On Efficient Noncommutative Polynomial Factorization via Higman Linearization

Vikraman Arvind ✉
Institute of Mathematical Sciences, Chennai, India

Pushkar S. Joglekar ✉
Vishwakarma Institute of Technology, Pune, India

Abstract

In this paper we study the problem of efficiently factorizing polynomials in the free noncommutative ring $\mathbb{F}\langle x_1, x_2, \ldots, x_n\rangle$ of polynomials in noncommuting variables $x_1, x_2, \ldots, x_n$ over the field $\mathbb{F}$. We obtain the following result:

- We give a randomized algorithm that takes as input a noncommutative arithmetic formula of size s computing a noncommutative polynomial $f \in \mathbb{F}\langle x_1, x_2, \ldots, x_n\rangle$, where $\mathbb{F} = \mathbb{F}_q$ is a finite field, and in time polynomial in s, n and $\log_2 q$ computes a factorization of f as a product $f = f_1 f_2 \cdots f_r$, where each f_i is an irreducible polynomial that is output as a noncommutative algebraic branching program.
- The algorithm works by first transforming f into a linear matrix L using Higman's linearization of polynomials. We then factorize the linear matrix L and recover the factorization of f. We use basic elements from Cohn's theory of free ideals rings combined with Ronyai's randomized polynomial-time algorithm for computing invariant subspaces of a collection of matrices over finite fields.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Noncommutative Polynomials, Arithmetic Circuits, Factorization, Identity testing

Digital Object Identifier 10.4230/LIPIcs.CCC.2022.12

Related Version *Full Version*: `https://arxiv.org/pdf/2202.09883.pdf`

Funding *Pushkar S. Joglekar*: Author would like to thank Science and Engineering Research Board for the funding through the MATRICS project, File no. MTR/2018/001214.

1 Introduction

Let $\mathbb{F}$ be any field and $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n free noncommuting variables. Let X^* denote the set of all free words (which are monomials) over the alphabet X with concatenation of words as the monoid operation and the empty word ϵ as identity element.

The *free noncommutative ring* $\mathbb{F}\langle X\rangle$ consists of all finite $\mathbb{F}$-linear combinations of monomials in X^*, where the ring addition $+$ is coefficient-wise addition and the ring multiplication $*$ is the usual convolution product. More precisely, let $f, g \in \mathbb{F}\langle X\rangle$ and let $f(m) \in \mathbb{F}$ denote the coefficient of monomial m in polynomial f. Then we can write $f = \sum_m f(m)m$ and $g = \sum_m g(m)m$, and in the product polynomial fg for each monomial m we have

$$fg(m) = \sum_{m_1 m_2 = m} f(m_1)g(m_2).$$

The *degree* of a monomial $m \in X^*$ is the length of the monomial m, and the degree $\deg f$ of a polynomial $f \in \mathbb{F}\langle X\rangle$ is the degree of a largest degree monomial in f with nonzero coefficient. For polynomials $f, g \in \mathbb{F}\langle X\rangle$ we clearly have $\deg(fg) = \deg f + \deg g$.

37th Computational Complexity Conference (CCC 2022).
Editor: Shachar Lovett; Article No. 12; pp. 12:1–12:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

A *nontrivial factorization* of a polynomial $f \in \mathbb{F}\langle X\rangle$ is an expression of f as a product $f = gh$ of polynomials $g, h \in \mathbb{F}\langle X\rangle$ such that $\deg g > 0$ and $\deg h > 0$. A polynomial $f \in \mathbb{F}\langle X\rangle$ is *irreducible* if it has no nontrivial factorization and is *reducible* otherwise. For instance, all degree 1 polynomials in $\mathbb{F}\langle X\rangle$ are irreducible. Clearly, by repeated factorization every polynomial in $\mathbb{F}\langle X\rangle$ can be expressed as a product of irreducibles.

In this paper we study the algorithmic complexity of polynomial factorization in the free ring $\mathbb{F}\langle X\rangle$. The factorization algorithm is by an application of Higman's linearization process followed by factorization of a matrix with linear entries (under some technical conditions) using Cohn's factorization theory.

It is interesting to note that Higman's linearization process [15] has been used to obtain a deterministic polynomial-time algorithm for the RIT problem. That is, the problem of testing if a noncommutative rational formula (which computes an element of the free skew field $\mathbb{F}\langle\!\langle X\rangle\!\rangle$) is zero on its domain of definition [13, 17, 18, 16].

1.1 Overview of the results

The main result of the paper is the following.

▶ **Theorem** (Main Theorem). *Given a multivariate noncommutative polynomial $f \in \mathbb{F}_q\langle X\rangle$ for a finite field*[1] *$\mathbb{F}_q$ by a noncommutative arithmetic formula of size s as input, a factorization of f as a product $f = f_1 f_2 \cdots f_r$ can be computed in randomized time* $\mathrm{poly}(s, \log_2 q, |X|)$*, where each $f_i \in \mathbb{F}_q\langle X\rangle$ is an irreducible polynomial that is output as an algebraic branching program.*

The proof has three broad steps described below.

- **Higman linearization and Cohn's factorization theory.** Briefly, given a noncommutative polynomial $f \in \mathbb{F}\langle X\rangle$ by a formula, we can transform it into a linear matrix L such that $f \oplus I = PLQ$, where P is an upper triangular matrix with polynomial entries and all 1's diagonal and Q is a lower triangular matrix with polynomial entries and all 1's diagonal, P and Q are the matrices implementing the sequence of row and column operations required for the Higman linearization process. Cohn's theory of factorization of noncommutative linear matrices gives us sufficient information about the structure of irreducible linear matrices.

- **Ronyai's common invariant subspace algorithm.** Next, the most important tool algorithmically, is Ronyai's algorithm for computing common invariant subspaces of a collection of matrices over finite fields [24]. We show that Ronyai's common invariant subspace algorithm can be repeatedly applied to factorize a linear matrix $L = A_0 + \sum_{i=1}^{n} A_i x_i$, into a product of irreducible linear matrices provided A_0 is invertible and $[A_1 A_2 \cdots A_n]$ has full row rank or $[A_1^T A_2^T \cdots A_n^T]^T$ has full column rank. The later conditions are called as right and left monicity of the linear matrix L respectively. With some technical work we can ensure these conditions for a linear matrix L that is produced from a polynomial f by Higman linearization. Then Ronyai's algorithm yields the factorization of L into a product of irreducible linear matrices (upto multiplication by units).

[1] We present the detailed randomized algorithm over large finite fields. In the case of small finite fields we obtain a deterministic $\mathrm{poly}(s, q, |X|)$ time algorithm with minor modifications.

- **Recovering the factors of f.** Finally, we design a simple linear algebraic algorithm for trivializing a matrix product $AB = 0$, where A is a linear matrix and B is a column vector of polynomials from $\mathbb{F}\langle X\rangle$, using which we are able to extract the irreducible factors of f from the factors of L. An invertible matrix M with polynomial entries *trivializes* the relation $AB = 0$ if the modified relation $(AM)(M^{-1}B) = 0$ has the property that for every index i either the i^{th} column of AM is zero or the i^{th} row of $M^{-1}B$ is zero. While such matrices M exist for any matrix product $AB = 0$ with entries from $\mathbb{F}\langle X\rangle$, we obtain an efficient algorithm in the special case when A is linear and B's entries are polynomials computed by small arithmetic circuits. This special case is sufficient for our application.

There are some additional technical aspects we need to deal with. Let $L = A_0 + \sum_{i=1}^{n} A_i x_i$ be the linear matrix obtained from $f \in \mathbb{F}_q\langle X\rangle$ by Higman linearization, where $X = \{x_1, x_2, \ldots, x_n\}$ and $A_i \in \mathbb{F}_q^{d\times d}, 0 \le i \le n$. If A_0 is an invertible matrix then it turns out that the problem of factorizing L can be directly reduced to the problem of finding a common invariant subspace for the matrices $A_0^{-1}A_i, 1 \le i \le n$. In general, however, A_0 is not invertible. Two cases arise:

(a) The polynomial f is *commutatively nonzero*. That is, it is nonzero on $\mathbb{F}_q^n$ (or on $\mathbb{F}^n$ for a small extension field $\mathbb{F}$). In this case, by the DeMillo-Lipton-Schwartz-Zippel Lemma [9, 25, 27], we can do a linear shift of the variables $x_i \leftarrow x_i + \alpha_i$ in the polynomial f, for α_i randomly picked from $\mathbb{F}_q$ (or $\mathbb{F}$). Let the resulting polynomial be f' and let its Higman linearization be $L_{f'}$. In $L_{f'}$ the constant matrix term A'_0 will be invertible with high probability, and the reduction steps outlined above will work for $L_{f'}$. Furthermore, from the factorization of f' we can efficiently recover the factorization of f. Section 4 deals with Case (a), with Theorem 32 summarizing the algorithm for factorizing f. Theorem 28 describes the algorithm for factorization of the linear matrix $L_{f'}$, and the factor extraction lemma (Lemma 31) allows us to efficiently recover the factorization of f' from the factorization of $L_{f'}$.

(b) In the second case, suppose f is zero on all scalars. Then, for example by Amitsur's theorem [1], for a random matrix substitution $x_i \leftarrow M_i \in \mathbb{F}^{2s\times 2s}$ the matrix $f(M_1, M_2, \ldots, M_n)$ is *invertible* with high probability, where s is the formula size of f.[2] [3] Accordingly, we can consider the factorization problem for shifted and dilated linear matrix $L' = A_0 \otimes I_\ell + \sum_{i=1}^{n} A_i \otimes (Y_i + M_i)$ which will have the constant matrix term invertible, where each Y_i is an $\ell \times \ell$ matrix of distinct noncommuting variables, where $\ell = 2s$. Recovering the factorization of L from the factorization of L' requires some additional algorithmic work based on linear algebra. A lemma from [14] (refer Section 5 and the Appendix for the details) turns out to be crucial here. The algorithm handling Case (b) is described in Section 5. Indeed, the new aspect of the algorithm is factorization of the dilated matrix L' from which we recover the factorization of the Higman linearization L_f of f. The remaining algorithm steps are exactly as in Section 4.

