

On the Complexity of Noncommutative Polynomial Factorization

V. Arvind* Pushkar S Joglekar† Gaurav Rattan‡

Abstract

In this paper we study the complexity of factorization of polynomials in the free noncommutative ring $\mathbb{F}\langle x_1, x_2, \dots, x_n \rangle$ of polynomials over the field \mathbb{F} and noncommuting variables x_1, x_2, \dots, x_n . Our main results are the following:

- Although $\mathbb{F}\langle x_1, \dots, x_n \rangle$ is not a unique factorization ring, we note that *variable-disjoint* factorization in $\mathbb{F}\langle x_1, \dots, x_n \rangle$ has the uniqueness property. Furthermore, we prove that computing the variable-disjoint factorization is polynomial-time equivalent to Polynomial Identity Testing (both when the input polynomial is given by an arithmetic circuit or an algebraic branching program). We also show that variable-disjoint factorization in the black-box setting can be efficiently computed (where the factors computed will be also given by black-boxes, analogous to the work [KT90] in the commutative setting).
- As a consequence of the previous result we show that homogeneous noncommutative polynomials and multilinear noncommutative polynomials have unique factorizations in the usual sense, which can be efficiently computed.
- Finally, we discuss a polynomial decomposition problem in $\mathbb{F}\langle x_1, \dots, x_n \rangle$ which is a natural generalization of homogeneous polynomial factorization and prove some complexity bounds for it.

1 Introduction

Let \mathbb{F} be any field and $X = \{x_1, x_2, \dots, x_n\}$ 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

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

*Institute of Mathematical Sciences, Chennai, India, email: arvind@imsc.res.in

†Vishwakarma Institute of Technology, Pune, India, email: joglekar.pushkar@gmail.com

‡Institute of Mathematical Sciences, Chennai, India, email: grattan@imsc.res.in

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 $\deg(fg) = \deg f + \deg g$.

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 $\mathbb{F}\langle X \rangle$.

Polynomial Factorization Problem

The problem of polynomial factorization in the *commutative* polynomial ring $\mathbb{F}[x_1, x_2, \dots, x_n]$ is a classical well-studied problem in algorithmic complexity culminating in Kaltofen's celebrated efficient factorization algorithm [Ka89]. Kaltofen's algorithm builds on efficient algorithms for univariate polynomial factorization; there are deterministic polynomial-time algorithms over rationals and over fields of unary characteristic and randomized polynomial-time over large characteristic fields (the textbook [GG] contains a comprehensive excellent treatment of the subject). The basic idea in Kaltofen's algorithm is essentially a randomized reduction from multivariate factorization to univariate factorization using Hilbert's irreducibility theorem. Thus, we can say that Kaltofen's algorithm uses randomization in two ways: the first is in the application of Hilbert's irreducibility theorem, and the second is in dealing with *univariate* polynomial factorization over fields of large characteristic. In a recent paper Kopparty et al [KSS14] have shown that the first of these requirements of randomness can be eliminated, assuming an efficient algorithm as subroutine for the problem of *polynomial identity testing* for small degree polynomials given by circuits. More precisely, it is shown in [KSS14] that over finite fields of unary characteristic (or over rationals) polynomial identity testing is deterministic polynomial-time equivalent to multivariate polynomial factorization.

Thus, in the commutative setting it turns out that the complexity of multivariate polynomial factorization is closely related to polynomial identity testing (whose deterministic complexity is known to be related to proving superpolynomial size arithmetic circuit lower bounds).

Noncommutative Polynomial Factorization

The study of noncommutative arithmetic computation was initiated by Nisan [N91] in which he showed exponential size lower bounds for algebraic branching programs that compute the noncommutative permanent or the noncommutative determinant. Noncommutative polynomial identity testing was studied in [BW05, RS05]. In [BW05] a randomized polynomial time algorithm is shown for identity testing of polynomial degree noncommutative arithmetic circuits. For algebraic branching programs [RS05] give a deterministic polynomial-time algorithm. Proving superpolynomial size lower bounds for noncommutative arithmetic circuits computing the noncommutative permanent is open. Likewise, obtaining a deterministic polynomial-time identity test for polynomial degree noncommutative circuits is open.

In this context, it is interesting to ask if we can relate the complexity of noncommutative

factorization to noncommutative polynomial identity testing. However, there are various mathematical issues that arise in the study of noncommutative polynomial factorization.

Unlike in the commutative setting, the noncommutative polynomial ring $\mathbb{F}\langle X \rangle$ is *not* a unique factorization ring. A well-known example is the polynomial

$$xyx + x$$

which has two factorizations: $x(yx + 1)$ and $(xy + 1)x$. Both $xy + 1$ and $yx + 1$ are irreducible polynomials in $\mathbb{F}\langle X \rangle$.

There is a detailed theory of factorization in noncommutative rings [Co85, Co]. We will mention an interesting result on the structure of polynomial factorizations in the ring $R = \mathbb{F}\langle X \rangle$.

Two elements $a, b \in R$ are *similar* if there are elements $a', b' \in R$ such that $ab' = a'b$, and (i) a and a' do not have common nontrivial left factors, (ii) b and b' do not have common nontrivial right factors.

Among other results, Cohn [Co] has shown the following interesting theorem about factorizations in the ring $R = \mathbb{F}\langle X \rangle$.

Theorem 1.1 (Cohn's theorem) *For a polynomial $a \in \mathbb{F}\langle X \rangle$ let*

$$a = a_1 a_2 \dots a_r \text{ and } a = b_1 b_2 \dots b_s$$

be any two factorizations of a into irreducible polynomials in $\mathbb{F}\langle X \rangle$. Then $r = s$, and there is a permutation π of the indices $\{1, 2, \dots, r\}$ such that a_i and $b_{\pi(i)}$ are similar polynomials for $1 \leq i \leq r$.

For instance, consider the two factorizations of $xyx + x$ above. We note that polynomials $xy + 1$ and $yx + 1$ are similar. It is easy to construct examples of degree d polynomials in $\mathbb{F}\langle X \rangle$ that have $2^{\Omega(d)}$ distinct factorizations. Cohn [Co85] discusses a number of interesting properties of factorizations in $\mathbb{F}\langle X \rangle$. But it is not clear how to algorithmically exploit these to obtain an efficient algorithm in the general case.

Our Results

In this paper, we study some *restricted* cases of polynomial factorization in the ring $\mathbb{F}\langle X \rangle$ and prove the following results.

- We consider *variable-disjoint* factorization of polynomials in $\mathbb{F}\langle X \rangle$ into *variable-disjoint irreducibles*. It turns out that such factorizations are unique and computing them is polynomial-time equivalent to polynomial identity testing (for both noncommutative arithmetic circuits and algebraic branching programs).
- It turns out that we can apply the algorithm for variable-disjoint factorization to two special cases of factorization in $\mathbb{F}\langle X \rangle$: homogeneous polynomials and multilinear polynomials. These polynomials do have unique factorizations and we obtain efficient algorithms for computing them.
- We also study a natural polynomial decomposition problem for noncommutative polynomials and obtain complexity results.

2 Variable-disjoint Factorization Problem

In this section we consider the problem of factorizing a noncommutative polynomial $f \in \mathbb{F}\langle X \rangle$ into variable disjoint factors.

For a polynomial $f \in \mathbb{F}\langle X \rangle$ let $Var(f) \subseteq X$ denote the set of all variables occurring in nonzero monomials of f .

Definition 2.1 *A nontrivial variable-disjoint factorization of a polynomial $f \in \mathbb{F}\langle X \rangle$ is a factorization*

$$f = gh$$

such that $\deg g > 0$ and $\deg h > 0$, and $Var(g) \cap Var(h) = \emptyset$.

