Efficient Black-Box Identity Testing for Free Group Algebras

V. Arvind
Institute of Mathematical Sciences (HBNI), Chennai, India
arvind@imsc.res.in

Abhranil Chatterjee
Institute of Mathematical Sciences (HBNI), Chennai, India
abhranilc@imsc.res.in

Rajit Datta
Chennai Mathematical Institute, Chennai, India
rajit@cmi.ac.in

Partha Mukhopadhyay
Chennai Mathematical Institute, Chennai, India
partham@cmi.ac.in

Abstract

Hrubeš and Wigderson [12] initiated the study of noncommutative arithmetic circuits with division computing a noncommutative rational function in the *free skew field*, and raised the question of rational identity testing. For noncommutative *formulas* with inverses the problem can be solved in deterministic polynomial time in the *white-box* model [10, 13]. It can be solved in randomized polynomial time in the *black-box* model [8], where the running time is polynomial in the size of the formula. The complexity of identity testing of noncommutative rational functions, in general, remains open for noncommutative circuits with inverses.

We solve the problem for a natural special case. We consider expressions in the free group algebra $\mathbb{F}\langle X, X^{-1}\rangle$ [1] where $X = \{x_1, x_2, \ldots, x_n\}$. Our main results are the following.

1. Given a degree d expression f in $\mathbb{F}\langle X, X^{-1}\rangle$ as a black-box, we obtain a randomized poly(n, d) algorithm to check whether f is an identically zero expression or not. The technical contribution is an Amitsur-Levitzki type theorem [1] for $\mathbb{F}\langle X, X^{-1}\rangle$. This also yields a deterministic identity testing algorithm (and even an expression reconstruction algorithm) that is polynomial time in the sparsity of the input expression.
2. Given an expression f in $\mathbb{F}\langle X, X^{-1}\rangle$ of degree D and sparsity s, as black-box, we can check whether f is identically zero or not in randomized poly$(n, \log s, \log D)$ time. This yields a randomized polynomial-time algorithm when D and s are exponential in n.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation

Keywords and phrases Rational identity testing, Free group algebra, Noncommutative computation, Randomized algorithms

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.57

Category RANDOM

Acknowledgements We thank the anonymous reviewers of RANDOM 2019 for their valuable comments.

[1] Here $\mathbb{F}\langle X, X^{-1}\rangle$ denotes $\mathbb{F}\langle x_1, \ldots, x_n, x_1^{-1}, \ldots, x_n^{-1}\rangle$.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019).
Editors: Dimitris Achlioptas and László A. Végh; Article No. 57; pp. 57:1–57:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Noncommutative computation is an important sub-area of arithmetic circuit complexity. In the usual arithmetic circuit model for noncommutative computation, the arithmetic operations are addition and multiplication, where each circuit input is either a variable from $X = \{x_1, x_2, \ldots, x_n\}$ or a scalar from a prescribed field $\mathbb{F}$. Each multiplication gate in the circuit respect the order of its inputs since the variables x_i are noncommuting. Such circuits compute precisely noncommutative polynomials in the *free noncommutative ring* denoted by $\mathbb{F}\langle X\rangle$.

Analogous to commutative arithmetic computation, the central questions are to show circuit size lower bounds for explicit noncommutative polynomials and derandomization of polynomial identity testing (PIT) for noncommutative circuits (or subclasses of circuits). There is nontrivial progress on these problems unlike in the commutative case. Nisan [16] has shown that any algebraic branching program (ABP) computing the $n \times n$ noncommutative Determinant or Permanent polynomial requires exponential (in n) size. Raz and Shpilka [17] have shown a deterministic polynomial-time PIT for noncommutative ABPs in the white-box model. A quasi-polynomial time derandomization is also known for the black-box model [9]. However, for general circuits there are no better results (either lower bound or PIT) than known in the commutative setting.

The randomized polynomial-time PIT algorithm for noncommutative circuits computing a polynomial of polynomially bounded degree [6] follows from the Amitsur-Levitzki theorem [1] which states that a nonzero polynomial $p \in \mathbb{F}\langle X\rangle$ of degree $< 2k$ cannot be an identity for the ring $\mathbb{M}_k(\mathbb{F})$ of $k \times k$ matrices over $\mathbb{F}$. It is also known [2] that a nonzero noncommutative polynomial does not vanish on matrices of dimension logarithmic in the sparsity of the polynomial. This yields a randomized polynomial-time identity test for noncommutative circuits computing polynomials of exponential degree and exponential sparsity.

Hrubeš and Wigderson [12] initiated the study of noncommutative computation with inverses. In the commutative world, it suffices to consider additions and multiplications. By Strassen's result [20] (extended to finite fields [11]), divisions can be efficiently replaced by polynomially many additions and multiplications. However, divisions in noncommutative computation are more intricate [12]. In the same paper [12], the authors introduce *rational identity testing*: Given a noncommutative formula involving addition, multiplication and division gates, efficiently check if the resulting rational expression is identically zero in the free skew-field of noncommutative rational functions. They show that it is reducible to the following SINGULAR problem:

Given a matrix $A_{n\times n}$ where the entries are linear forms over noncommuting variables $\{x_1, x_2, \ldots, x_n\}$, is A invertible in the free skew-field?

In the white-box model the problem is in deterministic polynomial time [10, 13], and in randomized polynomial time in the black-box model [8]. Specifically, for rational formulas of size s, random matrix substitutions of dimension linear in s suffices to test if the rational expression is identically zero [8].

The complexity of identity testing for general rational expressions remains open. For example, given a noncommutative circuit involving addition, multiplication and division gates, no efficient algorithm (even randomized!) is known to check if the resulting rational expression is identically zero in the free skew-field of noncommutative rational functions. In order to precisely formulate the problem, we define classes of rational expressions based on Bergman's definition [5] of *inversion height* which we now recall and elaborate upon with some notation.

▶ **Definition 1** ([5]). *Let X be a set of free noncommuting variables. Polynomials in the free ring $\mathbb{F}\langle X\rangle$ are defined to be rational expressions of* height 0*. A rational expression of* height $i+1$ *is inductively defined to be a polynomial in rational expressions of height at most i, and inverses of such expressions.*

Let $\mathcal{E}_{d,0}$ denote all polynomials of degree at most d in the free ring $\mathbb{F}\langle X\rangle$. We inductively define rational expressions in $\mathcal{E}_{d,i+1}$ as follows: Let $f_1, f_2, \ldots, f_r$ and $g_1, g_2, \ldots, g_s$ be rational expressions in $\mathcal{E}_{d,i}$ in the variables $x_1, x_2, \ldots, x_n$. Let $f(y_1, y_2, \ldots, y_s, z_1, z_2, \ldots, z_r)$ be a degree-d polynomial in $\mathbb{F}\langle X\rangle$. Then $f(g_1, g_2, \ldots, g_s, f_1^{-1}, f_2^{-1}, \ldots, f_r^{-1})$ is a rational expression (of inversion height $i+1$) in $\mathcal{E}_{d,i+1}$.