1.2 Small Finite fields

We now briefly explain the deterministic poly$(s, q, |X|)$ time factorization algorithm (when $\mathbb{F}_q$ is small). There are two places in the factorization algorithm outlined above where randomization is used: first, to obtain a matrix tuple $(M_1, M_2, \ldots, M_n)$ such that $f(M_1, M_2, \ldots, M_n)$

[2] Amitsur's theorem strengthens the Amitsur-Levitski theorem [2] often used in noncommutative PIT algorithms [5].

[3] In the actual algorithm we pick the matrices M_i using a result from [10]

is invertible, which ensures that the constant matrix term of the linear matrix L' is invertible. When $q = \Omega(d)$, where $d = \deg f$, it suffices to randomly pick $M_i \in \mathbb{F}_q^{2s\times 2s}$. However, if $q < d$ we can choose entries of the matrices M_i from a small extension field $\mathbb{F}_{q^k}$ such that $q^k = \Omega(d)$. Thereby, we will obtain factorization of L' and subsequently that of the polynomial f over the extension field $\mathbb{F}_{q^k}$. However, we can use the fact that the finite field $\mathbb{F}_{q^k}$ can be embedded using the regular representation of the elements of $\mathbb{F}_{q^k}$ in the matrix algebra $\mathbb{F}_q^{k\times k}$. Thus, we can obtain from $(M_1, M_2, \ldots, M_n)$ a matrix tuple $(M'_1, M'_2, \ldots, M'_n)$ with $M'_i \in \mathbb{F}_q^{2sk\times 2sk}$ such that $f(M'_1, M'_2, \ldots, M'_n)$ is invertible. This will ensure that the linear matrix L' can be factorized over the field $\mathbb{F}_q$ which will allow us to obtain a complete factorization of f into irreducible factors over $\mathbb{F}_q$.

In order to get a deterministic polynomial-time algorithm for finding such matrices $M'_i, 1 \leq i \leq n$ we will use the fact that the polynomial f is given by a small noncommutative formula and hence has a small algebraic branching program. Then, using ideas from [23, 11, 4] we can easily find such matrices M'_i in deterministic polynomial time.

Next, we notice that Ronyai's algorithm for finding common invariant subspaces of matrices over $\mathbb{F}_q$ is essentially a polynomial-time reduction to univariate polynomial factorization over $\mathbb{F}_q$. We can use Berlekamp's deterministic poly(q, D) algorithm for the factorization of univariate degree D polynomials over $\mathbb{F}_q$. Putting it together, we can obtain a deterministic poly$(s, q, |X|)$ time algorithm for factorization of $f \in \mathbb{F}_q\langle X\rangle$ as a product of irreducible factors over $\mathbb{F}_q$.

Unfortunately, the algorithm outlined above does not yield an efficient algorithm for noncommutative polynomial factorization over rationals. The bottleneck is the problem of computing common invariant subspaces for a collection of matrices over $\mathbb{Q}$. Ronyai's algorithm for the problem over finite fields [24] builds on the decomposition of finite-dimensional associative algebras over fields. Given an algebra $\mathcal{A}$ over a finite field $\mathbb{F}_q$ the algorithm decomposes $\mathcal{A}$ as a direct sum of minimal left ideals of $\mathcal{A}$ which is used to find nontrivial common invariant subspaces. However, as shown by Friedl and Ronyai [12], over rationals the problem of decomposing a *simple* algebra as a direct sum of minimal left ideals is at least as hard as factoring square-free integers.

1.3 Related research

The study of factorization in noncommutative rings is systematically investigated as part of Cohn's general theory of noncommutative free ideal rings [7, 8] which is based on the notion of the weak algorithm. In fact, there is a hierarchy of weak algorithms generalizing the division algorithm for commutative integral domains [7].

To the best of our knowledge, the complexity of noncommutative polynomial factorization has not been studied much, unlike the problem of commutative polynomial factorization [26, 19, 20]. Prior work on the complexity of noncommutative polynomial factorization we are aware of is [3] where efficient algorithms are described for the problem of factoring *homogeneous* noncommutative polynomials (which enjoy the unique factorization property, and indeed the algorithms in [3] crucially use the unique factorization property). When the input homogeneous noncommutative polynomial has a small noncommutative arithmetic circuit (even given by a black-box as in Kaltofen's algorithms [19, 20]) it turns out that the problem is efficiently reducible to commutative factorization by set-multilinearizing the given noncommutative polynomial with new commuting variables. This also works in the black-box setting and yields a randomized polynomial-time algorithm which will produce as output black-boxes for the irreducible factors (which will all be homogeneous). When the input homogeneous polynomial is given by an algebraic branching program there is even a deterministic polynomial-time factorization algorithm. Indeed, the noncommutative factorization problem in for homogeneous polynomials efficiently reduces to the noncommutative PIT problem [3], analogous to the commutative case [21], modulo the randomness required for univariate polynomial factorization in the case of finite fields of large characteristic. The motivation of the present paper is to extend the above results to the inhomogeneous case.

Plan of the paper. In Section 2 we present basic definitions and the background results from Cohn's work on factorization. In Section 3 we further present some results from Cohn's work relevant to the paper. In Section 4 we present the factorization algorithm for polynomials f that does not vanish on scalars and in Section 5 we present the algorithm for the general case. Omitted proofs from the conference version can be found in full version.

2 Preliminaries

In this section we give some basic definitions and results relevant to the paper, mainly from Cohn's theory of factorization. Analogous to integral domains and unique factorization domains in commutative ring theory, P.M. Cohn [7, 8] has developed a theory for noncommutative rings based on the weak algorithm (a noncommutative generalization of the Euclidean division algorithm) and the notion of free ideal rings. We present the relevant basic definitions and results, specialized to the ring $\mathbb{F}\langle X\rangle$ of noncommutative polynomials with coefficients in a (commutative) field $\mathbb{F}$, and also for matrix rings with entries from $\mathbb{F}\langle X\rangle$.

The results about $\mathbb{F}\langle X\rangle$ in Cohn's text [7, Chapter 5] are stated uniformly for algebraically closed fields $\mathbb{F}$. However, those we discuss hold for any field $\mathbb{F}$ (in particular for $\mathbb{F}_q$ or a small degree extension of it). The proofs are essentially based on linear algebra.

Since we will be using Higman's linearization [15] to factorize noncommutative polynomials, we are naturally lead to studying the factorization of linear matrices in $\mathbb{F}\langle X\rangle^{d\times d}$ using Cohn's theory.

▶ **Definition 1** ([7]). *A matrix M in $\mathbb{F}\langle X\rangle^{d\times d}$ is called* full *if it has (noncommutative) rank d. That is, it cannot be decomposed as a matrix product $M = M_1 \cdot M_2$, for matrices $M_1 \in \mathbb{F}\lessdot X \gtrdot^{d\times e}$ and $M_2 \in \mathbb{F}\lessdot X \gtrdot^{e\times d}$ with $e < d$.*

▶ Remark 2. Based on the notion of noncommutative matrix rank [7], the square matrix $M \in \mathbb{F}\langle X\rangle^{d\times d}$ is full precisely when it is invertible in the skew field $\mathbb{F}\lessdot X \gtrdot$. That is, M is full if and only if there is a matrix $N \in \mathbb{F}\lessdot X \gtrdot^{d\times d}$ such that $MN = NM = I_d$, where I_d is $d \times d$ identity matrix. We refer to [13] and [18] for different equivalent formulations of noncommutative matrix rank.

We note the distinction between full matrices and units in the matrix ring $\mathbb{F}\langle X\rangle^{d\times d}$.

▶ **Definition 3.** *A matrix $U \in \mathbb{F}\langle X\rangle^{d\times d}$ is a* unit *if there is a matrix $V \in \mathbb{F}\langle X\rangle^{d\times d}$ such that $UV = VU = I_d$, where I_d is $d \times d$ identity matrix.*

Clearly, units in $\mathbb{F}\langle X\rangle^{d\times d}$ are full. Examples of units in $\mathbb{F}\langle X\rangle^{d\times d}$, which have an important role in our factorization algorithm, are upper (or lower) triangular matrices in $\mathbb{F}\langle X\rangle^{d\times d}$ whose diagonal entries are all *nonzero scalars.* Full matrices, in general, need not be units: for example, the 1×1 matrix x, where x is a variable, is full but it is not a unit in the ring $\mathbb{F}\langle X\rangle^{1\times 1} = \mathbb{F}\langle X\rangle$.

▶ Remark 4. Full non-unit matrices are essentially non-unit non-zero-divisors. For the factorization of elements in $\mathbb{F}\langle X\rangle^{d\times d}$, units are similar to scalars in the factorization of polynomials in polynomial rings. Cohn's theory [7] considers factorizations of full non-unit elements in $\mathbb{F}\langle X\rangle^{d\times d}$.

We next define *atoms* in $\mathbb{F}\langle X\rangle^{d\times d}$, which are essentially the irreducible elements in it.

▶ **Definition 5.** *A full non-unit element A in $\mathbb{F}\langle X\rangle^{d\times d}$ is an* atom *if A cannot be factorized as $A = A_1 A_2$ for full non-unit matrices A_1, A_2 in $\mathbb{F}\langle X\rangle^{d\times d}$.*

Noncommutative polynomials do not have unique factorization in the usual sense of commutative polynomial factorization.[4] A classic example [7] is the polynomial $x + xyx$ with its two different factorizations

$$x + xyx = x(1 + yx) = (1 + xy)x,$$

where $1 + xy$ and $1 + yx$ are distinct irreducible polynomials.

