Variety Membership Testing in Algebraic Complexity Theory

A dissertation submitted towards the degree Doctor of Natural Sciences of the Faculty of Mathematics and Computer Science of Saarland University

submitted by

Anurag Pandey

Saarbrücken / 2021

Day of Colloquium:	17 June 2021
Dean of the Faculty:	Prof. Dr. Thomas Schuster
Chair of the Committee:	Prof. Dr. Frank-Olaf Schreyer
Reporters	
First reviewer:	Prof. Dr. Markus Bläser
Second reviewer:	Dr. Christian Ikenmeyer
Third reviewer:	Prof. Dr. Meena Mahajan
Academic Assistant:	Dr. Anand Kumar Narayanan

Abstract

Abstract In this thesis, we study some of the central problems in algebraic complexity theory through the lens of the variety membership testing problem.

In the first part, we investigate whether separations between algebraic complexity classes can be phrased as instances of the variety membership testing problem. For this, we compare some complexity classes with their closures. We show that monotone commutative single-(source, sink) ABPs are closed. Further, we prove that multi-(source, sink) ABPs are not closed in both the monotone commutative and the noncommutative settings. However, the corresponding complexity classes are closed in all these settings. Next, we observe a separation between the complexity classes VQP and $\overline{\mathsf{VNP}}$.

In the second part, we cover the blackbox polynomial identity testing (PIT) problem, and the rank computation problem of symbolic matrices, both phrasable as instances of the variety membership testing problem. For the blackbox PIT, we give a randomized polynomial time algorithm that uses the number of random bits that matches the information-theoretic lower bound, differing from it only in the lower order terms. For the rank computation problem, we give a deterministic polynomial time approximation scheme (PTAS) when the degrees of the entries of the matrices are bounded by a constant.

Finally, we show NP-hardness of two problems on 3-tensors, both of which are instances of the variety membership testing problem. The first problem is the orbit closure containment problem for the action of $\mathrm{GL}_k \times \mathrm{GL}_m \times \mathrm{GL}_n$ on 3-tensors, while the second problem is to decide whether the slice rank of a given 3-tensor is at most r.

Zusammenfassung In dieser Arbeit untersuchen wir einige der zentralen Probleme der algebraischen Komplexitätstheorie, die sich als eine Instanz des Problems des Enthaltenseins in einer Varietät formulieren lassen.

Im ersten Teil untersuchen wir, ob das Trennen von algebraischen Komplexitätsklassen als Instanzen des Problems des Enthaltenseins in Varietäten formuliert werden kann. Dazu vergleichen wir einige Komplexitätsklassen mit ihren Abschlüssen. Wir zeigen, dass monotone und kommutative Einzel-(Quelle, Senke)-ABPs abgeschlossen sind. Weiterhin beweisen wir, dass Multi-(Quelle, Senke)-ABPs sowohl in der monotonen und kommutativen als auch in der nicht-kommutativen Variante nicht abgeschlossen sind. Die entsprechenden Komplexitätsklassen sind jedoch in allen Varianten abgeschlossen. Schließlich zeigen wir, dass die Komplexitätsklassen VQP und $\overline{\mathsf{VNP}}$ verschieden sind.

Im zweiten Teil behandeln wir das Problem des Blackbox-Polynomial-Identity-Testing (PIT) und das Problem der Rangberechnung von symbolische Matrizen, die beide als eine Frage des Enthaltensein in einer Varietät formuliert werden können. Für das Blackbox-PIT geben wir einen randomisierten Polynomzeitalgorithmus an, der eine Anzahl von Zufallsbits verwendet, die mit der informationstheoretischen Untergrenze übereinstimmt bis auf Terme geringerer Ordnung. Für das Rangberechnungsproblem geben wir ein deterministisches Polynomialzeit-Approximationsschema (PTAS) an, wenn die Grade der Einträge der Matrizen durch eine Konstante begrenzt sind.

Schließlich zeigen wir die NP-Härte zweier Probleme auf 3-Tensoren. Beide Probleme können wieder als das Enthaltensein in einer Varietät formuliert werden. Das erste Problem ist die Frage, ob ein 3-Tensor im Abschluss der $\mathrm{GL}_k \times \mathrm{GL}_m \times \mathrm{GL}_n$-Bahn eines anderen Tensors liegt, während das zweite Problem darin besteht, zu entscheiden, ob der Slice-Rang eines bestimmten 3-Tensors höchstens r beträgt.

Acknowledgments

As a mango tree found itself bearing its first set of fruits, it got elated, It decided to acknowledge everything and everyone that made it possible. But what and whom is it supposed to acknowledge? Should it acknowledge the farmer who always made sure it got the necessary ambience and the crucial nutrients to the best of his knowledge and perception? Or should it acknowledge the farmer's little daughter who wanted a mango tree in their farm? Or rather her friend who introduced her to mangoes? Or should it thank everything and everyone responsible for the farmer and his family's health and well-being making sure that they successfully managed to care for it? Or the monkey who, after eating a mango, threw the seed in their garden which the farmer finally planted? Or should it be grateful to the numerous plants and animals who shed their wastes contributing to the richness of the soil? Or should it thank the weather cycles and the responsible planetary forces causing plants to shed leaves? Or the animals' biological system and the responsible life forces making them shed their wastes regularly? Or should it thank all the birds feeding on the worms that were eating it when it was still a sapling, or the ecological system leading the birds to do so? Or should it acknowledge the sun, or its source? Or should it thank the passerby who ended up urinating around its roots when the little mango plant was about to die because of lack of water? Or should it thank the storm which blew away the unnoticed weeds that had been growing around it and had been taking up all its nutrients?

Overwhelmed by all this, it could not help but feel grateful towards every single atom and every single living being that have ever existed, and all the forces causing and sustaining them. It kept quiet and simply lived on, continuing to bear new flowers every spring and to bear new fruits every summer.

Contents

CHAPTER 1

Introduction

Algebraic complexity theory mainly deals with algorithmic and complexity theoretic problems related to computing polynomials using various models of computations. Algebraic geometry, on the other hand, primarily studies the zeros of multivariate polynomials. The central object in algebraic geometry is an *algebraic variety*, which is the defined as the set of common zeros of a set of multivariate polynomials [1]. Naturally, one of the central and ubiquitous problems in algebraic geometry is to determine if a point of interest belongs to a variety of interest. This is what we refer to as the *variety membership testing* problem.

Since algebraic complexity theory is about computing polynomials, and algebraic geometry asserts that polynomials can be understood by understanding their zeros, it seems plausible that the understanding of zero sets of polynomials or *varieties* is crucial in understanding the algebraic complexity of polynomials.

In fact, algebraic geometry turns out to be intimately relevant to algebraic complexity theory, as many of the problems, both algorithmic problems and the problems about showing lower bounds, can be reduced to instances of variety membership testing problem, whereas for many other problems, it is of interest to find out if they can indeed be seen as instances of variety membership testing problem.

In this thesis, we emphasise on this viewpoint of looking at various central problems in algebraic complexity theory as instances of variety membership testing problem. In the rest of the chapter, we take some of the central problems in algebraic complexity theory and make this connection more transparent.

1.1 Separations among complexity classes and lower bounds

Let us start with the most central problems in algebraic complexity theory. They are the problems asking whether two algebraic complexity classes of interest, (for instance, VQP and VNP, discussed in Chapter 3) are the same. Since the algebraic complexity classes are essentially sets of families of polynomials and polynomials can be viewed as a vector of its coefficients in an appropriately defined ambient space, algebraic complexity classes can be viewed as families of set of points in a well-defined affine space. Hence, towards our question, one may ask whether the algebraic complexity classes themselves can be described as zero-sets of multivariate polynomials. In other words, does the question

[1] In many places in the literature, this is referred to as an *algebraic set*. There, an *irreducible* algebraic set is called a variety. Also, in the modern language of algebraic geometry, the definition of variety is more general. We will stick to the definition given above.

of whether a polynomial family belongs to a complexity class becomes a problem of whether a point lies in a variety. And, by extension, the question of whether the two complexity classes are equal, becomes a question of whether two seemingly different varieties, because of different definitions, are actually the same variety. This indeed happens for some models of computation. For instance, this happens in the case of noncommutative single-(source, sink) algebraic branching programs, which we discuss in Chapter 2.