Black-box identity testing for rational expressions is not well understood in general. In particular, no efficient randomized algorithm seems to be known even for identity testing of the class $\mathcal{E}_{d,1}$. One source of difficulty is the subtle behaviour of rational expressions when evaluated on matrix algebras. For example, a surprising result of Bergman [5, Proposition 5.1] shows that there are rational expressions that are nonzero over a dense subset of 2×2 matrices but evaluate to zero on dense subsets of 3×3 matrices.

▶ Remark 2. In this connection, we note that Hrubeš and Wigderson [12] have observed that testing if a *correct* rational expression Φ (see [12], Section 2) is not identically zero is equivalent to testing if the rational expression Φ^{-1} is *correct.* I.e. testing if a correct rational expression of *inversion height i* is identically zero or not can be reduced to testing if a rational expression of *inversion height $i+1$* is correct or not. Furthermore, testing if a rational expression of *inversion height one* is correct can be done by applying (to each inversion operation in this expression) a theorem of Amitsur (see [18, 15]) which implies that a nonzero degree $2d-1$ noncommutative polynomial evaluated on $d \times d$ matrices will be invertible with high probability. However, this does not yield an efficient randomized identity testing algorithm for rational expressions of inversion height one. Because that requires testing correctness of expressions of inversion height two which is a question left open in their paper [12, Section 9].

Free Group Algebras

This motivates the study of black-box identity testing for rational expressions in the *free group algebra* $\mathbb{F}\langle X, X^{-1}\rangle$ which is a natural subclass of rational expressions of inversion height one, as we explain next.

We consider expressions in the free group algebra $\mathbb{F}\langle X, X^{-1}\rangle$, where $(X, X^{-1})^*$ denotes the free group generated by the n generators $X = \{x_1, x_2, \ldots, x_n\}$ and their inverses

$$X^{-1} = \{x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}\}.$$

Elements of the free group $(X, X^{-1})^*$ are words in X, X^{-1}. The only relations satisfied by the generators is $x_i x_i^{-1} = x_i^{-1} x_i = 1$ for all i. Thus, the elements in the free group $(X, X^{-1})^*$ are the *reduced words* which are words to which the above relations are not applicable.

The elements of the *free group algebra* $\mathbb{F}\langle X, X^{-1}\rangle$ are $\mathbb{F}$-linear combinations of the form

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

where each $w \in (X, X^{-1})^*$ is a reduced word. The *degree* of the expression f is defined as the maximum length of a word w such that $\alpha_w \neq 0$. The expression f is said to have *sparsity* s if there are s many reduced words w such that $\alpha_w \neq 0$ in f. We also use the notation $[w]f$ to denote the coefficient α_w of the reduced word w in the expression f.

The free noncommutative ring $\mathbb{F}\langle X\rangle$ is a subalgebra of $\mathbb{F}\langle X, X^{-1}\rangle$. Clearly, the elements of $\mathbb{F}\langle X, X^{-1}\rangle$ are a special case of rational expressions of *inversion height one*. I.e., we note that:

▶ **Proposition 3.** $\mathbb{F}\langle X, X^{-1}\rangle \subset \cup_{d>0}\mathcal{E}_{d,1}$.

Note that the rational expressions in $\mathbb{F}\langle X, X^{-1}\rangle$ allows inverses only of the variables x_i, whereas the *free skew field* $\mathbb{F}\langle\!\!\langle X \rangle\!\!\rangle$ contains all possible rational expressions (with inverses at any nested level).

Our results

Our main goal is to obtain black-box identity tests for rational expressions in the free group algebra $\mathbb{F}\langle X, X^{-1}\rangle$.

Our first result is an Amitsur-Levitzki type theorem [1] for $\mathbb{F}\langle X, X^{-1}\rangle$. Let A be an associative algebra with identity over $\mathbb{F}$. An expression $f \in \mathbb{F}\langle X, X^{-1}\rangle$ is an *identity* for A if

$$f(a_1, \ldots, a_n) = 0,$$

for all $a_i \in A$ such that a_i^{-1} is defined for each $i \in [n]$.

▶ **Theorem 4.** *Let $\mathbb{F}$ be any field of characteristic zero and $f \in \mathbb{F}\langle X, X^{-1}\rangle$ be a nonzero expression of degree d. Then f is not an identity for the matrix algebra $\mathbb{M}_{2d}(\mathbb{F})$.*

The following corollary is immediate.

▶ **Corollary 5** (Black-box identity testing in free group algebras)**.** *There is a black-box randomized* $\mathrm{poly}(n, d)$ *identity test for degree d expressions in $\mathbb{F}\langle X, X^{-1}\rangle$.*

If the black-box contains a sparse expression, we show efficient deterministic algorithms for identity testing and interpolation algorithm.

▶ **Theorem 6** (Reconstruction for sparse expressions)**.** *Let $\mathbb{F}$ be any field of characteristic zero and f is an expression in $\mathbb{F}\langle X, X^{-1}\rangle$ of degree d and sparsity s given as black-box. Then we can reconstruct f in deterministic* $\mathrm{poly}(n, d, s)$ *time with matrix-valued queries to the black-box.*

Nonzero polynomials in $\mathbb{F}\langle X\rangle$ of sparsity s cannot vanish on $O(\log s)$ dimensional matrix algebras [2]. We obtain a similar result for $\mathbb{F}\langle X, X^{-1}\rangle$: nonzero expressions in $\mathbb{F}\langle X, X^{-1}\rangle$ of degree D and sparsity s do not vanish on $O(\log s)$ dimensional matrices. It yields a randomized polynomial-time identity test if the black-box contains an expression f of exponential degree and exponential sparsity.

▶ **Theorem 7.** *Let $\mathbb{F}$ be any field of characteristic zero. Then, a degree-D expression $f \in \mathbb{F}\langle X, X^{-1}\rangle$ of sparsity s is not an identity for the matrix algebra $\mathbb{M}_k(\mathbb{F})$ for $k \geq c\log s$ for some small constant c.*

▶ **Corollary 8.** *Given a degree-D expression $f \in \mathbb{F}\langle X, X^{-1}\rangle$ of sparsity s as black box, we can check whether f is identically zero or not in randomized* $\mathrm{poly}(n, \log D, \log s)$ *time.*

▶ Remark 9. We have stated our results for fields of characteristic zero for simplicity. With suitable modifications, the results easily extend to fields of positive characteristic as discussed in Section 4.

