THEORY OF COMPUTING, Volume 15 (7), 2019, pp. 1–36
www.theoryofcomputing.org

Randomized Polynomial-Time Identity Testing for Noncommutative Circuits

Vikraman Arvind Pushkar S. Joglekar Partha Mukhopadhyay
S. Raja*

Received August 13, 2017; Revised October 4, 2018; Published October 13, 2019

Abstract: In this paper we show that black-box polynomial identity testing (PIT) for n-variate noncommutative polynomials f of degree D with at most t nonzero monomials can be done in randomized $\operatorname{poly}(n, \log t, \log D)$ time, and consequently in randomized $\operatorname{poly}(n, \log t, s)$ time if f is computable by a circuit of size s. This result makes progress on a question that has been open for over a decade. Our algorithm is based on efficiently isolating a monomial using nondeterministic automata.

The above result does not yield an efficient randomized PIT for noncommutative circuits in general, as noncommutative circuits of size s can compute polynomials with a double-exponential (in s) number of monomials. As a first step, we consider a natural class of homogeneous noncommutative circuits, that we call $+$-regular circuits, and give a *white-box* polynomial-time deterministic PIT for them. These circuits can compute noncommutative polynomials with number of monomials double-exponential in the circuit size. Our algorithm combines some new structural results for $+$-regular circuits with known PIT results for noncommutative algebraic branching programs, a rank bound for commutative depth-3

A preliminary version of this paper appeared in the Proceedings of the 49th ACM Symp. on Theory of Computing (STOC'17) [4].

*Part of the work was done while the author was a postdoctoral fellow at Chennai Mathematical Institute, India.

ACM Classification: F.1.2

AMS Classification: 68W20

Key words and phrases: algebraic complexity, non-commutative computation, polynomial identity testing, randomized algorithm, Polynomial Identity Lemma

DOI: 10.4086/toc.2019.v015a007

identities, and an equivalence testing problem for words. Finally, we solve the *black-box* PIT problem for depth-3 $+$-regular circuits in randomized polynomial time. In particular, we show if f is a nonzero noncommutative polynomial in n variables over the field $\mathbb{F}$ computed by a depth-3 $+$-regular circuit of size s, then f cannot be a polynomial identity for the matrix algebra $\mathbb{M}_s(\mathbb{F})$ when $\mathbb{F}$ is sufficiently large depending on the degree of f.

1 Introduction

Noncommutative computation, introduced in complexity theory by Hyafil [12] and Nisan [19], is a central field of algebraic complexity theory. The main algebraic structure of interest is the free noncommutative ring $\mathbb{F}\langle Z\rangle$ over a field $\mathbb{F}$, where $Z = \{z_1, z_2, \ldots, z_n\}$, $z_i, 1 \le i \le n$ are free noncommuting variables.[1]

A fundamental problem in the subject is designing efficient algorithms for noncommutative Polynomial Identity Testing. The problem can be stated as follows:

Let $f \in \mathbb{F}\langle Z\rangle$ be a polynomial represented by a noncommutative arithmetic circuit C. The polynomial f can be either given by a black-box for C (using which we can evaluate the polynomial f on matrices with entries from $\mathbb{F}$ or an extension field), or the circuit C may be explicitly given. The algorithmic problem is to check if the polynomial computed by C is identically zero.

We recall the formal definition of a noncommutative arithmetic circuit.

Definition 1.1. A *noncommutative arithmetic circuit* C over a field $\mathbb{F}$ and indeterminates $z_1, z_2, \ldots, z_n$ is a directed acyclic graph (DAG) with each node of indegree zero labeled by a variable or a scalar constant from $\mathbb{F}$: the indegree 0 nodes are the input nodes of the circuit. Each internal node of the DAG is of indegree two and is labeled by either a $+$ or a $\times$ (indicating that it is a plus gate or multiplication gate, respectively). Furthermore, the two inputs to each $\times$ gate are designated as left and right inputs which is the order in which the gate multiplication is done. A gate of C is designated as *output*. Each internal gate computes a polynomial (by adding or multiplying its input polynomials), where the polynomial computed at an input node is just its label. The *polynomial computed* by the circuit is the polynomial computed at its output gate. An arithmetic circuit is a formula if the fan-out of every gate is at most one.

Notice that if the size of circuit C is s then the degree of the polynomial computed by C can be 2^s.

Bogdanov and Wee [8] have shown a randomized polynomial-time algorithm when the degree of the noncommutative circuit C is polynomially bounded in s and n. Their algorithm is based on a classical result of Amitsur-Levitzki [3] stated below.

Theorem 1.2 (Amitsur-Levitzki Theorem)**.** *For any field $\mathbb{F}$, a nonzero noncommutative polynomial $P \in \mathbb{F}\langle Z\rangle$ of degree $\le 2d-1$ is not a polynomial identity for the matrix algebra $\mathbb{M}_d(\mathbb{F})$. That is to say, P does not vanish on all $d \times d$ matrices over $\mathbb{F}$.*

Remark 1.3. There is a second part to the Amitsur-Levitzki theorem which states that $\mathbb{M}_d(\mathbb{F})$ has degree $2d$ identities. In particular, the *standard polynomial* $\mathbb{S}_{2d}(x_1, \ldots, x_{2d}) = \sum_{\sigma \in S_{2d}} \mathrm{sgn}(\sigma) x_{\sigma(1)} \cdots x_{\sigma(2d)}$, is a minimal identity for $\mathbb{M}_d(\mathbb{F})$.

[1] The z_i are also called indeterminates. We shall be using both terms interchangeably.

Bogdanov and Wee's randomized PIT algorithm [8] applies the above theorem to obtain a randomized PIT as follows: Let $C(z_1, z_2, \dots, z_n)$ be a circuit of syntactic degree bounded by $2d-1$. For each $i \in [n]$, substitute the variable z_i by a $d \times d$ matrix M_i of commuting indeterminates. More precisely, the (ℓ,k)-th entry of M_i is $z^{(i)}_{\ell,k}$ where $1 \le \ell, k \le d$. By Theorem 1.2, the matrix $M_f = f(M_1, M_2, \dots, M_n)$ is not identically zero. Hence, in M_f there is an entry (ℓ', k') which has the *commutative* nonzero polynomial $g_{\ell',k'}$ over the variables $\{z^{(i)}_{\ell,k} : 1 \le i \le n, 1 \le \ell, k \le d\}$. Notice that the degree of the polynomial $g_{\ell',k'}$ is at most $2d-1$. If we do random substitutions from an extension field of $\mathbb{F}$ of size at least $4d$, then we get a randomized polynomial identity testing algorithm, with error probability at most $1/2$, by the *Polynomial Identity Lemma*[2] [20, 26, 18, 9, 27, 28].

The problem with this approach for general noncommutative circuits (whose degree can be 2^s) is that the dimension of the matrices grows linearly with the degree of the polynomial. Therefore, this approach only yields a randomized exponential-time algorithm for the problem. Finding an efficient randomized identity test for general noncommutative circuits is a well-known open problem, as mentioned in a recent workshop on algebraic complexity theory (WACT 2016).

We also recall the definition of noncommutative algebraic branching programs [23].

Definition 1.4. A homogeneous *noncommutative algebraic branching program* ABP is a layered DAG with one in-degree zero source node s, and one out-degree zero sink node t. The vertices are partitioned into layers $0, 1, \dots, d$ (source at layer 0 and sink at layer d). Edges go only from layer i to layer $i+1$ for each $i \le d-1$. Each edge is labeled with a homogeneous linear form in the noncommuting variables $\{x_i \mid 1 \le i \le n\}$. For each s-to-t directed path $\gamma = e_1, e_2, \dots, e_d$, let $f_\gamma = \ell_1 \cdot \ell_2 \cdots \ell_d$ be the product of the linear forms ℓ_i labeling the edges in that order. The ABP computes the homogeneous degree d noncommutative polynomial $f = \sum_\gamma f_\gamma \in \mathbb{F}\langle X \rangle$, where the sum is over all directed paths γ from s to t.

We note that Raz and Shpilka [23] have shown a white-box deterministic polynomial-time PIT for noncommutative ABPs. Forbes-Shpilka [11] and Agrawal et al. [1] have given a quasi-polynomial-time black-box algorithm for small degree noncommutative ABPs.

2 Main results

The main result of the paper is the following theorem that we show about noncommutative identities which is of independent mathematical interest.

Theorem 2.1. *Let $\mathbb{F}$ be a field of size more than $(n+2)d$. Let $f \in \mathbb{F}\langle z_1, z_2, \dots, z_n \rangle$ be a nonzero polynomial of degree d and with t nonzero monomials. Then f cannot be a polynomial identity for the matrix ring $\mathbb{M}_k(\mathbb{F})$ for $k \ge \lceil \log t \rceil + 1$.*

The above theorem yields a randomized PIT for *black-box* noncommutative polynomials. To see this, suppose $f \in \mathbb{F}\langle z_1, z_2, \dots, z_n \rangle$ be a nonzero polynomial of degree d and with t nonzero monomials. We can assume $\mathbb{F}$ is of size more than $(n+2)d$ (if required, we take a suitable extension field). If $f \not\equiv 0$ then, by Theorem 2.1, the polynomial f does not vanish if we substitute for each z_i, $(\log t + 1) \times (\log t + 1)$ matrices

[2] The Polynomial Identity Lemma is widely known as the DeMillo-Lipton-Schwartz-Zippel Lemma. We have given it an alternative name for reasons explained in Section 3.1, where the lemma is also formally stated as Lemma 3.3.

of distinct commuting indeterminates. Indeed, f will evaluate to a nonzero $(\log t+1)\times(\log t+1)$ matrix whose entries are polynomials in commuting variables of degree at most d. For any nonzero entry of this matrix, by applying the Polynomial Identity Lemma (Lemma 3.3) for commutative polynomials, random substitution from $\mathbb{F}$ (or a suitable extension field) for the commuting variables is nonzero with high probability.

Corollary 2.2. *Let $f \in \mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$ be a noncommutative polynomial of degree d and sparsity t given by a black-box, where the black-box can be evaluated on matrices over $\mathbb{F}$ or an extension field. Then there is a randomized algorithm to check whether f is an identically zero polynomial that runs in time* $\mathrm{poly}(\log d, n, \log t)$.

In particular, suppose C is a noncommutative arithmetic circuit, given by black-box access, of size s computing a polynomial in n variables of sparsity t. There is a randomized algorithm that checks if $C \equiv 0$ in time $\mathrm{poly}(s, n, \log t)$.

To second part of the corollary follows because C computes a polynomial of degree bounded by 2^s.

Remark 2.3.

1. It is interesting to compare Theorem 2.1 with the classical Amitsur-Levitski theorem. Our result brings out the importance of the number of monomials in a polynomial identity for $d\times d$ matrices. It implies that any polynomial identity f for $d\times d$ matrices over a field $\mathbb{F}$ of size more than $\deg f$ must have more than 2^{d-1} monomials.

2. We also note that the dimension k of the matrix ring $\mathbb{M}_k(\mathbb{F})$ in Theorem 2.1 is nearly optimal up to a logarithmic factor. In fact, the second part of the Amitsur-Levitzki theorem states that the standard polynomial $\mathbb{S}_{2d}(x_1, \ldots, x_{2d})$, is a minimal identity for $\mathbb{M}_d(\mathbb{F})$. Notice that the number of monomials in the standard polynomial is $2^{O(d\log d)}$.

In general, a noncommutative circuit of size s can compute a polynomial that can have $2^{2^{\Omega(s)}}$ monomials. For example the polynomial $f(x,y)=(x+y)^{2^s}$ has noncommutative circuit of size $O(s)$ but the number of monomials is 2^{2^s}. We consider identity testing for a subclass of homogeneous noncommutative circuits, that we call $+$-*regular circuits*. These are syntactic homogeneous circuits where the $+$ gates can be partitioned into *layers* such that the following holds:

(i) There are no directed paths between the $+$ gates within a layer.

(ii) All $+$ gates in a layer have the same syntactic degree.

(iii) The output gate is a $+$ gate.

(iv) All input to output paths go through exactly one $+$ gate in each layer.

The $+$-*depth* of a $+$-regular circuit is the number of $+$-layers in it.

A couple of remarks about $+$-regular circuits are in order.

Remark 2.4.

- We note that the computational power of noncommutative $+$-regular circuits is quite limited. We can easily adapt Nisan's rank-based argument [19] to show that the noncommutative permanent cannot be computed by polynomial-size $+$-regular circuits. Such a result is not known for general noncommutative circuits.

- However, polynomial-size $+$-regular circuits of $+$-depth 2 can compute polynomials of exponential degree and a double-exponential number of monomials. Such polynomials cannot be computed by bounded-depth noncommutative circuits.

- As is evident from the example $(x+y)^{2^s}$, notice that $+$-regular circuits of size s can compute polynomials of degree 2^s with $2^{2^{\Omega(s)}}$ monomials. Nevertheless, we are able to exploit the circuit's structure to give a deterministic polynomial-time identity testing algorithm for such circuits.

Theorem 2.5. *Let C be a noncommutative $+$-regular circuit of size s given as a white-box computing a polynomial in $\mathbb{F}\langle X\rangle$. There is a deterministic polynomial-time algorithm that tests whether C computes the identically zero polynomial.*

Finally, we give a randomized polynomial identity test for $+$-regular circuits of $+$-depth 2 in the black-box model. We denote such circuits by $\Sigma\Pi^*\Sigma$.

Remark 2.6. We use this notation because the polynomials computed by $\Sigma\Pi^*\Sigma$ are sums of products of linear forms, like the well-studied $\Sigma\Pi\Sigma$ circuits. Our notation also brings out their $+$-regular structure: there are two $+$-layers. The top $+$ gate is the output gate and the bottom $+$-layer consists of gates computing homogeneous linear forms. The Π^* indicates that between the two $+$-layers we can have several $\times$ gates. However, $+$-regularity guarantees that all inputs to the top $+$ gate have the same syntactic degree.