One may ponder why should such a translation of the questions of algebraic complexity classes to a language of algebraic geometry be sought. The first answer lies in the definition of varieties that they are nothing but common zero sets of sets of polynomials. So, if a set turns out to be a variety, then the next question one may ask is what exactly are the corresponding polynomials whose common zero set this variety is. Even if we do not manage to find all the defining polynomials corresponding to a variety, every non-trivial polynomial can significantly help in having a tighter grip on the variety, and hence on the corresponding complexity class. For instance, every member of the complexity class must be a zero of every such polynomial, and hence being a zero of these defining polynomials serves as a necessary criterion for the membership in the complexity class. This might already be sufficient for proving non-trivial lower bounds and separations between complexity classes.

This is not merely speculative, since we do know separations between algebraic complexity classes. We will see one such separation in Chapter 3, where we observe that the proof that the complexity class VQP is not contained in the complexity class VNP goes along this line and hence also gives a separation between the complexity classes VQP and $\overline{\mathsf{VNP}}$.

The second reason why it is interesting to unleash if a set, in particular, an algebraic complexity class, is indeed a variety is because of the existence of a vast amount of research on geometric properties of a variety which can be used for the purposes of setting two varieties apart from each other, in particular when the varieties are given as group orbit closures (discussed in Chapter 6 and 7). Here also, using tools from algebraic geometry and representation theory, there are explicit lower bounds have been obtained. This is precisely the goal of the Geometric Complexity Theory program initiated by Mulumuley and Sohoni [142]. Within this framework, lower bounds against matrix multiplication problem [51, 103] and lower bounds against weaker models of computation, for instance, sums of powers of linear forms and product of linear forms [64, 110, 2] have been obtained. In fact, it is known that most known lower bounds in algebraic complexity theory can unified using this viewpoint [93].

Thus, it makes sense to ask whether a set defined complexity theoretically is also algebraically closed, that is, geometrically well-behaved. In this thesis, in Chapter 2, we will see a geometric method itself, that is, a method based on the dimension of a variety, helps also in deciding whether certain models of computations are indeed closed, that is whether the corresponding set is indeed a variety.

1.2 Algorithmic problems in algebraic complexity theory

The other question that we may ask is about the connection between the algorithmic problems of interest in algebraic complexity and the instances of the algorithmic version of the variety membership problem. We explore this connection as well in this thesis. In many cases, it turns out that the algorithmic problems in algebraic complexity theory are nothing but special instances of the variety membership testing problem. We see this in this thesis in part 2 and part 3 of the thesis. In some cases, this is obvious (polynomial identity testing problem) while in others it requires more work (matrix rank, slice rank and minrank). In the other direction, some special instances of variety membership testing problem, which were already interesting from the perspective of algebraic geometry has recently turned out to be quite important from the perspective of algebraic complexity theory. We explore this in Chapter 6, the chapter about orbit closure containment problem.

1.2.1 Polynomial identity testing

The polynomial identity testing problem (PIT) is one of the central algorithmic problems in complexity theory, capturing problems like perfect matching in graphs [54, 133, 144] and primality testing [4, 5, 6], while it is also known to be intimately connected to proving complexity theoretic lower bounds [117, 3]. The goal in PIT is, for a given multivariate polynomial in some implicit form, to decide if the polynomial is an identically zero polynomial. An identically zero polynomial is a polynomial whose coefficients in the standard monomial representation are all zero. When we view an n-variate, degree d, polynomial as an $N := \binom{n+d}{d}$ dimensional vector, that is, we view the polynomial as a point in $\mathbb{C}^N$, then the PIT problem is asking if the "given" point is the origin. Or in other words, it asks, whether the given point lies in the *zero variety*, comprising of the single point, the origin. Thus, PIT is also an instance of variety membership testing problem. We discuss an algorithm for polynomial identity testing in Chapter 4.

1.2.2 Rank of symbolic matrices

Let us say, we are given an $n \times n$ matrix whose entries are multivariate polynomials, and a natural number r, and the problem is to determine whether the commutative rank (over the function field) of the matrix $\leq r$. This problem is also fundamental in algebraic complexity theory, as it subsumes polynomial identity testing. This problem can again be seen as an instance of variety membership testing. This is because of the characterization of rank via vanishing of minors. That is, it is known that a matrix has rank $\leq r$ if and only if all its $(r+1) \times (r+1)$ minors are zero. In other words, the set of matrices with rank $\leq r$ is a variety, and the defining polynomials are the $(r+1) \times (r+1)$ minors. Thus, the problem becomes to find the biggest number r such that the given matrix belongs to the variety of matrices with rank $\leq r$. We discuss an algorithm for the rank computation of symbolic matrices in Chapter 5.

1.2.3 Orbit closure containment problem and the null cone problem

For a group G, acting[2] on a vector space V, the orbit of a vector $v \in V$, denoted as Gv, is defined to be the set $\{gv \mid g \in G\}$. That is, the orbit Gv is the set of points that v gets mapped to, on the action of G. The group problem that has received the widest attention in computer science is the orbit containment problem. This asks, given a group G acting on a vector space V, and two elements $u, v \in V$, to decide if $u \in Gv$. Thus it asks if a vector is in the orbit of another vector. This problem is quite general and captures many problems, for instance the graph isomorphism problem and the module isomorphism problem.

From the perspective of topology, it is more natural to consider *orbit closures* instead. For a group G acting on a vector space V, the orbit closure of $v \in V$, denoted as $\overline{Gv}$, is defined to be the smallest closed subset of V which contains Gv. In the standard Euclidean topology, this translates to $\overline{Gv}$ being the smallest superset of Gv which contains the limit points of all convergent sequences comprising of elements of Gv. In the Zariski topology, this translates to $\overline{Gv}$ being the smallest superset of Gv which is algebraically closed, that is, it contains all the common zeros of the set of polynomials that vanish on all the elements of Gv. In most of the cases of interest, in particular, when the underlying field is $\mathbb{C}$, the definitions of $\overline{Gv}$ obtained by considering the above two topologies, that is, the Euclidean (or analytic) closure and the Zariski (or algebraic) closure coincide [3]. Thus we can ask the following weakening of the orbit containment problem, that is, *the orbit closure containment problem*. This asks, for a group G acting on a vector space V, and two elements $u, v \in V$, decide if $u \in \overline{Gv}$. This problem again is quite general, and has appeared centrally in the algorithmic and the complexity theoretic problems related to algebra and combinatorial optimization, by capturing problems like the border rank of tensors and the *null cone problem*.

The null cone problem is a special case of the orbit closure containment problem where vector u is always the origin, the 0 vector. That, is, we ask the following: for a group G acting on a vector space V, and $v \in V$, decide if $0 \in \overline{Gv}$.

For an example set up of the null cone problem, let us think of a tensor $t \in \mathbb{F}^{n \times n \times m}$ as a set of m matrices $A_1, \ldots, A_m$ of size $n \times n$, stacked up on top of each other (also called slices). The group $\Gamma_n \coloneqq \mathsf{SL}_n \times \mathsf{SL}_n$ acts on t by simultaneously multiplying each of the matrices from the left and the right. King [120] showed that the *noncommutative* rank of the matrix space given by $A_1, \ldots, A_m$ is maximal iff $0 \in \overline{\Gamma_n t}$. (All such tensors t are said to lie in the *null cone*.)