A polynomial f is variable-disjoint irreducible if does not have a nontrivial variable-disjoint factorization.

Clearly, all irreducible polynomials are also variable-disjoint irreducible. But the converse is not true. For instance, the familiar polynomial $xyx + x$ is variable-disjoint irreducible but not irreducible. Furthermore, all univariate polynomials in $\mathbb{F}\langle X \rangle$ are variable-disjoint irreducible.

We will study the complexity of variable-disjoint factorization for noncommutative polynomials. First of all, it is interesting that although we do not have the usual unique factorization in the ring $\mathbb{F}\langle X \rangle$, we can prove that every polynomial in $\mathbb{F}\langle X \rangle$ has a *unique* variable-disjoint factorization into variable-disjoint irreducible polynomials.¹

We can exploit the properties we use to show uniqueness of variable-disjoint factorization for computing the variable-disjoint factorization. Given $f \in \mathbb{F}\langle X \rangle$ as input by a noncommutative arithmetic circuit the problem of computing arithmetic circuits for the variable-disjoint irreducible factors of f is polynomial-time reducible to PIT for noncommutative arithmetic circuits. An analogous result holds for f given by an algebraic branching programs (ABPs). Hence, there is a deterministic polynomial-time algorithm for computing the variable-disjoint factorization of f given by an ABP. Also in the case when the polynomial $f \in \mathbb{F}\langle X \rangle$ is given as input by a black-box (appropriately defined) we give an efficient algorithm that gives black-box access to each variable-disjoint irreducible factor of f .

Remark 2.2 *Factorization of commutative polynomials into variable-disjoint factors is studied by Shpilka and Volkovich in [SV10]. They show a deterministic reduction to polynomial identity testing. However, the techniques used in their paper are specific to commutative rings, involving scalar substitutions, and do not appear useful in the noncommutative case. Our techniques for factorization are simple, essentially based on computing left and right partial derivatives of noncommutative polynomials given by circuits or branching programs.*

2.1 Uniqueness of variable-disjoint factorization

Although the ring $\mathbb{F}\langle X \rangle$ is not a unique factorization domain we show that factorization into variable-disjoint irreducible factors is unique.

For a polynomial $f \in \mathbb{F}\langle X \rangle$ let $mon(f)$ denote the set of nonzero monomials occurring in f .

¹Uniqueness of the factors is upto scalar multiplication.

Lemma 2.3 *Let $f = gh$ such that $\text{Var}(g) \cap \text{Var}(h) = \phi$ and $|\text{Var}(g)|, |\text{Var}(h)| \geq 1$. Then*

$$\text{mon}(f) = \{mw \mid m \in \text{mon}(g), w \in \text{mon}(h)\}.$$

Moreover, the coefficient of mw in f is the product of the coefficients of m in g and w in h .

Proof. Let $m \in \text{mon}(g)$ and $w \in \text{mon}(h)$. We will argue that the monomial mw is in $\text{mon}(f)$ and can be obtained in a unique way in the product gh . Namely, by multiplying $m \in \text{mon}(g)$ with $w \in \text{mon}(h)$.

Assume to the contrary that for some $u \in \text{mon}(g)$ and $v \in \text{mon}(h)$, $u \neq m$ we have $mw = uv$. Note that m and u are words over $\text{Var}(g)$ and w, v are words over $\text{Var}(h)$. Clearly, $|m| \neq |u|$ because $m \neq u$. Without loss of generality, we assume that $|u| > |m|$. As $mw = uv$, it follows that for some word $s \in X^*$, $|s| > 0$, we have $u = ms$ and $w = sv$.

That implies $\text{Var}(s) \subseteq \text{Var}(g) \cap \text{Var}(h)$ which contradicts variable disjointness of g and h . \square

Lemma 2.4 *Let $f = g.h$ and $f = u.v$ be two nontrivial variable-disjoint factorizations of f . That is,*

$$\begin{aligned} \text{Var}(g) \cap \text{Var}(h) &= \phi \\ \text{Var}(u) \cap \text{Var}(v) &= \phi. \end{aligned}$$

Then either $\text{Var}(g) \subseteq \text{Var}(u)$ and $\text{Var}(h) \supseteq \text{Var}(v)$ or $\text{Var}(u) \subseteq \text{Var}(g)$ and $\text{Var}(v) \supseteq \text{Var}(h)$.

Proof. Suppose to the contrary that $x \in \text{Var}(g) \setminus \text{Var}(u)$ and $y \in \text{Var}(u) \setminus \text{Var}(g)$. Let m be any monomial of f in which variable y occurs, and let $m = m_1 m_2$ such that $m_1 \in \text{mon}(g)$ and $m_2 \in \text{mon}(h)$. Clearly, y must occur in m_2 . Similarly, for any monomial m' of f in which variable x occurs, if $m' = m'_1 m'_2$, for $m'_1 \in \text{mon}(g)$ and $m'_2 \in \text{mon}(h)$, then x must occur in m'_1 . Thus, $x \in \text{Var}(g)$ and $y \in \text{Var}(h)$. Similarly, it also holds that $y \in \text{Var}(u)$ and $x \in \text{Var}(v)$. Thus we have:

1. $x \in \text{Var}(g)$ and $y \in \text{Var}(h)$
2. $y \in \text{Var}(u)$ and $x \in \text{Var}(v)$.

Hence, there are monomials of f in which both x and y occur. Furthermore, (1) above implies that in each monomial of f containing both x and y , x always occurs before y . On the other hand, (2) implies y occurs before x in each such monomial of f . This contradicts our assumption. \square

Lemma 2.5 *Let $f \in \mathbb{F}\langle X \rangle$ and suppose $f = gh$ and $f = uv$ are two variable-disjoint factorizations of f such that $\text{Var}(g) = \text{Var}(u)$. Then $g = \alpha u$ and $h = \beta v$ for scalars $\alpha, \beta \in \mathbb{F}$.*

Proof. As $f = gh = uv$ are variable-disjoint factorizations, by Lemma 2.3 each monomial $m \in \text{mon}(f)$ is uniquely expressible as a product $m = m_1 m_2$ for $m_1 \in \text{mon}(g)$ and $m_2 \in \text{mon}(h)$, and as a product $m = m'_1 m'_2$ for $m'_1 \in \text{mon}(u)$ and $m'_2 \in \text{mon}(v)$.

As $\text{Var}(g) = \text{Var}(u)$, we notice that $\text{Var}(h) = \text{Var}(v)$, because $\text{Var}(f) = \text{Var}(g) \uplus \text{Var}(h)$ and $\text{Var}(f) = \text{Var}(u) \uplus \text{Var}(v)$. Now, from $m = m_1 m_2 = m'_1 m'_2$ it immediately follows that $m_1 = m'_1$ and $m_2 = m'_2$. Hence, $\text{mon}(g) = \text{mon}(u)$ and $\text{mon}(h) = \text{mon}(v)$.

Furthermore, if m is a monomial of maximum degree in g , by taking the left partial derivative of f w.r.t. m we have

$$\frac{\partial^\ell f}{\partial m} = \alpha' h = \beta' v,$$

where α' and β' are coefficient of m in g and u respectively. It follows that $h = \beta v$ for some $\beta \in \mathbb{F}$. Similarly, by taking the right partial derivative of f w.r.t. a maximum degree monomial in h we can see that $g = \alpha u$ for $\alpha \in \mathbb{F}$. \square

We now prove the uniqueness of variable-disjoint factorizations in $\mathbb{F}\langle X \rangle$.