▶ **Definition 6.** *Elements $A \in \mathbb{F}\langle X\rangle^{d\times d}$ and $B \in \mathbb{F}\langle X\rangle^{d'\times d'}$ are called* stable associates *if there are positive integers t and t' such that $d + t = d' + t'$ and units $P, Q \in \mathbb{F}\langle X\rangle^{(d+t)\times(d+t)}$ such that $A \oplus I_t = P(B \oplus I_{t'})Q$.*

It is easy to check that the polynomials $1 + xy$ and $1 + yx$ are stable associates.

Notice that if A and B are full non-unit matrices that are stable associates then A is atom if and only if B is atom. Furthermore, we note that stable associativity defines an equivalence relation between full matrices over the ring $\mathbb{F}\langle X\rangle$.

We observe that the problem of checking if two polynomials in $\mathbb{F}\langle X\rangle$ given as arithmetic formulas are stable associates or not has an efficient randomized algorithm (Lemma 17).

Now we turn to the problem of noncommutative polynomial factorization. By Higman's linearization [15, 7], given a polynomial $f \in \mathbb{F}\langle X\rangle$ there is a positive integer ℓ such that f is stably associated with a *linear matrix* $L \in \mathbb{F}\langle X\rangle^{\ell\times\ell}$, that is to say, the entries of L are affine linear forms.[5] Higman's linearization process is a simple algorithm obtaining the linear matrix L for a given f, and it plays a crucial role in our factorization algorithm. We describe it and state an effective version [13] which gives a simple polynomial-time algorithm to compute L when f is given as a noncommutative arithmetic formula.

Higman's linearization process

We describe a single step of the linearization process. Given an $m \times m$ matrix M over $\mathbb{F}\langle X\rangle$ such that $M[m, m] = f + g \times h$, apply the following:

- Expand M to an $(m + 1) \times (m + 1)$ matrix by adding a new last row and last column with diagonal entry 1 and remaining new entries zero:

$$\left[\begin{array}{c|c} M & 0 \\ \hline 0 & 1 \end{array}\right].$$

[4] However, as shown by Cohn, using the notion of stable associates there is a more general sense in which noncommutative polynomials have "unique" factorization [7].

[5] More generally, by Higman's linearization any matrix of polynomials M is stably associated with a linear matrix $L \in \mathbb{F}\langle X\rangle^{\ell\times\ell}$ for some ℓ.

- Then the bottom right 2×2 submatrix is transformed as follows by elementary row and column operations

$$\begin{pmatrix} f+gh & 0 \\ 0 & 1 \end{pmatrix} \to \begin{pmatrix} f+gh & g \\ 0 & 1 \end{pmatrix} \to \begin{pmatrix} f & g \\ -h & 1 \end{pmatrix}$$

Given a polynomial $f \in \mathbb{F}\langle X\rangle$ by repeated application of the above step we will finally obtain a *linear matrix* $L = A_0 + \sum_{i=1}^{n} A_i x_i$, where each $A_i, 0 \le i \le n$ is an $\ell \times \ell$ over $\mathbb{F}$, for some ℓ. The following theorem summarizes its properties.

▶ **Theorem 7** (Higman Linearization, [7])**.** *Given a polynomial $f \in \mathbb{F}\langle X\rangle$, there are matrices $P, Q \in \mathbb{F}\langle X\rangle^{\ell\times\ell}$ and a linear matrix $L \in \mathbb{F}\langle X\rangle^{\ell\times\ell}$ such that*

$$\left(\begin{array}{c|c} f & 0 \\ \hline 0 & I_{\ell-1} \end{array}\right) = PLQ \tag{1}$$

with P upper triangular, Q lower triangular, and the diagonal entries of both P and Q are all 1's (hence, P and Q are both units in $\mathbb{F}\langle X\rangle^{\ell\times\ell}$).

Instead of a single f, we can apply Higman linearization to a matrix of polynomials $M \in \mathbb{F}\langle X\rangle^{m\times m}$ to obtain a linear matrix L that is stably associated to M. We state the algorithmic version of Garg et al. [13] in this general form.

▶ **Theorem 8** ([13], Proposition A.2)**.** *Let $M \in \mathbb{F}\langle X\rangle^{m\times m}$ such that $M_{i,j}$ is computed by a noncommutative arithmetic formula of size at most s and bit complexity at most b. Then, for some $k = O(s)$, in time $\mathrm{poly}(s, b)$ we can compute the matrices P, Q and L in $\mathbb{F}\langle X\rangle^{\ell\times\ell}$ of Higman's linearization such that*

$$\left(\begin{array}{c|c} M & 0 \\ \hline 0 & I_k \end{array}\right) = PLQ.$$

where $\ell = m + k$. Moreover, the entries of the matrices P and Q as well as P^{-1} and Q^{-1} are given by polynomial-size algebraic branching programs which can also be obtained in polynomial time.

We will sometimes denote the block diagonal matrix $\left(\begin{array}{c|c} M & 0 \\ \hline 0 & I_k \end{array}\right)$ by $M \oplus I_k$.

As P and Q are units with diagonal entries all 1's, the matrix M is full iff the linear matrix L is full. Also, the scalar matrix $M(\overline{0})$ (obtained by setting all variables to zero) is invertible iff the scalar matrix $L(\overline{0})$, similarly obtained, is invertible.

Invariant Subspaces and Ronyai's Algorithm

▶ **Definition 9.** *Let $A_1, \ldots, A_n \in \mathbb{F}^{d\times d}$. A subspace $V \subseteq \mathbb{F}^n$ is called as common invariant subspace of $A_1, \ldots, A_n$ if $A_i v \in V$ for all $i \in [n]$ and $v \in V$.*

Clearly 0 and $\mathbb{F}^n$ are, trivially, common invariant subspaces for any collection of matrices. The algorithmic problem is to find a *non-trivial* common invariant subspace if one exists. Ronyai [24] gives a randomized polynomial-time algorithm for this problem when $\mathbb{F}$ is finite field.

▶ **Theorem 10** ([24])**.** *Given $A_1, \ldots, A_n \in \mathbb{F}_q^{d\times d}$ there is a randomized algorithm running in time polynomial in $n, d, \log q$ that computes with high probability a non-trivial common invariant subspace of $A_1, \ldots, A_n$ if such a subspace exists, and outputs "no" otherwise.*

▶ Remark 11. We should note here, the classical Burnside's theorem [6] for matrix algebras over algebraically closed fields. It essentially shows that the algebra generated by $A_1, A_2, \ldots, A_n$ is the full matrix algebra iff there is no nontrivial common invariant subspace.

▶ Remark 12. As already mentioned in the introduction, Friedl and Ronyai [12] have shown that over rationals the problem is at least as hard as factoring square-free integers, and hence likely to be intractable.

For standard definitions of noncommutative formulas and noncommutative algebraic branching programs (ABPs) we refer to [22].

3 Some Basic Results

In this section we present some basic results required for our factorization algorithm.

Monic linear matrices

▶ **Definition 13** ([7]). *Let $L = A_0 + A_1x_1 + \ldots + A_nx_n \in \mathbb{F}\langle X\rangle^{d\times d}$ be a linear matrix, where each A_i is a $d\times d$ scalar matrix over $\mathbb{F}$. Then L is called* right monic *if the $d\times nd$ scalar matrix $[A_1\ A_2\ \ldots\ A_n]$ has full row rank. Equivalently, if there are matrices $B_1, \ldots, B_n \in \mathbb{F}^{d\times d}$ such that $\Sigma_{i=1}^n A_iB_i = I_d$ (i.e. the matrix $[A_1\ A_2\ \ldots\ A_n]$ has right inverse).*

Similarly, L is left monic *if the $nd \times d$ matrix $[A_1^T\ A_2^T\ \ldots\ A_n^T]^T$ has full column rank. L is called* monic *if it is both left and right monic.*

The next two results from Cohn [7] are important properties of monic linear matrices.

▶ **Lemma 14** ([7]). *A right (or left) monic linear matrix in $\mathbb{F}\langle X\rangle^{d\times d}$ is not a unit in $\mathbb{F}\langle X\rangle^{d\times d}$.*

Let $f \in \mathbb{F}\langle X\rangle$ be a nonzero polynomial and L be a linear matrix obtained from f by Higman linearization as in Equation 1. Clearly, L is a full linear matrix. We show that we can transform L to obtain a full and right (or left) monic linear matrix L' that is stably associated to f. Furthermore, we can efficiently compute L' and the related transformation matrices.

▶ **Theorem 15** ([7]). *Let $L = A_0 + \sum_{i=1}^n A_ix_i$ be a full linear matrix in $\mathbb{F}\langle X\rangle^{d\times d}$ obtained by Higman linearization from a non constant polynomial $f \in \mathbb{F}\langle X\rangle$. Then there are deterministic* $\mathrm{poly}(n, d, \log_2 q)$ *time algorithms that compute units $U, U' \in \mathbb{F}\langle X\rangle^{d\times d}$ and invertible scalar matrices $S, S' \in \mathbb{F}_q^{d\times d}$ such that:*
1. *$ULS = L' \oplus I_r$, and L' is right monic. Moreover, if L is not right monic then $r > 0$.*
2. *$S'LU' = L' \oplus I_{r'}$, and L' is left monic. Moreover, if L is not left monic then $r' > 0$.*

▶ Remark 16. By repeated application of the algorithm in Theorem 15 we can compute units $U_1, U_2 \in \mathbb{F}\langle X\rangle^{d\times d}$ such that $U_1LU_2 = L' \oplus I_r$, where L' is *both left and right monic*. Such a two-sided monic L' is called *monic* in [7].

For our factorization algorithm, it suffices to compute an L' that is either left or right monic that is associated to L as in Theorem 15. It turns out that either a left monic or a right monic L' suffices to use Ronyai's common invariant subspace algorithm to factorize L' (and hence also L) as we show in Theorem 28. More importantly, the fact that matrices S and S' in Theorem 15 are scalar is important for the factor extraction algorithm as discussed in Theorem 32.