Orbit closure containment problems have played a central role in algebraic complexity theory in the recent years, primarily due to the role of border rank of tensors in several advancements in the fast matrix multiplication algorithms [26] and the formulation of the famous permanent versus determinant problem as an orbit closure containment problem in the geometric complexity program initiated by Mulmuley and Sohoni [142]. Very recently, the null cone problem has proved to be useful in giving polynomial time

[2] When we say a group G acts on the ambient space S, we have a mapping $\cdot : G \times S \to S$ that satisfies the axioms $1 \cdot s = s$ and $(gh) \cdot s = g \cdot (h \cdot s)$ for all $s \in S$ and $g, h \in G$. Here gh is the group operation.

[3] Unless stated otherwise, we assume the underlying field to be $\mathbb{C}$ in this paper

algorithms for special cases of the polynomial identity testing problem and several non-convex optimization problems [48, 82, 79, 47, 50, 9, 81, 49, 80] .

1.2.4 Border Rank and Slice rank of tensors

In this section, we see two notions of tensor ranks which give rise to a variety membership testing problem.

The first is the well known notion of the *border rank* of tensors. The border rank of tensor is, in fact, an instance of orbit closure containment problem, and hence an instance of variety membership testing problem too. To see the border rank problem as an orbit closure containment problem, let GL_n denote the group of all invertible $n \times n$ matrices. GL_n acts on $\mathbb{F}^n$ by the usual matrix-vector multiplication. $G_n := \mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ acts on rank-one tensors $u \otimes v \otimes w$ by $(A, B, C) \cdot u \otimes v \otimes w = Au \otimes Bv \otimes Cw$ and on arbitrary tensors by linear continuation. The orbit of a tensor t under G_n is the set $G_n t := \{g \cdot t \mid g \in G_n\}$ and its orbit closure is the closure $\overline{G_n t}$ in the Zariski topology. It is well known that the set of all tensors of border rank $\leq r$ can be written with the help of an orbit closure [46], namely $\overline{G_r e_r}$ where e_r is the so-called unit tensor in $\mathbb{F}^{r \times r \times r}$: A tensor $t \in \mathbb{F}^{n \times n \times n}$ has border rank $\leq r$ iff $\tilde{t} \in \overline{G_r e_r}$, where $\tilde{t}$ is an embedding of t into the larger ambient space $\mathbb{F}^{r \times r \times r}$.

The second notion, that is the notion of slice rank, is quite recent and was first used implicitly by Croot, Lev, and Pach and later explicitly by Tao [174]. The term "slice rank", however, was first used by Blasiak et al. [35] who used the term for the notion that Tao introduced. The methods based on slice rank have been very useful in the breakthrough works for several combinatorial problems like the sunflowers free sets, the tri-colored and multi-colored sum-free sets, the capsets and the progression-free problem, multiplicative matching in nonabelian groups, and also in proving barrier results against the group-theoretic approach to fast matrix multiplication.

We now describe the notion of slice rank and then the corresponding computational problem. For this, we consider the space $V_1 \otimes V_2 \otimes V_3$. It can also be written as $\bigotimes_{i=1}^{3} V_i$, and is generated by the decomposable (also called rank-one) tensors $v_1 \otimes v_2 \otimes v_3$, where $v_i \in V_i$. The usual tensor rank is the minimum number of decomposable tensors that is needed to write a given tensor as a sum of decomposable tensors. The slice rank is defined in a similar manner, however, the basic building blocks are not decomposable tensors but tensors that can be decomposed into a matrix and a single vector. More formally, consider the smaller tensor products $\bigotimes_{1 \leq i \leq 3 : i \neq j} V_i$ and the j-th tensor products $\otimes_j : V_j \times \bigotimes_{1 \leq i \leq 3 : i \neq j} V_i \to \bigotimes_{i=1}^{3} V_i$ with its natural definition. Now the rank one functions are the elements of the form $v_j \otimes_j v_{\hat{j}}$ for some $v_j \in V_j$ and $v_{\hat{j}} \in \bigotimes_{1 \leq i \leq 3 : i \neq j} V_i$. The slice rank (or srk for short) of a tensor $T \in \bigotimes_{i=1}^{3} V_i$ is the smallest nonnegative integer r such that T can be expressed as a linear combination of r rank one functions.

The algorithmic problem corresponding to the slice rank problem is the following: given $T \in \mathbb{F}^n \otimes \mathbb{F}^n \otimes \mathbb{F}^n$ and a number r, decide if $\mathrm{srk}(T) \leq r$.

It was shown by Tao and Sawin that the set of tensors with slice rank bounded by at most r is algebraically closed and hence is a variety. So, the slice rank problem is also an instance of variety membership testing problem. In Chapter 7, we show that it can be

phrased as a problem about containment in a union of orbit closures. We further show that this algorithmic problem is NP-hard as well.

PART I

Are algebraic complexity classes varieties?

This part is the result of close collaboration with Markus Bläser, Christian Ikenmeyer, Meena Mahajan and Nitin Saurabh. It is based on an article titled *Algebraic Branching Programs, Border Complexity, and Tangent Spaces* that appeared in the *Computational Complexity Conference*, 2020 [32].

CHAPTER 2

Border Complexity and Algebraic Branching Programs

In this chapter, we ask questions of the kind, "Can the membership problem of a polynomial family in an algebraic complexity class be formulated as instances of the variety membership problem?" This translates to asking if a complexity class, or more generally, a set, defined complexity theoretically can be described as a set of common solutions to some system of polynomial equations. In algebraic geometry, such sets are called algebraically closed. Thus, we ask whether the given set, defined complexity theoretically, is algebraically closed. If a complexity class, or more generally, any set defined complexity theoretically, is algebraically closed, then membership in that set is an instance of variety membership problem.

We explore this question in the setting of algebraic branching programs, that is, the complexity theoretically defined set is defined via algebraic branching programs. In particular, we explore it in the setting of noncommutative algebraic branching programs, monotone commutative algebraic branching programs and commutative algebraic branching programs.

Nisan [147] showed in 1991 that the width of a smallest noncommutative single-(source, sink) algebraic branching program (ABP) to compute a noncommutative polynomial is given by the ranks of specific matrices. This means that the set of noncommutative polynomials with ABP width complexity at most k is Zariski-closed, an important property in geometric complexity theory. It follows that approximations cannot help to reduce the required ABP width.

It was mentioned by Forbes [69] that this result would probably break when going from single-(source, sink) ABPs to trace ABPs. We prove that this is correct. Moreover, we study the commutative monotone setting and prove a result similar to Nisan, but concerning the analytic closure. We observe the same behavior here: The set of polynomials with ABP width complexity at most k is closed for single-(source, sink) ABPs and not closed for trace ABPs. The proofs reveal an intriguing connection between tangent spaces and the vector space of flows on the ABP.

2.1 Set-up and results

Algebraic branching programs (ABPs) are an elegant model of computation that is widely studied in algebraic complexity theory in its various incarnations (see e.g. [21, 177, 139, 141, 10, 14, 119, 124, 55, 76]) and is a focus of study in geometric complexity theory

[129, 85, 86]. In this chapter, we study the layered, homogeneous ABP (which we will simply refer to as ABP), since this is the most suitable incarnation of ABP to study from the viewpoint of variety membership testing and geometric complexity theory. An ABP is a layered directed graph with $d+1$ layers of vertices (edges only go from layers i to $i+1$) such that the first and last layer have exactly the same number of vertices. Each vertex in the first layer has exactly one so-called *corresponding* vertex in the last layer. One interesting classical case is when the first and last layer have exactly one vertex, which is usually studied in theoretical computer science. We call this the *single-(source, sink) model*. Among algebraic geometers working on ABPs it is common to not impose restrictions on the number of vertices in the first and last layer [129, 85, 130]. We call this the *trace model*. Every edge in an ABP is labeled by a homogeneous linear form (i.e., a linear polynomial without a constant term in the variables $x_1, \ldots, x_m$). If we denote by $\ell(e)$ the homogeneous linear form of edge e, then we say that the ABP computes $\sum_p \prod_{e \in p} \ell(e)$, where the sum is over all paths that start in the first layer and end in the last layer at the vertex corresponding to the start vertex.

