entry of $\Phi(f)$ is equal to $\phi(f_{d'})$ where $f_{d'}$ is the homogeneous degree d' component of f.*

Lemma 20. *Let Ψ be a nonassociative, noncommutative arithmetic circuit computing a polynomial $f \in \mathbb{F}_{\bar{A},\bar{C}}[X]$ of degree $d' \leqslant d$. Then, there exists an unambiguous circuit Ψ' computing $\phi(f_{d'})$ (where $f_{d'}$ is the homogeneous degree d' component of f). Furthermore, the size of Ψ' is at most $3d^4s$ and product depth at most Δ.*

4.2.2 Hitting sets for low-depth associative, unambiguous circuits

In this section, we construct hitting sets for *low-depth* unambiguous circuits in the commutative, associative setting. We first define the most important tool in the design of hitting sets for unambiguous circuits: *basis isolating weight assignments*. Agrawal et al. [23] defined basis isolating weight assignments and used them to construct hitting sets for Read-Once oblivious ABPs. These weight assignements were subsequently also used in a work of Saptharishi and Tengse [26] for PIT of noncommutative unique parse tree circuits.

Let $Z = \{z_1, \ldots, z_n\}$ and let Ψ an unambiguous circuit with s gates computing a polynomial $f \in \mathbb{F}[Z]$ of degree d. Define $f_\Psi \in \mathbb{F}^s[Z]$ to be the polynomial $\sum_{m \in \text{mons}(Z)} v_m m$ where for each monomial m, v_m, the coefficient of m in f_Ψ, is (as before) an s dimensional vector whose entries are indexed by gates of Ψ and for each gate g in Ψ, $v_m(g) \triangleq \text{coeff}_m(g)$. Let $w : Z \to \mathbb{N}$ be a weight function that assigns weights to variables in Z. w extends to $\text{mons}(Z)$ naturally as follows: $w(z_1^{i_1} z_2^{i_2} \ldots z_n^{i_n}) = \sum_{j=1}^n i_j w(z_j)$.

Definition 7 (Basis Isolating Weight Assignment)**.** A weight function $w : Z \to \mathbb{N}$ is said to be a *basis isolating weight assignment* for $f_\Psi \in \mathbb{F}^s[Z]$ if there exists a set M of *isolated* monomials in $\mathrm{mons}(Z)$ such that the following conditions hold:

1. $B = \{v_m \mid m \in M\}$ forms a basis for $\mathrm{span}\{v_m \mid m \in \mathrm{mons}(Z)\}$

2. For $m, m' \in M$ such that $m \neq m'$, we have that $w(m) \neq w(m')$

3. For each $m \notin M$, $v_m \in \mathrm{span}\{v_{m'} \mid m' \in M, w(m') < w(m)\}$.

In order to build basis isolating weight assignments, we need an efficient version of the Kronecker map described in [14], [23] that we now state.

Lemma 21 (Efficient Kronecker map, [14])**.** *Let* $Z = \{z_1, \ldots, z_n\}$ *be a set of commutative, associative variables. For each* $k \geqslant 1$*, there is a set* W_k *of* $N \leqslant n\binom{k}{2} \log(d+1)$ *weight functions* $w : Z \to [2N \log N]$ *such that for any set* A *of monomials in* Z *of individual degree* $\leqslant d$ *satisfying* $|A| \leqslant k$*, there exists a* $w \in W_k$ *that **separates** A, that is,* $\forall m \neq m' \in A$*,* $w(m) \neq w(m')$*. Furthermore, the set* W_k *is constructible in polynomial time.*

For convenience, we say that a set W of weight assignments to Z separates a set A of monomials if there exists a $w \in W$ that separates A. In what follows, we will construct a basis isolating weight assignment w for f_Ψ.