▶ **Lemma 17.** *Given polynomials $f, g \in \mathbb{F}\langle X\rangle$ as input by noncommutative arithmetic formulas, we can check in randomized polynomial time if f and g are stable associates.*

The next result shows how irreducibility (more generally, the property of being an atom) is preserved by Higman linearization.

▶ **Theorem 18.** *Let $f \in \mathbb{F}\langle X\rangle$ be a nonconstant polynomial and L be a full linear matrix stably associated with f (obtained via Higman linearization).Then the polynomial f is irreducible iff L is an atom.*

We give a self-contained proof of the above theorem, using the following (suitably paraphrased) result of Cohn.

▶ **Lemma 19** (Matrix Product Trivialization, [8], pp. 198)**.** *Let $A \in \mathbb{F}\langle X\rangle^{m\times n}$ and $B \in \mathbb{F}\langle X\rangle^{n\times s}$ be polynomial matrices such that their product $AB = 0$. Then there exists a unit $P \in \mathbb{F}\langle X\rangle^{n\times n}$ such that for every index $i \in [n]$ either the i^{th} column of the matrix product AP is all zeros or the i^{th} row of the matrix product $P^{-1}B$ is all zeros.*

Let $L \in \mathbb{F}\langle X\rangle^{d\times d}$ be a full and right (or left) monic linear matrix. Let $L = A_0 + \sum_{i=1}^{n} A_i x_i$. For a positive integer ℓ let $M_i, i \in [n]$ be $\ell \times \ell$ scalar matrices with entries from $\mathbb{F}$ (or a small degree extension of $\mathbb{F}$). Let $Y_i, i \in [n]$ be $\ell \times \ell$ matrices whose entries are distinct noncommuting variables $y_{ijk}, 1 \le j, k \le \ell$. Then the evaluation of the linear matrix L at $x_i \leftarrow Y_i + M_i, 1 \le i \le n$ is the $d\ell \times d\ell$ linear matrix in the y_{ijk} variables:

$$L' = A_0 \otimes I_\ell + \sum_{i=1}^{n} A_i \otimes M_i + \sum_{i=1}^{n} \sum_{j,k=1}^{\ell} (A_i \otimes E_{jk}) \cdot y_{ijk}$$

▶ **Lemma 20.** *There is a positive integer $\ell \le 2d$ such that for randomly picked $\ell \times \ell$ matrices $M_i, i \in [n]$ (with entries from $\mathbb{F}$ or a small degree extension field) the matrix $A_0 \otimes I_\ell + \sum_{i=1}^{n} A_i \otimes M_i$ is an invertible matrix.*

Proof. Since $L \in \mathbb{F}\langle X\rangle^{d\times d}$ is a full linear matrix, it has noncommutative rank d. Hence, by the result of [10] for the generic $2d \times 2d$ matrix substitution $x_i \leftarrow X_i, i \in [n]$, where X_i is a matrix of distinct commuting variables, the commutative rank of $L(X_1, X_2, \ldots, X_n)$ is $2d^2$ (which means it is invertible). Hence there is a least $\ell \le 2d$ such that the commutative rank of $L(X_1, X_2, \ldots, X_n)$ is $d\ell$, where X_i are generic $\ell \times \ell$ matrices with commuting variables. Hence, by the DeMillo-Lipton-Schwartz-Zippel lemma [9, 25, 27] the rank of the scalar matrix $L(M_1, M_2, \ldots, M_n)$ is $d\ell$, where M_i is a random scalar matrix with entries from $\mathbb{F}$ or a small extension. ◀

Finally, we state and prove a *modified version* of a result due to Cohn that allows us to relate the factorization of a polynomial $f \in \mathbb{F}\langle X\rangle$ to the factorization of its Higman linearization L. The proof is given in the appendix.

▶ **Theorem 21** ([7], Theorem 5.8.8)**.** *Let $C \in \mathbb{F}\langle X\rangle^{d\times d}$ be a full and right monic (or left monic) linear matrix for $d > 1$. Then C is not an atom if and only if there are $d \times d$ invertible scalar matrices S and S' such that*

$$SCS' = \begin{pmatrix} A & 0 \\ D & B \end{pmatrix} \tag{2}$$

where A is an $r \times r$ full right (respec. left) monic linear matrix and B is an $s \times s$ full right (respec. left) monic linear matrix such that $r + s = d$.

▶ Remark 22. In [7] the theorem is proved under the stronger assumption that C is monic. However, as we show, it holds even for C that is right monic or left monic with minor changes to Cohn's proof. We require the above version for our factorization algorithm.

4 Polynomial factorization: commutatively non-zero case

Recall that $\mathbb{F}\langle X\rangle$ denotes the free noncommutative polynomial ring $\mathbb{F}\langle x_1, x_2, \ldots, x_n\rangle$ and our goal is to give a randomized polynomial-time factorization algorithm for input polynomials in $\mathbb{F}\langle X\rangle$ given as arithmetic formulas when $\mathbb{F} = \mathbb{F}_q$ is a finite field of size q.

A polynomial $f \in \mathbb{F}\langle X\rangle$ is *commutatively nonzero* if $f(\alpha_1, \alpha_2, \ldots, \alpha_n) \neq 0$ for scalars $\alpha_i \in \mathbb{F}$ (or a small extension field of $\mathbb{F}$).

In this section we will present the factorization algorithm for commutatively nonzero polynomials.[6] It has three broad steps:

(i) We transform the given polynomial f to a full and *right (or left) monic* linear matrix L by first the Higman linearization of f followed by the algorithm in the proof of Theorem 15.

(ii) Next, we factorize the full and right (or left) monic linear matrix L into atoms.

(iii) Finally, we recover the irreducible factors of f from the atomic factors of L.

We formally state the three problems of interest in this paper.

▶ **Problem 23** (FACT($\mathbb{F}$)).
Input *A noncommutative polynomial $f \in \mathbb{F}\langle X\rangle$ given by an arithmetic formula.*
Output *Compute a factorization $f = f_1 f_2 \cdots f_r$, where each f_i is irreducible, and each f_i is output as an algebraic branching program.*

▶ **Problem 24** (LIN-FACT($\mathbb{F}$)).
Input *A full and right (or left) monic linear matrix $L \in \mathbb{F}\langle X\rangle^{d\times d}$.*
Output *Compute a factorization $L = F_1 F_2 \cdots F_r$, where each F_i is a full linear matrix that is an atom.*

▶ **Problem 25** (INV($\mathbb{F}$)).
Input *A list of scalar matrices $A_1, A_2, \ldots, A_n \in \mathbb{F}^{d\times d}$.*
Output *Compute a nontrivial invariant subspace $V \subset \mathbb{F}^d$ or report that the only invariant subspaces are 0 and $\mathbb{F}^d$.*

In the three-step outline of the algorithm, for the second step we will show that factoring a full and right (or left) monic linear matrix is randomized polynomial-time reducible to the problem of computing a common invariant subspace for a collection of scalar matrices. For the third step, we will give a polynomial-time algorithm (based on Lemma 19) to recover the irreducible factors of f from the atomic factors of L.

▶ Remark 26. We use Ronyai's randomized polynomial-time algorithm [24] to solve the problem of computing a a common invariant subspace for a collection of matrices over $\mathbb{F}_q$. Over rational numbers $\mathbb{Q}$, even for a special case the problem of computing a common invariant subspace turns out to be at least as hard as factoring square-free integers [12]. Hence, our approach to noncommutative polynomial factorization does not yield an efficient algorithm over $\mathbb{Q}$.

Suppose $f \in \mathbb{F}\langle X\rangle$ is given by a noncommutative arithmetic formula. Since f has small degree we can check if it is commutatively nonzero in randomized polynomial-time by the DeMillo-Lipton-Schwartz-Zippel test [9, 25, 27] and, if so, find $\alpha_i \in \mathbb{F}, i \in [n]$ such that $f(\alpha_1, \alpha_2, \ldots, \alpha_n) \neq 0$ (if $\mathbb{F}$ is small, we pick α_i from a small extension field). Furthermore,

[6] In the next section we will deal with the general case. The algorithm is more technical in detail, although in essence the same.

by a linear shift of the variables $x_i \leftarrow x_i + \alpha_i, i \in [n]$ followed by scaling we can assume $f(\overline{0}) = 1$. Note that from the factorization of the linear shift of f we can recover the factors of f by shifting the variables back, and irreducibility is preserved by linear shift. For the rest of this section we will assume $f(\overline{0}) = 1$.

Let $L = A_0 + \sum_{i=1}^{n} A_i x_i$. As $f(\overline{0}) = 1$, we have $L(\overline{0}) = A_0$ is an invertible matrix. We now present an efficient algorithm for factoring L as a product of linear matrices $L_1 L_2 \cdots L_r$, where each L_i is an atom.

▶ Remark 27. The factorization algorithm for arbitrary full and right (or left) monic linear matrices (in which A_0 need not be invertible) is similar but more involved. It is based on Lemma 20 and is dealt with in the next section.

4.1 Algorithm for a special case of LIN-FACT($\mathbb{F}_q$)

▶ **Theorem 28.** *There is a randomized polynomial-time algorithm for the following two special cases of the LIN-FACT($\mathbb{F}_q$) problem:*

1. *Given a full right monic matrix L as input such that $L(\overline{0})$ is an invertible matrix, the algorithm outputs a factorization of L as a product of linear matrices that are atoms.*
2. *Given a full left monic matrix L as input such that $L(\overline{0})$ is an invertible matrix, the algorithm outputs a factorization of L as a product of linear matrices that are atoms.*

Proof. We present the algorithm only for the first part, as the second part has essentially the same solution.

Let $L = A_0 + \sum_{i=1}^{n} A_i x_i$ in $\mathbb{F}\langle X \rangle^{d \times d}$ be such an instance of LIN-FACT($\mathbb{F}_q$). We can write $L = A_0 \cdot L'$ where L' is the full and right monic linear matrix

$$L' = I_d + \sum_{i=1}^{n} A_0^{-1} A_i x_i.$$

Clearly, it suffices to factorize the linear matrix L' into atoms.