Theorem 2.7. *Let $\mathbb{F}$ be a field of size more than D. Let $f(x_1,\ldots,x_n) \in \mathbb{F}\langle X\rangle$ be a nonzero homogeneous polynomial of degree D computed by a $\Sigma\Pi^*\Sigma$ circuit with top gate fan-in s and the fan-in of the product gates are D. Then f cannot be a polynomial identity for the matrix ring $\mathbb{M}_s(\mathbb{F})$.*

The *black-box* randomized polynomial identity test for $\Sigma\Pi^*\Sigma$ arithmetic circuits is an immediate corollary.

Corollary 2.8. *Let C be a depth-three $+$-regular circuit of size s computing a polynomial $f(x_1,\ldots,x_n) \in \mathbb{F}\langle X\rangle$, where the circuit C is given only by black-box access. Then, there is a randomized algorithm that checks whether f is identically zero, and the algorithm runs in time $\mathrm{poly}(s,n)$.*

2.1 Outline of the proofs

In this section, we give informal description of the proofs for Theorem 2.1, Theorem 2.5, and Theorem 2.7.

Black-box algorithm for noncommutative polynomials of exponential sparsity

We first describe the basic steps required for the proof of Theorem 2.1. Since we are working in the free noncommutative ring $\mathbb{F}\langle z_1,z_2,\ldots,z_n\rangle$, notice that monomials are free words over the alphabet

$\{z_1, z_2, \ldots, z_n\}$, and the polynomial f is an $\mathbb{F}$-linear combination of monomials. Degree d monomials are elements of $\{z_1, z_2, \ldots, z_n\}^d$. Let m be a degree-d monomial. We use the notation $m[i]$ to denote the i-th variable in the monomial. More precisely, if $m = z_{j_1} z_{j_2} \cdots z_{j_{i-1}} z_{j_i} \cdots z_{j_d}$ then $m[i] = z_{j_i}$.

Converting to a bivariate polynomial

It is convenient to convert, by a simple encoding trick, the given noncommutative polynomial into a noncommutative polynomial in $\mathbb{F}\langle x_0, x_1\rangle$, where x_0 and x_1 are two noncommuting variables. Let

$$f = \sum_{i=1}^{t} c_i w_i$$

with $c_i \in \mathbb{F}$, where w_i are monomials in variables $\{z_1, z_2, \ldots, z_n\}$. We encode each variable z_i using the bivariate substitution $\forall i \in [n] : z_i \to x_0 x_1^i x_0$. Thus, each monomial w_i is uniquely encoded as a monomial $\hat{w}_i$ in the two variables $\{x_0 . x_1\}$, where $\hat{w}_i$ is obtained from w_i by applying the bivariate substitution map to each variable. Let the resulting polynomial be $f'(x_0, x_1) \in \mathbb{F}\langle x_0, x_1\rangle$. Since this encoding of monomials is bijective, the following claim clearly holds.

Proposition 2.9. *For any polynomial $f \in \mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$ of degree bounded by d, the corresponding bivariate noncommutative polynomial $f'(x_0, x_1)$, of degree at most $(n+2)d$, is nonzero if and only if f is nonzero. Furthermore, if f is given by black-box access for evaluation on matrices, we can efficiently create from it black-box access for f'.*

The following definition is crucial for the main result in the paper.

Definition 2.10. Let $\mathcal{M} \subseteq \{x_0, x_1\}^D$ be a subset of degree D monomials over noncommuting variables $\{x_0, x_1\}$. A subset of indices $I \subseteq [D]$ is said to be an *isolating index set* for $\mathcal{M}$ if there is a monomial $m \in \mathcal{M}$ such that for each $m' \neq m, m' \in \mathcal{M}$, there is some index $i \in I$ for which $m[i] \neq m'[i]$. In other words, no other monomial in $\mathcal{M}$ agrees with monomial m on all positions in the index set I.

The following lemma says that every subset of monomials $\mathcal{M} \subseteq \{x_0, x_1\}^D$ has an isolating index set of size $\log |\mathcal{M}|$. The proof is a simple halving argument.

Lemma 2.11. *Let $\mathcal{M} \subseteq \{x_0, x_1\}^D$ be a finite set of degree D monomials over variables $\{x_0, x_1\}$. Then $\mathcal{M}$ has an isolating index set of size k which is bounded by* $\log |\mathcal{M}|$.

Proof. The monomials $m \in \mathcal{M}$ are seen as indexed from left to right, where $m[i]$ denotes the variable in the i-th position of m. Let $i_1 \leq D$ be the first index such that not all monomials agree on the i_1-th position. Let

$$\begin{aligned} S_0 &= \{m : m[i_1] = x_0\}, \\ S_1 &= \{m : m[i_1] = x_1\}. \end{aligned}$$

Either $|S_0|$ or $|S_1|$ is of size at most $|\mathcal{M}|/2$. Let S_{b_1} denote that subset, $b_1 \in \{0, 1\}$. We replace the monomial set $\mathcal{M}$ by S_{b_1} and repeat the same argument for at most $\log |\mathcal{M}|$ steps. Clearly, by this process we identify a set of indices $I = \{i_1, \ldots, i_k\}$, $k \leq \log |\mathcal{M}|$ such that the set shrinks to a singleton set $\{m\}$. Clearly, I is an isolating index set as witnessed by the *isolated monomial* m. $\square$

Remark 2.12. Notice that the size of the isolating index set denoted k is bounded by $\log t$ as well as the degree D of the polynomial $f(x_0,x_1)$.

NFA construction

In an earlier paper [6] (for sparse polynomial identity testing) they used a deterministic finite state automaton to isolate a monomial by designing an automaton which accepts a unique monomial. It seems to not work in its current form for the proof of Theorem 2.1 because the number of states that such a deterministic automaton requires is the length of the monomial which could be exponentially large. It turns out that we can use a small *nondeterministic* finite automaton which will guess the isolating index set for the set of nonzero monomials of f. The complication is that there are exponentially many wrong guesses. However, it turns out that if we make our NFA a *substitution automaton*, we can ensure that the monomials computed on different nondeterministic paths (which correspond to different guesses of the isolating index set) all have disjoint support. Once we have this property, it is easy to argue that for the correct nondeterministic path, the computed commutative polynomial is nonvanishing (because the isolated monomial cannot be canceled). With this intuition, we proceed with the simple technical details in Section 4.

White-box algorithm for $+$-regular circuits

Now we informally describe the proof of Theorem 2.5. We note a crucial observation: Let $T(z_1,\ldots,z_s)$ be a homogeneous noncommutative polynomial of degree d. Let $R_1,\ldots,R_s$ be homogeneous noncommutative polynomials each of degree d'. Consider any maximal $\mathbb{F}$-linearly independent subset of the polynomials $R_1,\ldots,R_s$. Let $R_1,\ldots,R_k$ be such a set. We can express $R_j=\sum_{i=1}^{k}\alpha_{ji}R_i$ for $k+1\le j\le s$ where $\alpha_{ji}\in\mathbb{F}$. Then, it turns out that $T(R_1,\ldots,R_k,\sum_{i=1}^{k}\alpha_{k+1i}R_i,\ldots,\sum_{i=1}^{k}\alpha_{si}R_i)=0$ if and only if $T(y_1,\ldots,y_k,\sum_{i=1}^{k}\alpha_{k+1i}y_i,\ldots,\sum_{i=1}^{k}\alpha_{si}y_i)=0$, where $y_1,\ldots,y_k$ are fresh noncommuting variables. As a consequence, it turns out that for a deterministic polynomial-time white-box identity testing for $+$-regular circuits, it suffices to solve the following computational problem:

Let $P_1,\ldots,P_\ell\in\mathbb{F}\langle X\rangle$ be products of homogeneous linear forms given by *multiplicative circuits* of size s. The degrees of the polynomials P_i could be exponential in s. Then find a maximal $\mathbb{F}$-linearly independent subset of the polynomials and express the others as linear combination of the independent polynomials. We solve the above problem in deterministic polynomial time. We prove that it suffices to replace P_i with $\tilde{P}_i$ which is obtained from P_i by retaining, in the product, only linear forms that appear in at most ℓ^5 locations (roughly). This is shown using a rank bound for commutative depth-three identities [25]. We also require algorithms [16, 21, 17] over words to efficiently find the linear forms appearing in those ℓ^5 locations. Since $\tilde{P}_i : 1\le i\le\ell$ are small degree, we are in the usual regime of low-degree noncommutative polynomials, and can adapt the noncommutative ABP identity testing [23] to solve the linear independence testing problem.

Black-box algorithm for depth-three $+$-regular circuits

We now outline the proof of Theorem 2.7. It is similar to the proof of Theorem 2.1. The main idea is the following. Suppose $P_1,P_2,\ldots,P_s$ are D-products of homogeneous linear forms in $\mathbb{F}\langle X\rangle$. Consider any $\mathbb{F}$-linear combination $\sum_{i=1}^{s} \beta_i P_i$ where without loss of generality $\forall i : \beta_i \in \mathbb{F} \setminus \{0\}$. Then there exists locations $I \subseteq [D]$ with $|I| \leq s-1$ with the following property: consider polynomials $P_{i,I}$ obtained from the P_i by treating only the variables appearing in positions in I as noncommutative, and the rest as commutative. Then

$$\sum_{i=1}^{s} \beta_i P_i = 0 \text{ iff } \sum_{i=1}^{s} \beta_i P_{i,I} = 0.$$

Now, we can design small nondeterministic substitution automata that guess the locations in I. The rest of the proof is similar to the proof of Theorem 2.1.

2.2 Organization

The paper is organized as follows. In Section 3, we give some simple properties of noncommutative polynomials. In Section 4 we prove Theorem 2.1. In Section 5 we prove Theorem 2.5. The proof of Theorem 2.7 is in Section 6.

3 Preliminaries

In this section we state a few simple properties of noncommutative polynomials useful for our proofs. Let $A \in \mathbb{F}^{n\times n}$ be an $n \times n$ invertible matrix, and $X = \{x_1, x_2, \ldots, x_n\}$ be n noncommuting variables. We note that homogeneous $\mathbb{F}$-linear forms $\sum_{i=1}^{n} u_i x_i, u_i \in \mathbb{F}^n$ is an n-dimensional vector space over $\mathbb{F}$. We can identify a vector $u \in \mathbb{F}^n$ with the homogeneous linear form $\sum_{i=1}^{n} u_i x_i$. Thus, we can think of A as an invertible linear transform on homogeneous linear forms: it maps x_i to $\sum_{\ell=1}^{n} A_{\ell i} x_\ell$. Let $f(x_1, x_2, \ldots, x_n) \in \mathbb{F}\langle X\rangle$ be a homogeneous degree-d noncommutative polynomial. Let

$$f = \sum_{w \in X^d} f_w \cdot w,$$

where $f_w \in \mathbb{F}^n$ is the coefficient of monomial w in f. Let

$$w = x_{i_1} x_{i_2} \cdots x_{i_d}.$$

We can apply the linear transform A to the j-th position of the monomial w by replacing x_{i_j} with the linear form $\sum_{\ell=1}^{n} A_{\ell i_j} x_\ell$ to obtain the polynomial

$$A_j(w) = x_{i_1} x_{i_2} \cdots x_{i_{j-1}} \cdot \left(\sum_{\ell=1}^{n} A_{\ell i_j} x_\ell \right) \cdots x_{i_d}.$$

By linearity, we define $A_j(f) = \sum_{w \in X^d} f_w \cdot A_j(w)$, and observe the following proposition.

Proposition 3.1. *Let $A : \mathbb{F}^n \to \mathbb{F}^n$ be any invertible linear transformation, and $f(x_1, x_2, \ldots, x_n) \in \mathbb{F}\langle X\rangle$ be any homogeneous polynomial of degree d. Let $A_j(f)$ be the polynomial obtained by applying the transform A to the j-th position of monomials of f, as defined above, for $j \in [d]$. Then $f \neq 0$ if and only if $A_j(f) \neq 0$.*

Proof. If $f \equiv 0$ then clearly $A_j(f) \equiv 0$.

For the other direction, suppose $f \not\equiv 0$. Choose and fix monomials $w_1 \in X^{j-1}$ and $w_2 \in X^{d-j}$ such that

$$W = \{w_1 x_i w_2 \mid x_i \in X, \text{ and } w_1 x_i w_2 \text{ is nonzero in } f\}.$$

By collecting monomials of f with prefix w_1 and suffix w_2 we can write

$$f = w_1 \cdot L \cdot w_2 + \sum_{w \notin W} f_w \cdot w,$$

where $L = \sum_{\ell=1}^{n} u_\ell x_\ell$ is some nonzero linear form. By linearity of A_j we have

$$A_j(f) = A_j(w_1 \cdot L \cdot w_2) + \sum_{w \notin W} f_w \cdot A_j(w).$$

Notice that $A_j(w_1 \cdot L \cdot w_2) = w_1 \cdot L' \cdot w_2$, and $L' = \sum_{\ell=1}^{n} v_\ell x_\ell$ where $v_\ell = \sum_{k=1}^{n} A_{k\ell} u_k$. Now, L' is nonzero because A is an invertible matrix. Hence, $A_j(w_1 \cdot L \cdot w_2)$ is a nonzero polynomial and all its nonzero monomials are in W. On the other hand, $\sum_{w \notin W} f_w \cdot A_j(w)$ has no nonzero monomials in W. It follows that $A_j(f) \not\equiv 0$. □

Given any noncommutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle X \rangle$ of degree d, we can rename the variable $x_i : 1 \le i \le n$ occurring in position $j \in [d]$ (from the left), by a new *commuting* variable x_{ij} and obtain a commutative polynomial g in the variables $\{x_{ij}\}$. The polynomial g is the set-multilinearized polynomial obtained from the noncommutative polynomial f. The following proposition states that this set-multilinearization preserves identities.