Each of our proofs typically involves the construction of a nondeterministic substitution automaton $\mathcal{A}$. Consider any expression $f \in \mathbb{F}\langle X, X^{-1}\rangle$. For $\mathcal{A}$, let M_i denote its transition matrix for variable $x_i \in X$. The automaton $\mathcal{A}$ has the property that $f \not\equiv 0$ iff a specified entry of the matrix $f(M_1, M_2, \ldots, M_n)$ is nonzero. This entry will actually be a commutative polynomial (or a ratio of two commutative polynomials). Automata constructions for noncommutative PIT have been used before [4, 3, 2]. In this work, an important difference is that we have to deal with $\mathbb{F}$-linear combinations of words in $\{X, X^{-1}\}$. Thus, if M_i is the transition matrix for x_i then M_i^{-1} will be substituted for x_i^{-1}. Hence, in the construction we have to ensure M_i is invertible. Furthermore, when the automaton reads x_i^{-1} its state transition will be governed by M_i^{-1}. In order to ensure that the final output matrix is nonzero, the transition matrices for the x_i have to be chosen carefully, taking into account the above aspects.

Organization

The paper is organized as follows. In Section 2, we prove Theorem 4, Corollary 5, and Theorem 6. In Section 3, we prove Theorem 7 and Corollary 8. Finally, in Section 4, we discuss suitable modifications to extend our results for finite fields.

2 An Amitsur-Levitzki Type Theorem

The main idea in our proof is to efficiently encode expressions in $\mathbb{F}\langle X, X^{-1}\rangle$ as polynomials in a suitable commutative ring preserving the identity. Let $\mathbb{F}[Y, Z]$ denote the commutative ring $\mathbb{F}[y_{ij}, z_{ij}]_{i\in[n], j\in[d]}$ for $n, d \in \mathbb{N}$, where $Y = \{y_{ij} \mid i \in [n], j \in [d]\}$ and $Z = \{z_{ij} \mid i \in [n], j \in [d]\}$.

▶ **Definition 10.** *Define a map $\varphi : \mathbb{F}\langle X, X^{-1}\rangle \to \mathbb{F}[Y, Z]$ such that φ is identity on $\mathbb{F}$, and for each reduced word $w = x_{i_1}^{b_1} x_{i_2}^{b_2} \cdots x_{i_d}^{b_d}$,*

$$\varphi(x_{i_1}^{b_1} x_{i_2}^{b_2} \cdots x_{i_d}^{b_d}) = \prod_{j=1}^{d} (\mathbb{1}_{[b_j=1]} \cdot y_{i_j j} + \mathbb{1}_{[b_j=-1]} \cdot z_{i_j j}),$$

where $\mathbb{1}_{[b_j=b]} = 1$ if $b_j = b$ and $\mathbb{1}_{[b_j=b]} = 0$ otherwise.

By linearity the map φ is defined on all expressions in $\mathbb{F}\langle X, X^{-1}\rangle$. We observe the following properties of φ.

1. The map φ is injective on the reduced words $(X, X^{-1})^*$. I.e., it maps each reduced word $w \in (X, X^{-1})^*$ to a unique monomial over the commuting variables $Y \cup Z$.
2. Consequently, φ is identity preserving. I.e., an expression f in $\mathbb{F}\langle X, X^{-1}\rangle$ is identically zero if and only if its image $\varphi(f)$ is the zero polynomial in $\mathbb{F}[Y, Z]$.
3. φ preserves the sparsity of the expression. I.e., f in $\mathbb{F}\langle X, X^{-1}\rangle$ is s-sparse iff $\varphi(f)$ in $\mathbb{F}[Y, Z]$ is s-sparse.
4. Given the image $\varphi(f) \in \mathbb{F}[Y, Z]$ in its sparse description (i.e., as a linear combination of monomials), we can efficiently recover the sparse description of $f \in \mathbb{F}\langle X, X^{-1}\rangle$.

Given polynomials $f, f' \in \mathbb{F}[Y, Z]$, we say f and f' are *weakly equivalent*, if for each monomial m, $[m]f = 0$ if and only if $[m]f' = 0$, where $[m]f$ denotes the coefficient of monomial m in f.

Given a black-box expression f in $\mathbb{F}\langle X, X^{-1}\rangle$, we show how to evaluate it on suitable matrices and obtain a polynomial in $\mathbb{F}[Y, Z]$ that is *weakly equivalent to* $\varphi(f)$ as a specific entry of the resulting matrix. The matrix substitutions are based on automata constructions.

Similar ideas have been used earlier to design PIT algorithms for noncommutative polynomials [4]. However, since we are dealing with rational expressions, some difficulties arise. The matrix substitutions for the variables $x_1, \ldots, x_n$ are obtained as the corresponding transition matrices M_i of the automaton. The matrix substitution for x_i^{-1} will be M_i^{-1}. Therefore, we must ensure that the transition matrices M_i are invertible and sufficiently structured to be useful for the identity testing.

We first illustrate our construction for an example degree-2 expression $f = x_1 x_2^{-1} + x_2 x_1^{-1}$, where $X = \{x_1, x_2\}$.

The basic "building block" for the transition matrix M_i is the 2×2 block matrix

$$\begin{bmatrix} 0 & y_{ij} \\ \frac{1}{z_{ij}} & 0 \end{bmatrix},$$

whose inverse is

$$\begin{bmatrix} 0 & z_{ij} \\ \frac{1}{y_{ij}} & 0 \end{bmatrix}.$$

When the 2×2 block is the j^{th} diagonal block in M_i, the corresponding automaton will go from state $2j-1$ to state $2j$ replacing x_i by y_{ij} (or if x_i^{-1} occurs, it will replace it by z_{ij}).

We will keep the transition matrix M_i for x_i a block diagonal matrix with such 2×2 invertible blocks as the principal minors along the diagonal. In order to ensure this we introduce two new variables $W = \{w_1, w_2\}$ and substitute x_i by the word $w_i x_i w_i$ in the expression. This will ensure that we do not have two consecutive x_i in the resulting reduced words. In fact, between two X variables (or their inverses) we will have inserted exactly two W variables (or their inverses). Now, we define M_i for the above example as

$$M_i = \begin{bmatrix} 0 & y_{i1} & 0 & 0 \\ \frac{1}{z_{i1}} & 0 & 0 & 0 \\ 0 & 0 & 0 & y_{i2} \\ 0 & 0 & \frac{1}{z_{i2}} & 0 \end{bmatrix}, \qquad M_i^{-1} = \begin{bmatrix} 0 & z_{i1} & 0 & 0 \\ \frac{1}{y_{i1}} & 0 & 0 & 0 \\ 0 & 0 & 0 & z_{i2} \\ 0 & 0 & \frac{1}{y_{i2}} & 0 \end{bmatrix}.$$

The corresponding transitions of the automaton is shown in Figure 1.

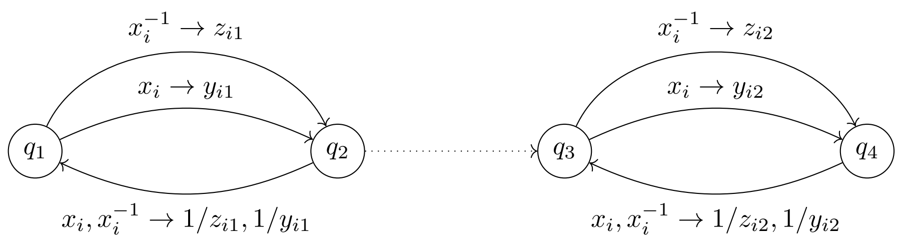

Figure 1 The transition diagram of the automaton for x variables.