First we show that L' is not an atom iff matrices $A_0^{-1} A_i$, $1 \le i \le n$ have a nontrivial common invariant subspace. By Theorem 21, L' is not an atom if and only if we can write $S_1 L' S_2 = \begin{pmatrix} B & 0 \\ D & C \end{pmatrix}$ for invertible scalar matrices S_1 and S_2, where B and C are full and right monic linear matrices, and D is some linear matrix. Equating the constant terms on both sides of the above equation we have $S_1 S_2 = \begin{pmatrix} B_0 & 0 \\ D_0 & C_0 \end{pmatrix}$ as the constant term of L' is I_d. Thus the matrices $S_1 S_2$ and its inverse also has the same block form which implies that $S_1 L' S_1^{-1} = S_1 L' S_2 (S_1 S_2)^{-1}$ also has the same block form. It follows that the n matrices $A_0^{-1} A_i, 1 \le i \le n$ have a nontrivial common invariant subspace. Conversely, if the matrices $A_0^{-1} A_i, 1 \le i \le n$ have a nontrivial common invariant subspace then we have a basic change scalar matrix S such that $S L' S^{-1}$ has the block form $\begin{pmatrix} L_1 & 0 \\ * & L_2 \end{pmatrix}$, where L_1 and L_2 are full and right monic linear matrices. So by Theorem 21 L' is not an atom. So we have established, L' (and hence L) is not an atom iff matrices $A_0^{-1} A_i$, $1 \le i \le n$ have a nontrivial common invariant subspace. We will use Ronyai's randomized polynomial-time algorithm for finding a nontrivial common invariant subspace for matrices $A_0^{-1} A_i, 1 \le i \le n$ over finite field $\mathbb{F}_q$.

If there is no nontrivial invariant subspace then the linear matrix L' (and hence L) is an atom. Otherwise, by repeated application of Ronyai's algorithm we will obtain a basis change scalar matrix T which when applied to L' yields a linear matrix in the following *atomic block diagonal form*:

$$TL'T^{-1} = \begin{pmatrix} L_1 & 0 & 0 & \dots & 0 \\ * & L_2 & 0 & \dots & 0 \\ * & * & L_3 & \dots & 0 \\ & & & \ddots & \\ * & * & * & \dots & L_r \end{pmatrix}, \tag{3}$$

where for each $j \in [r]$, the full right monic linear matrix $L_j \in \mathbb{F}\langle X\rangle^{d_j \times d_j}$ is an atom, and each $*$ stands for some unspecified linear matrix. It is now easy to factorize $TL'T^{-1}$ as a product of atoms by noting one step of the factorization of $TL'T^{-1}$ from its form:

$$TL'T^{-1} = \begin{pmatrix} A & 0 \\ D & L_r \end{pmatrix} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ D & I \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ 0 & L_r \end{pmatrix}.$$

We note that $\begin{pmatrix} I & 0 \\ D & I \end{pmatrix}$ is a unit. Since L_r is an atom the product $\begin{pmatrix} I & 0 \\ D & I \end{pmatrix} \cdot \begin{pmatrix} I & 0 \\ 0 & L_r \end{pmatrix}$ is also an atom and a linear matrix, and it is the rightmost factor of $TL'T^{-1}$. Continuing thus with A now, we can factorize $TL'T^{-1}$ as a product $F_1'F_2'\cdots F_r'$ of r atoms, each of which is a linear matrix. It follows that $L = A_0T^{-1}F_1'F_2'\cdots F_r'T$ is a complete factorization of L as a product of atomic linear matrices (both A_0 and T are scalar invertible matrices). ◀

▶ Remark 29. We note that Ronyai's algorithm [24] for $\mathrm{INV}(\mathbb{F}_q)$ is actually a deterministic polynomial-time *reduction* from $\mathrm{INV}(\mathbb{F}_q)$ to univariate polynomial factorization over $\mathbb{F}_q$.

Based on whether we want to work with right monic or left monic case we will express $f \oplus I_s$ in an appropriate form using Higman linearization and Theorem 15 as described in the equation below:

$$f \oplus I_s = \begin{cases} PU(L' \oplus I_t)SQ, & \text{in the right monic case} \\ PS(L' \oplus I_t)UQ, & \text{in the left monic case} \end{cases} \tag{4}$$

where $d + t = s + 1$, $L' \in \mathbb{F}\langle X\rangle^{d\times d}$ is a full and right (or left) monic linear matrix, P is upper triangular with all 1's diagonal, Q is lower triangular with all 1's diagonal, $U \in \mathbb{F}\langle X\rangle^{(d+t)\times(d+t)}$ is a unit, and $S \in \mathbb{F}^{(d+t)\times(d+t)}$ is an invertible scalar matrix.

Algorithm for FACT($\mathbb{F}_q$)

We are now ready to describe the polynomial factorization algorithm for commutatively nonzero polynomials in $\mathbb{F}\langle X\rangle$. Starting with the Higman linearization of the input polynomial $f \in \mathbb{F}\langle X\rangle$ as in Equation 4, by an application of the first parts of Theorems 15 and 28 we obtain the factorization $f \oplus I_s = PUF_1'F_2'\cdots F_r'SQ$ using the structure in Equation 3.

Alternatively, by applying the second part of Theorem 15 we can compute a left monic linear matrix L' that is a stable associate of f and, applying the second part of Theorem 28 we can compute the factorization

$$f \oplus I_s = PS'F_1'F_2'\cdots F_r'U'Q. \tag{5}$$

where each linear matrix F'_i is an atom, P is upper triangular with all 1's diagonal, Q is lower triangular with all 1's diagonal, U' is a unit and S' is a scalar invertible matrix. Equation 5 is the form we will use for the algorithm (we could equally well use the other factorization).

From the structure of the atomic block diagonal matrix $TL'T^{-1}$ in Equation 3 notice that the product $S'F'_1F'_2\cdots F'_i$ is a linear matrix for each $1 \leq i < r$.

The next lemma presents an algorithm that is crucial for extracting the factors of f.

▶ **Lemma 30.** *Let $C \in \mathbb{F}\langle X\rangle^{u\times d}$ be a linear matrix and $v \in \mathbb{F}\langle X\rangle^{d\times 1}$ be a column of polynomials such that $Cv = 0$. Each entry v_i of v is given by an algebraic branching program as input. Then, in polynomial time we can compute a invertible matrix $N \in \mathbb{F}\langle X\rangle^{d\times d}$ such that*

- *For $1 \leq i \leq d$ either the i^{th} column of CN is all zeros or the i^{th} row of $N^{-1}v$ is zero.*
- *Each entry of N is a polynomial of degree at most d^2 and is computed by a polynomial size ABP, and also each entry of N^{-1} is computed by a polynomial size ABP.*

Proof. We will describe the algorithm as a recursive procedure Trivialize that takes matrix C and column vector v as parameters and returns a matrix N as claimed in the statement.

Procedure Trivialize($C \in \mathbb{F}\langle X\rangle^{u\times d}, v \in \mathbb{F}\langle X\rangle^{d\times 1}$).

1. If $d = 1$ then (since $Cv = 0$ iff either $C = 0$ or $v = 0$) **return** the identity matrix.
2. If $d > 1$ then
3. write $C = C_0 + C_1$, where C_0 is a scalar matrix and C_1 is the degree 1 homogeneous part of C. Let k be the degree of the highest degree nonzero monomials in the polynomial vector v, and let m be a nonzero degree k monomial. Let $v(m) \in \mathbb{F}_q^{d\times 1}$ denote its (nonzero) coefficient in v. Then $Cv = 0$ implies $C_1v(m) = 0$. Let $T_0 \in \mathbb{F}_q^{d\times d}$ be a scalar invertible matrix with first column $v(m)$ obtained by completing the basis.
   a. If $C_0v(m) = 0$ then the first column of CT_0 is zero.
   b. Otherwise, CT_0 has first column as the nonzero scalar vector $Cv(m) = C_0v(m)$. Suppose i^{th} entry of $Cv(m)$ is a nonzero scalar α. With column operations we can drive the i^{th} entry in all other columns of CT_0 to zero. Let the resulting matrix be CT_0T_1 (where the matrix T_1 is invertible as it is a product of elementary matrices corresponding to these column operations, each of which is of the form $\mathrm{Col}_i \leftarrow (\mathrm{Col}_i + \mathrm{Col}_1 \cdot \alpha_0 + \sum_i \alpha_i x_i)$). Notice that CT_0T_1 is still linear.
   c. As $Cv = (CT_0T_1)(T_1^{-1}T_0^{-1}v)$, and in the i^{th} row of CT_0T_1 the only nonzero entry is α which is in its first column, we have that the first entry of $T_1^{-1}T_0^{-1}v$ is zero.
4. Let $C' \in \mathbb{F}\langle X\rangle^{u\times(d-1)}$ obtained by dropping the first column of CT_0T_1. Let $v' \in \mathbb{F}\langle X\rangle^{(d-1)\times 1}$ be obtained by dropping the first entry of $T_1^{-1}T_0^{-1}v$. Note that C' is still linear.
5. Recursively call Trivialize($C' \in \mathbb{F}\langle X\rangle^{u\times(d-1)}, v' \in \mathbb{F}\langle X\rangle^{(d-1)\times 1}$). and let the matrix returned by the call be $T_2 \in \mathbb{F}\langle X\rangle^{(d-1)\times(d-1)}$.
6. Putting it together, return the matrix $T_0T_1(I_1 \oplus T_2)$.