Theorem 22. *Let* Ψ *be an unambiguous circuit with* s *gates and product depth* Δ *computing* $f \in \mathbb{F}[Z]$ *of degree* d*. Let* $f_\Psi = \sum_{m \in \mathrm{mons}(Z)} v_m m$ *be as above. Then, we can construct a basis isolating weight assignment* $w : Z \to \mathbb{N}$ *for* f_Ψ *such that* $w(z_i) = (nds)^{O(\Delta)}$*, for each* $i \in [n]$*. Furthermore, we can construct* w *in time* $(nds)^{O(\Delta)}$*.*

Proof. For any monomial m, the *depth* of m (with degree $\geqslant 1$) in Ψ is the depth of T_m, the unique reduced parse tree in Ψ computing the monomial m. Since the product depth of Ψ is at most Δ, we have that for each monomial m, the depth of T_m is also $\leqslant \Delta$. Let $f_\Psi = \sum_{m \in \mathrm{mons}(Z)} v_m m$ where $v_m \in \mathbb{F}^s$ is as defined before. The basis isolating weight assignment w for f_Ψ is obtained by taking a carefully chosen linear combination of $\Delta + 1$ many weight functions. That is, the $\Delta + 1$ many weight functions are chosen and scaled appropriately to obtain the final basis isolating weight function as in Agrawal et al. [23]. Recall that W_k is the set of weight assignments from Lemma 21 that separates every set of at most k monomials of degree $\leqslant d$.

Claim 23. *There exist* $w_2, \ldots, w_{\Delta+1} \in W_{s^2\Delta}$ *such that*

$$w \triangleq \sum_{i=0}^{\Delta} B^{\Delta-i} w_{i+1}$$

is a basis isolating weight assignment for f_Ψ*, where* w_1 *sends each* z_i *to the value* i*, and* $B = 1 + \max\{w_i(m)\}$ *where the* $\max$ *is over all monomials* m *of degree* $\leqslant d$ *in* Z*, and all* i *belonging to* $[\Delta + 1]$*.*

Proof. We construct the weight function w by iteratively constructing $w_1, \ldots, w_\Delta$. For this, we will need the following function:

$$w^j \triangleq \sum_{i=0}^{j-1} B^{j-1-i} w_{i+1} \tag{4}$$

First, we sort the variables in Z in ascending order of their weight with respect to w_1 and pick a basis B_1 for $\mathsf{span}\{v_{z_i} \mid i \in [n]\}$ greedily starting from the lowest weight variable. Let the corresponding set of monomials (variables) be M_1. Clearly, $|M_1| \leqslant s$ and $w^1 = w_1$. Further, by construction of B_1 and M_1 and as $w^1 = w_1$, we have that for every monomial m of depth 1, $v_m \in \mathsf{span}\{v_{m'} \mid m' \in M_1, w^1(m') < w^1(m)\}$. Now, we extend this idea to monomials of depth $j \in \{2, \ldots, \Delta\}$. More precisely, we show the following:

For each $2 \leqslant i \leqslant \Delta$, there exists $w_i \in W_{s^2\Delta}$ such that for each $j = 1, \ldots, \Delta$ there exists a subset M_j of monomials of depth j and a corresponding set $B_j = \{v_m \mid m \in M_j\}$ of coefficients of f_Ψ such that for every monomial m of depth j, $v_m \in \mathsf{span}\{v_{m'} \mid m' \in M_j, w^j(m') < w^j(m)\}$. Furthermore, $|M_j| \leqslant s$ for each $j \in [\Delta]$, and w^j separates M_j.

We will prove this by induction on j. Observe that the base case $j = 1$ has already been discussed above. Suppose we already have weight functions $w_1, \ldots, w_{j-1}$ (with $w_2, \ldots, w_{j-1} \in W_{s^2\Delta}$) satisfying our requirements. Then we know that there exist corresponding sets $M_1, \ldots, M_{j-1}$ and $B_1, \ldots, B_{j-1}$ also satisfying the above mentioned conditions. Define M'_j and B'_j as follows:

$$M'_j \triangleq \bigcup_{k \leqslant j-1} \left\{ m' \cdot m'' \;\middle|\; \begin{array}{l} m' \in M_{j-1}, m'' \in M_k, \mathsf{depth}(T_{m'm''}) = j, \\ T_{m'm''} \text{ decomposes } m' \cdot m'' \text{ as } m' \times m'' \end{array} \right\}$$

$$B'_j \triangleq \{v_m \mid m \in M'_j\}$$

where we say that a parse tree T for a monomial $m = m_1 \cdot m_2$ *decomposes* m as $m_1 \times m_2$ if one of the children of the root of T computes m_1 and the other computes m_2.