We now describe the transition matrices N_i for w_i. The matrix N_i is also a 4×4 block diagonal matrix. There are three blocks along the diagonal. The first and third are 1×1 blocks of the identity. The second one is a 2×2 block for w_i-transitions from state q_2 to state q_3. It ensures that for any subword $w_1^{b_1} w_2^{b_2}$, $b_i \in \{1, -1\}$, in the resulting product matrix $N_1^{b_1} N_2^{b_2}$ the $(1,2)^{th}$ entry of the 2×2 block is nonzero. The corresponding transitions of the automaton is depicted in Figure 2.

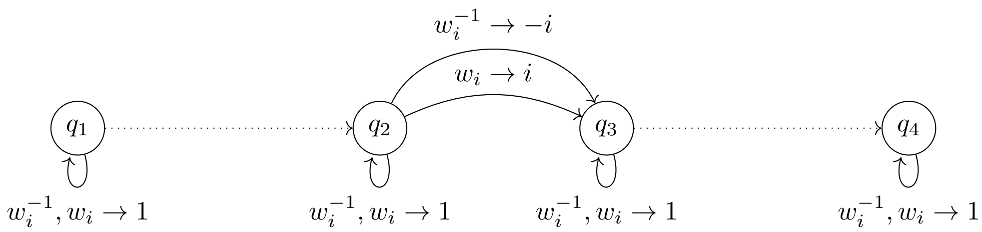

Figure 2 The transition diagram of the automaton for w variables.

$$N_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & i & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad N_i^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -i & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad N_i^{b_1} N_j^{b_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & b_1 i + b_2 j & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Hence, evaluating $f(N_1 M_1 N_1, N_2 M_2 N_2)$ we obtain (a polynomial weakly equivalent to) $\varphi(f)$ at the $(1,4)^{th}$ entry. The complete automaton is depicted in figure 3.

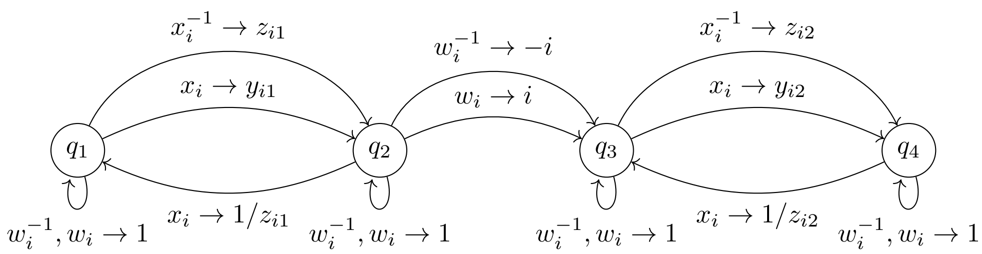

Figure 3 The transition diagram of the automaton.

We now explain the general construction. For $f \in \mathbb{F}\langle X, X^{-1}\rangle$ let $H_\ell(f)$ denote the degree-ℓ homogeneous part of f. We will denote by $\widehat{\varphi(H_\ell(f))}$ an arbitrary polynomial in $\mathbb{F}[Y, Z]$ weakly equivalent to $\varphi(H_\ell(f))$.

▶ **Lemma 11.** *Let $f \in \mathbb{F}\langle X, X^{-1}\rangle$ be a nonzero expression of degree d. There is an n-tuple of $2d \times 2d$ matrices $(M_1, M_2, \ldots, M_n)$ whose entries are either scalars, or variables $u \in Y \cup Z$, or their inverses $1/u$, such that*

$$(f(M_1, \ldots, M_n))_{1,2d} = \widehat{\varphi(H_d(f))}.$$

Furthermore, for each degree-d reduced word of $m = x_{i_1}^{b_1} x_{i_2}^{b_2} \cdots x_{i_d}^{b_d}$ in $\mathbb{F}\langle X, X^{-1}\rangle$,

$$[\varphi(m)]\widehat{\varphi(H_d(f))} = [m]f \cdot \prod_{j=1}^{d-1} (b_j \cdot i_j + b_{j+1} \cdot i_{j+1}). \tag{1}$$

Proof. Let e_{ij}, for $i, j \in [k]$, be the $(i,j)^{th}$ elementary matrix in $\mathbb{M}_k(\mathbb{F})$: its $(i,j)^{th}$ entry is 1 and other entries are 0.

We now define the transition matrices of the NFA for variables $\{w_i : 1 \le i \le n\}$ and $\{x_i : 1 \le i \le n\}$. For each $i \in [n]$, define 2×2 matrix $N'_i = e_{11} + e_{22} + i \cdot e_{12}$. Now N_i is a $2d \times 2d$ matrix defined as the block diagonal matrix,

$$N'_i = \begin{bmatrix} 1 & i \\ 0 & 1 \end{bmatrix}, \qquad N_i = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & N'_i & 0 & \dots & 0 & 0 \\ 0 & 0 & N'_i & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & N'_i & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

$$N_i'^{-1} = \begin{bmatrix} 1 & -i \\ 0 & 1 \end{bmatrix}, \qquad N_i^{-1} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & N_i'^{-1} & 0 & \dots & 0 & 0 \\ 0 & 0 & N_i'^{-1} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & N_i'^{-1} & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

Each $M_i, 1 \le i \le n$ is the $2d \times 2d$ block diagonal matrix where each 2×2 block $M'_{ij}, 1 \le j \le d$ is a 2×2 matrix defined as $M'_{i,j} = y_{ij} \cdot e_{12} + \frac{1}{z_{ij}} \cdot e_{21}$. Their inverses have a similar structure.

$$M'_{i,p} = \begin{bmatrix} 0 & y_{ip} \\ \frac{1}{z_{ip}} & 0 \end{bmatrix}, \qquad M_i = \begin{bmatrix} M'_{i,1} & 0 & 0 & \dots & 0 \\ 0 & M'_{i,2} & 0 & \dots & 0 \\ 0 & 0 & M'_{i,3} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & M'_{i,d} \end{bmatrix}.$$

$$M_{i,p}'^{-1} = \begin{bmatrix} 0 & z_{ip} \\ \frac{1}{y_{ip}} & 0 \end{bmatrix}, \qquad M_i^{-1} = \begin{bmatrix} M_{i,1}'^{-1} & 0 & 0 & \dots & 0 \\ 0 & M_{i,2}'^{-1} & 0 & \dots & 0 \\ 0 & 0 & M_{i,3}'^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & M_{i,d}'^{-1} \end{bmatrix}.$$

The corresponding NFA is depicted in Figure 4. We substitute each x_{i_j} by the $2d \times 2d$ matrix $N_{i_j} M_{i_j} N_{i_j}$. Each $x_{i_j}^{-1}$ is substituted by its inverse matrix $N_{i_j}^{-1} M_{i_j}^{-1} N_{i_j}^{-1}$.