Proposition 3.2. *Let $f(x_1, \ldots, x_n) \in \mathbb{F}\langle X \rangle$ be a noncommutative polynomial of degree d. For $1 \le i \le n$, replace the variable x_i occurring in position $j \ : \ 1 \le j \le d$ by a new commuting variable x_{ij}. Let $g(x_{11}, \ldots, x_{1d}, x_{21}, \ldots, x_{n1}, \ldots, x_{nd})$ be the polynomial obtained by this transformation. Then $f = 0$ if and only if $g = 0$.*

Proof. The proof follows easily as the above variable replacement uniquely encodes nonzero monomials of f into nonzero monomials of g. Thus, nonzero monomials in f and in g are in 1-1 correspondence. □

3.1 The Polynomial Identity Lemma

In this section we state the Polynomial Identity Lemma, which is widely known as the DeMillo-Lipton-Schwartz-Zippel Lemma, and attempt to briefly trace its history [20, 26, 18, 9, 27, 28].

Lemma 3.3 (The Polynomial Identity Lemma)**.** *Let $f(x_1, x_2, \ldots, x_n)$ be a nonzero degree-d n-variate polynomial over a field F, and $S \subset F$ be a finite subset. The number of zeros of f in the cartesian product set S^n is bounded by $d \cdot |S|^{n-1}$.*

This basic lemma has played an important role in randomized and algebraic computation. The chronology of this lemma is equally interesting. Until some years ago, it was known as the Schwartz–Zippel Lemma, although the paper by DeMillo and Lipton [9], which also proves the lemma, appeared before the papers by Zippel [28] and Schwartz [27].

However, the lemma can be traced further back. A variant of the Polynomial Identity Lemma, bounding the number of zeros of a nonzero n-variate degree-d polynomial over a finite field $\mathbb{F}_q$ by dq^{n-1}, is attributed to a 1922 article by Øystein Ore [20] in Lidl and Niederreiter's book [14, Theorem 6.13, p. 275]. The book contains the standard proof by induction. This is the version of Lemma 3.3 with $S = \mathbb{F}_q$, but the proof is practically identical. Additionally, we note that W. Schmidt's 1976 monograph [26, Lemma 3A, p. 147] also derives the dq^{n-1} bound using the same inductive argument, although he does not credit anyone for it. Thus, it appears that the lemma was rediscovered multiple times in the last century.

The article by Bishnoi et al. [7], which also mentions Ore's paper, contains a nice section on various versions of this lemma and its connections to the Alon–Füredi theorem [2]. Lipton's blog too [15] has an interesting discussion on the Polynomial Identity Lemma.

Finally, we note that a related result for multilinear polynomials over $\mathbb{F}_2$ was shown by Muller [18], in order to give a lower bound on the distance of Reed-Muller codes [18, 24] for multilinear polynomials over $\mathbb{F}_2$.

In the light of these findings, in the present paper we suggest an alternative name for this fundamental result: we refer to it as the "Polynomial Identity Lemma." It would be more expressive of the lemma's content than using a list of (now perhaps seven) names.

4 Black-box PIT for polynomials of exponential degree and sparsity

In this section we prove Theorem 2.1, which will yield a simple randomized black-box PIT algorithm with polynomial run time for noncommutative polynomials of exponential degree and sparsity.

We first give an intuitive sketch of the proof idea. In essence, our algorithm is a "nondeterministic degree reduction" technique, which transforms the given polynomial f into another polynomial $\hat{f}$ whose *noncommutative degree* is polynomially bounded such that $f \equiv 0$ iff $\hat{f} \equiv 0$. In any monomial of f, we can think of the algorithm as nondeterministically choosing polynomially many positions as special, and replacing variables in other positions by commuting variables. In fact, the noncommutative variables in the polynomially many special positions too are converted to commutative variables using set-multilinearization as in Proposition 3.2.

This transformation is carried out using finite automata. In an earlier paper [6], a deterministic black-box PIT algorithm was obtained for noncommutative sparse polynomials of polynomial degree, using a deterministic finite automaton that *isolates* a unique monomial, in the sense that it accepts a unique nonzero monomial of the input sparse polynomial.

This idea is not directly useful for us. However, small *nondeterministic* finite substitution automata help. We can design a substitution NFA that nondeterministically guesses polynomially many positions at which some nonzero monomial of f is uniquely defined, assuming f has only exponentially many monomials. The variables in these positions are then replaced by set-multilinearized versions (using an extra position index, as in Proposition 3.2), and in other positions the NFA replaces the variables with commuting variables. One complication is that there are exponentially many wrong guesses made by the NFA. However, we can ensure that monomials output on different nondeterministic paths (which correspond to different guesses of the isolating set of positions) have disjoint support. This guarantees that the overall polynomial output by the NFA is of polynomial noncommutative degree, and nonzero because the isolated monomial will not be canceled. We now present the formal details.

Definition 4.1. [5] A finite *nondeterministic substitution automaton* (abbreviated as substitution NFA) is a finite nondeterministic automaton $\mathcal{A}$ along with a substitution map

$$\delta : Q \times X \to \mathcal{P}(Q \times (Y \cup \mathbb{F})),$$

where Q is the set of states of $\mathcal{A}$, Y is a set of variables and for a set S, and $\mathcal{P}(S)$ is the power set of S. For $u \in Y \cup \mathbb{F}$, if $(j,u) \in \delta(i,x)$, it means that when the automaton $\mathcal{A}$ is in state i it can make transition to the state j on reading variable x and replacing x by u. In our construction, Y is a set of commuting variables.

Remark 4.2. In fact, the nondeterministic substitution automaton we will design is more restrictive. It has the following behavior: for $i \in Q$ and $x \in X$ if $(j,u),(j,v) \in \delta(i,x)$ then $u = v$. In other words, the substitution made for variable x on transition from state i to state j is uniquely determined by the state j. To emphasize this dependence on j we denote the substitution by u_j. Now, for $x \in X$ we have the $|Q| \times |Q|$ transition matrix M'_x:

$$M'_x(i,j) = u_j, 1 \le i,j \le |Q|, \text{ if } (j,u_j) \in \delta(i,x). \tag{4.1}$$

The substitution map δ is naturally extended to $\hat{\delta} : Q \times X^* \to \mathcal{P}(Q \times (Y \cup \mathbb{F})^*)$. For a state $j \in Q$ and $w' \in (Y \cup \mathbb{F})^*$, $(j,w') \in \hat{\delta}(i,w)$ means that the automaton starting state i, on input string $w \in X^*$ can nondeterministically move to state j by transforming the input string w to w' on some computation path.

The output of a substitution NFA on a polynomial $f \in \mathbb{F}\langle X \rangle$

We first explain the output of a substitution NFA $\mathcal{A}$ on a degree D monomial $w \in X^D$. Let s be a designated initial state and t a designated final state of $\mathcal{A}$. As defined in Equation (4.1), for each $x_i \in X$ we have a $|Q| \times |Q|$ transition matrix M'_x. Let

$$w = x_{i_1} x_{i_2} \cdots x_{i_D},$$

and define the matrix M'_w as

$$M'_w = M'_{x_{i_1}} M'_{x_{i_2}} \cdots M'_{x_{i_D}}.$$

The *output* of the substitution NFA $\mathcal{A}$ on monomial w as input is defined as the (s,t)-th entry of the matrix M'_w, which is a polynomial in the variable set Y.

Remark 4.3. We can also think of the (s,t)-th entry of the matrix product M'_w as the output of an algebraic branching program of width $|Q|$ and D layers, computing a polynomial in variable set Y.

For $f \in \mathbb{F}\langle X \rangle$, the output of NFA $\mathcal{A}$ is defined as the (s,t)-th entry of the matrix $f(M'_{x_1}, M'_{x_2}, \ldots, M'_{x_n})$ obtained by substituting matrix M'_{x_i} for $x_i, 1 \le i \le n$ in the polynomial f. The output is clearly a polynomial in variables Y.

Alternatively, writing $f \in \mathbb{F}\langle X \rangle$ as

$$f = \sum_w f_w \cdot w, \quad f_w \in \mathbb{F},$$

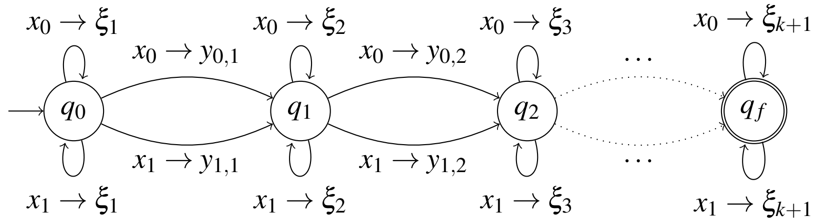

Figure 1: The transition diagram of the automaton

note that the (s,t)-th entry of the $|Q| \times |Q|$ matrix $f(M'_{x_1}, M'_{x_2}, \ldots, M'_{x_n})$ is the polynomial

$$g = \sum_{w \in W_t} f_w \cdot \left(\sum_{w' \in W'_t} w' \right),$$

where $W'_t = \{w' \mid (t, w') \in \hat{\delta}(s, w)\}$. The inner sum is over all $w' \in W'_t$, which are monomials output by the substitution NFA $\mathcal{A}$ along different nondeterministic computation paths from state s to state t.

Now we describe the construction of a substitution NFA $\mathcal{A}$ that substitutes, on its transition edges, new commuting/noncommuting variables for the variables (x_0 or x_1) that it reads. Let $Q = \{q_0, q_1, q_2, \ldots, q_k\}$ be the states of automaton $\mathcal{A}$ and q_0 be the initial state and $q_k = q_f$ be the final state.

We use the indices $i_1, \ldots, i_k$ from Lemma 2.11 to define the transition of A. The set of indices partition each monomial m into $k+1$ blocks as follows.

$$m[1, i_1 - 1]m[i_1]m[i_1 + 1, i_2 - 1]m[i_2] \cdot \cdots \cdot m[i_k]m[i_{k+1}, D],$$

where $m[i]$ denotes the variable in i-th position of m and $m[i, j]$ denotes the submonomial of m from positions i to j.

We define two new sets of variables. The *block variables* are a set of $k+1$ commuting variables $\{\xi_1, \xi_2, \ldots, \xi_{k+1}\}$. There are $2k$ many commuting *index variables* $\bigcup_{j \in [k]} \{y_{0,j}, y_{1,j}\}$.

Now we describe the transitions of the substitution NFA $\mathcal{A}$. While $\mathcal{A}$ is reading input variables in block j, it replaces each $x_b, b \in \{0,1\}$ read by the block variable ξ_j. It *nondeterministically* decides if block j is over and the variable seen in the current position is an index in the isolating set. If so, $\mathcal{A}$ replaces the input variable x_b by the index variable $y_{b,j}$, and it increments the block number to $j+1$. Now $\mathcal{A}$ will make its transitions in the $(j+1)^{st}$ block as described above. The substitution map for $\mathcal{A}$ is given below.

For $0 \le i \le k-1$ and $b \in \{0,1\}$:

$$\begin{aligned} \delta(q_i, x_b) &= \{(q_i, \xi_{i+1}), (q_{i+1}, y_{b,i+1})\}, \\ \delta(q_k, x_b) &= \{(q_k, \xi_{k+1})\}. \end{aligned}$$

Figure 1 depicts the automaton. The substitution NFA $\mathcal{A}$ is described by two $(k+1) \times (k+1)$ transition matrices M_{x_0} and M_{x_1} as already explained.

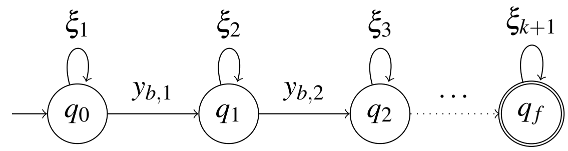

Figure 2: Transition diagram for the variable x_b for $b \in \{0,1\}$

More precisely, for variable x_b, we take the adjacency matrix M_{x_b} of the labeled directed graph in Figure 2, extracted from the automaton in Figure 1.

The corresponding matrix M_{x_b} of dimension $(k+1)\times(k+1)$, we substitute for x_b is the following.

$$\mathbf{M_{x_b}} = \begin{pmatrix} \xi_1 & y_{b,1} & 0 & \dots & 0 & 0 \\ 0 & \xi_2 & y_{b,2} & \dots & 0 & 0 \\ 0 & 0 & \xi_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \xi_k & y_{b,k} \\ 0 & 0 & 0 & \dots & 0 & \xi_{k+1} \end{pmatrix}$$

The rows and the columns of the matrices $M_{x_b}, b \in \{0,1\}$ are indexed by the states of the automaton and the entries are either block variables or index variables as indicated in the transition diagram. Let

$$f(x_0,x_1) = \sum_{i=1}^{t} c_i w_i.$$

Recall that $f(x_0,x_1)$ is a polynomial with sparsity t, degree D and w_i represents a monomial with coefficient c_i for $i \in [t]$. Define the matrix product $w_i(M_{x_0},M_{x_1})$ obtained by substituting the matrix M_{x_b} for x_b, $b \in \{0,1\}$ in the monomial w_i, and multiplying these matrices. The following proposition is immediate as f is a linear combination of the w_i.

Proposition 4.4. $M_f = f(M_{x_0},M_{x_1}) = \sum_{i=1}^{t} c_i w_i(M_{x_0},M_{x_1})$.

Now we are ready to prove Theorem 2.1.

Proof of Theorem 2.1. By Proposition 2.9, we can assume that the input polynomial (given by black-box) is a bivariate polynomial $f(x_0,x_1)$ over noncommuting variables x_0 and x_1 of sparsity t. Let $\mathcal{M} \neq \emptyset$ denote the set of nonzero monomials of degree D occurring in f, where $D = \deg(f)$. Then we can write the polynomial $f = \sum_{j=1}^{t} c_j w_j$ in two parts

$$f = \sum_{w_j \in \mathcal{M}} c_j w_j + \sum_{w_j \notin \mathcal{M}} c_j w_j,$$

where $\sum_{w_j \in \mathcal{M}} c_j w_j$ is the homogeneous degree D part of f.

Let us assume, without loss of generality, that w_1 is in $\mathcal{M}$ and it is the monomial isolated in Lemma 2.11. Let the isolating index set be $I = \{i_1, i_2, \dots, i_k\}$, which means that for all $w_j \in \mathcal{M}$, $w_j|_I \neq w_1|_I$ (i. e., the projections of each $w_j, \forall j \neq 1$ on index set I differs from the projection of w_1). Let

$$w_1 = x_{b_1} x_{b_2} \cdots x_{b_D},$$

where $b_j \in \{0,1\}$.