Let m be any monomial of depth j with reduced parse tree T_m such that T_m decomposes m as $m = m_1 \times m_2$ with $\mathsf{depth}(m_1) = j - 1$ and $\mathsf{depth}(m_2) = k \leqslant j - 1$. Let us define M_{m_1} and M_{m_2} as follows:

$$M_{m_1} \triangleq \{m' \in M_{j-1} \mid w^{j-1}(m') < w^{j-1}(m_1)\} \subseteq M_{j-1}$$
$$M_{m_2} \triangleq \{m'' \in M_k \mid w^k(m'') < w^k(m_2)\} \subseteq M_k$$

Let us also define the set $M_{m_1 m_2} \subseteq M'_j$ as follows:

$$M_{m_1 m_2} \triangleq \left\{ m' \cdot m'' \;\middle|\; \begin{array}{l} m' \in M_{m_1}, m'' \in M_{m_2}, \mathsf{depth}(T_{m'm''}) = j, \\ T_{m'm''} \text{ decomposes } m' \cdot m'' \text{ as } m' \times m'' \end{array} \right\}$$

We will show that for each monomial m as defined above, we have

$$v_m \in \mathsf{span}\{v_{m'm''} \mid m' \cdot m'' \in M_{m_1 m_2}\} \tag{5}$$

24

The proof of Equation 5 is similar to the proof of Claim 14, although in this case we will need to be careful about the depth of the monomials involved. By the induction hypothesis, we have that $v_{m_1} \in \mathsf{span}\{v_{m'} \mid m' \in M_{m_1}\}$ and $v_{m_2} \in \mathsf{span}\{v_{m''} \mid m'' \in M_{m_2}\}$. That is,

$$v_{m_1} = \sum_{m' \in M_{m_1}} \alpha_{m'} v_{m'} \tag{6}$$

$$v_{m_2} = \sum_{m'' \in M_{m_2}} \beta_{m''} v_{m''} \tag{7}$$

where the $\alpha_{m'}$'s and $\beta_{m''}$'s are scalars in $\mathbb{F}$. In order to prove Equation 5, we will show that for each gate g in Ψ,

$$v_{m_1 m_2}(g) = \sum_{\substack{m' \in M_{m_1}, m'' \in M_{m_2} \\ m'm'' \in M_{m_1 m_2}}} \alpha_{m'} \beta_{m''} v_{m'm''}(g) \tag{8}$$

We do this, as in the proof of Claim 14, by induction on the depth of the gate g in Ψ.

1. If g is a leaf, then both the LHS and RHS in Equation 8 are 0 (since $j \geqslant 2$). Therefore, Equation 8 is true in this case.

2. If $g = g_1 + g_2$ is a sum gate, we have

$$v_{m_1 m_2}(g) = v_{m_1 m_2}(g_1) + v_{m_1 m_2}(g_2)$$

By induction on the depth of the gates g_1, g_2, Equation 8 is true for $v_{m_1 m_2}(g_1)$ and $v_{m_1 m_2}(g_2)$ and therefore it is also true for $v_{m_1 m_2}(g)$.

3. The interesting case is when $g = g_1 \times g_2$ is a product gate. In this case,

$$v_{m_1 m_2}(g) = v_{m_1}(g_1) v_{m_2}(g_2) + v_{m_1}(g_2) v_{m_2}(g_1) + v_{m_1 m_2}(g_1) v_1(g_2) + v_1(g_1) v_{m_1 m_2}(g_2)$$

Substituting (6) for v_{m_1} and (7) for v_{m_2} we get

$$v_{m_1}(g_1) v_{m_2}(g_2) + v_{m_1}(g_2) v_{m_2}(g_1) = \sum_{\substack{m' \in M_{m_1} \\ m'' \in M_{m_2}}} \alpha_{m'} \beta_{m''} (v_{m'}(g_1) v_{m''}(g_2) + v_{m'}(g_2) v_{m''}(g_1))$$

$$\tag{9}$$

The key observation here is that if for some $m' \in M_{m_1}$ and $m'' \in M_{m_2}$ it is the case that $m'm'' \notin M_{m_1 m_2}$, then $v_{m'}(g_1) v_{m''}(g_2) = 0$ and $v_{m'}(g_2) v_{m''}(g_1) = 0$, for otherwise $T_{m'm''}$ it would decompose $m'm''$ as $m' \times m''$ and it would have depth j (because $\mathsf{depth}(T_{m'}) = j - 1$ and $\mathsf{depth}(T_{m''}) = k \leqslant j - 1$).