Correctness:

Consider a degree-d reduced word $m = x_{i_1}^{b_1} x_{i_2}^{b_2} \cdots x_{i_d}^{b_d}$.

Following the automaton construction of Figure 4, $x_i^{b_i}$ occurring at position j is substituted by $(\mathbb{1}_{[b_i=1]} y_{ij} + \mathbb{1}_{[b_i=-1]} z_{ij})$. Moreover, for each position $j \in [d-1]$, the adjacent pair $x_{i_j}^{b_j} x_{i_{j+1}}^{b_{j+1}}$ produces a scalar factor $(b_j \cdot i_j + b_{j+1} \cdot i_{j+1})$ due to the product $N_{i_j}^{b_j} N_{i_{j+1}}^{b_{j+1}}$. Consequently, it follows that

$$(m(M_1, \dots, M_n))_{1,2d} = \prod_{j=1}^{d-1} (b_j \cdot i_j + b_{j+1} \cdot i_{j+1}) \prod_{j=1}^{d} ([b_j = 1] y_{i_j j} + [b_j = -1] z_{i_j j}).$$

As φ is a linear map, the lemma follows. ◀

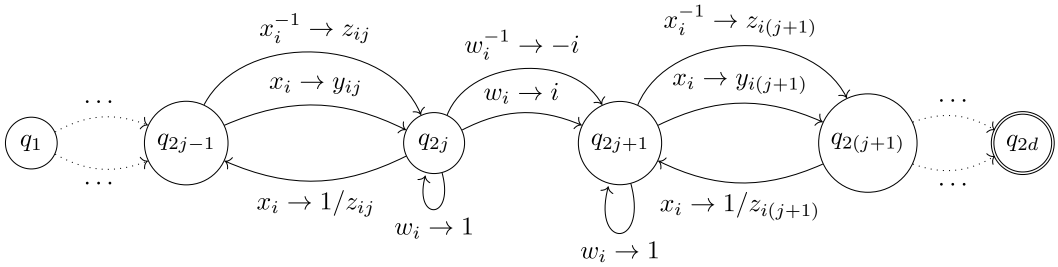

Figure 4 The transition diagram of the automaton.

2.1 Black-box identity testing for circuits in free group algebras

Proof of Theorem 4. The proof follows easily from Lemma 11. Lemma 11 says that if $f \in \mathbb{F}\langle X, X^{-1}\rangle$ is nonzero of degree d then the $(1, 2d)$ entry of the matrix $p(N_1M_1N_1, \ldots, N_nM_nN_n)$ is a nonzero polynomial in $\mathbb{F}[Y, Z]$. Hence f can not be an identity for $M_{2d}(\mathbb{F})$. ◀

Proof of Corollary 5. The identity testing algorithm follows from Theorem 4. We can randomly substitute for the variables and apply the Schwartz-Zippel-Demillo-Lipton Theorem [19, 21, 7]. This completes the proof of the Corollary 5. ◀

2.2 Reconstruction of sparse expressions

If the black-box contains an s-sparse expression in $\mathbb{F}\langle X, X^{-1}\rangle$, we give a poly$(s, n, d)$ deterministic interpolation algorithm (which also gives a deterministic identity testing for such expressions). We use a result of Klivans-Spielman [14, Theorem11] that constructs a test set in deterministic polynomial time for sparse commutative polynomials, which is used for the interpolation algorithm.

Proof of Theorem 6. Let the black-box expression f be s-sparse of degree d. By Lemma 11, a polynomial $\varphi(\widehat{H_d(f)})$ in $\mathbb{F}[Y, Z]$ is obtained at the $(1, 2d)^{th}$ entry of the matrix $f(M_1, \ldots, M_n)$, where $M_i \in \mathbb{M}_{2d}(\mathbb{F}[Y, Z])$ is as defined in Lemma 11. By Definition 10, $\varphi(f) \in \mathbb{F}[Y, Z]$ is s-sparse and has $2nd$ variables. Let $\mathcal{H}_{2nd,d,s}$ be the corresponding test set from [14] to interpolate a polynomial of degree d and s-sparse over $2nd$ variables. Querying the black-box on $M_1(\vec{h}), M_2(\vec{h}), \ldots, M_n(\vec{h})$ for each $\vec{h} \in \mathcal{H}_{2nd,d,s}$ we can interpolate the commutative polynomial $\varphi(\widehat{H_d(f)})$ and obtain an expression for $\varphi(\widehat{H_d(f)}) = \sum_{t=1}^{s} c_{m_t} m_t$ as a sum of monomials.

We will now adjust the extra scalar factors for each monomial in $\varphi(\widehat{H_d(f)})$ to obtain $\varphi(H_d(f))$. We can adjust this for each monomial as Lemma 11 shows that the extra scalar factor for the word $m = x_{i_1}^{b_1} x_{i_2}^{b_2} \cdots x_{i_\ell}^{b_\ell}$ is just $\alpha_{\varphi(m)} = \prod_{j=1}^{\ell-1}(b_j \cdot i_j + b_{j+1} \cdot i_{j+1})$. So we construct $\varphi(H_d(f)) = \sum_{t=1}^{s} \frac{c_{m_t}}{\alpha_{m_t}} m_t$ by removing the factors α_{m_t} for each monomial m_t. We now *invert* the map φ (using the 4^{th} property of Definition 10) on every monomial m_t to obtain $H_d(f)$ as a sum of degree d reduced words. This yields the expression for highest degree homogeneous component of f. We can repeat the above procedure on $f - H_d(f)$ and reconstruct the remaining homogeneous components of f. ◀

3 Black-box Identity Testing for Expressions of Exponential Degree and Exponential Sparsity

It is known [2] that a nonzero noncommutative polynomial of sparsity s cannot be an *identity* for $O(\log s)$ dimensional matrix algebras. In this section, we show a similar result for free group algebras. In particular, we prove that the dimension of the matrix algebra for which a nonzero free group algebra expression f does not vanish is logarithmic in the sparsity of f. It yields a randomized $\text{poly}(\log D, \log s, n)$ time identity testing algorithm when the black-box contains an expression of degree D and sparsity s.

We first recall the notion of *isolating index set* from [2].

▶ **Definition 12.** *Let $\mathcal{M} \subseteq \{X, X^{-1}\}^D$ be a subset of reduced words of degree D. An index set $I \subseteq [D]$ is an* isolating index set *for $\mathcal{M}$ if there is a word $m \in \mathcal{M}$ such that for each $m' \in \mathcal{M} \setminus \{m\}$ there is an index $i \in I$ for which $m[i] \neq m'[i]$. I.e. no other word in $\mathcal{M}$ agrees with m on all positions in the index set I. We say m is an* isolated word*.*

In the following lemma we show that $\mathcal{M}$ has an isolating index set of size $\log |\mathcal{M}|$. The proof is identical to [2]. Nevertheless, we give the simple details for completeness as we deal with both variables and their inverses.