Theorem 2.6 *Every polynomial in $\mathbb{F}\langle X \rangle$ has a unique variable-disjoint factorization as a product of variable-disjoint irreducible factors, where the uniqueness is upto scalar multiples of the irreducible factors.*

Proof. We prove the theorem by induction on the degree d of polynomials in $\mathbb{F}\langle X \rangle$. The base case $d = 1$ is obvious, since degree 1 polynomials are irreducible and hence also variable-disjoint irreducible. Assume as induction hypothesis that the theorem holds for polynomials of degree less than d . Let $f \in \mathbb{F}\langle X \rangle$ be a polynomial of degree d . If f is variable-disjoint irreducible there is nothing to prove. Suppose f has nontrivial variable-disjoint factors. Let $f = gh$ be a *nontrivial* variable-disjoint factorization such that $\text{Var}(g)$ is *minimum*. Such a factorization must exist because of Lemmas 2.3 and 2.4. Furthermore, the set $\text{Var}(g)$ is uniquely defined and by Lemma 2.5 the left factor g is unique upto scalar multiples. Applying induction hypothesis to the polynomial h now completes the induction step. \square

2.2 Equivalence with PIT

Theorem 2.7 *Let $f \in \mathbb{F}\langle X \rangle$ be a polynomial as input instance for variable-disjoint factorization. Then*

1. *If f is input by an arithmetic circuit of degree d and size s there is a randomized $\text{poly}(s, d)$ time algorithm that factorizes f into variable-disjoint irreducible factors.*
2. *If f is input by an algebraic branching program there is a deterministic polynomial-time algorithm that factorizes f into its variable-disjoint irreducible factors.*

Proof. We first consider the most general case of the algorithm when $f \in \mathbb{F}\langle X \rangle$ is given by an arithmetic circuit of polynomially bounded syntactic degree. The algorithm specializes to the other cases too. The algorithm must compute an arithmetic circuit for each variable-disjoint factor of f . We explain the polynomial-time reduction to Polynomial Identity Testing (PIT) for noncommutative arithmetic circuits.

Let $d = \deg f$ which can be computed by first computing circuits for the polynomially many homogeneous parts of f , and then using PIT as subroutine on each of them.

Next, we compute a monomial $m \in \text{mon}(f)$ of maximum degree d with polynomially many subroutine calls to PIT. The basic idea here is to do a prefix search for m . More precisely, suppose we have computed a prefix m' such that the left partial derivative $\frac{\partial^\ell f}{\partial m'}$ computes a

polynomial of degree $d - |m'|$ then we extend m' with the first variable $x_i \in X$ such that $\frac{\partial^\ell f}{\partial m' x}$ computes a polynomial of degree $d - |m'| - 1$. Proceeding thus, we will compute the lexicographically first monomial $m \in \text{mon}(f)$ of degree d .

Now, starting with $d_1 = 1$ we look at all factorizations $m = m_1.m_2$ of monomial m with $|m_1| = d_1$.

Let $h = \frac{\partial^\ell f}{\partial m_1}$ and $g = \frac{\partial^r f}{\partial m_2}$, circuits for which can be efficiently computed from the given circuit for f . Let α , β and γ be the coefficients of m in f , m_1 in g , and m_2 in h , respectively (which can be computed in deterministic polynomial time [AMS10]).

Next we compute $\text{Var}(g)$ and $\text{Var}(h)$. Notice that $x_i \notin \text{Var}(g)$ if and only if g and $g|_{x_i=0}$ are not identical polynomials. Thus, we can easily determine $\text{Var}(g)$ and $\text{Var}(h)$ with n subroutine calls to PIT.

Clearly, if $\text{Var}(g) \cap \text{Var}(h) \neq \emptyset$ then we do not have a candidate variable-disjoint factorization that splits m as $m_1 m_2$ and we continue with incrementing the value of d_1 . Else, we check if

$$f = \frac{\alpha}{\beta\gamma} gh, \quad (1)$$

with a subroutine call to PIT. If $f = \frac{\alpha}{\beta\gamma} gh$ then $\frac{\alpha}{\beta\gamma} g$ is the unique leftmost variable-disjoint irreducible factor of f (upto scalar multiplication), and we continue the computation. Otherwise, we continue the search with incrementing the value of d_1 .

In the general step, suppose we have already computed $f = g_1 g_2 \dots g_i h_i$, where g_1, g_2, \dots, g_i are the successive variable-disjoint irreducible factors from the left. There will be a corresponding factorization of the monomial m as

$$m = m_1 m_2 \dots m_i m'_i$$

where m_j occurs in g_j , $1 \leq j \leq i$ and m'_i occurs in h_i . Notice that the polynomial $h_i = \frac{\partial^\ell f}{\partial m_1 m_2 \dots m_i}$ can be computed by a small noncommutative arithmetic circuit obtained from f by partial derivatives. This describes the overall algorithm proving that variable-disjoint factorization of f is deterministic polynomial-time reducible to PIT when f is given by arithmetic circuits. The algorithm outputs arithmetic circuits for each of the variable-disjoint irreducible factors of f .

We note that the algorithm specializes to the case when f is given by an ABP. The variable-disjoint irreducible factors are computed by ABPs in this case.

Finally, to complete the proof we note that in the case of noncommutative arithmetic circuits of polynomial degree there is a randomized polynomial time PIT algorithm [BW05]. For polynomials given as ABPs, there is a deterministic polynomial-time PIT algorithm [RS05]. \square

Next we consider variable-disjoint factorization of polynomials input in sparse representation and show that the problem is solvable in deterministic logspace (even by constant-depth circuits). Recall that AC^0 circuits mean a family of circuits $\{C_n\}$, where C_n for length n inputs, such that C_n has polynomially bounded size and constant-depth and is allowed unbounded fanin AND and OR gates. The class of TC^0 circuits is similarly defined, but is additionally allowed unbounded fanin majority gates. The logspace uniformity condition means that there is a logspace transducer that outputs C_n on input 1^n for each n .

Theorem 2.8 *Let $f \in \mathbb{F}\langle X \rangle$ be a polynomial input instance for variable-disjoint factorization given in sparse representation.*

- (a) When \mathbb{F} is a fixed finite field the variable-disjoint factorization is computable in deterministic logspace (more precisely, even by logspace-uniform AC^0 circuits).
- (b) When \mathbb{F} is the field of rationals the variable-disjoint factorization is computable in deterministic logspace (even by logspace-uniform TC^0 circuits).

Proof. We briefly sketch the simple circuit constructions. Let us first consider the case when \mathbb{F} is a fixed finite field. The idea behind the AC^0 circuit for it is to try all pairs of indices $1 \leq a < b < d$ in parallel and check if f has an irreducible factor of degree $b - a + 1$ located in the (a, b) range. More precisely, we will check in parallel for a variable-disjoint factorization $f = g_1 g_2 g_3$, where g_1 is of degree $a - 1$, g_2 is variable-disjoint irreducible of degree $b - a + 1$, and g_3 is of degree $d - b$. If such a factorization exists then we will compute the unique (upto scalar multiples) irreducible polynomial g_3 located in the (a, b) range of degrees.

In order to carry out this computation, we take a highest degree nonzero monomial m of f and factor it as $m = m_1 m_2 m_3$, where m_2 is located in the (a, b) range of m . Clearly, the polynomial

$$\frac{\partial^\ell}{\partial m_1} \left(\frac{\partial^r f}{\partial m_3} \right)$$

has to be g_2 upto a scalar multiple. Likewise, $\frac{\partial^\ell f}{\partial m_1 m_2}$ is g_3 and $\frac{\partial^r f}{\partial m_2 m_3}$ is g_1 upto scalar multiples. Since f is given in sparse representation, we can compute these partial derivatives for each monomial in parallel in constant depth.