To complete the proof, we note that a highest degree monomial m such that $v(m) \neq 0$ is easy to compute in deterministic polynomial time if each v_i is given by an algebraic branching program using the PIT algorithm of Raz and Shpilka [23]. Notice that for the recursive call we need C' to be also a linear matrix and each entry of v' to have a small ABP. C' is linear because CT_0T_1 is a linear matrix since CT_0 is linear, its first column is scalar, and each column operation performed by T_1 is scaling the first column of CT_0 by a linear form and subtracting from another column of CT_0. Each entry of v' has a small ABP because T_0^{-1} is scalar and it is easy to see that the entries of T_1^{-1} have ABPs of polynomial size. Finally, we note that T_1 is a product of at most $d-1$ linear matrices (each corresponding to a column operation), and N is an iterated product of d such matrices. Hence, each entry of N as well as N^{-1} is a polynomial of degree at most d^2 and is computable by a small ABP. ◀

Turning back to our algorithm for FACT($\mathbb{F}_q$), in the next lemma we design an efficient algorithm that will allow us to extract all the irreducible factors of f (given Equation 5).

▶ **Lemma 31** (Factor Extraction). *Let $f \in \mathbb{F}\langle X\rangle$ be a polynomial and $G \in \mathbb{F}\langle X\rangle^{(d-1)\times(d-1)}$ be a unit such that*

$$\begin{pmatrix} f & u \\ 0 & G \end{pmatrix} = PCD, \tag{6}$$

such that

- *C is a full linear matrix that is a non-unit, P is upper triangular with all 1's diagonal, and $D \in \mathbb{F}\langle X\rangle^{d\times d}$ is a full non-unit matrix which is also an atom.*
- *The polynomial f, and the entries of u, G, P, D are all given as input by algebraic branching programs.*

Then we can compute in deterministic polynomial time a nontrivial factorization $f = g \cdot h$ of the polynomial f such that h is an irreducible polynomial.

Proof. Let

$$C = \begin{pmatrix} c_1 & c_3 \\ c_2 & c_4 \end{pmatrix} \text{ and } D = \begin{pmatrix} d_1 & d_3 \\ d_2 & d_4 \end{pmatrix},$$

written as 2×2 block matrices where c_1 and d_1 are 1×1 blocks. By dropping the first row of the matrix in the left hand side of Equation 6 and the first row of P we get

$$(0\ G) = (0\ P')CD,$$

where P' is also an upper triangular matrix with all 1's diagonal. Equating the first columns on both sides we have

$$\begin{aligned} 0 &= (0\ P')\begin{pmatrix} c_1 & c_3 \\ c_2 & c_4 \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \text{ which implies that} \\ 0 &= P'(c_2\ c_4)\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \text{ and hence} \\ 0 &= (c_2\ c_4)\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \text{ since } P' \text{ is invertible.} \end{aligned}$$

Since $(c_2\ c_4) \in \mathbb{F}\langle X\rangle^{(d-1)\times d}$ is a matrix with linear entries and $\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \in \mathbb{F}\langle X\rangle^{d\times 1}$ is a column vector of polynomials which are given by ABPs as input, we can apply the algorithm of Lemma 30 to compute a unit N such that its entries are all given by ABPs such that for $1 \le i \le d$, either the i^{th} column of $(c_2'\ c_4') = (c_2\ c_4)N$ is zero or the i^{th} row of $\begin{pmatrix} d_1' \\ d_2' \end{pmatrix} = N^{-1}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$ is zero.

Now the following argument is almost identical with the argument towards the end of the proof of the Theorem 18. We give it below for completeness. Since D is a full matrix, the matrix $N^{-1}D$ is also full which implies its first column $\begin{pmatrix} d_1' \\ d_2' \end{pmatrix}$ cannot be all zeros. So there is at least one nonzero entry in $\begin{pmatrix} d_1' \\ d_2' \end{pmatrix}$ and the corresponding column in $(c_2'\ c_4')$ is all zero. This implies there exist a permutation matrix Π such that the first column of $C(c_2'\ c_4')\Pi$ is all zero and first entry of $\Pi^{-1}\begin{pmatrix} d_1' \\ d_2' \end{pmatrix}$ is non zero.

Consider the matrices $C'' = CN\Pi = \begin{pmatrix} c_1'' & c_3'' \\ c_2'' & c_4'' \end{pmatrix}$ and $D'' = \Pi^{-1}N^{-1}D = \begin{pmatrix} d_1'' & d_3'' \\ d_2'' & d_4'' \end{pmatrix}$. We have

$$\begin{pmatrix} f & * \\ 0 & G' \end{pmatrix} = P^{-1}\begin{pmatrix} f & u \\ 0 & G \end{pmatrix} = \begin{pmatrix} c_1'' & c_3'' \\ c_2'' & c_4'' \end{pmatrix}\begin{pmatrix} d_1'' & d_3'' \\ d_2'' & d_4'' \end{pmatrix},$$

where $G' = (P')^{-1}G$ is a unit, c_2'' is all zero column matrix and d_1'' is non-zero. Now observing $(2,1)^{th}$ matrix block in the above equation, we get d_2'' is all zero column. Hence, by looking at $(2,2)^{th}$ block in the above equation, we can see that c_4'' and d_4'' are units as G' is a unit. Clearly, we have $f = c_1'' \cdot d_1''$. Now, since C and D are non-units (by assumption), the matrices C'' and D'' are also non-units. Therefore, c_1'' is not a scalar for otherwise C'' would be a unit. Similarly, d_1'' is not a scalar. It follows that $f = c_1''d_1''$ is a nontrivial factorization of f.

Furthermore, since D is an atom by assumption and D'' is a stable associate of D, D'' is an atom. As $D'' = \begin{pmatrix} d_1'' & d_3'' \\ 0 & d_4'' \end{pmatrix}$ and d_4'' is invertible, we get $\begin{pmatrix} 1 & 0 \\ 0 & (d_4'')^{-1} \end{pmatrix} \cdot D'' = \begin{pmatrix} d_1'' & d_3'' \\ 0 & I_s \end{pmatrix}$. Now applying suitable row operations to the matrix $(1 \oplus (d_4'')^{-1})D''$ we can drive d_3'' to zero. So we have $U(1 \oplus (d_4'')^{-1})D'' = (d_1'' \oplus I_s)$ for a unit U. Hence d_1'' is an associate of D'' and therefore d_1'' is irreducible as D'' is an atom. ◀

Finally, we describe the factorization algorithm for commutatively nonzero polynomials $f \in \mathbb{F}\langle X\rangle$ over finite fields $\mathbb{F}_q$.

▶ **Theorem 32.** *Let $\mathbb{F}\langle X\rangle = \mathbb{F}_q\langle X\rangle$ and $f \in \mathbb{F}\langle X\rangle$ be a commutatively nonzero polynomial given by an arithmetic formula of size s as input instance of FACT$(\mathbb{F}_q)$. Then there is a* $\mathrm{poly}(s, \log q)$ *time randomized algorithm that outputs a factorization $f = f_1 f_2 \cdots f_r$ such that each f_i is irreducible and is output as an algebraic branching program.*

Proof. Given f as input, we apply Higman linearization followed by the algorithm for LIN-FACT$(\mathbb{F}_q)$ described in Theorem 28. This will yield the factorization of $f \oplus I_s = PSS_1F_1F_2 \ldots F_rS_2UQ$ where each linear matrix F_i is an atom, P is upper triangular with all 1's diagonal, Q is lower triangular with all 1's diagonal, U is a unit and S is a scalar invertible matrix, as given in Equation 5. We can now apply Lemma 31 to extract irreducible factors of f (one by one from the right).

For the first step, let $C = SS_1F_1F_2 \cdots F_{r-1}$ and $D = F_rS_2UQ$ in Lemma 31. The proof of Lemma 31 yields the matrix $N_r = N\Pi$ such that both matrices $C'' = PSS_1F_1F_2 \cdots F_{r-1}N_r$ and $D'' = N_r^{-1}F_rS_2UQ$ has the first column all zeros except the $(1,1)^{th}$ entries c_1'' and d_1'' which yields the nontrivial factorization $f = c_1''d_1''$, where $d_1'' = f_r$ is irreducible. Renaming c_1'' as g_r we have from the structure of C'':

$$\begin{pmatrix} g_r & * \\ 0 & G_r \end{pmatrix} = P(SS_1F_1F_2 \cdots F_{r-2})(F_{r-1}N_r).$$

Setting $C = SS_1F_1F_2 \cdots F_{r-2}$ and $D = F_{r-1}N_r$ in Lemma 31 we can compute the matrix N_{r-1} using which we will obtain the next factorization $g_r = g_{r-1}f_{r-1}$, where f_{r-1} is irreducible because the linear matrix F_{r-1} is an atom. Lemma 31 is applicable as all conditions are met by the matrices in the above equation (note that G_r will be a unit).

Continuing thus, at the i^{th} stage we will have $f = g_{r-i+1} f_{r-i+1} f_{r-i+2} \cdots f_r$ after obtaining the rightmost i irreducible factors by the above process. At this stage we will have

$$\begin{pmatrix} g_{r-i+1} & * \\ 0 & G_{r-i+1} \end{pmatrix} = P(SS_1 F_1 F_2 \cdots F_{r-i-1})(F_{r-i} N_{r-i+1}),$$

where G_{r-i+1} is a unit and all other conditions are met to apply Lemma 31.

Thus, after r stages we will obtain the complete factorization $f = f_1 f_2 \cdots f_r$. For the running time, it suffices to note that the matrix N computed in Lemma 31 is a product of degree at most d^2 many linear matrices (corresponding to the column operations). Thus, at the i^{th} of the above iteration, the sizes of the ABPs for the entries of N_{r-i+1} are independent of the stages. Hence, the overall running time is easily seen to be polynomial in s and $\log q$. ◀

▶ **Corollary 33.** *If $f \in \mathbb{F}\langle X\rangle$ is commutatively nonzero polynomial given as input in sparse representation (as an $\mathbb{F}_q$-linear combination of its monomials) then in randomized polynomial time we can compute a factorization into irreducible factors in sparse representation.*

Proof. Let f be given as input in sparse representation. Suppose $\deg f = d$ and it is t-sparse. Then there are at most td^2 many monomials that can occur as a substring of the monomials of f. We can apply the randomized algorithm of Theorem 32 to obtain the factorization $f = f_1 f_2 \cdots f_r$, where each f_i is given by an ABP. Now, for each of the td^2 many candidate monomials of f_i we can find its coefficient in f_i in polynomial time (using the Raz-Shpilka algorithm [23]). Hence we can obtain the factorization $f = f'_1 f'_2 \cdots f'_r$, where each f'_i is a t-sparse polynomial. ◀

5 Factorization of Commutatively zero polynomials

In this section we will describe the general case of the factorization algorithm when the input polynomial $f \in \mathbb{F}\langle X\rangle$ is a commutatively zero polynomial. That is, f evaluates to zero on all scalar substitutions from $\mathbb{F}_q$ or any (commutative) extension field.