We will now analyze the polynomial computed at the (q_0, q_f)-th entry of the matrix $f(M_{x_0}, M_{x_1})$.

Firstly, notice from the definition of the substitution NFA $\mathcal{A}$ that the only nondeterminism is in picking the index set. Therefore, an index set $J = \{j_1, j_2, \dots, j_k\}$ nondeterministically picked by $\mathcal{A}$ determines a unique computation path for it. Let $w_j \in \mathcal{M}$ be a nonzero degree-D monomial of f. It is transformed by $\mathcal{A}$ into a degree-D commutative monomial $w_{j,J}$ (which is over the block and index variables) as follows. Let

$$\begin{aligned} \xi_J &= \xi_1^{j_1 - 1} \xi_2^{j_2 - j_1 - 1} \cdots \xi_{k+1}^{D - j_k}, \\ y_{j,J} &= y_{a_1,1} y_{a_2,2} \cdots y_{a_k,k},\ a_i \in \{0,1\}, \end{aligned}$$

where $a_i = b$ if j_i-th variable of w_j is x_b. Then we have the following claim from the construction of $\mathcal{A}$.

Claim 4.5. *For the index set $J = \{j_1, j_2, \dots, j_k\}$ nondeterministically chosen by the substitution NFA $\mathcal{A}$, the monomial output by $\mathcal{A}$ on input w_j is*

$$w_{j,J} = \xi_J \cdot y_{j,J}.$$

Notice that for two distinct index sets J and J' we clearly have

$$\xi_J \neq \xi_{J'}.$$

To see this, let j_ℓ be the first index where J and J' are different. Then the power of ξ_ℓ will be different in ξ_J and $\xi_{J'}$. We thus have:

Claim 4.6. *For any two index sets J, J' and any monomial $w_j \in \mathcal{M}$, the corresponding commutative monomials $w_{j,J}$ and $w_{j,J'}$ are distinct.*

We also note that the degree-k monomial $y_{j,J}$ essentially encodes the projection of the degree-D monomial w_j to the k indices in $J = \{j_1, j_2, \dots, j_k\}$. If variable x_b occurs in position j_i of monomial w_j it is encoded as variable $y_{b,i}$ in monomial $y_{j,J}$.

Claim 4.7. *For each $w_j \in \mathcal{M}$ we note that the (q_0, q_f)-th entry of the matrix $w_j(M_{x_0}, M_{x_1})$ is the sum*

$$\sum_J w_{j,J} = \sum_J \xi_J y_{j,J}$$

over all nondeterministically picked size-k index sets $J \subset [D]$.

To see the above claim we note that each computation path of the NFA $\mathcal{A}$ is determined by an index set J along which the monomial $\xi_J y_{j,J}$ is output by $\mathcal{A}$. The matrix product $w_j(M_{x_0}, M_{x_1})$ sums up the monomials produced over all the paths.

Now, let f_J be the polynomial

$$f_J = \sum_{j=1}^{t} c_j w_{j,J} = \sum_{w_j \in \mathcal{M}} c_j w_{j,J} + \sum_{w_j \notin \mathcal{M}} c_j w_{j,J}.$$

Claim 4.8. *After the matrix substitution $x_0 = M_{x_0}$ and $x_1 = M_{x_1}$ in the polynomial f we note that the (q_0, q_f)-th entry of the matrix $f(M_{x_0}, M_{x_1})$ is $\sum_J f_J$.*

To see the above claim we need to change the order of summation in computing the (q_0, q_f)-th entry of the matrix $f(M_{x_0}, M_{x_1})$. Notice that for a fixed index set J guessed by the NFA $\mathcal{A}$, for each monomial w_j the NFA outputs $w_{j,J}$. Hence, by linearity, fixing the guessed index set J, the NFA computes f_J on input f. Summing over all guessed index sets J, it follows that (q_0, q_f)-th entry of the matrix $f(M_{x_0}, M_{x_1})$ is $\sum_J f_J$.

Finally, we focus on the monomial $w_{1,I}$ occurring in the polynomial $\sum_J f_J$, where w_1 is the isolated monomial and I is the isolated index set.

Claim 4.9. *The coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$ is c_1. As a consequence, the (q_0, q_f) entry of the matrix $M_f = f(M_{x_0}, M_{x_1})$ which computes $\sum_J f_J$ is nonzero.*

To see the above claim we note the following:

1. For a nonzero monomial $w_j \notin \mathcal{M}$ of f, for each index set J of size k the contribution to the (q_0, q_f)-th entry of the matrix $f(M_{x_0}, M_{x_1})$ is a monomial of degree $\deg(w_j)$, and $\deg(w_j) < D$. Hence, these monomials have no influence on the coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$.

2. Consider monomials $w_j \in \mathcal{M}$. Notice that $w_{1,I} = \xi_I y_{1,I}$ and for $j \neq 1$ $w_{j,I} = \xi_I y_{j,I}$. Now,

$$y_{1,I} \neq y_{j,I}, \text{ for all } w_j \in \mathcal{M} \setminus \{w_1\},$$

because I is an isolating index set for $\mathcal{M}$ and the monomial w_1 is isolated. Therefore, $w_{1,I} \neq w_{j,I}$ for all $w_j \in \mathcal{M} \setminus \{w_1\}$.

It follows that the monomials $w_j \in \mathcal{M} \setminus \{w_1\}$ also have no influence on the coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$.

Hence, we conclude that the (q_0, q_f)-th entry of the matrix $M_f = f(M_{x_0}, M_{x_1})$ is a nonzero polynomial $\sum_J f_J$ in the commuting variables $\bigcup_{j \in [k+1]} \{\xi_j\}$ and $\bigcup_{j \in [k]} \{y_{0,j}, y_{1,j}\}$. Moreover, the degree of polynomial $\sum_J f_J$ is D. By the Polynomial Identity Lemma 3.3 it follows that the polynomial $\sum_J f_J$ is nonzero in a suitable extension of size more than $(n+2)d$ of the field $\mathbb{F}$.

Since the polynomial f is nonzero over the algebra $\mathbb{M}_{k+1}(\mathbb{F})$, it is also nonzero over the algebra $\mathbb{M}_{\log t+1}(\mathbb{F})$. This completes the proof of Theorem 2.1. □

5 A deterministic PIT for regular circuits

In this section we consider polynomial identity testing for noncommutative $+$-regular circuits. As mentioned earlier, these circuits can compute polynomials of exponential degree and a double-exponential number of monomials. However, we exploit the circuit structure and obtain a white-box deterministic polynomial-time identity test for $+$-regular circuits (Theorem 2.5).

Definition 5.1. A noncommutative circuit C, computing a polynomial in $\mathbb{F}\langle X\rangle$ where $X = \{x_1, x_2, \ldots, x_n\}$, is $+$-*regular* if it satisfy the following properties:

- The circuit is homogeneous. The gates are of unbounded fanin.
- The $+$ gates in the circuit are partitioned into layers (termed $+$-layers) such that if g_1 and g_2 are $+$ gates in a $+$-layer then there is no directed path in the circuit between g_1 and g_2.
- All gates in a $+$-layer are of the same syntactic degree.
- The output gate is a $+$ gate.
- Every input-to-output path in the circuit goes through a gate in each $+$-layer.
- Additionally, we allow scalar edge labels in the circuit. For example, suppose g is a $+$ gate in C whose inputs are gates $g_1, g_2, \ldots, g_t$ such that $\beta_i \in \mathbb{F}$ labels edge $(g_i, g), i \in [t]$. If polynomial P_i is computed at gate $g_i, i \in [t]$, then g computes the polynomial $\sum_{i=1}^{t} \beta_i P_i$.

Remark 5.2. We draw the reader's attention to the crucial fifth condition in Definition 5.1. An important consequence of the condition is that each $+$-layer (other than the output and input) is a separating set for the underlying DAG of the circuit C. This property allows us to do a deterministic depth reduction of such circuits, preserving polynomial identity.

We list below some observations about $+$-regular circuits, and also introduce some notation in the process.

1. In a $+$-regular circuit, for any $+$ gate g of degree more than 1, we can assume that all inputs to g are $\times$-gates. This can be effected with a simple transformation to the circuit without increasing its size.
2. The $+$-*depth* of a $+$-regular circuit C is the number of $+$-layers in it. Let d be the $+$-depth of C. We number the $+$-layers of C from bottom upward. For $1 \leq i \leq d$, let $\mathcal{L}_i^+$ denote the set of $+$ gates in the i-th $+$-layer, and let $\mathcal{L}_i^\times$ denote the set of $\times$-gates which are inputs to gates in $\mathcal{L}_i^+$. Notice that all gates in $\mathcal{L}_i^\times$ and $\mathcal{L}_i^+$ have the same syntactic degree D_i. The degree of the circuit C is D_d.
3. Continuing from above, we can assume that all $+$ gates of degree D_i in the circuit are in layer $\mathcal{L}_i^+$. Similarly, all $\times$-gates of degree D_i in the circuit are in $\mathcal{L}_i^\times$.
4. It is clear from the structure of circuit C that the part of C between $+$-layers $\mathcal{L}_i^+$ and $\mathcal{L}_{i+1}^+$ is a multiplicative circuit whose inputs are the gates in layer $\mathcal{L}_i^+$ and outputs are the gates in layer $\mathcal{L}_{i+1}^\times$, $1 \leq i \leq d-1$. We note that this property follows from the fifth condition in Definition 5.1.
5. Finally, consider the bottommost $+$-layer $\mathcal{L}_1^+$. We assume without loss of generality that $\mathcal{L}_1^\times$ are the input variables, and the gates in $\mathcal{L}_1^+$ compute homogeneous linear forms. We can ensure this by adding a dummy $+$-layer as the first layer 1.

Remark 5.3. The term "regular formulas" is also used in a somewhat different sense by Kayal et al. [13]. We note that their model of regular formulas is much more restricted than the $+$-regular circuits considered here.

In our study of $+$-regular circuits, it is important to consider certain special $+$-regular circuits of $+$-depth 2. We term these as homogeneous $\Sigma\Pi^*\Sigma$ circuits (as already mentioned in Section 2), in analogy with the usual $\Sigma\Pi\Sigma$ arithmetic circuits (see, e. g., [25]).

Definition 5.4. $\Sigma\Pi^*\Sigma$ circuit are special $+$-regular circuits of $+$-depth 2 defined as follows:

- The output gate is a $+$ gate (which constitutes the second $+$-layer).
- All inputs to the output gate are $\times$ gates of the same syntactic degree.
- The gates in the first $+$-layer compute homogeneous linear forms (taking as input the circuit inputs).

We will require the following definition of multiplicative circuits.

Definition 5.5 (Multiplicative Circuit)**.** Let $\Gamma = \{y_1, y_2, \ldots, y_n\}$ be a finite alphabet. A *multiplicative circuit* over the free monoid $(\Gamma^*, .)$ is a directed acyclic graph with each indegree zero nodes labeled by letters from Γ (the inputs to the circuit). Internal nodes are concatenation gates of fanin two: its two inputs are designated left and right child, and the gate concatenates the output of the left child with the output of the right child. A gate of the circuit is designated the *output* gate.

Remark 5.6.

- Multiplicative circuits computing words as described above are also known as a compressed context-free grammars, see, e. g., [16].
- If we let Γ be the set $\{y_1, y_2, \ldots, y_n\}$ of noncommuting variables, then a multiplicative circuit can be seen as a special kind of noncommutative arithmetic circuit that computes a single monomial in this variable set. The internal concatenation gates of the circuit are $\times$ gates.
- We sometimes allow a multiplicative circuit C to have multiple output gates. Thus, at each output gate g, the output $C_g(y_1, y_2, \ldots, y_n)$ computed by C is a monomial (i. e., a word) in variables $y_1, y_2, \ldots, y_n$.
- We can define $\Pi^*\Sigma$ circuit as multiplicative circuits with homogeneous linear forms as input. More precisely, let $C(y_1, y_2, \ldots, y_n)$ be a multiplicative circuit with output gate g, and $L_1, L_2, \ldots, L_n$ be homogeneous linear forms in noncommuting variables $x_1, x_2, \ldots, x_n$. Then $C_g(L_1, L_2, \ldots, L_n)$ is a $\Pi^*\Sigma$ circuit.

The following theorem is crucial to our PIT algorithm for $+$-regular circuits. It will allow us to decompose $+$-regular circuits of $+$-depth d into $+$-regular circuits of smaller $+$-depth, preserving homogeneous polynomial identities.

Theorem 5.7. *Let $T(z_1, z_2, \ldots, z_s)$ be a noncommutative homogeneous degree-d polynomial over a field $\mathbb{F}$ in noncommuting variables $z_1, z_2, \ldots, z_s$. Let $R_1, R_2, \ldots, R_s$ be noncommutative homogeneous degree d' polynomials in variables $x_1, x_2, \ldots, x_n$ over $\mathbb{F}$ such that $\{R_1, R_2, \ldots, R_k\}$ are a maximal linearly independent subset of $\{R_1, R_2, \ldots, R_s\}$ over $\mathbb{F}$, where*

$$R_j = \sum_{i=1}^{k} \alpha_{ji} R_i, \quad k+1 \le j \le s, \quad \alpha_{ji} \in \mathbb{F}.$$

For fresh noncommuting variables $y_1, y_2, \ldots, y_k$ define linear forms

$$\ell_j = \sum_{i=1}^{k} \alpha_{ji} y_i, \quad k+1 \le j \le s.$$

Then $T(R_1, R_2, \ldots, R_s) \equiv 0$ if and only if $T(y_1, y_2, \ldots, y_k, \ell_{k+1}, \ldots, \ell_s) \equiv 0$.

Proof. The reverse implication is immediate. For, suppose $T(y_1, y_2, \ldots, y_k, \ell_{k+1}, \ldots, \ell_s) \equiv 0$. Then, by substituting R_i for $y_i, 1 \le i \le k$ we obtain $T(R_1, R_2, \ldots, R_s) \equiv 0$.