We can then check using a polynomial size constant-depth circuit if $f = g_1 g_2 g_3$ upto an overall scalar multiple (since we have assumed \mathbb{F} is a constant size finite field). For a given a , the least b for which such a factorization occurs will give us the variable-disjoint irreducible factor located in the (a, b) range. For each $a < d$ we can carry out this logspace computation looking for the variable-disjoint irreducible factor (if there is one) located at (a, b) range for some $b > a$.

In this manner, the entire constant-depth circuit computation will output all variable-disjoint irreducible factors of f from left to right. This concludes the proof for the constant size finite field case.

Over the field of rationals, we use the same approach as above. The problem of checking if $f = g_1 g_2 g_3$ upto an overall scalar multiple will require integer multiplication of a constant number of integers. This can be carried out by a constant-depth threshold circuit. This gives the TC^0 upper bound. \square

We now observe that PIT for noncommutative arithmetic circuits is also deterministic polynomial-time reducible to variable-disjoint factorization, making the problems polynomial-time equivalent.

Theorem 2.9 *Polynomial identity testing for noncommutative polynomials $f \in \mathbb{F}\langle X \rangle$ given by arithmetic circuits (of polynomial degree) is deterministic polynomial-time equivalent to variable-disjoint factorization of polynomials given by noncommutative arithmetic circuits.*

Proof. Clearly proof of Theorem 2.7 gives a reduction from variable-disjoint factorization to PIT. To see the other direction, let $f \in \mathbb{F}\langle X \rangle$ and $y, z \notin X$ be two new variables. We observe that the polynomial $f \in \mathbb{F}\langle X \rangle$ is identically zero iff the polynomial $f + y.z$ has a nontrivial variable-disjoint factorization. This gives a reduction from PIT to variable-disjoint factorization. \square

2.3 Black-Box Variable-Disjoint Factorization Algorithm

In this subsection we give an algorithm for variable-disjoint factorization when the input polynomial $f \in \mathbb{F}\langle X \rangle$ is given by black-box access. We explain the black-box model below:

In this model, the polynomial $f \in \mathbb{F}\langle X \rangle$ can be evaluated on any n -tuple of matrices (M_1, M_2, \dots, M_n) where each M_i is a $t \times t$ matrix over \mathbb{F} (or a suitable extension field of \mathbb{F}) and get the resulting $t \times t$ matrix $f(M_1, M_2, \dots, M_n)$ as output.

The algorithm for black-box variable-disjoint factorization takes as input such a black-box access for f and outputs black-boxes for each variable-disjoint irreducible factor of f . More precisely, the factorization algorithm, on input i , makes calls to the black-box for f and works as black-box for the i^{th} variable-disjoint irreducible factor of f , for each i .

The efficiency of the algorithm is measured in terms of the number of calls to the black-box and the size t of the matrices. In this section we will design a variable disjoint factorization algorithm that makes polynomially many black-box queries to f on matrices of polynomially bounded dimension. We state the theorem formally and then prove it in the rest of this section.

Theorem 2.10 *Suppose $f \in \mathbb{F}\langle X \rangle$ is a polynomial of degree bounded by D , given as input via black-box access. Let $f = f_1 f_2 \dots f_r$ be the variable-disjoint factorization of f . Then there is a polynomial-time algorithm that, on input i , computes black-box access to the i^{th} factor f_i .*

Proof. In the proof we sketch the algorithm to give an overview. We then explain the main technical parts in the next two lemmas. The black-box algorithm closely follows the white-box algorithm already described in the first part of this section. The main steps of the algorithm are:

1. We compute a nonzero monomial m of highest degree $d \leq D$ in f by making black-box queries to f . We can also compute the coefficient $f(m)$ of m by making black-box queries to f .
2. Then we compute the factorization of this monomial

$$m = m_1 m_2 \dots m_r$$

which, as in the white-box case, is precisely how the monomial m factorizes according to the unique variable-disjoint factorization $f = f_1 f_2 \dots f_r$ of f (that is $m_i \in \text{mon}(f_i)$).

3. In order to find this factorization of the monomial m , it suffices to compute all factorizations $m = m' m''$ of m such that $f = (\frac{\partial^\ell f}{\partial m'^\ell})(\frac{\partial^r f}{\partial m''^r})$ is a variable disjoint factorization (this is done using Equation 1, exactly as in the proof of Theorem 2.7).

Finding all such factorizations will allow us to compute the m_i . However, since we have only black-box access to f we will achieve this by creating and working with black-box access to the partial derivatives $(\frac{\partial^\ell f}{\partial m'^\ell}$ and $\frac{\partial^r f}{\partial m''^r})$ for each factorization $m = m' m''$. We explain the entire procedure in detail.

Starting with $d_1 = 1$ we look at all factorizations $m = m' m''$ of monomial m with $|m'| = d_1$.

Suppose we have computed black-boxes for $h = \frac{\partial^\ell f}{\partial m'^\ell}$ and $g = \frac{\partial^r f}{\partial m''^r}$. Let α , β and γ be the coefficients of m in f , m' in g , and m'' in h , which can be computed using black-box queries to f , g and h , respectively.

Next we compute $Var(g)$ and $Var(h)$. Notice that $x_i \notin Var(g)$ if and only if g and $g|_{x_i=0}$ are not identical polynomials. This amounts to performing PIT with black-box access to g . Thus, we can easily determine $Var(g)$ and $Var(h)$ by appropriately using black-box PIT.

Clearly, if $Var(g) \cap Var(h) \neq \emptyset$ then we do not have a candidate variable-disjoint factorization that splits m as m_1m_2 and we continue with incrementing the value of d_1 . Else, we check if

$$f = \frac{\alpha}{\beta\gamma}gh, \quad (2)$$

with a subroutine call to PIT (like Equation 1 in Theorem 2.7). If $f = \frac{\alpha}{\beta\gamma}gh$ then $\frac{\alpha}{\beta\gamma}g$ is the unique leftmost variable-disjoint irreducible factor of f (upto scalar multiplication), and we continue the computation. Otherwise, we continue the search with incrementing the value of d_1 .

Suppose we have computed the monomial factorization $m = m_1m_2 \dots m_im'_i$. There is a corresponding factorization of f as $f = f_1f_2 \dots f_ih_i$, where f_1, f_2, \dots, f_i are the successive variable-disjoint irreducible factors from the left, and m_j occurs in f_j , $1 \leq j \leq i$ and m'_i occurs in h_i . Then we have black-box access to $h_i = \frac{\partial^\ell f}{\partial m_1 m_2 \dots m_i}$ and we can find its leftmost irreducible variable-disjoint factor as explained above. This completes the overall algorithm sketch for efficiently computing black-boxes for the irreducible variable-disjoint factors of f given by black-box. \square

Remark 2.11 *The variable-disjoint irreducible factors of f are unique only upto scalar multiples. However, we note that the algorithm in Theorem 2.10 computes as black-box some fixed scalar multiple for each variable-disjoint irreducible factor f_i . Since we can compute the coefficient of the monomial m in f (where m is computed in the proof of Theorem 2.10), we can even ensure that the product $f_1f_2 \dots f_r$ equals f by appropriate scaling.*

Lemma 2.12 *Given a polynomial f of degree d with black-box access, we can compute a degree- d nonzero monomial of f , if it exists, with at most nd calls to the black-box on $(D+1)2d \times (D+1)2d$ matrices.*

Proof. Similar to the white-box case, we will do a prefix search for a nonzero degree- d monomial m . We explain the matrix-valued queries we will make to the black-box for f in two stages. In the first stage the query matrices will have the noncommuting variables from X as entries. In the second stage, which we will actually use to query f , we will substitute matrices (with scalar entries) for the variables occurring in the query matrices.