Therefore, we have that the only surviving terms in the RHS of Equation 9 are those corresponding to the $m' \in M_{m_1}$ and $m'' \in M_{m_2}$ such that $m'm'' \in M_{m_1 m_2}$. Also, by

induction on the depth of the gates g_1 and g_2, Equation 8 is true for g_1 and g_2. Combing these observations, we see that $v_{m_1 m_2}(g)$ is equal to

$$\sum_{\substack{m' \in M_{m_1}, m'' \in M_{m_2} \\ m' m'' \in M_{m_1 m_2}}} \alpha_{m'} \beta_{m''} (v_{m'}(g_1) v_{m''}(g_2) + v_{m'}(g_2) v_{m''}(g_1) + v_m(g_1) v_1(g_2) + v_1(g_1) v_m(g_2))$$

This quantity is exactly equal the RHS of Equation 8.

These three cases together prove Equation 8, and therefore, Equation 5. Note that Equation 5 puts v_m in the span of B'_j by definition of M'_j.

Furthermore, observe the following:

- For any $m'' \in M_{m_2}$, $w^k(m'') < w^k(m_2)$, by the definition of set M_{m_2}.

- For any two monomials $\tilde{m}_1, \tilde{m}_2$ and any $1 \leqslant l \leqslant t \leqslant \Delta$, $w^l(\tilde{m}_1) < w^l(\tilde{m}_2) \implies w^t(\tilde{m}_1) < w^t(\tilde{m}_2)$. This is by the choice of B: $w^t(\tilde{m}_1) = B^{t-l} w^l(\tilde{m}_1) + B^{t-l-1} w_{l+1}(\tilde{m}_1) + \ldots + w_t(\tilde{m}_1) \leqslant B^{t-l} w^l(\tilde{m}_1) + B^{t-l} - 1 < B^{t-l} w^l(\tilde{m}_2) \leqslant w^t(\tilde{m}_2)$.

- Therefore we have that for any $m'' \in M_{m_2}$, $w^{j-1}(m'') < w^{j-1}(m_2)$ (since $k \leqslant j-1$).

From the observations above, we see that for any $m' \in M_{m_1}, m'' \in M_{m_2}$ we have:

$$w^{j-1}(m) = w^j(m_1 \cdot m_2) = w^{j-1}(m_1) + w^{j-1}(m_2) > w^{j-1}(m') + w^{j-1}(m'') = w^{j-1}(m' \cdot m'') \quad (10)$$

Hence, from Equations (5) and (10), for any monomial m of depth j we have that $v_m \in \mathsf{span}\{v_{m'} \mid m' \in M'_j, w^{j-1}(m') < w^{j-1}(m)\}$. To complete the induction step, pick a $w_j \in W_{s^2 \Delta}$ that separates M'_j (this fixes w^j as well). By the way w^j is defined, w^j also separates M'_j. Sort the elements of M'_j (and therefore B'_j) in ascending order of weight with respect to w^j and pick a greedy basis for $\mathsf{span}\{B'_j\}$, going over vectors in B'_j from left to right starting with the lowest weight monomial. Set that basis to be B_j and the corresponding set of monomials to be M_j. Clearly, for all monomials m of depth j, $v_m \in \mathsf{span}\{v_{m'} \mid m' \in M_j, w^j(m') < w^j(m)\}$, and $|M_j| \leqslant s$. Also, w^j separates M_j by construction. This completes the inductive step.

At $j = \Delta$, we will have constructed w^Δ with the following properties:

(a) w^Δ separates M_j for all $1 \leqslant j \leqslant \Delta$. This is because w^j separates M_j and therefore so do all w^k for $j \leqslant k \leqslant \Delta$.

(b) For all $j \in [\Delta]$ and every monomial m of depth j, $v_m \in \mathsf{span}\{v_{m'} \mid w^\Delta(m') < w^\Delta(m)\}$. Again, this is because $w^j(m') < w^j(m)$ implies $w^k(m') < w^k(m)$ for all $j \leqslant k \leqslant \Delta$.