We now show the forward implication. Suppose $T(R_1, R_2, \ldots, R_s) \equiv 0$. As $R_1, R_2, \ldots, R_k$ are linearly independent over $\mathbb{F}$ we can find degree-d' monomials $m_1, m_2, \ldots, m_k$ such that the $k \times k$ matrix B of their coefficients is of full rank. More precisely, if β_{ji} is the coefficient of m_i in R_j then the matrix

$$B = (\beta_{ji})_{1 \le j, i \le k}$$

is full rank.

Define polynomials

$$R'_j = \sum_{i=1}^{k} \beta_{ji} m_i, 1 \le j \le k, \tag{5.1}$$

$$R'_j = \sum_{i=1}^{k} \alpha_{ji} R'_i, k+1 \le j \le s. \tag{5.2}$$

Notice that $T(R_1, R_2, \ldots, R_s) \equiv 0$ implies $T(R'_1, R'_2, \ldots, R'_s) \equiv 0$. This is because every nonzero monomial occurring in $T(R'_1, R'_2, \ldots, R'_s)$ precisely consists of all monomials from the set $\{m_1, m_2, \ldots, m_k\}^d$ occurring in $T(R_1, R_2, \ldots, R_s)$ (with the same coefficient).

Replacing m_i by variable $y_i, 1 \le i \le k$ transforms each R'_j to linear forms

$$\ell'_j = \sum_{i=1}^{k} \beta_{ji} y_i, \text{ for } 1 \le j \le k,$$

and

$$\ell'_j = \sum_{i=1}^{k} \alpha_{ji} \sum_{q=1}^{k} \beta_{iq} y_q, \text{ for } k+1 \le j \le s.$$

Note that the coefficient of any monomial $y_{i_1} y_{i_2} \ldots y_{i_d}$ in $T(\ell'_1, \ell'_2, \ldots, \ell'_k, \ell'_{k+1}, \ldots, \ell'_s)$ is the same as the coefficient of the corresponding monomial $m_{i_1} m_{i_2} \ldots m_{i_d}$ in $T(R'_1, R'_2, \ldots, R'_s)$ which is zero. Hence $T(\ell'_1, \ell'_2, \ldots, \ell'_k, \ell'_{k+1}, \ldots, \ell'_s) \equiv 0$. Now, since B is invertible, we can apply the linear map B^{-1} to each of the d positions in the polynomial $T(\ell'_1, \ell'_2, \ldots, \ell'_k, \ell'_{k+1}, \ldots, \ell'_s)$ and obtain $T(y_1, y_2, \ldots, y_k, \ell_{k+1}, \ldots, \ell_s)$, which must be identically zero by Proposition 3.1. This completes the proof. $\square$

Outline of the algorithm

In the following lemma we will apply Theorem 5.7 to decompose $+$-regular circuits. Using this we will be able to design a deterministic polynomial-time PIT algorithm for $+$-regular circuits.

Lemma 5.8. *Suppose C is a $+$-regular circuit of size s, syntactic degree D, and $+$-depth d, computing a polynomial in $\mathbb{F}\langle X\rangle$. For $i \in [d]$, let $\mathcal{L}_2^{\times} = \{g_1, g_2, \ldots, g_m\}$. Suppose, $P_1, P_2, \ldots, P_m$ are the polynomials computed at the gates $g_1, g_2, \ldots, g_m$, respectively. Assume, without loss of generality, that $\{P_1, P_2 \ldots, P_t\}$ is a maximal $\mathbb{F}$-linearly independent subset of $\{P_1, P_2, \ldots, P_m\}$, and*

$$P_j = \sum_{i=1}^{t} \alpha_{ji} P_i, \quad t+1 \leq j \leq m.$$

Let C' be the circuit obtained from C as follows:

(i) Delete all gates below $\mathcal{L}_2^{\times}$.

(ii) Replace $g_1, g_2, \ldots, g_m$ by input variables $y_1, y_2, \ldots, y_m$, respectively.

Then $C'(y_1, y_2, \ldots, y_m)$ is a $+$-regular circuit of $+$-depth $d-1$ and size at most s, computing a homogeneous polynomial in $y_1, y_2, \ldots, y_m$. Let C'' be the circuit obtained from C' by replacing y_j by the linear form $\sum_{i=1}^{t} \alpha_{ji} y_i$, for each $t+1 \leq j \leq m$. Then the circuit C is identically zero if and only if the circuit C'' is identically zero.

Proof. The proof is an easy consequence of Theorem 5.7. Let $P(x_1, x_2, \ldots, x_n) \in \mathbb{F}\langle X\rangle$ be the homogeneous degree D polynomial computed by C, where $X = \{x_1, x_2, \ldots, x_n\}$. Let $T(y_1, y_2, \ldots, y_m)$ be the polynomial computed by the circuit C'. Let $P_1, P_2, \ldots, P_m$ be the polynomials computed by the gates $g_1, g_2, \ldots, g_m$, respectively. Since C' is also a $+$-regular circuit, T is homogeneous of degree,say, D'. Each P_i is homogeneous of degree $D'' = D/D'$. It follows from the structure of C that

$$P(x_1, x_2, \ldots, x_n) = T(P_1, P_2, \ldots, P_m).$$

Now, applying Theorem 5.7 to the polynomials $P, T, P_1, P_2, \ldots, P_m$, it follows that the circuit C is identically zero if and only if the circuit

$$C'(y_1, y_2, \ldots, y_t, \sum_{i=1}^{t} \alpha_{i+1i} y_i, \ldots, \sum_{i=1}^{t} \alpha_{mi} y_i) \equiv 0. \qquad \square$$

Given a size s $+$-regular circuit C of $+$-depth d, we can apply Lemma 5.8 to transform C to another $+$-regular circuit C' of size s and $+$-depth $d-1$ such that

$$C \equiv 0 \text{ if and only if } C' \equiv 0.$$

The pseudo-code for this procedure **Prune-plus-depth** is given in Algorithm A. This procedure runs in polynomial time assuming that we have a deterministic polynomial-time subroutine **Basis** that computes a maximum $\mathbb{F}$-linearly independent subset of a set of polynomials $\{P_1, P_2, \ldots, P_m\} \subset \mathbb{F}\langle X\rangle$, where each P_i is given by a $\Pi^*\Sigma$ circuit of size at most s computing it. Without loss of generality, suppose

$\{P_1, P_2, \ldots, P_t\}$ is the maximum linearly independent subset computed by **Basis**. We will assume that the subroutine **Basis** will also compute scalars $\alpha_{ji} \in \mathbb{F}$ such that

$$P_j = \sum_{i=1}^{t} \alpha_{ji} P_i, \quad t+1 \leq j \leq m.$$

In the next subsection we will describe subroutine **Basis**, which will complete the overall algorithm.

Prune-plus-depth

Input: A $+$-regular circuit C of degree D, size s, and $+$-depth d.
Output: A $+$-regular circuit C' of $+$-depth $d-1$ and size at most s such that $C \equiv 0$ iff $C' \equiv 0$.

1. Let $\mathcal{L}_2^{\times} = \{g_1, g_2, \ldots, g_m\}$ and P_i be the polynomial computed at $g_i, 1 \leq i \leq m$.

2. Notice that each P_i is computed by a $\Pi^*\Sigma$ circuit. Using subroutine **Basis** we compute a maximum linearly independent subset, say, $\{P_1, P_2, \ldots, P_t\}$ of $\{P_i \mid 1 \leq i \leq m\}$, and also compute scalars $\alpha_{ji} \in \mathbb{F}$ such that:

$$P_j = \sum_{i=1}^{t} \alpha_{ji} P_i, \quad t+1 \leq j \leq m.$$

3. Construct the pruned circuit C'' as follows: in circuit C remove all gates below $\mathcal{L}_2^{\times}$, and replace g_i by $y_i, 1 \leq i \leq t$. These variables are the new inputs to C'.

4. Furthermore, in C, let the 2^{nd} $+$-layer gates be $\mathcal{L}_2^{+} = \{h_1, h_2, \ldots, h_q\}$. These gates compute linear combinations of the gates $g_i, 1 \leq i \leq m$. We do the rewiring so that in C'' the gates $h_1, h_2, \ldots, h_q$ compute the appropriate linear combinations of $y_1, y_2, \ldots, y_t$. Notice that $h_1, h_2, \ldots, h_q$ becomes the first $+$-layer for C'.

5. Output C'.

Algorithm A

5.1 Linear independence testing of $\Pi^*\Sigma$ circuits

In this subsection we solve the above mentioned linear independence testing problem. Namely, given as input $\Pi^*\Sigma$ circuits computing noncommutative polynomials $P_1, P_2, \ldots, P_m \in \mathbb{F}\langle X\rangle$, we give a deterministic polynomial-time algorithm that finds a maximal linearly independent subset A of these polynomials , and express the others as $\mathbb{F}$-linear combinations of the P_i in A.

We first need to analyze $\Pi^*\Sigma$ circuits. Let $\{L_1, L_2, \ldots, L_t\}$ be the set of homogeneous linear forms (in noncommuting variables $x_1, x_2, \ldots, x_n$) defined by the bottom Σ layers of the given $\Pi^*\Sigma$ circuits computing polynomials $P_i, 1 \le i \le m$.

Without loss of generality, let $\{L_1, L_2, \ldots, L_r\}$ be a maximal subset of $\{L_1, L_2, \ldots, L_t\}$ such that L_i and L_j are not scalar multiples of each other for $1 \le i < j \le r$. Thus, for each $L_i, i > r$ there is some $L_j, j \le r$ such that L_i is a scalar multiple of L_j. Therefore, we can express each P_i as a product of linear forms from $L_1, \ldots, L_r$, upto a scalar multiple:

$$P_i = \alpha_i L_{i1} L_{i2} \ldots L_{iD}, \quad 1 \le i \le m, \; \forall u : L_{iu} \in \{L_1, \ldots, L_r\}, \; \alpha_i \in \mathbb{F}.$$

Thus, each of these polynomials P_i is, upto a scalar multiple, a product of D many linear forms from the set $\{L_1, L_2, \ldots, L_r\}$.

Proposition 5.9. *Suppose polynomials P_i and P_j factorize as*

$$\begin{aligned} P_i &= \alpha_i L_{i1} L_{i2} \ldots L_{iD}, \\ P_j &= \alpha_j L_{j1} L_{j2} \ldots L_{jD}, \end{aligned}$$

where $\forall u : L_{iu}, L_{ju} \in \{L_1, \ldots, L_t\}$. Then, P_i and P_j are scalar multiples of each other iff $L_{ju} = L_{iu}, 1 \le u \le D$.

Proof. The polynomial P_i and P_j are homogeneous degree D polynomials in the free noncommutative ring $\mathbb{F}\langle X \rangle$. Now, it turns out that *homogeneous* polynomials in $\mathbb{F}\langle X \rangle$ have unique factorization (see, e. g., [5, Theorem 3.4]). Furthermore, the irreducible factors are all homogeneous and their order is also uniquely determined (only scalar multiples can vary). The proposition immediately follows from this observation. $\square$

In order to analyze $\Pi^*\Sigma$ circuits further, it is useful to think of a $\Pi^*\Sigma$ circuit computing a product of linear forms from $\{L_1, L_2, \ldots, L_r\}$ as a multiplicative circuit over the letters $\{L_1, L_2, \ldots, L_r\}$. More formally, corresponding to the linear forms $L_1, L_2, \ldots, L_r$ we define an alphabet $\Gamma = \{a_1, a_2, \ldots, a_r\}$ of r letters, where a_i stands for $L_i, 1 \le i \le r$.

Clearly, a multiplicative circuit over $(\Gamma^*, .)$ computes a word in Γ^*. Conversely, given a word $w \in \Gamma^*$ we could ask for its compression as a multiplicative circuit C_w. Such word compressions are well-studied in the literature and are closely related to the Lempel–Ziv text compression [16, 21, 17]. Some basic algorithmic results in this area turn out to be useful for our algorithm. We formally state these below.

Theorem 5.10 (Lohrey, Plandowski, Mehlhorn et al.)**.** *Let $\Gamma = \{a_1, a_2, \ldots, a_r\}$ be a finite alphabet.*

- *There is a deterministic polynomial-time algorithm that takes as input two multiplicative circuits C_i and C_j over Γ and tests if the words computed by them are identical. If not the algorithm returns the leftmost index k where the two words differ.*
- *Given a word w by a multiplicative circuit C as input over Γ, the following tasks can be done in deterministic polynomial time:*
  - *(i) computing the length $|w|$ of w.*

(ii) *given as input an index k in binary computing the k-th letter $w[k]$.*

(iii) *Multiplicative circuit $C_{k,k'}$ for the subword $w[k\dots k']$ for indices k and k' given in binary. In particular, it follows that the circuit $C_{k,k'}$ is of size polynomial in the sizes of C, k and k'. The parameters k, k' are given in binary.*

Remark 5.11. For $1 \le i \le m$, we transform the $\Pi^*\Sigma$ circuit computing P_i into a multiplicative circuit C_i computing a word w_i of length D in $\{a_1, a_2, \dots, a_r\}^D$ as follows: replace linear form L_j in the $\Pi^*\Sigma$ circuit by letter a_k if L_j is a scalar multiple of L_k, for $k \in [r]$. Clearly, C_i has the same number of multiplication gates as the $\Pi^*\Sigma$ circuit for P_i.

The following lemma is immediate.

Lemma 5.12.

- *The polynomials P_i and P_j are scalar multiples of each other if and only if the multiplicative circuits C_i and C_j compute the same word over $\{a_1, a_2, \dots, a_r\}$ (i. e., iff $w_i = w_j$).*
- *Thus, given as input $\Pi^*\Sigma$ circuits computing polynomials P_i and P_j, we can check if P_i and P_j are scalar multiples of each other in deterministic polynomial time.*

Proof. The first part of the lemma follows directly from Proposition 5.9. The second part follows immediately from Theorem 5.10. □

Our aim is to efficiently determine a maximal linearly independent subset A of $P_1, P_2, \dots, P_r$, and express each $P_i \notin A$ as a linear combination of polynomials in A. The difficulty is that the P_i, which are given by $\Pi^*\Sigma$ circuits of size s, can have degree exponential in s.