The *width* of an ABP is the number of vertices in its largest layer. We denote by $\mathsf{w}(f)$ the minimal width required to compute f in the trace model and we call $\mathsf{w}(f)$ the *trace ABP width complexity* of f. We denote by $\mathsf{w}_1(f)$ the minimal width required to compute f in the single-(source, sink) model and we call $\mathsf{w}_1(f)$ the *single-(source, sink) ABP width complexity* of f.

The complexity class VBP is defined as the set of sequences of polynomials (f_m) for which the sequences $\mathsf{w}(f_m)$ and $\deg(f_m)$ are polynomially bounded, where $\deg(f)$ denotes the degree of the polynomial f. Let $\mathrm{per}_m := \sum_{\pi \in \mathfrak{S}_m} \prod_{i=1}^m x_{i,\pi(i)}$ be the permanent polynomial. Valiant's famous VBP $\neq$ VNP conjecture can concisely be stated as "The sequence of natural numbers $\big(\mathsf{w}(\mathrm{per}_m)\big)_m$ is not polynomially bounded." Alternatively, this is phrased with w_1 or other polynomially related complexity measures in a completely analogous way. In geometric complexity theory (GCT), one searches for lower bounds on algebraic complexity measures over $\mathbb{C}$ such as w and w_1 for explicit polynomials such as the permanent. All lower bounds methods in GCT and most lower bounds methods in algebraic complexity theory are *continuous*, which means that if f_ε is a curve of polynomials with $\lim_{\varepsilon \to 0} f_\varepsilon = f$ (coefficient-wise limit) and $\mathsf{w}(f_\varepsilon) \leq w$, then these methods cannot be used to prove $\mathsf{w}(f) > w$. This is usually phrased in terms of so-called *border complexity* (see e.g. [52, 129]): The *border trace ABP width complexity* $\underline{\mathsf{w}}(f)$ is the smallest w such that f can be approximated arbitrarily closely by polynomials f_ε with $\mathsf{w}(f_\varepsilon) \leq w$. Analogously, we define the *border single-(source, sink) ABP width complexity* $\underline{\mathsf{w}_1}(f)$ as the smallest w such that f can be approximated arbitrarily closely by polynomials f_ε with $\mathsf{w}_1(f_\varepsilon) \leq w$. We define $\overline{\mathrm{VBP}}$ as the set of sequences of polynomials such that $(\underline{\mathsf{w}}(f_m))$ is polynomially bounded. Clearly $\mathrm{VBP} \subseteq \overline{\mathrm{VBP}}$. Mulmuley and Sohoni [142, 143, 52] (see also [45, 43] for a related conjecture) conjectured a strengthening of Valiant's conjecture, namely that $\mathrm{VNP} \notin \overline{\mathrm{VBP}}$. In principle it could be that $\underline{\mathsf{w}}(f) < \mathsf{w}(f)$; the gap could even be superpolynomial, which would mean that $\mathrm{VBP} \subsetneq \overline{\mathrm{VBP}}$. If $\mathrm{VBP} = \overline{\mathrm{VBP}}$, then Valiant's conjecture is the same as the Mulmuley-Sohoni conjecture, which would mean that if $\mathrm{VBP} \neq \mathrm{VNP}$, then continuous lower bounds methods exist that show this separation.

Border complexity is an old area of study in algebraic geometry. In theoretical computer science it was introduced by Bini et al. [25], where [24] proves that in the study of fast matrix multiplication, the gap between complexity and border complexity is not too large. The study of the gap between complexity and border complexity of algebraic complexity measures in general has started recently [94, 41, 123] as an approach to understand if strong algebraic complexity lower bounds can be obtained from continuous methods.

In this chapter we study two very different settings of ABPs: The noncommutative and the monotone setting. To capture commutative, noncommutative, and monotone computation, let R be a graded semiring with homogeneous components R_d. In our case the settings for R_d are

- $R_d = \mathbb{F}[x_1, \dots, x_m]_d$ the set of homogeneous degree d polynomials in m variables over a field $\mathbb{F}$,
- $R_d = \mathbb{F}\langle x_1, \dots, x_m\rangle_d$ the set of homogeneous degree d polynomials in m noncommuting variables over a field $\mathbb{F}$,
- $R_d = \mathbb{R}_+[x_1, \dots, x_m]_d$ the set of homogeneous degree d polynomials in m variables with nonnegative coefficients.

As it is common in the theoretical computer science literature, we call elements of R_d *polynomials*. Note that $\mathbb{F}\langle x_1, \dots, x_m\rangle_d$ is naturally isomorphic to the d-th tensor power of $\mathbb{F}^m$, so *tensor* would be the better name. We hope that no confusion arises when in the later sections (where we use concepts from multilinear algebra) we use the tensor language. In the *homogeneous setting*, all ABP edge labels are in R_1, and hence the polynomial that is computed is in R_d. In the *affine setting*, all ABP edge labels are in $R_0 + R_1$, and hence the polynomial that is computed is in $\bigoplus_{d' \le d} R_{d'}$.

Noncommutative ABPs

Let $R_d = \mathbb{F}\langle x_1, \dots, x_m\rangle_d$ and consider the homogeneous setting. We write ncw instead of w and ncw_1 instead of w_1 to highlight that we are in the noncommutative setting. Nisan [147] proved:

Theorem 2.1. *Let M_i denote the $m^i \times m^{d-i}$ matrix[1] whose $((k_1, \dots, k_i), (k_{i+1}, \dots, k_d))$-th entry is the coefficient of the monomial $x_{k_1}x_{k_2}\cdots x_{k_d}$ in f. Then every single-(source, sink) ABP computing f has at least $\operatorname{rk}(M_i)$ many vertices in layer i. Conversely, there exists a single-(source, sink) ABP computing f with exactly $\operatorname{rk}(M_i)$ many vertices in layer i.*

Nisan used this formulation to prove strong complexity lower bounds for the noncommutative determinant and permanent. Forbes [70] remarked that Theorem 2.1 implies that for fixed w

$$\text{the set } \{f \mid \mathsf{ncw}_1(f) \le w\} \text{ is Zariski-closed}^2 \tag{2.1.1}$$

[1] We identify the rows (resp. columns) all degree i (resp. degree $d-i$) monomials in the variables $x_1, \dots, x_m$.

and hence that

$$\underline{\mathsf{ncw}_1}(f) = \mathsf{ncw}_1(f) \text{ for all } f. \tag{2.1.2}$$

Proving a similar result (even up to polynomial blowups) in the commutative setting would be spectacular: It would imply VBP = $\overline{\text{VBP}}$ and hence that Valiant's conjecture is the same as the Mulmuley-Sohoni conjecture. By a general principle, for all standard algebraic complexity measures, over $\mathbb{C}$ we have that the Zariski-closure of a set of polynomials of complexity at most w equals the Euclidean closure [145, S2.C].

Forbes mentioned that he believes that Nisan's proof cannot be lifted to the trace model. In this chapter we prove that Forbes is correct, by constructing a polynomial f_0 with

$$\underline{\mathsf{ncw}}(f_0) < \mathsf{ncw}(f_0). \tag{2.1.3}$$

The proof is given in Section 2.5–Section 2.8. It is a surprisingly subtle application of differential geometry (inspired by [109]) and interprets tangent spaces to certain varieties as vector spaces of flows on an ABP digraph.