The factorization algorithm will follow the three broad steps described in Section 4 for the commutatively nonzero case: first, using Higman linearization and Theorem 15, transform the polynomial f to a stably associated linear matrix L that is full and left (or right) monic. Next, factorize the linear matrix L into atoms. Finally, recover the irreducible factors of f from the atomic factors of the linear matrix L using the factor extraction procedure described in Lemma 31.

The step that requires a new algorithm is factorizing a full and right (or left) monic linear matrix $L \in \mathbb{F}\langle X\rangle$ into atoms when f is commutatively zero, which means there is no scalar substitution $x_i \leftarrow \alpha_i, i \in [n]$ such that $L(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is invertible. Note that in this case we cannot apply the algorithm for factorizing a linear matrix as discussed in the proof of Theorem 28).

5.1 Factorization of full and monic linear matrices

Let $f \in \mathbb{F}\langle X\rangle$ be the input polynomial given by a size s formula and let $L \in \mathbb{F}\langle X\rangle^{d\times d}$ be a full, right monic linear matrix stably associated with f obtained via Higman linearization and an application of Theorem 15.

Recall, by Equation 4 we have $f \oplus I_s = PU(L \oplus I_t)SQ$ where, P, Q are respectively upper triangular and lower triangular units with diagonal entries 1, U is a unit and S is scalar invertible.

Let $L = A_0 + \sum_{i=1}^{n} A_n x_i \in \mathbb{F}\langle X\rangle^{d\times d}$ be the given full and right monic linear matrix. First, by Lemma 20, we will find a suitable scalar matrix n-tuple $\bar{M} = (M_1, M_2, \ldots, M_n)$, each $M_i \in \mathbb{F}_q^{\ell\times\ell}$ for $\ell \le 2d$, such that under the substitution $x_i \leftarrow M_i$ the matrix $L(\bar{M})$ is invertible.

For $1 \le i \le n$ let Y_i be an $\ell \times \ell$ matrix of distinct noncommuting variables y_{ijk}. We consider the dilated linear matrix

$$L' = A_0 \otimes I_\ell + \sum_{i=1}^{n} A_i \otimes (Y_i + M_i). \tag{7}$$

It is not hard to see that L' is full and L' is right monic as L is right monic. Additionally, its constant term is invertible. So, we can apply Theorem 28 to factorize L' as a product of two linear matrices, both non-units.

The following lemma [14] has an important role in our algorithm for recovering the factorization for L from a factorization of L'.

▶ **Lemma 34** ([14]). *Let $L \in \mathbb{F}\langle X\rangle^{d\times d}$ be a full linear matrix with $L = A_0 + A_1x_1 + \ldots + A_nx_n$ such that $A_i \ne 0$ for at least one i, $1 \le i \le n$ and $L' \in R^{d\ell\times d\ell}$ be a matrix obtained from L by substituting variable x_i by Y_i for $i \in [n]$, where Y_i is $\ell \times \ell$ matrix whose $(j,k)^{th}$ entry is a fresh noncommuting variable $y_{i,j,k}$ for $1 \le j,k \le \ell$. Then*

1. *If L' is of the form $GL'H = \left(\begin{array}{c|c} A' & 0 \\ \hline D' & B' \end{array}\right)$, where A' is $d' \times d'$ matrix and B' is $d'' \times d''$ matrix for $0 < d', d''$, with $d' + d'' = d\ell$ and G, H are $d\ell \times d\ell$ invertible scalar matrices then there exist $d \times d$ invertible scalar matrices U, V such that $ULV = \left(\begin{array}{c|c} A & 0 \\ \hline D & B \end{array}\right)$, where A is $e' \times e'$ matrix and B is $e'' \times e''$ matrix for $0 < e', e''$, with $e' + e'' = d$.*
2. *Moreover, given L' explicitly along with its representation mentioned above, we can find the matrices U, V in deterministic polynomial time (in n, ℓ, d).*

▶ Remark 35. We give a self-contained complete proof of the above linear-algebraic lemma in the appendix for $\mathbb{F}_q$, because the proof given in [14] is sketchy in parts with some details missing, and also their lemma is stated only for complex numbers and they are not concerned about computing the matrices U and V.

Now, we can apply Lemma 34 to transform the factorization of L' to a factorization of L as a product of two linear matrices, both non-units. Repeating the above on both the factors of L we will get a complete atomic factorization of L. Formally, we prove the following.

▶ **Theorem 36.** *On input a full and right (or left) monic linear matrix $L = A_0 + \sum_{i=1}^{n} A_i x_i$ where $A_i \in \mathbb{F}^{d\times d}$ for $i \in [n]$, there is a randomized polynomial time ($poly(n,d)$) algorithm to compute scalar invertible matrices S, S' such that SLS' has atomic block diagonal form.*

Proof. We present the algorithm only for right monic L; the left monic case has essentially the same solution.

If the input L is not full or right monic the algorithm can efficiently detect that and output "failure". If L is an atom the algorithm will output that L is an atom and set the matrices S and S' to I_d. Otherwise, the algorithm will compute invertible scalar matrices S and S' such that

$$SLS' = \begin{pmatrix} L_1 & 0 & 0 & \dots & 0 \\ * & L_2 & 0 & \dots & 0 \\ * & * & L_3 & \dots & 0 \\ & & & \ddots & \\ * & * & * & \dots & L_r \end{pmatrix}, \tag{8}$$

where the matrix on the right is in atomic block diagonal form, that is, each linear matrix L_i is an atom.

Procedure Factor(L).

1. Test if L has full noncommutative rank using the algorithm in [17] or [13]. Test if L is right monic by checking if the matrix $[A_1 A_2 \dots A_n]$ has full row rank (which is d). If L is not full and right monic the algorithm outputs "fail".
2. Assume L is full and right monic. Using Lemma 20, find smallest positive integer $\ell \leq 2d$ and $\ell \times \ell$ scalar matrices $M_i, i \in [n]$ with entries from $\mathbb{F}$ (or a small degree extension of $\mathbb{F}$) such that $W = L(\bar{M})$ is $d \cdot \ell \times d \cdot \ell$ invertible scalar matrix. Compute the dilated linear matrix L' in the y_{ijk} variables as in Equation 7 which can be rewritten as:
$$L' = A_0 \otimes I_\ell + \sum_{i=1}^{n} A_i \otimes M_i + \sum_{i=1}^{n} \sum_{j,k=1}^{\ell} (A_i \otimes E_{jk}) \cdot y_{ijk}.$$
Let $L'' = W^{-1}L'$. Clearly $L''(\overline{0}) = I_{d\ell}$. Hence, by the algorithm of Theorem 28 we can either detect that L'' is an atom or factorize L''. If L'' is an atom then L is also an atom and the algorithm can output that and stop. Otherwise, L' is not an atom and by Theorem 28 we will obtain a basis change matrix T such that $TW^{-1}L'T^{-1} = TL''T^{-1} = \begin{pmatrix} C'' & 0 \\ * & D'' \end{pmatrix}$ where C'' and D'' are linear matrices of dimension $c'' \times c''$ and $d'' \times d''$ respectively, such that $c'' + d'' = d\ell$.
3. By linear shift of variables $y_{ijk} \leftarrow y_{ijk} - M_i(j,k)$ we obtain $\tilde{T}\tilde{L}\tilde{T}' = \begin{pmatrix} C' & 0 \\ * & D' \end{pmatrix}$ for some scalar invertible matrices $\tilde{T}, \tilde{T}'$ where $\tilde{L} = L(Y_1, \dots, Y_n)$.
4. Applying the algorithm of Lemma 34 to $\tilde{L}$, $\tilde{T}$, and $\tilde{T}'$, in deterministic polynomial time we obtain scalar invertible matrices $\tilde{S}, \tilde{S}'$ such that $\tilde{S}L\tilde{S}' = \begin{pmatrix} C & 0 \\ * & D \end{pmatrix}$ where C, D are square matrices of dimensions $e \times e$ and $g \times g$, respectively, such that $e + g = d$.
5. Recursively call Factor(C) and Factor(D). Let S_1, S_1' be the matrices returned by Factor(C) and S_2, S_2' be the matrices returned by Factor(D).
6. Let $S = (S_1 \oplus S_2)\tilde{S}$ and $S' = \tilde{S}'(S_1' \oplus S_2')$. Return the invertible scalar matrices S and S'. Note that at this stage SLS' has the desired atomic block diagonal form.

Next we give a brief argument for proving correctness of the above algorithm. Firstly, the algorithm declares L as an atom iff L is indeed an atom. To see this, we will prove L is not an atom iff L'' is not an atom. Forward direction is obvious. To prove the reverse direction of implication, let L'' is not an atom. Which implies $L' = WL''$ is not an atom. $\tilde{L}$ is a linear matrix obtained by substituting $M_i(j,k) = 0$ for all i, j, k in L'. Clearly, $\tilde{L}$ is not an atom as L' is not an atom. Using Lemma 34 it follows that L is not an atom. So we have established L is not a atom iff L'' is not an atom. So if input linear matrix L is an atom, the algorithm will correctly declare it to be an atom in step 2.

Now we argue that we will get correct atomic block diagonal form in the last step of the algorithm. Firstly, for giving recursive calls to the Factor procedure for the matrices C, D, we must have C, D to be right monic as stated in the claim below. This is proved by the same argument as in the proof of Theorem 21.