In order to check if there is a non-zero degree d monomial in f with x_i as first variable, we evaluate f on $(D+1) \times (D+1)$ matrices X_1, \dots, X_n defined as follows.

$$X_1 = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & 0 & x_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & x_1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}$$

and for $i > 1$,

$$X_i = \begin{pmatrix} 0 & 0 & \dots & \dots & 0 \\ \vdots & 0 & x_i & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & x_i \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}.$$

Therefore, $f(X_1, \dots, X_n)$ is a $(D+1) \times (D+1)$ matrix with entries in $\mathbb{F}\langle X \rangle$. In particular, by our choice of the matrices above, the $(1, d+1)^{th}$ entry of $f(X_1, \dots, X_n)$ will precisely be $\frac{\partial^\ell f_d}{\partial x_1}$, where f_d is the degree d homogeneous part of f . To check if $\frac{\partial^\ell f_d}{\partial x_1}$ is nonzero, we can substitute random $t \times t$ matrices for each variable x_i occurring in the matrices X_i and evaluate f to obtain a $((D+1)t) \times ((D+1)t)$ matrix. In this matrix $(1, d+1)^{th}$ block of size $t \times t$ will be nonzero with high probability for $t = 2d$ [BW05]. In general, suppose m' is a degree d monomial that is nonzero in f . We search for the first variable x_i such that $m'x_i$ is a prefix of some degree d nonzero monomial by similarly creating black-box access to $\frac{\partial^\ell f_d}{\partial m'x_i}$ and then substituting random $t \times t$ matrices for the variables and evaluating f to obtain a $((D+1)t) \times ((D+1)t)$ matrix. The $(1, d+1)^{th}$ block of size $t \times t$ is nonzero with high probability [BW05] if $m'x_i$ is a prefix for some nonzero degree d monomial. Continuing thus, we can obtain a nonzero monomial of highest degree d in f . \square

We now describe an efficient algorithm that creates black box access for left and right partial derivatives of f w.r.t. monomials. Let $m \in X^*$ be a monomial. We recall that the left partial derivative $\frac{\partial^\ell f}{\partial m}$ of f w.r.t. m is the polynomial

$$\frac{\partial^\ell f}{\partial m} = \sum_{f(mm') \neq 0} f(mm')m'.$$

Similarly, the right partial derivative $\frac{\partial^r f}{\partial m}$ of f w.r.t. m is the polynomial

$$\frac{\partial^r f}{\partial m} = \sum_{f(m'm) \neq 0} f(m'm)m'.$$

Lemma 2.13 *Given a polynomial f of degree d with black-box access, there are efficient algorithms that give black-box access to the polynomials $\frac{\partial^\ell f}{\partial m_1}$, $\frac{\partial^r f}{\partial m_2}$ for any monomials $m_1, m_2 \in X^*$. Furthermore, there is also an efficient algorithm giving black-box access to the polynomial $\frac{\partial^\ell}{\partial m_1} \left(\frac{\partial^r f}{\partial m_2} \right)$.*

Proof. Given the polynomial f and a monomial m_1 , we first show how to create black-box access to $\frac{\partial^\ell f}{\partial m_1}$. Let the monomial m_1 be $x_{i_1} \dots x_{i_k}$. Define matrices X_1, \dots, X_n as follows:

$$X_i = \left(\begin{array}{cccc|cccc} 0 & T_i[1] & 0 & \dots & 0 & \dots & \dots & 0 \\ \vdots & 0 & T_i[2] & \ddots & \ddots & \vdots & \dots & \vdots \\ \vdots & \ddots & 0 & T_i[k] & 0 & \dots & \dots & 0 \\ \hline 0 & \dots & \dots & 0 & x_i & \dots & \dots & 0 \\ \vdots & \dots & \dots & \vdots & 0 & x_i & 0 & 0 \\ \vdots & \dots & \dots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots & \vdots & \ddots & \ddots & x_i \\ 0 & \dots & \dots & 0 & 0 & \dots & \dots & 0 \end{array} \right).$$

For $1 \leq r \leq k$, the entry $T_i[r] = 1$ if $x_{i_r} = x_i$ and $T_i[r] = 0$ otherwise.

The black-box access to $\frac{\partial^\ell f}{\partial m_1}$ on input matrices M_1, \dots, M_n of size $t \times t$ can be created as follows. Note that in the $(D+1) \times (D+1)$ matrix $f(X_1, \dots, X_n)$, the $(1, j+1)^{th}$ location contains $\frac{\partial^\ell f_j}{\partial m_1}$, where f_j is the degree j^{th} homogeneous part of f . Now, suppose for each variable x_i in the matrix X_i we substitute a $t \times t$ scalar-entry matrix M_i and compute the resulting $f(X_1, \dots, X_n)$ which is now a $(D+1) \times (D+1)$ block matrix whose entries are $t \times t$ scalar-entry matrices. Then the block matrix located at the $(1, j+1)^{st}$ entry for $j \in \{2, \dots, d+1\}$ is the evaluation of $\frac{\partial^\ell f_j}{\partial m_1}$ on M_1, \dots, M_n . We output the sum of these matrix entries over $2 \leq j \leq d+1$ as the black-box output.

Next, we show how to create black-box access to $\frac{\partial^r f}{\partial m_2}$. Let the monomial $m_2 = x_{i_1} x_{i_2} \dots x_{i_k}$. We will define matrices $X_i^{(j)}$ for $i \in [n], j \in \{k, k+1, \dots, D\}$ as follows:

$$X_i^{(j)} = \left(\begin{array}{cccc|cccc} 0 & x_i & 0 & \dots & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \dots & \dots & \vdots \\ \vdots & \ddots & 0 & x_i & 0 & \dots & \dots & 0 \\ \hline 0 & \dots & \dots & 0 & T_i[1] & \dots & \dots & 0 \\ \vdots & \dots & \dots & \vdots & 0 & T_i[2] & 0 & 0 \\ \vdots & \dots & \dots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \dots & \dots & \vdots & \vdots & \ddots & \ddots & T_i[k] \\ 0 & \dots & \dots & 0 & 0 & \dots & \dots & 0 \end{array} \right).$$

The matrix $X_i^{(j)}$ is a $(j+1) \times (j+1)$ matrix where, in the figure above the top left block is of dimension $(j-k) \times (j-k+1)$. Here $T_i[r] = 1$ if $x_{i_r} = x_i$ and $T_i[r] = 0$ otherwise.

Finally, define the block diagonal matrix $X_i = \text{diag}(X_i^{(k)}, \dots, X_i^{(D)})$.

We now describe the black-box access to $\frac{\partial^r f}{\partial m_2}$. Let $x_i \leftarrow M_i$ be the input assignment of $t \times t$ matrices M_i to the variables x_i . This results in matrices $\hat{X}_i, 1 \leq i \leq n$, where \hat{X}_i is obtained from X_i by replacing x_i by M_i .

The query $f(\hat{X}_1, \hat{X}_2, \dots, \hat{X}_n)$ to f gives a block diagonal matrix $\text{diag}(N_k, N_{k+1}, \dots, N_D)$. Here the matrix N_j is a $(j+1) \times (j+1)$ block matrix with entries that are $t \times t$ matrices (over \mathbb{F}). The $(1, j+1)^{th}$ block in N_j is $\frac{\partial^r f_j}{\partial m_2}$ evaluated at M_1, M_2, \dots, M_n .

Hence the sum of the $(1, j + 1)^{th}$ blocks of the different $N_j, k \leq k \leq D$ gives the desired black-box access to $\frac{\partial^r f_j}{\partial m_2}$. □

3 Factorization of Multilinear and Homogeneous Polynomials

In this section we briefly discuss two interesting special cases of the standard factorization problem for polynomials in $\mathbb{F}\langle X \rangle$. Namely, the factorization of multilinear polynomials and the factorization of homogeneous polynomials. It turns out, as we show, that factorization of multilinear polynomials coincides with their variable-disjoint factorization. In the case of homogeneous polynomials, it turns out that by renaming variables we can reduce the problem to variable-disjoint factorization.