The gap between $\underline{\mathsf{ncw}}(f)$ and $\mathsf{ncw}(f)$ can never be very large though:

$$\underline{\mathsf{ncw}}(f) \le \mathsf{ncw}(f) \le \mathsf{ncw}_1(f) \overset{(2.1.2)}{=} \underline{\mathsf{ncw}_1}(f) \overset{3}{\le} \big(\underline{\mathsf{ncw}}(f)\big)^2 \text{ for all } f. \tag{2.1.4}$$

It is worth noting that for our separating polynomial f_0, the gap is even less; $\underline{\mathsf{ncw}}(f_0) < \mathsf{ncw}(f_0) \le 2\underline{\mathsf{ncw}}(f_0)$. This is the first algebraic model of computation where complexity and border complexity differ, but their gap is known to be polynomially bounded! For most models of computation almost nothing is known about the gap between complexity and border complexity. For commutative width 2 affine ABPs the gap is even as large as between computable and non-computable [41]!

Monotone ABPs

Let $R_d = \mathbb{R}_+[x_1, \ldots, x_m]_d$ and consider the affine or homogeneous setting.

Since $\mathbb{R}$ is not algebraically closed, we switch to a more algebraic definition of approximation. Let $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ denote the ring of Laurent polynomials that are nonnegative for all sufficiently small $\varepsilon > 0$. Clearly, elements from $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ can have a pole at $\varepsilon = 0$ of

[2] We identify each m-variate homogeneous degree d polynomial with its coefficient vector. There is a standard topology on the vector space of these coefficient vectors that we call the Euclidean topology. The Zariski-closure of a set X of vectors is the smallest set of vectors that contains X and that is the common zero set of a finite set of polynomials in the coordinate variables, see e.g. [30, Ch. 4] for the commutative case.

[3] Given a trace ABP Γ computing f and a pair of corresponding start and end vertices, we can extract a single-(source, sink) ABP by deleting all other start and end vertices. If we do this for each pair of start and end vertices, and if we then idenfity all start vertices to a single start vertex, and also all end vertex to a single end vertex, then we obtain a single-(source, sink) ABP computing f. The width has grown by a factor of w, where w is the number of start vertices in Γ.

arbitrarily high order. We define $\underline{\mathsf{mw}}(f)$ to be the smallest w such that there exists a polynomial f' over the ring $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ such that

- there exists a width w ABP over $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ that computes f',
- no coefficient in f' contains an ε with negative exponent, and setting ε to 0 in f' yields f, i.e., $f'|_{\varepsilon=0} = f$.

We prove a result that is comparable to (2.1.2), but uses a very different proof technique:

$$\underline{\mathsf{mw}_1}(f) = \mathsf{mw}_1(f) \text{ for all } f. \tag{2.1.5}$$

In terms of complexity classes, this implies

$$\mathrm{MVBP} = \overline{\mathrm{MVBP}}^{\mathbb{R}}.$$

Our proof also works if the ABP is not layered and the labels are affine.

Intuitively, in this monotone setting, one would think that approximations do not help, because there cannot be cancellations. But quite surprisingly the same construction as in (2.1.3) can be used to find f_0 such that

$$\underline{\mathsf{mw}}(f_0) < \mathsf{mw}(f_0). \tag{2.1.6}$$

By the same reasoning as in (2.1.4), we obtain

$$\underline{\mathsf{mw}}(f) \le \mathsf{mw}(f) \le \left(\underline{\mathsf{mw}}(f)\right)^2 \text{ for all } f. \tag{2.1.7}$$

This gives a natural monotone model of computation where approximations speed up the computation. Again, the gap is polynomially bounded!

Structure of the chapter

In Section 2.4, we prove (2.1.5). Section 2.5 to Section 2.8 are dedicated to proving (2.1.3) and (2.1.6) via a new connection between tangent spaces and flow vector spaces.

2.2 Related work

Grenet [90] showed that $\mathsf{mw}(\mathrm{per}_m) \le \binom{m}{\lceil m/2 \rceil}$ by an explicit construction of a monotone single-(source, sink) ABP. Even though the construction is monotone, its size is optimal for $m = 3$ [13] (for 4 this is already unknown). The noncommutative version of this setting has been studied in [76]. [183] recently showed that the monotone circuit classes MVP and MVNP are different. We refer the reader to [183] and [172] and the references therein to get more information about monotone algebraic models of computation and their long history.

Hüttenhain and Lairez [109] present a method that can be used to show that a complexity measure and its border variant are not the same. They used it to prove that an explicit polynomial has border determinantal complexity 3, but higher determinantal complexity. We use their ideas as a starting point in Section 2.5 and the later sections.

2.3 Preliminaries

For a homogeneous degree d ABP Γ, we denote by V the set of vertices of Γ and by V^i the set of vertices in layer i, $1 \le i \le d+1$. We choose an explicit bijection between the sets V^1 and V^{d+1}, so that each vertex v in V^1 has exactly one *corresponding* vertex $\mathsf{corr}(v)$ in V^{d+1}. We denote by E^i the set of edges from V^i to V^{i+1}. Let E denote the union of all E^i.

There is a classical interpretation in terms of iterated matrix multiplication: Fix some arbitrary ordering of the vertices within each layer, such that the i-th vertex in V^1 corresponds to the i-th vertex in V^{d+1}. For $1 \le k \le d$ let M_k be the $|V^k| \times |V^{k+1}|$ matrix whose entry at position (i,j) in M_k is the label of the edge from the i-th vertex in V^k to the j-th vertex in V^{k+1}. Then Γ computes the trace

$$\sum_{\substack{1\le k_1\le |V^1|\\ 1\le k_2\le |V^2|\\ \vdots\\ 1\le k_d\le |V^d|}} (M_1)_{k_1,k_2}(M_2)_{k_2,k_3}\cdots(M_{d-1})_{k_{d-1},k_d}(M_d)_{k_d,k_1} = \mathsf{tr}\big(M_1M_2\cdots M_d\big). \tag{2.3.1}$$

Hence the name *trace model*. In the single-(source, sink) model, the trace is taken of a 1×1 matrix.

2.4 Monotone commutative single-(source, sink) ABPs are closed

For fixed $w \in \mathbb{N}$ we study

$$\text{the set } \{f \in \mathbb{R}_+[x_1,\dots,x_n]_d \mid \mathsf{mw}_1(f) \le w\}. \tag{2.4.1}$$

We first start with the simple observation that it is *not* Zariski-closed.

Proposition 2.4.2. *$\{f \in \mathbb{R}_+[x_1,\dots,x_n]_d \mid \mathsf{mw}_1(f) \le w\}$ is not Zariski-closed.*

Proof. Note that a homogeneous degree d single-(source, sink) width w ABP has $2w + w^2(d-2)$ many edges. The label on each edge is a linear form in n variables, so such an ABP is determined by $N := n(2w + w^2(d-2))$ many parameters. Let $F : \mathbb{C}^N \to \mathbb{C}[x_1,\dots,x_n]_d$ be the map that maps these parameters to the polynomial computed by the ABP. Every coordinate function of F is given by polynomials in N variables, so F is Zariski-continuous. Therefore

$$\overline{F((\mathbb{R}_+)^N)} = \overline{F(\overline{(\mathbb{R}_+)^N})} = \overline{F(\mathbb{C}^N)} \supseteq F(\mathbb{C}^N) \supsetneq F((\mathbb{R}_+)^N),$$

where the overline means the Zariski-closure. We remark that we did not use any special feature of the model of computation other than the fact that it is defined over $\mathbb{R}$. □

Recall that an ABP has $d+1$ layers of vertices. If an ABP has w_i many vertices in layer i, $1 \le i \le d$, we say the ABP has *format* $w = (w_1, w_2, \dots, w_d)$. We further recall that $w_{d+1} = w_1$. The following theorem is our closure result, which proves (2.1.5) and hence $\mathrm{MVBP} = \overline{\mathrm{MVBP}}^{\mathbb{R}}$.