▶ **Lemma 13** ([2])**.** *Let $\mathcal{M} \subseteq \{X, X^{-1}\}^D$ be reduced degree-D words. Then $\mathcal{M}$ has an isolating index set of size k which is bounded by $\log |\mathcal{M}|$.*

Proof. The words $m \in \mathcal{M}$ are indexed, where $m[i]$ denotes the variable (or the inverse of a variable) in the i^{th} position of m. Let $i_1 \leq D$ be the first index such that not all words agree on the i_1^{th} position. Let

$$S_j^+ = \{m : m[i_1] = x_j\}$$
$$S_j^- = \{m : m[i_1] = x_j^{-1}\}.$$

For some j, $|S_j^+|$ or $|S_j^-|$ is of size at most $|\mathcal{M}|/2$. Let S_j^b denote that subset, $b \in \{+, -\}$. We replace $\mathcal{M}$ by S_j^b 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 *isolating word* m. ◀

Proof of Theorem 7

Proof. Let $k = 4(k' + 1)$ where k' is the size of the isolating set I. As in Section 2, we substitute each x_i by $w_i x_i w_i$, where $w_i, i \in [n]$ are n new variables. The transition matrices for w_i and x_i are denoted by N_i and M_i respectively.

For $1 \leq i \leq n$, we define $k \times k$ matrix N_i as a block diagonal matrix of $k' + 1$ many copies of a 4×4 matrix N'_i where $N'_i = I_4 + i(e_{12} + e_{34} + e_{32} + e_{14})$.

$$N'_i = \begin{bmatrix} 1 & i & 0 & i \\ 0 & 1 & 0 & 0 \\ 0 & i & 1 & i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad N_i = \begin{bmatrix} N'_i & 0 & 0 & \ldots & 0 \\ 0 & N'_i & 0 & \ldots & 0 \\ 0 & 0 & N'_i & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & N'_i \end{bmatrix},$$

$$N_i'^{-1} = \begin{bmatrix} 1 & -i & 0 & -i \\ 0 & 1 & 0 & 0 \\ 0 & -i & 1 & -i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad N_i^{-1} = \begin{bmatrix} N_i'^{-1} & 0 & 0 & \dots & 0 \\ 0 & N_i'^{-1} & 0 & \dots & 0 \\ 0 & 0 & N_i'^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & N_i'^{-1} \end{bmatrix}.$$

Notice that

$$N_i'^{b_1} N_j'^{b_2} = \begin{bmatrix} 1 & (b_1 i + b_2 j) & 0 & (b_1 i + b_2 j) \\ 0 & 1 & 0 & 0 \\ 0 & (b_1 i + b_2 j) & 1 & (b_1 i + b_2 j) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We now define the $k \times k$ transition matrix M_i as a block diagonal matrix,

$$M'_{i,j} = \begin{bmatrix} 0 & y_{ij} \\ \frac{1}{z_{ij}} & 0 \end{bmatrix}, \qquad M'_{\xi_i} = \begin{bmatrix} 0 & \xi_i \\ \frac{1}{\xi_i} & 0 \end{bmatrix},$$

$$M_i = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & M_{\xi_1} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & M'_{i,1} & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & M_{\xi_2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & M_{\xi_{k'+1}} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

These matrices can be seen as the transitions of a suitable NFA. We sketch the construction of this NFA.

Let $I = \{i_1, \dots, i_{k'}\}$ be an isolating set such that $i_1 < \dots < i_{k'}$. Intuitively, the NFA does one of two operations on each symbol (a variable or its inverse) of the input expression: a *Skip* or an *Encode*. In a *Skip* stage, the NFA deals with positions that are not part of the (guessed) isolating index set. In this stage, the NFA substitutes the w_i variables by suitable scalars (coming from the N'_i matrices) and x_i variables by block variables $\{\xi_1, \dots \xi_{k'+1}\}$. The NFA nondeterministically decides whether the *Skip* stage is over and it enters the *Encode* stage for a guessed index of the isolating set. It substitutes x_i and x_i^{-1} variables by y_{ij} and z_{ij} respectively. Fig. 5 summarizes the action of the NFA.

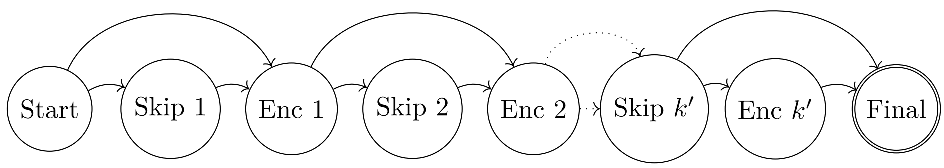

Figure 5 The transition diagram of the automaton.

Define $\hat{f}$ in $\mathbb{F}(Y, Z, \overline{\xi})$ to be rational function we obtain at the $(1,k)^{th}$[2] entry by evaluating the expression $f(N_1 M_1 N_1, \ldots, N_n M_n N_n)$. Notice that, the isolating word m of degree D will be of following form $m = W_1 x_{i_1}^{b_{i_1}} W_2 x_{i_2}^{b_{i_2}} \cdots W'_k x_{i'_k}^{b_{i'_k}} W_{k'+1}$ where each subword $W_j = x_{j_1}^{b_1} x_{j_2}^{b_2} \cdots x_{j_{\ell_j}}^{b_{\ell_j}}$ is of length $\ell_j \geq 0$, where some of the W_j could be the empty word as well.

We refer to an NFA transition $q_i \to q_j$ as a *forward edge* if $i < j$ and a *backward edge* if $i > j$. We classify the backward edges in three categories based on the substitution on the edge-label. We say, a backward edge is of *type A* if a variable is substituted by a scalar value; a backward edge is of *type B* if a variable is substituted by $\frac{1}{\xi_j}$ for some j; a backward edge is of *type C* if a variable is substituted by $\frac{1}{y_{ij}}$ or $\frac{1}{z_{ij}}$ for some i, j.

Consider a walk of the NFA on an input word m that reaches state k using only *type A* backward edges. In that case, m is substituted by $\alpha \cdot \hat{m}$ where $\hat{m}$ is a monomial over $\{Y, Z, \xi\}$ of same degree,

$$\hat{m} = \prod_{j=1}^{k'+1} \xi_j^{\ell_j} \cdot \prod_{j=1}^{k'} ([b_{i_j} = 1] y_{i_j j} + [b_{i_j} = -1] z_{i_j j}).$$

and α is some nonzero constant obtained as a product of $[m]f$ with the scalars obtained as substitutions from the edges involving the w_i variables in the *Skip* stages. Indeed, as we can see from the entries of product matrices $N_i'^{b_1} \cdot N_j'^{b_2}$, where $b_1, b_2 \in \{-1, 1\}$, the scalar α is a product of $[m]f$ with terms of the form $b_1 i + b_2 j$, for $i \neq j$, each of which is nonzero for any reduced word.