Consider the set $M' = \bigcup_{j=1}^{\Delta} M_j$ (note that $|M'| \leqslant s\Delta$). Pick a $w_{\Delta+1} \in W_{s^2 \Delta}$ that separates M'. Consider the weight assignment $w = w^{\Delta+1} \triangleq Bw^\Delta + w_{\Delta+1}$. It is not hard to see that w is a basis isolating weight assignment for f_Ψ. Sort the monomials in M' in ascending order of weight with respect to $w^{\Delta+1}$, and pick a greedy basis for $\mathsf{span}\{v_m \mid m \in M'\}$, as before. Call this set B and the corresponding set of monomials M. M is the set of monomials *isolated* by $w^{\Delta+1}$:

1. $w^{\Delta+1}$ separates M.

2. Let m be any monomial. Let depth of m be j. Then by observation (b) above, know that $v_m \in \{v_{m'} \mid m' \in M_j, w^{\Delta}(m') < w^{\Delta}(m)\}$. This implies that $v_m \in \{v_{m'} \mid m' \in M_j, w^{\Delta+1}(m') < w^{\Delta+1}(m)\}$. But for each $m' \in M_j \subseteq M'$, $v_{m'} \in \mathsf{span}\{v_{m''} \mid m'' \in M, w^{\Delta+1}(m'') < w^{\Delta+1}(m')\}$. Therefore, for each monomial m, we have $v_m \in \mathsf{span}\{v_{m''} \mid m'' \in M, w^{\Delta+1}(m'') < w^{\Delta+1}(m)\}$.

3. Since the vectors in B are linearly independent, B is in fact a basis for $\{v_m | m \in \mathrm{mons}(Z)\}$.

This finishes the proof of Claim 23. $\qquad\square$

Now, having proved existence, in order to construct the basis isolating weight assignment from Claim 23, simply try all tuples $(w_2, \ldots, w_{\Delta+1})$ in $(W_{s^2\Delta})^{\Delta}$. At least one of them is sure to work. The cost of doing this is $\mathsf{poly}(n, s, d)^{\Delta}$, and the weight assignments arising from these tuples give at most $\mathsf{poly}(n, s, d)^{\Delta}$ weight to any variable. $\qquad\square$

Next, we will show that a basis isolating weight assignment is *useful* for PIT.

Lemma 24. *Let Ψ be a circuit computing $f(z_1, \ldots, z_n) \in \mathbb{F}[Z]$ and $f_{\Psi} = \displaystyle\sum_{m \in mons(Z)} v_m m$ be as defined earlier. Suppose w is a basis isolating weight assignment for f_{Ψ}. Let ϕ be the polynomial map that sends z_i to $t^{w(z_i)}$ where t is a new variable. Then $\phi(f) \not\equiv 0 \iff f \not\equiv 0$.*

Proof. Clearly, $f \equiv 0 \implies \phi(f) \equiv 0$. For the other direction, first notice that $\phi(m) = t^{w(m)}$ for any monomial m. Let $M \subseteq \mathrm{mons}(Z)$ be the set of monomials isolated by w. Let $B = \{v_m \mid m \in M\}$. Let g be the output gate of Ψ which computes f. If $f \not\equiv 0$ then f must contain a monomial from M with non-zero coefficient, because B is a basis for $\mathsf{span}\{v_m \mid m \in \mathrm{mons}(Z)\}$. Let $M' \subseteq M$ be the monomials from M that occur in f with non-zero coefficient. Let $m \triangleq \mathrm{argmin}_{m \in M'}\{w(m)\}$. m is the unique minimizer, since w separates M. Now suppose for $m' \notin M$, we have that $w(m') = w(m)$. We know that $v_{m'} \in \mathsf{span}\{v_{m''} \mid m'' \in M, w(m'') < w(m')\}$. But for all such m'', $v_{m''}(g) = 0$ by minimality of m. Therefore, $v_{m'}(g) = 0$. Therefore, m is the only non-zero monomial in f that receives weight $w(m)$, and so it survives ϕ. $\qquad\square$

PIT for unambiguous circuits of low depth follows easily from Lemmas 22 and 24, just by noting that the degree of the univariate polynomial $\phi(f)$ (as above) is $\mathsf{poly}(n, d, s)^{\Delta}$. To check if a univariate is identically zero, we can query it on $\mathsf{degree} + 1$ points in $\mathbb{F}$ (recall that a non-zero univariate has at most degree many roots). If Δ is polylogarithmic in n, d, s, we get the quasipolyomial running time bound. We record this as our next theorem.