Theorem 2.2. *Given a polynomial f over $\mathbb{R}$ and given a format w single-(source, sink) ABP with affine linear labels over $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ computing f_ε such that $\lim_{\varepsilon \to 0} f_\varepsilon = f$. Then there exists a format w monotone single-(source, sink) ABP that computes f.*

Proof. The proof is constructive and done by a two-step process. In the first step (which is fairly standard and works in many computational models) we move all the ε with negative exponents to edges adjacent to the source. The second step then uses the monotonicity.

Given Γ with affine linear labels over $\mathbb{R}[\varepsilon, \varepsilon^{-1}]_+$ we repeat the following process until all labels that contain an ε with a negative exponent are incident to the source vertex.

- Let e be an edge whose label contains ε with a negative exponent $-i < 0$. Moreover, assume that e is not incident to the source vertex. Let v be the start vertex of e. We rescale all edges outgoing of v with ε^i and we rescale all edges incoming to v with ε^{-i}.

If we always choose the edge with the highest layer, then it is easy to see that this process terminates. Since every path from the source to the sink that goes through a vertex v must use exactly one edge that goes into v and exactly one edge that comes out of v, throughout the process the value of Γ does not change. We finish this first phase by taking the highest negative power i among all labels of edges that are incident to the source and then rescale all these edges with ε^i. The resulting ABP Γ^i computes $\varepsilon^i f_\varepsilon$ and no label contains an ε with negative exponent. We now start phase 2 that transforms Γ^i into Γ^{i-1} that computes $\varepsilon^{i-1} f_\varepsilon$ without introducing negative exponents of ε. We repeat phase 2 until we reach Γ^0 in which we safely set ε to 0. Throughout the whole process we do not change the structure of the ABP and only rescale edge labels with powers of ε, which preserves monotonicity, so the proof is finished. It remains to show how Γ^i can be transformed into Γ^{i-1}. An edge whose label is divisible by ε is called an *ε-edge*. Consider the set Δ of vertices that are reachable from the source using only non ε-edges in Γ^i. The crucial insight is that since Γ^i is monotone and computes a polynomial that is divisible by ε, we know that every path in Γ^i from the source to the sink uses an ε-edge. Therefore Δ cannot contain the sink. We call a vertex in Δ whose outdegree is zero a *leaf* vertex. We repeat the following procedure until the source is the only leaf vertex:

- Let v be a non-source leaf vertex in Δ. We rescale all edges outgoing of v with ε^{-1} and we rescale all edges incoming to v with ε.

It is easy to see that this process terminates with the source being the only leaf vertex. Since the source is a leaf vertex, all edges incident to the source are ε-edges. We divide all their labels by ε to obtain Γ^{i-1}. □

2.5 Explicit construction of f_0 with higher complexity than border complexity

Fix some $d \geq 3$. In this section for every $m \geq 2$ we construct f_0 such that

$$m = \underline{\mathsf{ncw}}(f_0) < \mathsf{ncw}(f_0). \tag{2.5.1}$$

A completely analogous construction can be used to find f_0 with $\underline{\mathsf{w}}(f_0) < \mathsf{w}(f_0)$ and with $\underline{\mathsf{mw}}(f_0) < \mathsf{mw}(f_0)$. For the sake of simplicity, we carry out only the proof for (2.5.1).

Recall that in a format w ABP we have $w_{d+1} = w_1$. In each layer i we enumerate the vertices $V^i = \{v^i_1, \dots, v^i_{w_i}\}$ and we assume without loss of generality that the correspondence bijection between V^{d+1} and V^1 is the identity on the indices j of v^1_j, i.e., the jth vertex in V^1 corresponds to the jth vertex in V^{d+1}.

Fix an ABP format $w = (w_1, w_2, \dots, w_d)$ such that for all i, $w_i \geq 2$. Let Γ_{com} denote[4] the directed acyclic graph underlying an ABP of format w. An edge can be described by the triple (a, b, i), where $1 \leq i \leq d$, $1 \leq a \leq w_i$ and $1 \leq b \leq w_{i+1}$. Consider the following labeling of the edges with triple-indexed variables: $\ell_{\mathrm{com}}((a,b,i)) = x^{(i)}_{(a,b)}$. Define f_{com} to be the polynomial computed by Γ_{com} with edge labels ℓ_{com}.

We now construct f_0 as follows. Let d be odd (the case when d is even works analogously). Since in each layer we enumerated the vertices, we can now assign to each vertex its parity: even or odd. We call an edge between two even or two odd vertices *parity preserving*, while we call the other edges *parity changing*. Let us consider the following labeling of Γ_{com}: We set $\ell_0((a,b,i)) := x^{(i)}_{(a,b)}$ if (a,b,i) is parity changing (i.e., $a \not\equiv b \pmod 2$) and set the label $\ell_0((a,b,i)) := \varepsilon x^{(i)}_{(a,b)}$ otherwise, where $\varepsilon \in \mathbb{C}$. Let f'_ε be the polynomial computed by Γ_{com} with edge labels ℓ_0 and set $f_\varepsilon := \frac{1}{\varepsilon} f'_\varepsilon$ for $\varepsilon \neq 0$. We define $f_0 := \lim_{\varepsilon \to 0} f_\varepsilon$ (convergence follows from the construction, because d is odd). By definition, for all $\varepsilon \neq 0$, f_ε can be computed by a format w ABP. However, we will now prove that this property fails for the limit point f_0.

Theorem 2.3. *Fix an ABP format $w = (w_1, w_2, \dots, w_d)$ such that for all i, $w_i \geq 2$. Let f_0 be defined as above. Then, f_0* cannot *be computed by an ABP of format w.*

Note that for a format where $m = w_1 = \cdots = w_d$, this gives the f_0 which was desired in (2.5.1). (Note, however, that f_0 can be computed by an ABP of width $2m$ as follows. Construct an ABP Γ' that has, for each vertex $v \in \Gamma_{\mathrm{com}}$, vertices v' and v''. For each parity changing edge $(a,b) \in \Gamma_{\mathrm{com}}$ with label ℓ_0, add edges (a', b') and (a'', b'') with the same label ℓ_0. For each parity preserving edge $(a,b) \in \Gamma_{\mathrm{com}}$ with label ℓ_0, add edge (a', b'') with label $(\frac{1}{\varepsilon})\ell_0$. For corresponding vertices u, v in Γ_{com}, let v'' be the corresponding vertex for u' and v' be the corresponding vertex for u'' in Γ'. All paths between corresponding vertices in this ABP use exactly one parity preserving edge of Γ_{com}, and so this ABP computes f_0.)

The proof of Theorem 2.3 works as follows. Let $\mathsf{G} := \mathrm{GL}_{w_1 w_2} \times \mathrm{GL}_{w_2 w_3} \times \cdots \times \mathrm{GL}_{w_d w_{d+1}}$. Let $\mathsf{End} := \overline{G}$ denote its Euclidean closure, i.e., tuples of matrices in which one or several matrices can be singular.

We consider noncommutative homogeneous polynomials in the variables $x^{(i)}_{(a,b)}$ such that the i-th variable in each monomial is $x^{(i)}_{(a,b)}$ for some $a \in [w_i]$ and $b \in [w_{i+1}]$. The vector space of these polynomials is isomorphic to $W := \mathbb{C}^{w_1 w_2} \otimes \mathbb{C}^{w_2 w_3} \otimes \cdots \otimes \mathbb{C}^{w_d w_{d+1}}$ and the

[4] Here and in subsequent usages, the subscript "com" is to denote that the underlying graph is a *complete* layered graph. It should not be confused with being *commutative*.

monoid End (and thus also the group G) acts on this space in the canonical way. The set

$$\{f \in W \mid f \text{ can be computed by a format } w \text{ ABP}\}$$

is precisely the orbit $\mathsf{End} f_{\text{com}}$. We follow the overall proof strategy in [109]. The monoid orbit $\mathsf{End} f_{\text{com}}$ decomposes into two disjoint orbits:

$$\mathsf{End} f_{\text{com}} = \mathsf{G} f_{\text{com}} \cup (\mathsf{End} \setminus \mathsf{G}) f_{\text{com}}.$$

Our goal is to show two things independently:

(1) $f_0 \notin (\mathsf{End} \setminus \mathsf{G}) f_{\text{com}}$, and

(2) $f_0 \notin \mathsf{G} f_{\text{com}}$,

which finishes the proof of Theorem 2.3.