▷ Claim 37. Let $L \in \mathbb{F}\langle X\rangle^{d\times d}$ be a full and right monic linear matrix such that $P'LQ' = \begin{pmatrix} C & 0 \\ E & D \end{pmatrix}$ where C and D are linear matrices of dimensions $e \times e$, $g \times g$, respectively, such that $e + g = d$. Then both C, D are right monic.

By recursive calls Factor(C) and Factor(D) obtain matrices S_1, S_1', S_2, S_2' such that $S_1CS_1' = C'$ and $S_2DS_2' = D'$ are in atomic block diagonal form. We can write $\tilde{S}L\tilde{S}'$ as

$$
\begin{aligned}
&= \begin{pmatrix} C & 0 \\ E & D \end{pmatrix}\\
&= \begin{pmatrix} C & 0 \\ 0 & I_g \end{pmatrix}\begin{pmatrix} I_e & 0 \\ E & I_g \end{pmatrix}\begin{pmatrix} I_e & 0 \\ 0 & B \end{pmatrix}\\
&= (S_1^{-1} \oplus I_g)(C' \oplus I_g)({S'}_1^{-1} \oplus I_g)\begin{pmatrix} I_e & 0 \\ E & I_g \end{pmatrix}(I_e \oplus S_2^{-1})(I_e \oplus D')(I_e \oplus {S'}_2^{-1})\\
&= (S_1^{-1} \oplus I_g)(C' \oplus I_g)(I_e \oplus {S'}_2^{-1})\begin{pmatrix} I_e & 0 \\ S_2ES_1' & I_g \end{pmatrix}({S'}_1^{-1} \oplus I_g)(I_e \oplus D')(I_e \oplus {S'}_2^{-1})\\
&= (S_1^{-1} \oplus I_g)(I_e \oplus S_2^{-1})(C' \oplus I_g)\begin{pmatrix} I_e & 0 \\ S_2ES_1' & I_g \end{pmatrix}(I_e \oplus D')({S'}_1^{-1} \oplus I_g)(I_e \oplus {S'}_2^{-1})\\
&= (S_1^{-1} \oplus I_g)(I_e \oplus S_2^{-1})\begin{pmatrix} C' & 0 \\ S_2ES_1' & D' \end{pmatrix}({S'}_1^{-1} \oplus I_g)(I_e \oplus {S'}_2^{-1})\\
&= (S_1^{-1} \oplus S_2^{-1})\begin{pmatrix} C' & 0 \\ S_2ES_1' & D' \end{pmatrix}({S'}_1^{-1} \oplus {S'}_2^{-1}).
\end{aligned}
$$

Thus we have

$$(S_1 \oplus S_2)\tilde{S}L\tilde{S}'(S_1' \oplus S_2') = \begin{pmatrix} C' & 0 \\ S_2ES_1' & D' \end{pmatrix}.$$

As C' and D' are in atomic block diagonal form, it follows that $\begin{pmatrix} C' & 0 \\ S_2ES_1' & D' \end{pmatrix}$ is also in atomic block diagonal form. Letting $S = (S_1 \oplus S_2)\tilde{S}$ and $S' = \tilde{S}'(S_1' \oplus S_2')$, it follows that SLS' is in the desired atomic block diagonal form which proves the correctness of Factor procedure. In each call to the procedure (excluding the recursive calls) the algorithm takes $\mathrm{poly}(n, d, \log_2 q)$ time. The total number of recursive calls overall is bounded by d. Hence, the overall running time is $\mathrm{poly}(n, d, \log_2 q)$. This completes the proof of the theorem. ◀

For the factorization of f, we assume the stably associated full linear matrix L is left monic. After we obtain atomic block diagonal form as in Equation 8, we can factorize L into atomic factors by Theorem 28. Combined with Equation 5 we have

$$f \oplus I_s = PS'F_1'F_2' \cdots F_r'U'Q,$$

where each linear matrix F_i' is an atom, P is upper triangular with all 1's diagonal, Q is lower triangular with all 1's diagonal, and S' is scalar invertible and U' is a unit. Now, applying Lemma 31 and Theorem 32 we obtain the complete factorization of f into irreducible factors. This is summarized in the following.

▶ **Theorem 38.** *Let $f \in \mathbb{F}\langle X\rangle$ be a polynomial given by an arithmetic formula as input instance of $FACT(\mathbb{F}_q)$. Then there is a* $\mathrm{poly}(s, \log q, |X|)$ *time randomized algorithm that outputs a factorization $f = f_1 f_2 \cdots f_r$ such that each f_i is irreducible and is output as an algebraic branching program.*

Analogous to Corollary 33, when the polynomial is given in a sparse representation, we have

▶ **Corollary 39.** *If $f \in \mathbb{F}\langle X\rangle$ is a polynomial given as input in sparse representation (that is, an $\mathbb{F}_q$-linear combination of its monomials) then in randomized polynomial time we can compute a factorization into irreducible factors in sparse representation.*

5.2 Factorization over small finite fields

Finally, we briefly discuss the factorization problem over small finite fields. As explained in Section 1.2, the two steps in our factoring algorithm requiring randomization can be replaced with deterministic $\mathrm{poly}(s, q, |X|)$ time computation. Furthermore, as explained in Section 1.2, the matrix shift $(M_1, M_2, \ldots, M_n)$ required for the Theorem 36 can be obtained in deterministic polynomial time such that the entries of the matrices M_i are from $\mathbb{F}_q$ for each i. Putting it together, it gives us a deterministic factorization algorithm for noncommutative polynomials that are input as arithmetic formulas over $\mathbb{F}_q$. In summary, we have the following.

▶ **Theorem 40.** *Given as input a multivariate polynomial $f \in \mathbb{F}_q\langle X\rangle$ for a finite field $\mathbb{F}_q$ by a noncommutative arithmetic formula of size s, a factorization of f as a product $f = f_1 f_2 \cdots f_r$ can be computed in deterministic time* $\mathrm{poly}(s, q, |X|)$*, where each $f_i \in \mathbb{F}_q\langle X\rangle$ is an irreducible polynomial that is output as an algebraic branching program.*

6 Concluding Remarks

In this paper we present a randomized polynomial-time algorithm for the factorization of noncommutative polynomials *over finite fields* that are input as *arithmetic formulas*. The irreducible factors are output as algebraic branching programs.

Several open questions arise from our work. We mention two of them. The first question is the complexity of factorization over rationals of noncommutative polynomials given as arithmetic formulas. Our approach involves the crucial use of Ronyai's algorithm for invariant subspace computation which turns out to be a hard problem over rationals. We believe a different approach may be required for the rational case.

The use of Higman linearization prevents us from generalizing this approach to noncommutative polynomials given as arithmetic circuits. We do not know any non-trivial complexity upper bound for the factorization problem for noncommutative polynomials given as arithmetic circuits.

References

1 S.A Amitsur. Rational identities and applications to algebra and geometry. *Journal of Algebra*, 3(3):304–359, 1966. doi:10.1016/0021-8693(66)90004-4.

2 S.A. Amitsur and J. Levitzki. Minimal identities for algebras. *Proceedings of the American Mathematical Society*, 4(2):449–463, 1950.

3 V. Arvind, Pushkar S. Joglekar, and Gaurav Rattan. On the complexity of noncommutative polynomial factorization. *Inf. Comput.*, 262:22–39, 2018. doi:10.1016/j.ic.2018.05.009.

4 Vikraman Arvind, Abhranil Chatterjee, Rajit Datta, and Partha Mukhopadhyay. A special case of rational identity testing and the brešar-klep theorem. In Javier Esparza and Daniel Král', editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.10.

5 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.

6 W. Burnside. On the condition of reducibility of any group of linear substitutions. *Proceedings of London Mathematical Society*, 3:430–434, 1905.

7 P. M. Cohn. *Free Ideal Rings and Localization in General Rings.* New Mathematical Monographs. Cambridge University Press, 2006. doi:10.1017/CBO9780511542794.

8 P. M. Cohn. *Introduction To Ring Theory.* Springer, 2011.

9 Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

10 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.

11 Michael A. Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2014. URL: http://hdl.handle.net/1721.1/89843.

12 Katalin Friedl and Lajos Rónyai. Polynomial time solutions of some problems in computational algebra. In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 153–162. ACM, 1985. doi:10.1145/22145.22162.

13 Ankit Garg, Leonid Gurvits, Rafael Mendes de Oliveira, and Avi Wigderson. Operator scaling: Theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020. doi:10.1007/s10208-019-09417-z.

14 J Helton, Igor Klep, and Jurij Volčič. Factorization of Noncommutative Polynomials and Nullstellensätze for the Free Algebra. *International Mathematics Research Notices*, 2022(1):343–372, June 2020. doi:10.1093/imrn/rnaa122.

15 Graham Higman. The units of group-rings. *Proceedings of the London Mathematical Society*, s2-46(1):231–248, 1940. doi:10.1112/plms/s2-46.1.231.

16 Pavel Hrubes and Avi Wigderson. Non-commutative arithmetic circuits with division. *Theory Comput.*, 11:357–393, 2015. doi:10.4086/toc.2015.v011a014.

17 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative edmonds' problem and matrix semi-invariants. *Comput. Complex.*, 26(3):717–763, 2017. doi:10.1007/s00037-016-0143-x.

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

19 Erich Kaltofen. Factorization of polynomials given by straight-line programs. *Adv. Comput. Res.*, 5:375–412, 1989.

20 Erich Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comput.*, 9(3):301–320, 1990. doi:10.1016/S0747-7171(08)80015-6.

21 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and polynomial factorization. *Comput. Complex.*, 24(2):295–331, 2015. doi:10.1007/s00037-015-0102-y.

22 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.

23 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.

24 Lajos Rónyai. Computing the structure of finite algebras. *J. Symb. Comput.*, 9(3):355–373, 1990. doi:10.1016/S0747-7171(08)80017-X.

25 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

26 Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra (3. ed.)*. Cambridge University Press, 2013.

27 Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.