Theorem 5. *There exists a set $H_{s,n,d,\Delta} \subseteq \mathbb{F}^n$ such that for any unambiguous circuit Ψ of size s and product depth Δ computing a non-zero polynomial $f \in \mathbb{F}[z_1, \ldots, z_n]$ of degree $\leqslant d$, f is non-zero on some point of $H_{s,n,d,\Delta}$. Furthermore, $|H_{s,n,d,\Delta}| = (nds)^{O(\Delta)}$ and $H_{s,n,d,\Delta}$ can be constructed in time $(nds)^{O(\Delta)}$.*

Theorems 4 and 15 are a direct consequence of combining the reduction from nonassociative circuits to commutative, associative, unambiguous circuits and the hitting set for unambiguous circuits.

Proof of Theorem 4. To embed the hitting set from Theorem 5 into the algebra $\mathbb{C}_d$, we let $Z = \{z_{i,j,k} \mid i \in [n], j \in [d]\}$ as in Section 3.2. Let $\mathbb{C}_d$ and $\mathbb{C}'$ be the algebras defined in Section 3.2, over the extension field $\mathbb{F}(Z)$. Recall the map $\Phi : \mathbb{F}_{\bar{A},C}[X] \to \mathbb{C}'_d$ as defined in Section 4.2.1. Φ sends x_i to $Z_i \in \mathbb{C}'_d$ for each $i \in [n]$ and each $j, k \in [d]$, the $(j, j + 1, k)$-th entry of Z_i is one and all other entries are zero. We will embed Φ into a map $\Phi' : \mathbb{F}_{\bar{A},C}[X] \to \mathbb{C}_d$ defined as follows: we let Φ' map x_i to $(Z_i, 0)$, $Z_i \in \mathbb{C}'_d$, $c \in \mathbb{F}$ to $(\mathbf{0}, c)$ (where $\mathbf{0} \in \mathbb{C}'_d$) and extend it to all of $\mathbb{C}_d$ by linearity. Note that for a polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$ with constant term c we have that $\Phi'(f) = (\Phi(f - c), c)$.

Let Ψ be a commutative, nonassociative circuit of size s and product depth Δ computing a polynomial $f \in \mathbb{F}_{\bar{A},C}[X]$, with $\mathsf{degree}(f) = d' \leqslant d$. Consider the image $\Phi'(f) = (\Phi(f - c), c)$ of f under Φ', where c is the constant term in f. We know from Lemma 16 that the $(1, d' + 1, 1)$-th entry of $\Phi(f - c)$ is $\phi(f_{d'})$ (see Section 4.2.1 for details). By Lemma 17, $\phi(f_{d'})$ has an unambiguous circuit Ψ' of size $\leqslant 3d^4 s$ and depth $\leqslant \Delta$. Note that $\Phi(f_{d'})$ is a polynomial in the $d^2 n$ many Z variables. Therefore, embedding the hitting set $H_{3d^4 s, d^2 n, d, \Delta}$ on the Z variables into the map Φ' gives us the theorem. $\qquad\square$

The proof of Theorem 15 is also exactly the same as above, except in this case we will obtain a hitting set with elements in $(\mathbb{A}_d)^n$.

5   Discussion

Two interesting question that stem from our work are the following:

1. *Depth reduction for unambiguous circuits:* As briefly mentioned in the introduction, we leave open the question of whether unambiguous circuits can be depth reduced without too much blowup in size. More precisely, suppose we have an unambiguous circuit Ψ of size s, computing a polynomial $f \in \mathbb{F}[z_1, \ldots, z_n]$ of degree d. Does there exist another unambiguous circuit Ψ' computing f, with size quasipolynomial in n, s, d and depth polylogarithmic in n, s, d? A positive answer to this question would imply that the size of the hitting sets from Theorem 4 and Theorem 15 can be improved to quasipolynomial, irrespective of the depth of the circuit. As far as we know, standard depth reduction techniques due to Valiant et al. [8] and Brent [2] do not preserve unambiguity of circuits, even if we allow quasipolynomial blowup in size.