All elements in $(\mathsf{End} \setminus \mathsf{G}) f_{\text{com}}$ are *not concise*, a term that we define in Section 2.6, where we also prove that f_0 is concise. Therefore $f_0 \notin (\mathsf{End} \setminus \mathsf{G}) f_{\text{com}}$.

All elements in $\mathsf{G} f_{\text{com}}$ have *full orbit dimension*, a term that we define in Section 2.7 and we prove that f_0 does *not* have full orbit dimension in Section 2.8. This finishes the proof of Theorem 2.3.

2.6 Conciseness

In this section we show that $f_0 \notin (\mathsf{End} \setminus \mathsf{G}) f_{\text{com}}$. To do so we use a notion called *conciseness*. Informally, it captures whether a polynomial depends on all variables independent of a change of basis, or a tensor cannot be embedded into a tensor product of smaller spaces.

Given a tensor f in $\mathbb{C}^{m_1} \otimes \mathbb{C}^{m_2} \otimes \cdots \otimes \mathbb{C}^{m_d}$, we associate the following matrices with f. For $j \in [d]$, define a matrix M_f^j of dimension $m_j \times (\prod_{i \in [d] \setminus \{j\}} m_i)$ with rows labeled by the standard basis of $\mathbb{C}^{m_j}$, and columns by elements in the Cartesian product $\{\text{standard basis of } \mathbb{C}^{m_1}\} \times \cdots \times \{\text{standard basis of } \mathbb{C}^{m_{j-1}}\} \times \{\text{standard basis of } \mathbb{C}^{m_{j+1}}\} \times \cdots \times \{\text{standard basis of } \mathbb{C}^{m_d}\}$. We write the tensor f in the standard basis

$$f = \sum_{\substack{1 \le i_1 \le m_1 \\ 1 \le i_2 \le m_2 \\ \vdots \\ 1 \le i_d \le m_d}} \alpha_{i_1,\ldots,i_d} e_{i_1} \otimes \cdots \otimes e_{i_d}$$

and associate to it the matrix M_f^j whose entry at position $((i_j), (i_1, i_2, \ldots, i_{j-1}, i_{j+1}, \ldots, i_d))$ is $\alpha_{i_1,\ldots,i_d}$.

Definition 2.1. *We say that a tensor f in $\mathbb{C}^{m_1} \otimes \mathbb{C}^{m_2} \otimes \cdots \otimes \mathbb{C}^{m_d}$ is* concise *if and only if for all $j \in [d]$, M_f^j has full rank.* [5]

As a warm-up exercise we now show that f_{com} is concise.

[5] When f is viewed as a set-multilinear polynomial (see [154, Section 1.4]), this condition translates to the linear independence of the partial derivatives of f. In particular, M_f^j is testing if the partial derivatives of f with respect to the j-th block of variables are all linearly independent. This partial derivatives based criterion for testing if a polynomial depends on all the variables, independent of a change of basis, is pretty standard: see, for instance, [108, Corollary 5.1.4].

Proposition 2.6.1. *f_{com} is concise.*

Proof. We know that $f_{\mathrm{com}} \in W$ (recall that $W := \mathbb{C}^{w_1 w_2} \otimes \mathbb{C}^{w_2 w_3} \otimes \cdots \otimes \mathbb{C}^{w_d w_{d+1}}$) and the monoid End. Let us consider the matrix $M^j_{f_{\mathrm{com}}}$ for some $j \in [d]$. To establish that $M^j_{f_{\mathrm{com}}}$ has full rank, it suffices to show that rows are linearly independent. In order to show that, we argue that every row is non-zero and every column has at most one non-zero entry. In other words, rows are supported on disjoint sets of columns.

A row of $M^j_{f_{\mathrm{com}}}$ is labeled by an edge in the j-th layer of the ABP Γ_{com}. Recall that only paths that start at a vertex in V^1 and end at the corresponding vertex in V^{d+1} contribute to the computation in Γ_{com}. We call such paths *valid paths*. An entry in $M^j_{f_{\mathrm{com}}}$ is non-zero iff the corresponding row and column labels form a valid path in Γ_{com}. Thus, it is easily seen that a row is non-zero iff there is a valid path in Γ_{com} that *passes through* the edge given by the row label. By the structure of Γ_{com}, in particular that every layer is a complete bipartite graph, we observe that passing through every edge there is some valid path. Hence, we obtain that every row is non-zero.

The second claim now follows from the observation that fixing $d-1$ edges either defines a unique dth edge so that these d edges form a valid path, or for these $d-1$ edges there is no such dth edge. □

As mentioned in Section 2.5, to establish $f_0 \notin (\mathsf{End} \setminus \mathsf{G}) f_{\mathrm{com}}$ we will show that f_0 is concise while any element in $(\mathsf{End} \setminus \mathsf{G}) f_{\mathrm{com}}$ is not.

Lemma 2.1. *f_0 is concise.*

Proof. Analogous to the proof of Proposition 2.6.1, we again show that every row of $M^j_{f_0}$ is non-zero and every column of it has at most one non-zero entry. That is, rows of $M^j_{f_0}$ are supported on disjoint sets of columns.

From the construction of f_0 it is seen that a path in Γ_{com} contributes to the computation of f_0 iff it is a valid path that comprises of *exactly one* parity preserving edge. The second claim of every column having at most one non-zero entry now follows for the same reason as in the proof of Proposition 2.6.1.

Before proving the first claim, we recall two assumptions in the construction of f_0. The first is that the format $w = (w_1, w_2, \ldots, w_d)$ is such that $w_i \geq 2$ for all $i \in [d]$ and the second is that d is odd. To argue that a row is non-zero it suffices to show that a valid path comprising of only one parity preserving edge passes through the edge given by the row level. Let us consider an arbitrary edge e in Γ_{com}. We have two cases to consider depending on whether it is parity *preserving* or *changing*.

Case 1. Suppose e is parity preserving and it belongs to a layer $j \in [d]$. The number of layers on the left of e is $j-1$ and on the right is $d-j$. Since d is odd, these numbers are either both even or both odd. We now argue for the case when they are even (the odd case is analogous). Choose a vertex v in V^1 that has the same parity (different in the odd case) as one of the end points of e. (Such a choice exists because $w_1 \geq 2$.) We now claim that there exists a valid path starting at v that passes through e and contains exactly one parity preserving edge. Since e is parity preserving, all edges in

the claimed path must be parity changing. We observe that e can be easily extended in both directions using parity changing edges such that the path ends at $\mathsf{corr}(v)$. The existence of parity changing edges at each layer uses the assumption that $w_i \geq 2$.