Our approach, that circumvents this difficulty, will crucially require a rank bound due to Saxena and Seshadhri [25] obtained in their study of *commutative* $\Sigma\Pi\Sigma$ arithmetic circuits. We recall the Saxena–Seshadhri result. Consider a $\Sigma\Pi\Sigma$ arithmetic circuit C', where the top Σ gate has fanin k, all Π gates are of fanin D, and each Π gate computes a product $Q_i, 1 \le i \le k$ of homogeneous $\mathbb{F}$-linear forms in *commuting* variables $y_1, y_2, \dots, y_n$. Circuit C' is said to be *simple* if the $\gcd(Q_1, Q_2, \dots, Q_k) = 1$. Circuit C' is said to be *minimal* if for any proper subset $S \subset [k]$ the sum $\sum_{i \in S} Q_i \neq 0$.

Theorem 5.13 (Saxena, Seshadhri)**.** *Let C' be a simple and minimal $\Sigma\Pi\Sigma$ arithmetic circuit of the form*

$$C'(y_1, y_2, \dots, y_n) = \sum_{i=1}^{k} \prod_{j=1}^{D} L_{ij},$$

where L_{ij} are homogeneous linear forms in commuting variables $y_1, y_2, \dots, y_n$. If the polynomial computed by C' is identically zero then the rank of the set $\{L_{ij} \mid 1 \le i \le k, 1 \le j \le D\}$ of all $\mathbb{F}$-linear forms occurring at the bottom Σ layer of the circuit is bounded by $O(k^2 \log D)$.

This line of research, analyzing the rank of linear forms occurring in $\Sigma\Pi\Sigma$ identities, was initiated by Dvir and Shpilka [10].

Noncommutative to commutative

For applying the rank bound of Theorem 5.13, we make the polynomials $P_i, 1 \le i \le \ell$ commutative and set-multilinear in variables $\{x_{im} \mid 1 \le i \le n, 1 \le m \le D\}$ as follows:

For each linear form $L_j = \sum_{i=1}^n \gamma_{ji} x_i, 1 \le j \le r$ define D many linear forms:

$$L'_{jm} = \sum_{i=1}^n \gamma_{ji} x_{im},\ 1 \le m \le D,$$

where x_{im}, for $1 \le i \le n$ and $1 \le m \le D$ are new distinct *commuting* variables. Define polynomials $\hat{P}_i$ from P_i as

$$\hat{P}_i = \prod_{m=1}^D L''_{im},$$

where $L''_{im} = L'_{jm}$ if $L_{im} = L_j, j \in [r]$. In other words, we obtain the set-multilinear polynomial $\hat{P}_i$ from P_i by replacing the m-th linear form, say L_j, with L'_{jm}, for $1 \le m \le D$.[3]

The following observation follows easily from the above definition of the polynomials $\hat{P}_i$.

Lemma 5.14. *Consider the set-multilinear polynomials $\hat{P}_i, 1 \le i \le m$ defined above. A linear form L' divides the gcd of $\{\hat{P}_i \mid i \in S\}$ for a subset $S \subseteq [m]$ if and only if $L' = L'_{jm}$ for some $j \in [r]$ and $m \in [D]$, and the linear form L_j occurs in the m-th position of each $P_i, i \in S$.*

Proof. If L' divides gcd of $\{\hat{P}_i \mid i \in S\}$, it follows that $L' = L''_{im}$ for some $m \in [D]$ for each i. But given i, $L''_{im} = L'_{jm}$ for some $j \in [r]$. It follows that L_j must occur at m-th position in each of the noncommutative products P_i. The reverse implication is also immediate. □

The next lemma explains the importance of Theorem 5.13 to the problem of computing a maximum independent subset of $\{P_i \mid i \in [m]\}$.

Lemma 5.15. *Suppose the polynomial P_i can be expressed as an $\mathbb{F}$-linear combination of the polynomials $P_1, P_2, \ldots, P_{i-1}$. Let*

$$S = \{j \in [i-1] \mid \deg(\gcd(\hat{P}_i, \hat{P}_j)) \ge D - c \cdot m^4\},$$

where $c > 0$ is some absolute constant. Then P_i can be expressed as a linear combination of polynomials from the subset $\{P_j \mid j \in S\}$.

Proof. Suppose P_i is expressible as an $\mathbb{F}$-linear combination of $P_1, P_2, \ldots, P_{i-1}$. Let $S \subseteq [i-1]$ be a minimal subset such that we can write

$$P_i = \sum_{j \in S} \gamma_j P_j, \quad \gamma_j \ne 0 \quad \text{for all } j.$$

Now, consider the set-multilinear circuit C' defined by the sum of products

$$\hat{P}_i - \sum_{j \in S} \gamma_j \hat{P}_j.$$

[3]The conversion to the set-multilinear polynomial is only for the sake of analysis and not for the actual algorithm.

By minimality of subset S, circuit C' is minimal. Suppose for some $j \in S$, P_i and P_j disagree on ρ positions, where $\rho > \ell^4$. That is, for more than ℓ^4 positions m, the linear forms occurring in the m-th position in P_i and P_j are different. Let the gcd of the polynomials in the set $\{\hat{P}_j \mid j \in S\} \cup \{\hat{P}_i\}$ be P, and let $\deg(P) = \delta$. By Lemma 5.14 it follows that

$$\delta \leq D - \rho.$$

Define polynomials $Q_i = \hat{P}_i/P$ and $Q_j = \hat{P}_j/P, j \in S$. Notice that each Q_i is a product of $D-\delta$ linear forms. Furthermore,

$$\hat{P}_i - \sum_{j \in S} \gamma_j \hat{P}_j = P(Q_i - \sum_{j \in S} \gamma_j Q_j).$$

Consider the simple and minimal circuit C'' defined by the sum

$$Q_i - \sum_{j \in S} \gamma_j Q_j.$$

Clearly, C' is zero iff C'' is zero. Since C'' is a *set-multilinear* circuit, the rank of the set of all $\mathbb{F}$-linear forms is at least $D-\delta$ (as linear forms L'_{jm} and $L'_{j'm'}$ for $m \neq m'$ have disjoint support). Now, if $C'' \equiv 0$ then by the Saxena-Seshadhri rank bound [25] we have $\rho \leq D - \delta \leq O(\ell^2 \log(D-\delta))$. Hence, it follows from the inequality $\log_2 x \leq x^{1/2}$ that $\rho \leq c \cdot \ell^4$, for some constant $c > 0$. Thus, for each $j \in S$, P_i and P_j can disagree on at most $c \cdot \ell^4$ positions. This proves the lemma. □

Using Theorem 5.10 we can efficiently determine if P_i and P_j given by $\Pi^*\Sigma$ circuits disagree on at most, say, q positions, q given in unary.

Lemma 5.16. *Let P_i and P_j be noncommutative homogeneous degree-D polynomials in $\mathbb{F}\langle X \rangle$, given as $\Pi^*\Sigma$ circuits of size at most s, where the first $+$-layer computes homogeneous linear forms $L_1, L_2, \ldots, L_r$ which are not scalar multiples of each other. Then in deterministic time polynomial in s, n, q we can determine if P_i and P_j differ in at most q positions, and if so, we can also determine those positions and the linear forms at those positions.*

Proof. The proof is a simple application of Theorem 5.10. First, as explained in Remark 5.11 we obtain size s multiplicative circuits C_i and C_j over letters a_i corresponding to the linear forms $L_i, 1 \leq i \leq r$. Let w_i and w_j be the length D words computed by C_i and C_j respectively. Applying the algorithm of Theorem 5.10 we can check if C_i and C_j compute identical words or find the leftmost index k such that $w_i[k] \neq w_j[k]$. By Theorem 5.10 we can compute circuits C'_i and C'_j for the suffixes $w_i[k+1 \cdots D]$ and $w_j[k+1 \cdots D]$ of w_i and w_j starting at $(k+1)$-th position, and again look for the leftmost index where they differ. We need to run this iteration at most $q+1$ times to determine all q positions where P_i and P_j differ, and if so, also find the linear forms at those locations. □

Before we design the algorithm for finding a maximum independent subset of $\{P_i \mid i \in [m]\}$, we need an observation which is a simple consequence of the Raz-Shpilka deterministic polynomial-time PIT algorithm [23] for *noncommutative* algebraic branching programs (see Definition 1.4).

Lemma 5.17. *Let P', $P'_1, P'_2, \ldots, P'_s$ be homogeneous degree-d noncommutative polynomials in $\mathbb{F}\langle X\rangle$, each given by an ABP as input. Then, in deterministic polynomial time we can determine if P' can be expressed as an $\mathbb{F}$-linear combination of the $P'_i, 1 \le i \le s$, and if so, we can compute scalars $\alpha_i \in \mathbb{F}$ such that*

$$P' = \sum_{i=1}^{s} \alpha_i P'_i.$$

Proof. Let B be the input ABP for P', and B_i be the given ABP for $P'_i, 1 \le i \le s$, respectively. Let $X = \{x_1, x_2, \ldots, x_n\}$.

For each monomial $m \in X^d$, let $v_m \in \mathbb{F}^s$ denote the vector of the coefficients of m in the polynomials $P'_i, 1 \le i \le s$. More precisely, $v_m[i]$ is the coefficient of m in $P'_i, 1 \le i \le s$. Let c_m denote the coefficient of m in the polynomial P'. Then we observe that

$$P' = \sum_{i=1}^{s} \alpha_i P'_i$$

if and only if for all monomials $m \in X^d$

$$c_m = \sum_{i=1}^{s} v_m[i] \cdot \alpha_i. \tag{5.3}$$

Equation (5.3) cannot direct give an efficient algorithm as there are n^d many equations to contend with. However, we will show, exploiting the fact that the P'_i are given as input by ABPs, that we can efficiently reduce the number of equations to polynomially many, which can then be efficient solved for the α_i.

Let y and z be two fresh noncommuting variables. Consider the noncommutative polynomial $\hat{P} = \sum_{i=1}^{s} y P'_i z$. It is homogeneous of degree $d+2$ and is computed by an ABP $\hat{B}$ obtained by combining the ABPs $B_i, 1 \le i \le s$ with a new start node and sink node as follows:

- Create a new start node s_0 and introduce directed edges from s_0 to the start node of B_i labeled by variable y for $1 \le i \le s$.

- Create a new sink node t_0 and introduce directed edges from the sink node of B_i to t_0 labeled by variable z for $1 \le i \le s$.

Clearly, $\hat{B}$ computes the polynomial $\hat{P}$.

Based on Raz and Shpilka's work [23], we now define a *Raz-Shpilka basis* for layer i of the ABP $\hat{P}$. Let the number of nodes at layer i be n_i and let $\{p_1, p_2, \ldots, p_{n_i}\}$ be the polynomials computed at the nodes. Consider the $n^i \times n_i$ matrix L_i whose rows are indexed by degree-i monomials, and the j-th column, $1 \le j \le n_i$, is the coefficient vector of p_j. A *Raz-Shpilka basis* for layer i is a set of at most n_i degree-i monomials such that the corresponding rows of L_i is a basis for the row space of L_i. Notice that every degree-i monomial is identified with a row vector of L_i. Following Raz and Shpilka [23], as also elaborated subsequently [6, Theorem 3.4], we can compute a Raz-Shpilka basis for every layer of the given ABP $\hat{P}$ in deterministic polynomial time. In particular, consider a Raz-Shpilka basis $\mathcal{M}$ for the $(d+1)$-th layer of $\hat{P}$ computed by the algorithm. The $(d+1)$-th layer of $\hat{P}$ consists of the s sink nodes of the ABPs $B_1, B_2, \ldots, B_s$. Thus, $\mathcal{M}$ is a collection of at most s monomials. Without loss of generality, we can assume that $\mathcal{M} = \{ym_1, ym_2, \ldots, ym_s\}$, where $m_i \in X^d, 1 \le i \le s$.

Since $\mathcal{M}$ is a Raz-Shpilka basis, it follows that for any monomial $m \in X^d$ the coefficient vector v_m is in the linear span of $\{v_{m_i} \mid 1 \le i \le s\}$. Next, we can compute the coefficient c_{m_i} of monomial m_i in the polynomial P' (from the input ABP B) in deterministic polynomial time [23, 6]. It follows that $\alpha_1, \alpha_2, \ldots, \alpha_s$ satisfies Equation (5.3) if and only if it satisfies the following s linear equations:

$$c_{m_j} = \sum_{i=1}^{s} v_{m_j}[i] \cdot \alpha_i, \ 1 \le j \le s. \tag{5.4}$$

Now, using Gaussian elimination, in polynomial time we can check feasibility of Equation (5.4) and compute a solution. □

We are now ready to prove the result of this subsection.

Theorem 5.18. *Given as input $\Pi^*\Sigma$ circuits computing noncommutative polynomials $P_1, P_2, \ldots, P_m \in \mathbb{F}\langle X\rangle$, there is a deterministic polynomial-time algorithm that will find a maximal linearly independent subset A of the polynomials $P_i, 1 \le i \le m$, and also express the others as $\mathbb{F}$-linear combinations of the P_i in A.*

Proof. The algorithm for computing a maximal linearly independent subset A of $P_1, P_2, \ldots, P_m$ works as follows: Include P_1 in A. For $i \ge 2$, include P_i in A iff it is linearly independent of $\{P_1, P_2, \ldots, P_{i-1}\}$.

Thus, the problem boils down to checking if P_i is linearly independent of $P_1, P_2, \ldots, P_{i-1}$. Following Lemma 5.15 let

$$S = \{j \in [i-1] \mid \deg(\gcd(\hat{P}_i, \hat{P}_j)) \ge D - c \cdot m^4\},$$

where D is the degree of each of the polynomials $P_i, 1 \le i \le m$ and $c > 0$ is some absolute constant. By Lemma 5.15 it suffices to check if P_i is linearly independent of $\{P_j \mid j \in [i-1]$ such that $j \in S\}$.