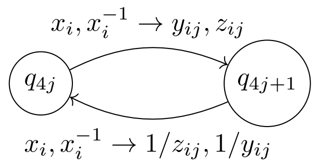

Figure 6 The transition diagram of the automaton at *Encode* stage.

▷ Claim 14.

$[\hat{m}]\hat{f} \neq 0$ iff $[m]f \neq 0$.

Proof. It suffices to show that for any word $m' \neq m$, where m' has degree $\leq D$, no walks of the NFA accepting m' generate $\hat{m}$ after substitution. For a computation path J, the monomial m_J in $\hat{f}$ has two parts, let us call it $skip_J$ and $encode_J$ where $skip_J$ is a monomial over $\{\xi_1, \ldots, \xi_{k'+1}\}$ and $encode_J$ is a monomial over $\{y_{ij}, z_{ij}\}_{i \in [n], j \in [k']}$. If the computation path J (which is different from the computation path described above for $\hat{m}$) uses only *type A* backward edges, then necessarily $m_J \neq \hat{m}$ from the definition of *isolating index set*. This argument is analogous to the argument given in [2].

Now consider a walk J which involves backward edges of other types. Let us first consider those walks that take backward edges only of *type A* and *type B*. Such a walk still produces a monomial over $\{y_{ij}, z_{ij}\}_{i \in [n], j \in [k']}$ and $\{\xi_i\}_{1 \leq i \leq k'+1}$ because division only by ξ_i variables

[2] Recall that $k = 4(k'+1)$ where k' is the size of an isolating set.

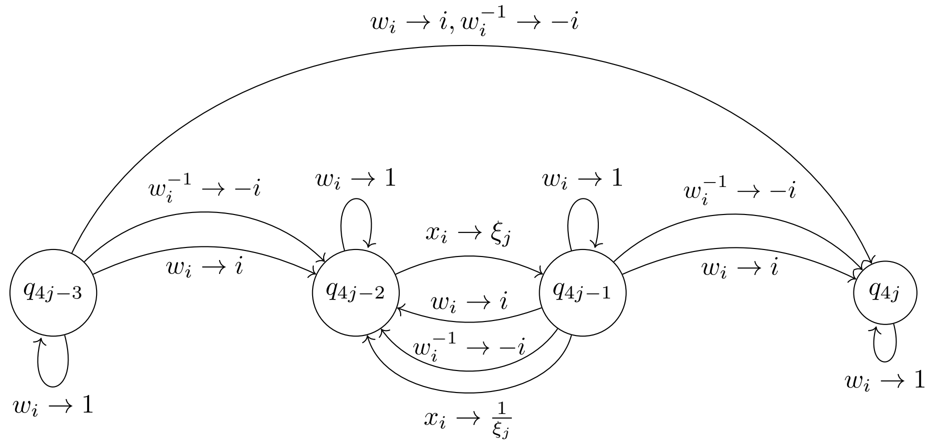

Figure 7 The transition diagram of the automaton at *Skip* stage.

occur in the resulting expression. Since $\hat{m}$ is of highest degree, the total degree of these monomials is strictly lesser than degree of $\hat{m}$. For those walks that take at least one backward edge of *type* C, a rational expression in $\{y_{ij}, z_{ij}\}_{i\in[n],j\in[k']}$ and $\{\xi_i\}_{1\le i\le k'+1}$ is produced (as there is division by y_{ij} or z_{ij} variables). As the sum of the degree of the numerator and degree of the numerator is bounded by the total degree, the degree of the numerator is smaller than degree of $\hat{m}$.

Thus the $(1,k)^{th}$ entry of the output matrix is of the form $\sum_{i=1}^{N_1} c_i m_i + \sum_{j=1}^{N_2} r_j$ where $\{m_1, \ldots, m_{N_1}\}$ are monomials arising from different walks (w.l.o.g. assume that $m_1 = \hat{m}$) and $\{r_1, \ldots, r_{N_2}\}$ are the rational expressions from the other walks (due to the backward edges of *type* C). Note that, denominator in each r_j is a monomial over Y, Z of degree at most D. Let $L = \prod_{i=1}^{n}\prod_{j=1}^{k'} y_{i,j}^D \cdot z_{i,j}^D$. Now, we have,

$$\sum_{i=1}^{N_1} c_i m_i + \sum_{j=1}^{N_2} r_j = \frac{1}{L}\cdot\left(\sum_{i=1}^{N_1} c_i m_i L + \sum_{j=1}^{N_2} p_j\right).$$

Since $\hat{m}L \neq m_i L$ for any $i \in \{2, \ldots, N_1\}$ and degree of each $p_j <$ degree of $\hat{m}L$ for any $j \in \{1, \ldots, N_2\}$, the numerator of the final expression is a nonzero polynomial in $\mathbb{F}[Y, Z, \overline{\xi}]$. $\lhd$

The above proof shows that the matrix $f(N_1M_1N_1, \ldots, N_nM_nN_n)$ is nonzero with rational entries in $\mathbb{F}[Y, Z, \overline{\xi}]$. Each entry is a linear combination of terms of the form m_1/m_2, where m_1 and m_2 are monomials in $Y \cup Z \cup \{\xi_1, \ldots, \xi_{k'+1}\}$ of degree bounded by D. Note that, the matrix dimension is $k = c\log s$ for some constant c. This completes the proof of Theorem 7. ◀

To get an identity testing algorithm, we can do random substitutions.The matrix dimension is $\log s$ and the overall running time of the algorithm is $\mathrm{poly}(n, \log s, \log D)$. This also proves Corollary 8. ◀

▶ Remark 15. For algorithmic purposes, we note that Theorem 4 is sometimes preferable to Theorem 7. For instance, the encoding used in Theorem 7 does not preserve the sparsity of the polynomial as required in the sparse reconstruction result (see Theorem 6).

4 Adaptation for Fields of Positive Characteristic

Let $\mathbb{F}$ be any finite field of characteristic p. We will ensure that for each word m in the free group algebra, the scalar α_m (see Equation 1) produced by the automaton described in Section 2 is not zero in $\mathbb{F}$. Recall that, reading $w_i^{b_i} w_j^{b_j}$ for two consecutive positions, the automaton produces a scalar $(b_i \cdot i + b_j \cdot j)$ where $b_i, b_j \in \{-1, +1\}$. Moreover, this is the only way the automaton produces a scalar and for each m, α_m is a product of such terms. Hence, it suffices to ensure that for each pair $i, j \in [n]$, $(b_i \cdot i + b_j \cdot j) \neq 0$. Similarly, it ensures that the scalar produced by the automaton described in Section 3 is nonzero.