2. *Tightness of the dimension of $\mathbb{C}_d$ and $\mathbb{A}_d$:* Consider the following, purely mathematical question: What is the smallest possible dimension $k(d)$ of a unital algebra that does not satisfy *any* identity $f \in \mathbb{F}_{\bar{A},C}[X]$ of degree $\leqslant d$? We show $k(d) \leqslant d(d + 1)^2 + 1$. One could also ask the analogous question over the algebra $\mathbb{F}_{\bar{A},\bar{C}}[X]$. The Amitsur-Levitzki Theorem (Theorem 7) gives such a tight characterization over $\mathbb{F}\langle X \rangle$, for matrix algebras.

References

[1] A. S. Amitsur and J. Levitzki. "Minimal Identities for Algebras". In: *Proceedings of the American Mathematical Society* 1.4 (1950), pp. 449–463. ISSN: 00029939, 10886826. URL: http://www.jstor.org/stable/2032312.

[2] Richard P. Brent. "The Parallel Evaluation of General Arithmetic Expressions". In: *J. ACM* 21.2 (Apr. 1974), pp. 201–206. ISSN: 0004-5411. DOI: 10.1145/321812.321815. URL: https://doi.org/10.1145/321812.321815.

[3] Leslie G. Valiant. "On Non-Linear Lower Bounds in Computational Complexity". In: *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*. STOC '75. Albuquerque, New Mexico, USA: Association for Computing Machinery, 1975, pp. 45–53. ISBN: 9781450374194. DOI: 10.1145/800116.803752. URL: https://doi.org/10.1145/800116.803752.

[4] Richard A. DeMillo and Richard J. Lipton. "A Probabilistic Remark on Algebraic Program Testing". In: *Inf. Process. Lett.* 7.4 (1978), pp. 193–195. DOI: 10.1016/0020-0190(78)90067-4. URL: https://doi.org/10.1016/0020-0190(78)90067-4.

[5] R. Zippel. "Probabilistic algorithms for sparse polynomials". In: *Proc. of the Int. Sym. on Symbolic and Algebraic Computation*. 1979, pp. 216–226.

[6] Jacob T. Schwartz. "Fast Probabilistic algorithm for verification of polynomial identities". In: *J. ACM.* 27.4 (1980), pp. 701–717.

[7] Walter Baur and Volker Strassen. "The complexity of partial derivatives". In: *Theoretical Computer Science* 22.3 (1983), pp. 317–330. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(83)90110-X. URL: https://www.sciencedirect.com/science/article/pii/030439758390110X.

[8] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. "Fast Parallel Computation of Polynomials Using Few Processors". In: *SIAM Journal on Computing* 12.4 (1983), pp. 641–644.

[9] K. A. Kalorkoti. "A Lower Bound for the Formula Size of Rational Functions". In: *SIAM Journal on Computing* 14.3 (1985), pp. 678–687. DOI: 10.1137/0214050. eprint: https://doi.org/10.1137/0214050. URL: https://doi.org/10.1137/0214050.

[10] Nicolas Bourbaki. *Lie Groups and Lie Algebras*. Springer Science, 1989. ISBN: 9783540642428.

[11] Noam Nisan. "Lower Bounds for Non-Commutative Computation (Extended Abstract)". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*. Ed. by Cris Koutsougeras and Jeffrey Scott Vitter. ACM, 1991, pp. 410–418. DOI: 10.1145/103418.103462. URL: https://doi.org/10.1145/103418.103462.

[12] Nathan Jacobson, Daniel Zelinsky, Richard E. Block, David J. Saltman, and J. Marshall Osborn. *A. Adrian Albert Collected Mathematical Papers*. Vol. 3. American Mathematical Society, 1993. ISBN: 978-0-8218-0003-4.

[13] Avi Wigderson Noam Nisan. "Hardness vs randomness". In: *Journal of Computer and System Sciences* 49.2 (1994), pp. 149–167. ISSN: 0022-0000.

[14] Manindra Agrawal and Somenath Biswas. "Primality and identity testing via Chinese remaindering". In: *J. ACM* 50.4 (July 2003), pp. 429–443. ISSN: 0004-5411. DOI: `10.1145/792538.792540`. URL: `https://doi.org/10.1145/792538.792540`.

[15] Valentine Kabanets and Russell Impagliazzo. "Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds". In: *Comput. Complex.* 13.1-2 (2004), pp. 1–46. DOI: `10.1007/s00037-004-0182-6`. URL: `https://doi.org/10.1007/s00037-004-0182-6`.