By Lemma 5.16 we can efficiently compute the subset S, and the subset $I \subset [D]$ of at most $c \cdot m^5$ many positions such that P_i agrees with every $P_j, j \in S$ in all positions in $[D] \setminus I$. Define $P'_i = \prod_{u \in I} L_{iu}$ and $P'_j = \prod_{u \in I} L_{ju}$ for all $j \in S$. The following claim is immediate. It follows from the fact that for every position $u \in [D] \setminus I$ the linear forms L_{iu} and $L_{ju}, j \in S$ are all identical.

Claim 5.19. *P_i is linearly independent of $\{P_j \mid j \in S\}$ if and only if P'_i is linearly independent of $\{P'_j \mid j \in S\}$.*

Thus it suffices to check if P'_i is linearly independent of $\{P'_j \mid j \in S\}$. Each of these polynomials is a product of at most cm^5 many linear forms. Since a product of linear forms can clearly be seen as an ABP, we can apply the algorithm described in Lemma 5.17 to check this in polynomial time.

To conclude the overall proof we note that Algorithm B below can be applied to determine the leftmost maximal linearly independent subset A of the input polynomials $P_1, \ldots, P_m$ and also express the others as linear combinations of polynomials in A.

Linear-Span

Input: $\Pi^*\Sigma$ circuits for polynomials $P_1, P_2, \ldots, P_i$.
Output: Check whether P_i can be expressed as a linear combination of $\{P_1, P_2 \ldots, P_{i-1}\}$ or not. If so, find such an expression.

1. Let $L_i, i \in [r]$ be the linear forms computed at the bottom $+$-layer of the input circuits such that they are not multiples of each other.

2. Find corresponding multiplicative circuits $C_j, j \leq i$ over letters $a_i, i \in [r]$ corresponding to the L_i. Using Lemma 5.16, determine $S \subseteq [i-1]$ such that C_i and C_j differ in at most cm^4 positions and find the letters occurring in those positions.

3. Let $I \subset [D]$ be the set of at most cm^5 positions where P_i differs from some $P_j, j \in S$, and compute the polynomials P_i' and $P_j', j \in S$ by dropping the linear forms occurring in all other positions.

4. Using Lemma 5.17 determine if P_i' is can be expressed as a linear combination of $P_j', j \in S$. If so, then the same linear expression will hold for P_i and $P_j, j \in S$ and is the output.

Algorithm B

Finally, we describe Algorithm C, which is the basis finding algorithm.

Basis

Input: $\Pi^*\Sigma$ circuits for polynomials $P_1, P_2, \ldots, P_m$.
Output: A maximum $\mathbb{F}$-linearly independent subset A of $\{P_1, P_2, \ldots, P_m\}$.

1. $A := \{P_1\}$.

2. For $i = 2$ to m do: If P_i is linearly independent of $\{P_1, P_2, \ldots, P_{i-1}\}$ then $A := A \cup \{P_i\}$. (This is tested using Algorithm B).

3. Output A.

Algorithm C

□

The main result of this section is now a simple consequence of Lemma 5.8.

Proof of Theorem 2.5. Let the input $+$-regular circuit C be of size s and $+$-depth d. By Lemma 5.8, using Algorithm A we can transform, in deterministic polynomial time, C into a $+$-regular circuit C' of size bounded by s and $+$-depth $d-1$ such that $C \equiv 0$ if and only if $C' \equiv 0$.

On repeated application we finally obtain a $\Sigma\Pi^*\Sigma$ circuit $\hat{C}$ of size at most s, for which we need to do polynomial identity testing. The polynomial $\hat{P}$ computed by $\hat{C}$ is of the form

$$\hat{P} = \sum_{i=1}^{m} \alpha_i P_i, \tag{5.5}$$

where each P_i is computed by a $\Pi^*\Sigma$ circuit of size at most s, and $P_m = \prod_{j=1}^{D} L_{jm}$, where each L_{jm} is from the set of linear forms $\{L_1, L_2, \ldots, L_t\}, t \leq s$ occurring at the bottom $+$-layer of circuit C.

We can now apply Theorem 5.18 to find a maximal linearly independent subset $\{P_j \mid j \in S\}$, $S \subseteq [m]$ of $\{P_1, P_2, \ldots, P_m\}$, and express each $P_i, i \notin S$ as a linear combination of polynomials in $\{P_j \mid j \in S\}$. Substituting these expressions in Equation (5.5) and collecting coefficients we will obtain

$$\hat{P} = \sum_{j \in S} \beta_j P_j,$$

which, by linear independence of $\{P_j \mid j \in S\}$ is identically zero iff each β_j is zero. □

6 Black-box randomized PIT for homogeneous $\Sigma\Pi^*\Sigma$

In this section we give a randomized black-box PIT algorithm for $\Sigma\Pi^*\Sigma$ circuits.

By Theorem 2.5 we can test if a given homogeneous $\Sigma\Pi^*\Sigma$ circuit (white-box) is identically zero in deterministic polynomial time. However, suppose we have only black-box access to a $\Sigma\Pi^*\Sigma$ circuit C computing a polynomial in $\mathbb{F}\langle X \rangle$. In other words, we can evaluate C on square matrices M_i substituted for $x_i, 1 \leq i \leq n$, where the cost of an evaluation is the dimension of the M_i. The results in Section 5 exploit the input circuit C, and hence are not applicable. Specifically, C may compute an exponential degree noncommutative polynomial, but it is not clear if we can test whether $C \equiv 0$ by evaluating it on polynomial dimension matrices. Neither is the black-box PIT of Section 4 applicable, since $\Sigma\Pi^*\Sigma$ circuits can compute polynomials with a double-exponential number of monomials.

Suppose C is an s-sum $P_1 + P_2 + \cdots + P_s$ of D-products of linear forms in variables X. That is,

$$P_i = L_{i1} L_{i2} \ldots L_{iD},$$

where D is exponential. Then we show that $C \not\equiv 0$ iff C is nonzero on random $O(s) \times O(s)$ matrices with entries from $\mathbb{F}$ or a suitably large extension of $\mathbb{F}$. The proof is based on the notion of projected polynomials defined below.

6.1 Projected polynomials

Definition 6.1. Let $P \in \mathbb{F}\langle X\rangle$ be a homogeneous degree-D polynomial. For an index set $I \subseteq [D]$ the *I-projection* of polynomial P is the polynomial P_I which is defined by letting all variables occurring in positions indexed by the set I as noncommuting. In all other positions we make the variables commuting, by replacing x_i with a corresponding commuting variable z_i for $1 \le i \le n$. Thus, the I-projected polynomial P_I is in $\mathbb{F}[Z]\langle X\rangle$, and the (noncommutative) degree of P_I, which is its degree only in the X variables, is $|I|$.

Lemma 6.2. *Let $P_1, P_2, \ldots, P_s \in \mathbb{F}\langle X\rangle$ each be a product of D homogeneous linear forms*

$$P_i = L_{i,1}L_{i,2}\ldots L_{i,D},$$

where $\{L_{i,j} : 1 \le i \le s, 1 \le j \le D\}$ are linear forms in $\mathbb{F}\langle X\rangle$. Then, given any scalars $\beta_1, \beta_2, \ldots, \beta_s \in \mathbb{F}$ there exists a subset $I \subseteq [D]$ of size at most $s-1$ such that

$$\sum_{i=1}^{s} \beta_i P_i = 0 \text{ iff } \sum_{i=1}^{s} \beta_i P_{i,I} = 0,$$

where $P_{i,I} \in \mathbb{F}[Z]\langle X\rangle$ is the I-projection of the polynomial P_i.

Proof. The proof is by induction on s. The lemma clearly holds for the base case $s = 1$, because $\mathbb{P}_{1,\emptyset}$, which is a product of linear forms in $\mathbb{F}[Z]$, is nonzero iff P_1 is nonzero.

By induction hypothesis, we assume that an index set of size at most $s-2$ exists for any set of at most $s-1$ polynomials in $\mathbb{F}\langle X\rangle$, each of which is a product of D homogeneous linear forms. The forward implication is obvious, because making variables commuting can only facilitate cancellations. We prove the reverse implication.

Suppose $\sum_{i=1}^{s} \beta_i P_i \neq 0$ for $\beta_i \in \mathbb{F}, 1 \le i \le s$. Let $j_0 \in [D]$ be the least index, if it exists, such that $\operatorname{rank}\{L_{1,j_0}, \ldots, L_{s,j_0}\} > 1$. If no such index exists then the P_i are all scalar multiples of each other. Then $\sum_{i=1}^{s} \beta_i P_i = \alpha P_1$, for some $\alpha \in \mathbb{F}$, which is zero if and only if $\alpha P_{1,\emptyset}$ is zero (by the base case), proving the implication.

We can assume, by renumbering the polynomials, that $\{L_{1,j_0}, \ldots, L_{t,j_0}\}$ is a maximal linearly independent set in $\{L_{1,j_0}, \ldots, L_{s,j_0}\}$, where $t > 1$.

Then,

$$P_i = c_i P L_{i,j_0} L_{i,j_0+1} \ldots L_{i,D} : 1 \le i \le t,$$
$$P_i = c_i P \cdot \Big(\sum_{k=1}^{t} \gamma_k^{(i)} L_{k,j_0}\Big) L_{i,j_0+1} \ldots L_{i,D} : t+1 \le i \le s,$$

where $\{c_i \in \mathbb{F} : 1 \le i \le s\}$, $\{\gamma_k^{(i)} \in \mathbb{F} : 1 \le k \le t, t+1 \le i \le D\}$, and $P \in \mathbb{F}\langle X\rangle$ is a product of homogeneous linear forms (or a scalar). For $1 \le i \le s$, let

$$P_i' = c_i \prod_{j=j_0+1}^{D} L_{i,j}.$$

We can then write

$$\sum_{i=1}^{s}\beta_i P_i = P\cdot\Big(\sum_{i=1}^{t}\beta_i L_{i,j_0}P'_i\Big) + P\cdot\Big(\sum_{i=t+1}^{s}\beta_i L_{i,j_0}P'_i\Big).$$

Note that

$$P\cdot\Big(\sum_{i=t+1}^{s}\beta_i L_{i,j_0}P'_i\Big) = P\cdot\left(\sum_{i=t+1}^{s}\beta_i\Big(\sum_{k=1}^{t}\gamma_k^{(i)}L_{k,j_0}\Big)P'_i\right).$$

Now by rearranging terms, we have:

$$\sum_{i=1}^{s}\beta_i P_i = P\cdot\Big(\sum_{k=1}^{t}L_{k,j_0}P''_k\Big) \text{ where,} \tag{6.1}$$

$$P''_k = \beta_k P'_k + \beta_{t+1}\gamma_k^{(t+1)}P'_{t+1} + \cdots + \beta_s\gamma_k^{(s)}P'_s, \text{ for } 1\le k\le t. \tag{6.2}$$

Now, $L_{k,j_0}, 1\le k\le t$ are linearly independent. Let A be an invertible linear transform such that $A: L_{k,j_0}\mapsto x_k, 1\le k\le t$. Applying Proposition 3.1, consider the map A_{j_0} applied to the polynomial $\sum_{i=1}^{s}\beta_i P_i$. By Proposition 3.1, $A_{j_0}(\sum_{i=1}^{t}\beta_i P_i)\neq 0$. Now, noting that A_{j_0} applies A to the position j_0 of the polynomial $\sum_{i=1}^{t}\beta_i P_i$, we have:

$$A_{j_0}\Big(\sum_{i=1}^{s}\beta_i P_i\Big) = P\cdot\Big(\sum_{k=1}^{t}x_k P''_k\Big)\neq 0.$$

Thus, not all P''_k are zero. Without loss of generality, suppose $P''_1\neq 0$. Since $t>1$, by Equation (6.1), P''_1 is a sum of at most $s-1$ polynomials, each of which is a product of homogeneous linear forms. Hence, by induction hypothesis, there is an index set $I'\subseteq\{j_0+1,\ldots,D\}$ of size at most $s-2$ such that

$$P''_{1,I'} = \left(\beta_1 P'_{1,I'} + \beta_{t+1}\gamma_1^{(t+1)}P'_{t+1,I'} + \cdots + \beta_s\gamma_1^{(s)}P'_{s,I'}\right)\neq 0.$$

Let $I=I'\cup\{j_0\}$. Now consider the polynomial $\sum_{i=1}^{s}\beta_i P_{i,I}$, which we claim is nonzero. By Equation (6.1), note that

$$\sum_{i=1}^{s}\beta_i P_{i,I} = \hat{P}\cdot\Big(\sum_{k=1}^{t}L_{k,j_0}P''_{k,I'}\Big),$$

where $\hat{P}$ is the commutative polynomial obtained by replacing x_i by z_i in P. Since $\hat{P}$ is a product of linear forms it is nonzero. Thus, it suffice to argue that $\sum_{k=1}^{t}L_{k,j_0}P''_{k,I'}$ is nonzero. Notice that, $\sum_{k=1}^{t}L_{k,j_0}P''_{k,I'}$ is a homogeneous noncommutative polynomial of noncommutative degree $|I|$ in $\mathbb{F}[Z]\langle X\rangle$. By Proposition 3.1, applying the linear transform A to the first position (because the original position j_0 is now the first position), observe that $A_1(\sum_{k=1}^{t}L_{k,j_0}P''_{i,I'}) = \sum_{k=1}^{t}x_k P''_{i,I'}$. This sum is zero if and only if each $P''_{k,I'}$ is zero. However, $P''_{1,I'}$ is nonzero. This completes the induction step and the proof. □

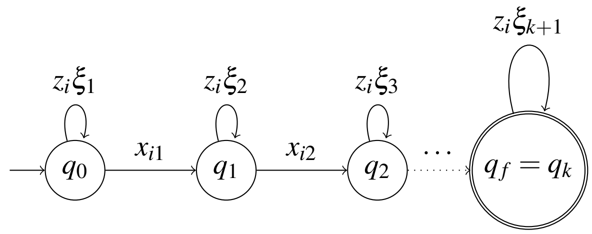

Figure 3: The transition diagram for the variable $x_i : 1 \le i \le n$

6.2 The black-box identity test

We now prove Theorem 2.7 which will yield a *black-box* randomized polynomial-time identity testing algorithm for $\Sigma\Pi^*\Sigma$ circuits.