In summary, multilinear polynomials as well as homogeneous polynomials have unique factorizations in $\mathbb{F}\langle X \rangle$, and by the results of the previous section these can be efficiently computed.

A polynomial $f \in \mathbb{F}\langle X \rangle$ is *multilinear* if in every nonzero monomial of f every variable in X occurs at most once. We begin by observing some properties of multilinear polynomials. Let $\text{Var}(f)$ denote the set of all indeterminates from X which appear in some nonzero monomial of f .

It turns out that factorization and variable-disjoint factorization of multilinear polynomials coincide.

Lemma 3.1 *Let $f \in \mathbb{F}\langle X \rangle$ be a multilinear polynomial and $f = gh$ be any nontrivial factorization of f . Then, $\text{Var}(g) \cap \text{Var}(h) = \emptyset$.*

Proof. Suppose $x_i \in \text{Var}(g) \cap \text{Var}(h)$ for some $x_i \in X$. Let m_1 be a monomial in g of maximal degree which also has the indeterminate x_i occurring in it. Similarly, let monomial m_2 be of maximal degree in h with x_i occurring in it. The product monomial $m_1 m_2$ is not multilinear and it cannot be nonzero in f . This monomial must therefore be cancelled in the product gh , which means there are nonzero monomials m'_1 of g and m'_2 of h such that $m_1 m_2 = m'_1 m'_2$. Since $\deg(m'_1) \leq \deg(m_1)$ and $\deg(m'_2) \leq \deg(m_2)$ the only possibility is that $m_1 = m'_1$ and $m_2 = m'_2$ which means the product monomial $m_1 m_2$ has a nonzero coefficient in f , contradicting the multilinearity of f . This completes the proof. □

Thus, by Theorem 2.6 multilinear polynomials in $\mathbb{F}\langle X \rangle$ have unique factorization. Furthermore, the algorithms described in Section 2.2 can be applied to efficiently factorize multilinear polynomials.

We now briefly consider factorization of homogeneous polynomials in $\mathbb{F}\langle X \rangle$.

Definition 3.2 *A polynomial $f \in \mathbb{F}\langle X \rangle$ is said to be homogeneous of degree d if every nonzero monomial of f is of degree d .*

Homogeneous polynomials do have the unique factorization property. This is attributed to J.H. Davenport in [Ca10]. However, we argue this by reducing the problem to variable-disjoint factorization.

Given a degree- d homogeneous polynomial $f \in \mathbb{F}\langle X \rangle$, we apply the following simple transformation to f : For each variable $x_i \in X$ we introduce d variables $x_{i1}, x_{i2}, \dots, x_{id}$. For

each monomial $m \in \text{mon}(f)$, we replace the occurrence of variable x_i in the j^{th} position of m by variable x_{ij} . The new polynomial f' is in $\mathbb{F}\langle\{x_{ij}\}\rangle$. The crucial property of homogeneous polynomials we use is that for any factorization $f = gh$ both g and h must be homogeneous.

Lemma 3.3 *Let $f \in \mathbb{F}\langle X \rangle$ be a homogeneous degree d polynomial and f' be the polynomial in $\mathbb{F}\langle\{x_{ij}\}\rangle$ obtained as above. Then*

- *The polynomial f' is variable-disjoint irreducible iff f is irreducible.*
- *If $f' = g'_1 g'_2 \dots g'_t$ is the variable-disjoint factorization of f' , where each g'_k is variable-disjoint irreducible then, correspondingly $f = g_1 g_2 \dots g_t$ is a factorization of f into irreducibles g_k , where g_k is obtained from g'_k by replacing each variable x_{ij} in g'_k by x_i .*

Proof. The first part follows because if f is reducible and $f = gh$ then $f' = g'h'$, where g' is obtained from g by replacing the variables x_i by x_{ij} , and h' is obtained from h by replacing the variables x_i by $x_{i,j+s}$, where $s = \deg g$.

For the second part, consider the product $g_1 g_2 \dots g_t$. As all the factors g_k are homogeneous, it follows that each g_k is irreducible for otherwise g_k is not variable disjoint irreducible. Furthermore, any monomial m in $g_1 g_2 \dots g_t$ can be uniquely expressed as $m = m_1 m_2 \dots m_t$, where $m_k \in \text{mon}(g_k)$ for each k . Thus, for each i and j , replacing the j^{th} occurrence of x_i by x_{ij} in the product $g_1 g_2 \dots g_t$ will give us f' again. Hence $g_1 g_2 \dots g_t$ is f . \square

It follows easily that factorization of homogeneous polynomials is reducible to variable-disjoint factorization and we can solve it efficiently using Theorems 2.7 and 2.8, depending on how the polynomial is input. We summarize this formally.

Theorem 3.4 *Homogeneous polynomials $f \in \mathbb{F}\langle X \rangle$ have unique factorizations into irreducible polynomials. Moreover, this factorization can be efficiently computed:*

- *Computing the factorization of a homogeneous polynomial f given by an arithmetic circuit of polynomial degree is polynomial-time reducible to computing the variable-disjoint factorization of a polynomial given by an arithmetic circuit.*
- *Factorization of f given by an ABP is constant-depth reducible to variable-disjoint factorization of polynomials given by ABPs.*
- *Factorization of f given in sparse representation is constant-depth reducible to variable-disjoint factorization of polynomials given by sparse representation.*

4 A Polynomial Decomposition Problem

Given a degree d homogeneous noncommutative polynomial $f \in \mathbb{F}\langle X \rangle$, a number k in unary as input we consider the following decomposition problem, denoted by SOP (for sum of products decomposition):

Does f admit a decomposition of the form

$$f = g_1 h_1 + \dots + g_k h_k?$$

where each $g_i \in \mathbb{F}\langle X \rangle$ is a homogeneous polynomial of degree d_1 and each $h_i \in \mathbb{F}\langle X \rangle$ is a homogeneous polynomial of degree d_2 . Notice that this problem is a generalization of homogeneous polynomial factorization. Indeed, homogeneous factorization is simply the case when $k = 1$.

Remark 4.1 *As mentioned in [Ar14], it is interesting to note that for commutative polynomials the complexity of SOP is open even in the case $k = 2$. However, when f is of constant degree then it can be solved efficiently by applying a very general algorithm [Ar14] based on a regularity lemma for polynomials.*

When the input polynomial f is given by an arithmetic circuit, we show that SOP is in $\text{MA} \cap \text{coNP}$. On the other hand, when f is given by an algebraic branching program then SOP can be solved in deterministic polynomial time by some well-known techniques (we can even compute ABPs for the g_i and h_i for the minimum k).

Theorem 4.2 *Suppose a degree d homogeneous noncommutative polynomial $f \in \mathbb{F}\langle X \rangle$, and positive integer k encoded in unary are the input to SOP:*

- (a) *If f is given by a polynomial degree arithmetic circuit then SOP is in $\text{MA} \cap \text{coNP}$.*
- (b) *If f is given by an algebraic branching program then SOP is in deterministic polynomial time (even in randomized NC^2).*
- (c) *If f is given in the sparse representation then SOP is equivalent to the problem of checking if the rank of a given matrix is at most k . In particular, if \mathbb{F} is the field of rationals, SOP is complete for the complexity class C=L .²*

We first focus on proving part (a) of the theorem. If (f, k) is a “yes” instance to SOP, then we claim that there exist small arithmetic circuits for the polynomials $g_i, h_i, i \in [k]$.