[16] R. Raz and A. Shpilka. "Deterministic polynomial identity testing in non commutative models". In: *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.* 2004, pp. 215–222. DOI: `10.1109/CCC.2004.1313845`.

[17] A. Giambruno and Mikhail Zaicev. *Polynomial Identities and Asymptotic Methods.* Vol. 122 of Mathematical surveys and monographs. American Mathematical Soc., 2005. ISBN: 9780821838297.

[18] H. Wee and A. Bogdanov. "More on Noncommutative Polynomial Identity Testing". In: *Proceedings. Twentieth Annual IEEE Conference on Computational Complexity.* Los Alamitos, CA, USA: IEEE Computer Society, June 2005, pp. 92–99. DOI: `10.1109/CCC.2005.13`. URL: `https://doi.ieeecomputersociety.org/10.1109/CCC.2005.13`.

[19] Pavel Hrubes, Avi Wigderson, and Amir Yehudayoff. "Relationless Completeness and Separations". In: *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, USA, June 9-12, 2010.* IEEE Computer Society, 2010, pp. 280–290. DOI: `10.1109/CCC.2010.34`. URL: `https://doi.org/10.1109/CCC.2010.34`.

[20] Amir Shpilka and Amir Yehudayoff. "Arithmetic Circuits: A survey of recent results and open questions". In: *Found. Trends Theor. Comput. Sci.* 5.3-4 (2010), pp. 207–388. DOI: `10.1561/0400000039`. URL: `https://doi.org/10.1561/0400000039`.

[21] Michael A. Forbes and Amir Shpilka. "Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science.* 2013, pp. 243–252. DOI: `10.1109/FOCS.2013.34`.

[22] Nitin Saxena. "Progress on Polynomial Identity Testing - II". In: *CoRR* abs/1401.0976 (2014). arXiv: `1401.0976`. URL: `http://arxiv.org/abs/1401.0976`.

[23] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. "Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits". In: *SIAM Journal on Computing* 44.3 (2015), pp. 669–697. DOI: `10.1137/140975103`. eprint: `https://doi.org/10.1137/140975103`. URL: `https://doi.org/10.1137/140975103`.

[24] V. Arvind and S. Raja. "Some Lower Bound Results for Set-Multilinear Arithmetic Computations". In: *Chicago Journal of Theoretical Computer Science* 2016 (Apr. 2016).

[25] Vikraman Arvind, Rajit Datta, Partha Mukhopadhyay, and S. Raja. "Efficient Identity Testing and Polynomial Factorization in Nonassociative Free Rings". In: *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*. Ed. by Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin. Vol. 83. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 38:1–38:13. DOI: `10.4230/LIPICS.MFCS.2017.38`.

[26] Ramprasad Saptharishi and Anamay Tengse. "Quasipolynomial Hitting Sets for Circuits with Restricted Parse Trees". In: *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2018)*. Ed. by Sumit Ganguly and Paritosh Pandya. Vol. 122. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 6:1–6:19. ISBN: 978-3-95977-093-4. DOI: `10.4230/LIPIcs.FSTTCS.2018.6`. URL: `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.FSTTCS.2018.6`.

[27] Guillaume Lagarde, Nutan Limaye, and Srikanth Srinivasan. "Lower Bounds and PIT for Non-commutative Arithmetic Circuits with Restricted Parse Trees". In: *computational complexity* 28.3 (Sept. 2019), pp. 471–542. ISSN: 1420-8954. DOI: `10.1007/s00037-018-0171-9`. URL: `https://doi.org/10.1007/s00037-018-0171-9`.

[28] Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. "Non-commutative computations: lower bounds and polynomial identity testing". In: *Chicago Journal of Theoretical Computer Science* 2019.2 (Sept. 2019).

[29] Nathanaël Fijalkow, Guillaume Lagarde, Pierre Ohlmann, and Olivier Serre. "Lower Bounds for Arithmetic Circuits via the Hankel Matrix". In: *computational complexity* 30.2 (Oct. 2021), p. 14. ISSN: 1420-8954. DOI: `10.1007/s00037-021-00214-1`. URL: `https://doi.org/10.1007/s00037-021-00214-1`.

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