Proof of Theorem 2.7. Theorem 2.7 is an easy consequence of Lemma 6.2.

Let $C = \sum_{i=1}^{s} \beta_i R_i \in \mathbb{F}\langle X\rangle$ be the polynomial computed by the black-box $\Sigma\Pi^*\Sigma$ circuit, where each R_i is a product of D homogeneous linear forms. By Lemma 6.2 there is a set $I \subseteq [D]$ of size at most $s-1$ such that $C = \sum_{i=1}^{s} \beta_i R_i = 0$ if and only if $\tilde{C} = \sum_{i=1}^{s} \beta_i R_{i,I} = 0$. As done in Section 4 we can use a small size nondeterministic automaton to guess this subset I of locations, and substitute suitable commuting variables at all locations in $[D]\setminus I$. It will turn out that the transition matrices for each variable x_i corresponding to this automaton will give us a matrix substitution for the variables where C is nonzero. We now present the details.

Let $|I| = k \le s-1$. Consider the following nondeterministic $k+1$-state finite automaton A whose transition diagram we depict for $x_i : 1 \le i \le n$ in Figure 3. For locations in $[D]\setminus I$, the automaton uses the block variables $Z = \{z_i : 1 \le i \le n\}$, $\xi = \{\xi_i : 1 \le i \le k+1\}$ which are commuting variables. Let $I = \{i_1, i_2, \ldots, i_k\}$ listed in increasing order. For variable x_i occurring in position i_j (i. e., the j-th position guessed to be in I) the automaton substitutes x_i by $x_{ij}, 1 \le i \le n$. These index variables $Z' = \{x_{ij} : 1 \le i \le n, 1 \le j \le k\}$ are also commuting variables.

Remark 6.3. Notice that in Lemma 6.2, the variables occurring in positions in I were left as noncommuting. However, the automaton we construct replaces x_i in position $j \in I$ by commuting variable x_{ij}. This transformation for homogeneous polynomials is known to preserves identities by Proposition 3.2.

Let
$$\forall i \in [s] : R_i = L_{i,1} \ldots L_{i,D}.$$

Let M_{x_i} be the matrix corresponding to variable $x_i, 1 \le i \le n$ (these are $(k+1)\times(k+1)$ matrices). When we do this matrix substitution to variables in R_i, the $(0,k)$-th entry of the resulting matrix M_{R_i} is
$$\widehat{R}_i = \sum_{(j_1, j_2, \ldots, j_k) \in [D]^k} P_{1,j_1-1}\, P_{j_1+1,j_2-1} \cdots$$
where
$$P_{1,j_1-1} = \prod_{j=1}^{j_1-1} L_{i,j}(Z)\xi_1{}^{j_1-1} L_{i,j_1}(Z'), \qquad P_{j_1+1,j_2-1} = \prod_{j=j_1+1}^{j_2-1} L_{i,j}(Z)\xi_2{}^{j_2-j_1-1} L_{i,j_2}(Z'),$$

and so on. For each $i \in [s]$, the polynomial $\widehat{R}_i \in \mathbb{F}[Z,\xi,Z']$. The $(0,k)$-th entry of the resulting matrix M_C is

$$\sum_{i=1}^{s} \beta_i \widehat{R}_i = \sum_{J\in[D]^k} \beta_i P_J \xi_J,$$

where $\xi_J = \xi_1^{j_1-1}\xi_2^{j_1-j_2-1}\ldots\xi_k^{D-j_k}$ and $P_J = \sum_{i=1}^{s} P_{i,J}$.

By Lemma 6.2, we know that $P_I = \sum_{i=1}^{s} P_{i,I} \neq 0$. Thus, $\sum_{i=1}^{s} \beta_i \widehat{R}_i$ is nonzero, as the monomials sets for different P_J are disjoint (ensured by the terms ξ_J). The degree of $\sum_{i=1}^{s} \beta_i \widehat{R}_i$ is D. So if $|\mathbb{F}|$ is more than D, it can not evaluate to zero on $\mathbb{F}$. This completes the proof of Theorem 2.7. □

Now the randomized identity testing algorithm follows by simply random substitution for variables in the commutative polynomial computed at the $(0,k)$-th entry of the resulting matrix M_C. This completes the proof of Corollary 2.8.

7 Conclusion

The Amitsur-Levitzki theorem is a cornerstone result in the theory of polynomial identities [22]. Our result that the dimension of the non-vanishing matrix algebra is at most logarithmic in the *sparsity* of the polynomial, should be interesting even from a purely algebraic perspective.

The main open problem is to extend our technique to solve the identity testing problem for all noncommutative circuits in randomized polynomial time (even in the white-box model). Our result for $+$-regular circuits is a first step towards that. Finding a randomized black-box identity testing algorithm for $+$-regular circuits is an interesting problem. We have such a result only for depth-three $+$-regular circuits.

Acknowledgements

We are grateful to the referees for their detailed comments and suggestions which have helped us improve the presentation. We also thank the anonymous STOC 2017 referees for their valuable comments. We thank Markus Lohrey, Amir Shpilka, and Srikanth Srinivasan for their comments on an earlier version of this work. Finally, we thank the editor-in-chief, László Babai, for pointing out Muller's paper [18], suggesting the name "Polynomial Identity Lemma," and numerous other editorial comments.

References

[1] MANINDRA AGRAWAL, ROHIT GURJAR, ARPITA KORWAR, AND NITIN SAXENA: Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. [doi:10.1137/140975103, arXiv:1406.7535] 3

[2] NOGA ALON AND ZOLTÁN FÜREDI: Covering the cube by affine hyperplanes. *Eur. J. Comb.*, 14:79–83, 1993. [doi:10.1006/eujc.1993.1011] 10

[3] AVRAHAM SHIMSHON AMITSUR AND JACOB LEVITZKI: Minimal identities for algebras. *Proc. Amer. Math. Soc.*, 1:449–463, 1950. [doi:10.2307/2032312] 2

[4] VIKRAMAN ARVIND, PUSHKAR S. JOGLEKAR, PARTHA MUKHOPADHYAY, AND S. RAJA: Randomized polynomial time identity testing for noncommutative circuits. In *Proc. 49th STOC*, pp. 831–841. ACM Press, 2017. [doi:10.1145/3055399.3055442] 1

[5] VIKRAMAN ARVIND, PUSHKAR S. JOGLEKAR, AND GAURAV RATTAN: On the complexity of noncommutative polynomial factorization. *Inform. and Comput.*, 262(1):22–39, 2018. Preliminary version in MFCS'15. [doi:10.1016/j.ic.2018.05.009, arXiv:1501.00671] 11, 21

[6] VIKRAMAN ARVIND, PARTHA MUKHOPADHYAY, AND SRIKANTH SRINIVASAN: New results on noncommutative and commutative polynomial identity testing. *Comput. Complexity*, 19(4):521–558, 2010. Preliminary version in CCC'08. [doi:10.1007/s00037-010-0299-8, arXiv:0801.0514] 7, 10, 25, 26

[7] ANURAG BISHNOI, PETE L. CLARK, ADITYA POTUKUCHI, AND JOHN SCHMITT: On zeros of a polynomial in a finite grid. *Combinatorics, Probability and Computing*, 27(3):310–333, 08 2018. [doi:10.1017/S0963548317000566, arXiv:1508.06020] 10

[8] ANDREJ BOGDANOV AND HOETECK WEE: More on noncommutative polynomial identity testing. In *Proc. 20th IEEE Conf. on Computational Complexity (CCC'05)*, pp. 92–99. IEEE Comp. Soc. Press, 2005. [doi:10.1109/CCC.2005.13] 2, 3

[9] RICHARD A. DEMILLO AND RICHARD J. LIPTON: A probabilistic remark on algebraic program testing. *Inform. Process. Lett.*, 7(4):193–195, 1978. [doi:10.1016/0020-0190(78)90067-4] 3, 9

[10] ZEEV DVIR AND AMIR SHPILKA: Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007. Preliminary version in STOC'05. [doi:10.1137/05063605X] 22

[11] MICHAEL A. FORBES AND AMIR SHPILKA: Quasipolynomial-time identity testing of noncommutative and read-once oblivious algebraic branching programs. In *Proc. 54th FOCS*, pp. 243–252. IEEE Comp. Soc. Press, 2013. [doi:10.1109/FOCS.2013.34, arXiv:1209.2408] 3

[12] LAURENT HYAFIL: The power of commutativity. In *Proc. 18th FOCS*, pp. 171–174. IEEE Comp. Soc. Press, 1977. [doi:10.1109/SFCS.1977.31] 2

[13] NEERAJ KAYAL, CHANDAN SAHA, AND RAMPRASAD SAPTHARISHI: A super-polynomial lower bound for regular arithmetic formulas. In *Proc. 46th STOC*, pp. 146–153. ACM Press, 2014. [doi:10.1145/2591796.2591847] 16

[14] RUDOLF LIDL AND HARALD NIEDERREITER: *Finite Fields*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 1996. [doi:10.1017/CBO9780511525926] 10

[15] RICHARD J. LIPTON: The curious history of the Schwartz–Zippel Lemma. `https://rjlipton.wordpress.com/2009/11/30/the-curious-history-of-the-schwartz-zippel-lemma/`, 2009. 10

[16] MARKUS LOHREY: Equality testing of compressed strings. In *Proc. 10th Int. Conf. on Combinatorics on Words (WORDS'15)*, pp. 14–26. Springer, 2015. [doi:10.1007/978-3-319-23660-5_2] 7, 17, 21

[17] KURT MEHLHORN, R. SUNDAR, AND CHRISTIAN UHRIG: Maintaining dynamic sequences under equality tests in polylogarithmic time. *Algorithmica*, 17(2):183–198, 1997. Preliminary version in SODA'94. [doi:10.1007/BF02522825] 7, 21

[18] DAVID E. MULLER: Application of boolean algebra to switching circuit design and to error detection. *Trans. Inst. Radio Engineers Professional Group on Electronic Computers*, EC-3:6–12, 1954. [doi:10.1109/IREPGELC.1954.6499441] 3, 9, 10, 32

[19] NOAM NISAN: Lower bounds for non-commutative computation (extended abstract). In *Proc. 23rd STOC*, pp. 410–418. ACM Press, 1991. [doi:10.1145/103418.103462] 2, 5

[20] ØYSTEIN ORE: Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter Ser. I*, 7(15):27, 1922. 3, 9, 10

[21] WOJCIECH PLANDOWSKI: Testing equivalence of morphisms on context-free languages. In *Proc. 2nd Europ. Symp. on Algorithms (ESA'94)*, pp. 460–470. Springer, 1994. [doi:10.1007/BFb0049431] 7, 21

[22] CLAUDIO PROCESI: On the theorem of Amitsur–Levitzki. *Israel J. Math.*, 207(1):151–154, 2015. [doi:10.1007/s11856-014-1118-8, arXiv:1308.2421] 32

[23] RAN RAZ AND AMIR SHPILKA: Deterministic polynomial identity testing in non-commutative models. *Comput. Complexity*, 14(1):1–19, 2005. Preliminary version in CCC'04. [doi:10.1007/s00037-005-0188-8] 3, 7, 24, 25, 26

[24] IRVING S. REED: A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory*, 4(4):38–49, 1955. [doi:10.1109/TIT.1954.1057465] 10

[25] NITIN SAXENA AND C. SESHADHRI: From Sylvester–Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33:1–33:33, 2013. Preliminary version in FOCS'10. [doi:10.1145/2528403, arXiv:1002.0145] 7, 17, 22, 24

[26] WOLFGANG M. SCHMIDT: *Equations over Finite Fields: An Elementary Approach*. Volume 536 of *Lecture Notes in Math.* Springer, 1 edition, 1976. [doi:10.1007/bfb0080437] 3, 9, 10

[27] JACOB T. SCHWARTZ: Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. Preliminary version in EUROSAM'79. [doi:10.1145/322217.322225] 3, 9

[28] RICHARD E. ZIPPEL: Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation (EUROSAM'79)*, volume 72 of *LNCS*, pp. 216–226. Springer, 1979. [doi:10.1007/3-540-09519-5_73] 3, 9

AUTHORS

Vikraman Arvind
Professor
Institute of Mathematical Sciences (HBNI)
Chennai, India,
arvind@imsc.res.in
www.imsc.res.in/~arvind

Pushkar S. Joglekar
Assistant professor
Vishwakarma Institute of Technology
Pune, India
joglekar.pushkar@gmail.com

Partha Mukhopadhyay
Associate professor
Chennai Mathematical Institute
Chennai, India,
partham@cmi.ac.in
www.cmi.ac.in/~partham

S. Raja
Assistant professor
Indian Institute of Technology Tirupati
Tirupati, India
raja@iittp.ac.in
www.iittp.ac.in/dr-s-raja

ABOUT THE AUTHORS

VIKRAMAN ARVIND, called Arvind by friends and colleagues, graduated from the Indian Institute of Technology, Kanpur, India in 1988. His advisor was Somenath Biswas. He is currently professor at The Institute of Mathematical Sciences, Chennai, India. His research interests are broadly in computational complexity and algorithms, and especially in problems of an algebraic flavor. He likes reading; his favorite authors include G. K. Chesterton and Ray Bradbury, and he enjoys watching movies.

PUSHKAR JOGLEKAR is currently an assistant professor at Vishwakarma Institute of Technology, Pune, India. He received his Ph. D. in Theoretical Computer Science from the Institute of Mathematical Sciences, Chennai, India in 2011 under the supervision of V. Arvind. Hobbies: reading, mathematical puzzle solving, gardening.

PARTHA MUKHOPADHYAY graduated from the Institute of Mathematical Sciences, Chennai, India in 2009. His advisor was V. Arvind. Currently he is an associate professor at the Chennai Mathematical Institute, India. His research interests are mainly in algorithmic problems in the intersection of computational complexity and algebra. He enjoys reading, long-distance running, and playing table tennis.

S. RAJA graduated from the Institute of Mathematical Sciences, Chennai, India in 2017. His advisor was V. Arvind. His current research interests lie in arithmetic circuit complexity. He is presently an assistant professor in the Computer Science and Engineering department at IIT Tirupati.