We define the *partial derivative matrix* A_f for the polynomial f as follows. The rows of A_f are indexed by degree d_1 monomials and the columns of A_f by degree d_2 monomials (over variables in X). For the row labeled m and column labeled m' , the entry $A_{m,m'}$ is defined as

$$A_{m,m'} = f(mm').$$

The key to analyzing the decomposition of f is the rank of the matrix A_f .

Claim 4.3 *Let $f \in \mathbb{F}\langle X \rangle$ be a homogeneous degree d polynomial.*

- (a) *Then f can be decomposed as $f = g_1h_1 + \dots + g_kh_k$ for homogeneous degree d_1 polynomials g_i and homogeneous degree d_2 polynomials h_i if and only if the rank of A_f is bounded by k .*
- (b) *Furthermore, if f is computed by a noncommutative arithmetic circuit C then if the rank of A_f is bounded by k there exist polynomials $g_i, h_i \in \mathbb{F}\langle X \rangle, i \in [k]$, such that $f = g_1h_1 + \dots + g_kh_k$, where g_i and h_i have noncommutative arithmetic circuits of size $\text{poly}(|C|, n, k)$ satisfying the above conditions.*

Proof of Claim 4.3.

For a homogeneous degree d_1 polynomial g let \bar{g} denote its coefficient column vector whose rows are indexed by degree d_1 monomials exactly as the rows of A_f . Similarly, for a homogeneous degree d_2 polynomial h there is a coefficient row vector \bar{h} with columns indexed by degree d_2 monomials (as the columns of A_f).

²The logspace counting class C=L captures the complexity of matrix rank over rationals [ABO99].

Observe that if f can be decomposed as the sum of products $f = g_1 h_1 + \dots + g_k h_k$, where each g_i is degree d_1 homogeneous and each h_i is degree d_2 homogeneous then the matrix A_f can be decomposed into a sum of k rank-one matrices:

$$A_f = \bar{g}_1 \bar{h}_1^T + \dots + \bar{g}_k \bar{h}_k^T.$$

It follows that the rank of A_f is bounded by k . Conversely, if the rank of A_f is k then A_f can be written as the sum of k rank 1 matrices. Since each rank one matrix is of the form $\bar{g} \bar{h}^T$, we obtain an expression as above which implies the decomposition of f as $g_1 h_1 + \dots + g_k h_k$.

Now, if the rank of A_f is k then there are degree d_1 monomials m_1, \dots, m_k and degree d_2 monomials m'_1, \dots, m'_k such that the $k \times k$ minor of A_f corresponding to these rows and columns is an invertible matrix K . W.l.o.g, we can write the $p \times q$ matrix A_f as

$$\begin{pmatrix} K & \Delta_2 \\ \Delta_1 & J \end{pmatrix}$$

for suitable matrices Δ_1, Δ_2, J . Moreover, since A_f is rank k , we can row-reduce to obtain

$$\begin{pmatrix} I & O \\ -\Delta_1 K^{-1} & I \end{pmatrix} \begin{pmatrix} K & \Delta_2 \\ \Delta_1 & J \end{pmatrix} = \begin{pmatrix} K & \Delta_2 \\ O & O \end{pmatrix}$$

and column reduce to obtain

$$\begin{pmatrix} I & O \\ -\Delta_1 K^{-1} & I \end{pmatrix} \begin{pmatrix} K & \Delta_2 \\ \Delta_1 & J \end{pmatrix} \begin{pmatrix} I & -K^{-1} \Delta_2 \\ O & I \end{pmatrix} = \begin{pmatrix} K & O \\ O & O \end{pmatrix}$$

It is easy to verify that this yields the following factorization for $A_f = UI_k V$

$$A_f = \begin{pmatrix} K & O \\ \Delta_1 & I \end{pmatrix} \begin{pmatrix} I & O \\ O & O \end{pmatrix} \begin{pmatrix} I & K^{-1} \Delta_2 \\ O & I \end{pmatrix}$$

Since we can write I_k as the sum $e_1 e_1^T + \dots + e_k e_k^T$ for standard basis vectors of suitable dimensions, we can express A_f as the sum of k rank-one matrices $(U e_i)(e_i^T V)$. We observe that the column vector $U e_i$ is the i^{th} column of matrix U , which is identical to the i^{th} column of matrix A_f . Therefore, the polynomial represented by this vector corresponds to the partial derivative of f w.r.t the monomial m_i , which can be computed by a circuit of desired size. Similarly, any row vector $e_i^T V$ is the i^{th} row of matrix V , which is identical to the i^{th} row of matrix A_f , scaled by the transformation K^{-1} . Moreover, the entries of K^{-1} are efficiently computable, and have size bounded by polynomial in the input size for the fields of our interest. Therefore, the polynomial represented by this vector corresponds to a linear combination of the partial derivatives of f w.r.t the monomials m'_1, \dots, m'_k , which can be computed by a circuit of desired size. Therefore, there exist polynomials g_i, h_i for $i \in [k]$ which satisfy the conditions of the second part of the claim. \square

Proof of Theorem 4.2.

Part (a). We first show that SOP is in MA. Given as input polynomial $f \in \mathbb{F}\langle X \rangle$ by an arithmetic circuit, the MA protocol works as follows. Merlin sends the description of arithmetic circuits for the polynomials $g_i, h_i, 1 \leq i \leq k$. By the second part of Claim 4.3, this message is polynomial sized as the polynomials g_i and h_i have circuits of size $\text{poly}(|C|, n, k)$.

Arthur verifies that $f = g_1h_1 + g_2h_2 + \dots + g_kh_k$ using a randomized noncommutative PIT algorithm with suitable success probability. This establishes the MA upper bound.

Next, we show that the complement problem $\overline{\text{SOP}}$ is in NP. I.e. given as input a polynomial $f \in \mathbb{F}\langle X \rangle$ such that f is not in SOP, we show that there is a $\text{poly}(|C|, n, k)$ -sized proof, verifiable in polynomial time, that f cannot be written as $f = g_1h_1 + \dots + g_kh_k$. It suffices to exhibit a short proof of the fact that the rank of matrix A_f is at least $k+1$. This can be done by listing $k+1$ degree d_1 monomials m_1, \dots, m_{k+1} and degree d_2 monomials m'_1, \dots, m'_{k+1} such that the $(k+1) \times (k+1)$ minor of A_f indexed by these monomials has full rank. This condition can be checked in polynomial time by verifying the determinant of the minor to be non-zero.

Part (b). When the polynomial is given as an ABP P , we sketch the simple polynomial-time algorithm for SOP.

Our goal is to compute a decomposition

$$f = g_1h_1 + \dots + g_kh_k \tag{3}$$

for minimum k , where $\deg(g_i) = d_1$ and $\deg(h_i) = d_2$. In order to compute this decomposition, we can suitably adapt the multiplicity automaton learning algorithm of Beimel et al [BB+00]. We can consider the input homogeneous degree- d polynomial $f \in \mathbb{F}\langle X \rangle$ as a function $f : X^d \rightarrow \mathbb{F}$, where $f(m)$ is the coefficient of m in the polynomial f . The learning algorithm works in Angluin's model. More precisely, when given black-box access to the function $f(m)$ for monomial queries m , it uses equivalence queries with counterexamples and learns the minimum size ABP computing f in polynomial time. Given a hypothesis ABP P' for f , we can simulate the equivalence queries and finding a counterexample by using the Raz-Shpilka PIT algorithm [RS05] on $P - P'$. The minimized ABP that is finally output by the learning algorithm will have the optimal number of nodes at each layer. In particular, layer d_1 will have the minimum number of nodes k which will give the decomposition in Equation 3. Furthermore, the ABPs for the polynomials g_i and h_i can also be immediately obtained from the minimum size ABP.