We note that, if p is more than $2n$ then each term $(b_i \cdot i + b_j \cdot j) \neq 0 \pmod p$ where $b_i, b_j \in \{-1, +1\}$ and $i, j \in [n]$. This results in a dependence on the characteristic of the base field for the analogous statements of Theorems 4, 7 for finite field. Additionally, for Theorem 4, the $(1, 2d)^{th}$ entry of the output matrix is a polynomial of degree d, and for Theorem 7, the degrees of the numerator polynomials in the rational expression of the output matrix is bounded by some scalar multiple of $nD \log s$. This lower bounds the size of the fields in the application. We summarize the above discussion in the following.

▶ **Observation 16.** *We can obtain results analogous to Theorem 4 and Theorem 7 for finite fields of characteristic more than $2n$ and sizes at least $d+1$ and $\Omega(nD \log s)$ respectively.*

However, the algorithms presented in Theorem 6 and Corollaries 5, 8 can be modified to work for finite fields of any characteristic. To this end, we first notice the following simple fact.

▶ **Proposition 17.** *Let $\mathbb{F}$ be a finite field of characteristic $p \leq 2n$. In We can find elements $\alpha_1, \alpha_2, \ldots, \alpha_n$ from a suitable (deterministically constructed) small extension field $\mathbb{F}'$ of $\mathbb{F}$ in deterministic $\mathrm{poly}(n)$ time, such that for any $b_i \in \{-1, 1\}, 1 \leq i \leq n$ we have*

For each $1 \leq i < j \leq n, \; b_i\alpha_i + b_j\alpha_j \neq 0.$

Let $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}'$ as given by the above proposition. We modify the matrix N'_i in the proof of Theorem 6 and Corollary 5 as

$$N'_i = \begin{bmatrix} 1 & \alpha_i \\ 0 & 1 \end{bmatrix},$$

and in Corollary 8 we modify N'_i as

$$N'_i = \begin{bmatrix} 1 & \alpha_i & 0 & \alpha_i \\ 0 & 1 & 0 & 0 \\ 0 & \alpha_i & 1 & \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For each pair $i, j \in [n]$, $(b_i \cdot \alpha_i + b_j \cdot \alpha_j) \neq 0$ by Proposition 17. Thus, for each word m, the scalar α_m produced by the automata are nonzero in the extension field $\mathbb{F}'$ as well. Furthermore, the test set of [14] works for all fields. Hence Theorem 6 holds for all finite fields too. To obtain Corollaries 5 and 8, we will do random substitutions from a suitable small degree extension field and use Schwartz-Zippel-Demillo-Lipton Theorem [19, 21, 7]. In summary, our algorithms in the paper can be adapted to work for all fields.

Proof of Proposition 17. Define polynomial $g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ as

$$g(x_1, x_2, \ldots, x_n) = \prod_{1 \leq i < j \leq n} (x_i + x_j) \cdot (x_i - x_j).$$

We substitute y^i for $x_i, 1 \leq i \leq n$. Then $g(y, y^2, \ldots, y^n) = G(y) \in \mathbb{F}[y]$ is a univariate polynomial of degree at most $2n^3$. Using standard techniques, in deterministic polynomial time we can construct an extension field $\mathbb{F}'$ of $\mathbb{F}$ such that $|\mathbb{F}'|$ is of $\mathrm{poly}(n) \geq 2n^3 + 1$ size. We can find an element $\alpha \in \mathbb{F}'$ such that $G(\alpha) \neq 0$ and set $\alpha_i = \alpha^i, 1 \leq i \leq n$. ◀

References

1 A. S. Amitsur and J. Levitzki. Minimal Identities for Algebras. *Proceedings of the American Mathematical Society*, 1(4):449–463, 1950. URL: `http://www.jstor.org/stable/2032312`.

2 Vikraman Arvind, Pushkar S. Joglekar, Partha Mukhopadhyay, and S. Raja. Randomized polynomial time identity testing for noncommutative circuits. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 831–841, 2017. `doi:10.1145/3055399.3055442`.

3 Vikraman Arvind and Partha Mukhopadhyay. The ideal membership problem and polynomial identity testing. *Inf. Comput.*, 208(4):351–363, 2010. `doi:10.1016/j.ic.2009.06.003`.

4 Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New Results on Non-commutative and Commutative Polynomial Identity Testing. *Computational Complexity*, 19(4):521–558, 2010. `doi:10.1007/s00037-010-0299-8`.

5 George M Bergman. Rational relations and rational identities in division rings. *Journal of Algebra*, 43(1):252–266, 1976. `doi:10.1016/0021-8693(76)90159-9`.

6 Andrej Bogdanov and Hoeteck Wee. More on Noncommutative Polynomial Identity Testing. In *20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA*, pages 92–99, 2005. `doi:10.1109/CCC.2005.13`.

7 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. `doi:10.1016/0020-0190(78)90067-4`.

8 Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics*, 310:44–63, 2017. `doi:10.1016/j.aim.2017.01.018`.

9 Michael Forbes and Amir Shpilka. Quasipolynomial-time Identity Testing of Non-Commutative and Read-Once Oblivious Algebraic Branching Programs. *Foundations of Computer Science, 1975., 16th Annual Symposium on*, September 2012. `doi:10.1109/FOCS.2013.34`.

10 Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. A Deterministic Polynomial Time Algorithm for Non-commutative Rational Identity Testing. *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117, 2016.

11 Pavel Hrubes and Amir Yehudayoff. Arithmetic Complexity in Ring Extensions. *Theory of Computing*, 7:119–129, 2011.

12 Pavel Hrubeš and Avi Wigderson. Non-commutative arithmetic circuits with division. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 49–66, 2014. `doi:10.1145/2554797.2554805`.

13 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *computational complexity*, 27(4):561–593, December 2018. `doi:10.1007/s00037-018-0165-7`.

14 Adam R. Klivans and Daniel Spielman. Randomness Efficient Identity Testing of Multivariate Polynomials. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 216–223, New York, NY, USA, 2001. ACM. `doi:10.1145/380752.380801`.

15 Tsiu-Kwen Lee and Yiqiang Zhou. Right ideals generated by an idempotent of finite rank. *Linear Algebra and its Applications*, 431:2118–2126, November 2009. `doi:10.1016/j.laa.2009.07.005`.

16 Noam Nisan. Lower Bounds for Non-Commutative Computation (Extended Abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418, 1991. `doi:10.1145/103418.103462`.

17 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. doi:10.1007/s00037-005-0188-8.

18 Louis Halle Rowen. *Polynomial identities in ring theory*. Pure and Applied Mathematics. Academic Press, 1980.

19 Jacob T. Schwartz. Fast Probabilistic algorithm for verification of polynomial identities. *J. ACM.*, 27(4):701–717, 1980.

20 Volker Strassen. Vermeidung von Divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973. URL: http://eudml.org/doc/151394.

21 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. of the Int. Sym. on Symbolic and Algebraic Computation*, pages 216–226, 1979.