Part (c). When the polynomial $f \in \mathbb{F}\langle X \rangle$ is given in sparse representation, we can explicitly write down the partial derivative matrix A_f and check its rank. Moreover, we can even compute the decomposition by computing a rank-one decomposition of the partial derivative matrix. The equivalence arises from the fact that the rank of a matrix A is equal to the minimum number of summands in the decomposition of the noncommutative polynomial $\sum_{i,j \in [n]} a_{ij}x_i x_j$ as a sum of products of homogeneous linear forms. □

An NP-hard decomposition problem

We now briefly discuss a generalization of SOP. Given a polynomial $f \in \mathbb{F}\langle X \rangle$ as input along with k in unary, can we decompose it as a k -sum of products of *three* homogeneous polynomials:

$$f = a_1b_1c_1 + a_2b_2c_2 + \dots + a_kb_kc_k,$$

where each a_i is degree d_1 , each b_i is degree d_2 , and each c_i is degree d_3 ?

It turns out that even in the simplest case when f is a cubic polynomial and the a_i, b_i, c_i are all homogeneous linear forms, this problem is NP-hard. The tensor rank problem: given a 3-dimensional tensor $A_{ijk}, 1 \leq i, j, k \leq n$ checking if the tensor rank of A is bounded by k , which is known to be NP-hard [Has90] is easily shown to be polynomial-time reducible to this decomposition problem.

Indeed, we can encode a three-dimensional tensor A_{ijk} as a homogeneous cubic noncommutative polynomial $f = \sum_{i,j,k \in [n]} A_{ijk} x_i y_j z_k$, such that any summand in the decomposition, which is product of three homogeneous linear forms, corresponds to a rank-one tensor. This allows us to test whether a tensor can be decomposed into at most k rank-one tensors, which is equivalent to testing whether the rank of the tensor is at most k .

5 Concluding Remarks and Open Problems

The main open problem is the complexity of noncommutative polynomial factorization in the general case. Even when the input polynomial $f \in \mathbb{F}\langle X \rangle$ is given in sparse representation we do not have an efficient algorithm nor any nontrivial complexity-theoretic upper bound. Although polynomials in $\mathbb{F}\langle X \rangle$ do not have unique factorization, there is interesting structure to the factorizations [Co85, Co] which can perhaps be exploited to obtain efficient algorithms.

In the case of irreducibility testing of polynomials in $\mathbb{F}\langle X \rangle$ we have the following observation that contrasts it with commutative polynomials. Let \mathbb{F} be a fixed finite field. We note that checking if $f \in \mathbb{F}\langle X \rangle$ given in sparse representation is *irreducible* is in coNP. To see this, suppose f is s -sparse of degree D . If f is reducible and $f = gh$ is any factorization then each monomial in g or h is either a prefix or a suffix of some monomial of f . Hence, both g and h are sD -sparse polynomials. An NP machine can guess g and h (since coefficients are constant-sized) and we can verify if $f = gh$ in deterministic polynomial time.

On the other hand, it is an interesting contrast to note that given an s -sparse polynomial f in the *commutative* ring $\mathbb{F}[x_1, x_2, \dots, x_n]$ we do not know if checking irreducibility is in coNP. However, checking irreducibility is known to be in RP (randomized polynomial time with one sided-error) as a consequence of the Hilbert irreducibility criterion [Ka89]. If the polynomial f is irreducible, then if we assign random values from a suitably large extension field of \mathbb{F} to variables x_2, \dots, x_n (say, $x_i \leftarrow r_i$) the resulting univariate polynomial $f(x_1, r_2, \dots, r_n)$ is irreducible with high probability.

Another interesting open problem that seems closely related to noncommutative sparse polynomial factorization is the problem of finite language factorization [SY00]. Given as input a finite list of words $L = \{w_1, w_2, \dots, w_s\}$ over the alphabet X the problem is to check if we can factorize L as $L = L_1 L_2$, where L_1 and L_2 are finite sets of words over X and $L_1 L_2$ consists of all strings uv for $u \in L_1$ and $v \in L_2$. This problem can be seen as noncommutative sparse polynomial factorization problem where the coefficients come from the *Boolean ring* $\{0, 1\}$. No efficient algorithm is known for this problem in general, neither is any nontrivial complexity bound known for it [SY00]. On the other hand, analogous to factorization in $\mathbb{F}\langle X \rangle$, we can solve it efficiently when L is *homogeneous* (i.e. all words in L are of the same length). Factorizing L as $L_1 L_2$, where L_1 and L_2 are variable-disjoint can also be efficiently done by adapting our approach from Section 2. It would be interesting if we can relate language factorization to sparse polynomial factorization in $\mathbb{F}\langle X \rangle$ for a field \mathbb{F} . Is one efficiently reducible to the other?

References

- [ABO99] E. ALLENDER, R. BEALS, M. OGIHARA, The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8:2, 99-126, 1999.
- [Ar14] ARNAB BHATTACHARYA, Polynomial Decompositions in Polynomial Time. *Proceedings of the ESA Conference*, pages 125-136, 2014.
- [AMS10] V. ARVIND, P. MUKHOPADHYAY, S. SRINIVASAN, New Results on Noncommutative and Commutative Polynomial Identity Testing. *Computational Complexity*, 19(4):521-558 (2010).
- [BB+00] A. BEIMEL, F. BERGADANO, N.Q H. BSHOUTY, E. KUSHILEVITZ, S. VARRICCHIO, Learning functions represented as multiplicity automata. *Journal of the ACM*, 47(3): 506-530 (2000).
- [BW05] A. BOGDANOV, H. WEE More on Noncommutative Polynomial Identity Testing *In Proc. of 20th Annual Conference on Computational Complexity*, 92-99, 2005.
- [Ca10] F. CARUSO, Factorization of Noncommutative Polynomials, *CoRR abs/1002.3180*, 2010.
- [Co85] P. M. COHN, *Free rings and their relations*, Academic Press, London Mathematical Society Monograph No. 19, 1985.
- [Co] P. M. COHN, Noncommutative unique factorization domains, *Transactions of the American Math. Society*, 313-331: 109 (1963).
- [GG] J. GATHEN, J. GERHARD, *Modern Computer Algebra* 2nd edition, *Cambridge University Press*.
- [Has90] JOHAN HASTAD, Tensor rank is NP-complete. *Journal of Algorithms*, 11(4):644-654,1990.
- [Ka89] E. KALTOFEN, Factorization of Polynomials given by straight-line programs, *Randomness in Computation*, vol. 5 of *Advances in Computing Research*, 375-412, 1989.
- [KT90] E. KALTOFEN AND B. TRAGER, Computing with polynomials given by black-boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symbolic Comput.*, 9(3):301-320, 1990.
- [KSS14] S. KOPPARTY, S. SARAF, A. SHPILKA, Equivalence of Polynomial Identity Testing and Deterministic Multivariate Polynomial, *Electronic Colloquium on Computational Complexity (ECCC) 21:1*, 2014.
- [N91] N. NISAN, Lower bounds for noncommutative computation *In Proc. of 23rd ACM Sym. on Theory of Computing*, 410-418, 1991.
- [RS05] R. RAZ, A. SHPILKA, Deterministic polynomial identity testing in non commutative models, *Computational Complexity*,14(1):1-19, 2005.
- [SV10] A. SHPILKA, I. VOLKOVICH, On the Relation between Polynomial Identity Testing and Finding Variable Disjoint Factors, *ICALP* , 2010.

[SY00] ARTO SALOMAA AND SHENG YU, On the decomposition of finite languages, *In Proc. Developments in Language Theory, Foundations, Applications, and Perspectives*, editors Grzegorz Rozenberg and Wolfgang Thomas, *World Scientific*, 22-31, 2000.