Case 2. Otherwise $e = (a, b)$ is parity changing. Again as before there are two cases based on whether both $j-1$ and $d-j$ are even or odd. Consider the case when they are even (the odd case being analogous). We first assume that $j \neq d$. Choose a vertex v in V^1 that has the same parity as a. We now construct a valid path from v to $\mathsf{corr}(v)$ that passes through e and contains exactly one parity preserving edge. It is easily seen that there exists a path from v to a using only parity changing edges. We choose a parity preserving outgoing edge incident to b. We call its endpoint v_1. Since v_1 and v have different parities, we can connect v_1 to $\mathsf{corr}(v)$ in V^{d+1} using only parity changing edges. Thus we obtain the following valid path $v \to \cdots \to a \to b \to v_1 \to \cdots \to \mathsf{corr}(v)$ passing through exactly one parity preserving edge (b, v_1). In the case that $j = d$, choose an incoming parity preserving edge incident on a instead of an outgoing edge on b. □

Remark 2.1. *We note that if the format $w = (w_1, \ldots, w_d)$ defining f_0 is such that for some $j \in [d]$, $w_j = 1$, then f_0 is* not *concise. This can be seen as follows.*

Let $w_j = 1$, and let v denote the unique vertex in V^j. Let e be the edge $e = (1, 1, j)$. If $j < d$, let e' be the edge $e' = (1, 1, j+1)$, otherwise let e' be the edge $e' = (1, 1, j-1)$. Both e, e' are parity preserving edges. By construction, every valid path using e' must also use e. Hence the corresponding row in the matrix $M_{f_0}^{j+1}$ if $j < d$, and in $M_{f_0}^{j-1}$ otherwise, is zero. Therefore f_0 is not concise.

This is an interesting observation, because this is the point where our proof fails for single-(source, sink) ABPs, and this is expected, because Nisan [147] had shown that the set of polynomials computed by such ABPs of format w is a closed set.

Lemma 2.2. *Let $f \in (\mathsf{End} \setminus \mathsf{G}) f_{\mathrm{com}}$. Then f is not concise.*

Proof. This statement is true in very high generality. In our specific case a proof goes as follows. If $f \in (\mathsf{End} \setminus \mathsf{G}) f_{\mathrm{com}}$, then $f = g f_{\mathrm{com}}$ for some $g \in \mathsf{End} \setminus \mathsf{G}$. Let $g = (g_1, \ldots, g_d)$, where $g_i \in \mathbb{C}^{w_i w_{i+1} \times w_i w_{i+1}}$. Since $g \notin \mathsf{G}$, at least one of the g_i must be singular. The crucial property is $M^i_{g f_{\mathrm{com}}} = g_i M^i_{f_{\mathrm{com}}}$, which finishes the proof. □

2.7 Orbit dimension, tangent spaces, and flows

In this section we introduce tangent spaces and study their dimensions. We especially study them in the context of $\mathsf{G} f_{\mathrm{com}}$, and $\mathsf{G} f_0$.

The *orbit dimension* of a tensor $f \in \mathbb{C}^{w_1 w_2} \otimes \mathbb{C}^{w_2 w_3} \otimes \cdots \otimes \mathbb{C}^{w_d w_{d+1}}$ is the dimension of the orbit $\mathsf{G} f$ as an affine variety. It can be determined as the dimension of the tangent space T_f of the action of G at f, which is a vector space defined as follows. Let $\mathfrak{g} := \mathbb{C}^{w_1 w_2 \times w_1 w_2} \times \cdots \times \mathbb{C}^{w_d w_{d+1} \times w_d w_{d+1}}$. For $A \in \mathfrak{g}$ we define the *Lie algebra action* $Af := \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}((\mathrm{id} + \varepsilon A) f - f)$, where $\mathrm{id} \in \mathsf{G}$ is the identity element. We define the vector space

$$T_f := \mathfrak{g} f = \{Af \mid A \in \mathfrak{g}\}.$$

2.7.1 Claim. The dimension $\dim T_h$ is the same for all $h \in \mathsf{G}f$.

Proof. Since the action of G is linear, for all $g \in \mathsf{G}$ and $A \in \mathfrak{g}$ we have

$$\begin{aligned} A(gf) &= \lim_{\varepsilon\to 0} \tfrac{1}{\varepsilon}\left((\mathrm{id} + \varepsilon A)(gf) - gf\right) = \lim_{\varepsilon\to 0} \tfrac{1}{\varepsilon}\left(gg^{-1}(\mathrm{id} + \varepsilon A)gf - gf\right) \\ &= g \lim_{\varepsilon\to 0} \tfrac{1}{\varepsilon}\left((\mathrm{id} + \varepsilon(g^{-1}Ag))f - f\right) = g((g^{-1}Ag)f) \end{aligned}$$

Since $A \mapsto g^{-1}Ag$ is a bijection on $\mathfrak{g}$, it follows that $T_{gf} = gT_f$. Hence the claim follows. □

In the following we will use Claim 2.7.1 to argue $f_0 \notin \mathsf{G}f_{\mathrm{com}}$ by showing that $\dim T_{f_{\mathrm{com}}}$ and $\dim T_{f_0}$ are different.

Let $e, e' \in E^i$ and let $A^{(i)}_{e,e'} \in \mathfrak{g}$ denote the matrix tuple where the i-th matrix has a 1 at position (e, e') and all other entries (also in all other matrices) are 0. Since these matrices form a basis of $\mathfrak{g}$, it follows that

$$\mathfrak{g}f = \mathrm{linspan}\{A^{(i)}_{e,e'}f\}.$$

For a tensor f we define the *support* of f as the set of monomials (i.e., standard basis tensors) for which f has nonzero coefficient. For a linear subspace $V \subseteq \mathbb{C}^{w_1w_2} \otimes \mathbb{C}^{w_2w_3} \otimes \cdots \otimes \mathbb{C}^{w_dw_{d+1}}$ we define the *support* of V as the union of the supports of all $f \in V$.

We write $e \cap e' = \varnothing$ to indicate that two edges e and e' do not share any vertex. We write $|e \cap e'| = 1$ if they share exactly one vertex. We observe that for $f \in \{f_{\mathrm{com}}, f_0\}$ the vector space T_f decomposes into a direct sum of three vector spaces,

$$\begin{aligned} \mathfrak{g}_2 &:= \mathrm{linspan}\{A^{(i)}_{e,e'} \mid 1 \le i \le d, 1 \le e, e' \le w_iw_{i+1}, e \cap e' = \varnothing\} \\ \mathfrak{g}_1 &:= \mathrm{linspan}\{A^{(i)}_{e,e'} \mid 1 \le i \le d, 1 \le e, e' \le w_iw_{i+1}, |e \cap e'| = 1\} \\ \mathfrak{g}_0 &:= \mathrm{linspan}\{A^{(i)}_{e,e} \mid 1 \le i \le d, 1 \le e \le w_iw_{i+1}\}. \\ \mathfrak{g} &= \mathfrak{g}_0 \oplus \mathfrak{g}_1 \oplus \mathfrak{g}_2 \\ T_f &= \mathfrak{g}_0 f \oplus \mathfrak{g}_1 f \oplus \mathfrak{g}_2 f \end{aligned}$$

The last direct sum decomposition follows from the fact that $\mathfrak{g}_0 f$, $\mathfrak{g}_1 f$, and $\mathfrak{g}_2 f$ have pairwise disjoint supports.

We show in this section that $\dim \mathfrak{g}_2 f_{\mathrm{com}} = \dim \mathfrak{g}_2 f_0$, and that $\dim \mathfrak{g}_1 f_{\mathrm{com}} = \dim \mathfrak{g}_1 f_0$. In Section 2.8 we show that $\dim \mathfrak{g}_0 f_{\mathrm{com}} > \dim \mathfrak{g}_0 f_0$, which then implies $f_0 \notin \mathsf{G}f_{\mathrm{com}}$ by Claim 2.7.1. In fact, Theorem 2.4 gives the exact dimension of $\mathfrak{g}_0 f_{\mathrm{com}}$ by proving that $\mathfrak{g}_0 f_{\mathrm{com}}$ is isomorphic to the vector space of flows on the ABP digraph when identifying vertices in V^1 with their corresponding vertices in V^{d+1}. Theorem 2.5 establishes an additional equation based on the vertex parities that shows that $\mathfrak{g}_0 f_0$ is strictly lower dimensional than $\mathfrak{g}_0 f_{\mathrm{com}}$.

We start with Lemma 2.3, which shows that $\dim \mathfrak{g}_2 f_{\mathrm{com}}$ and $\dim \mathfrak{g}_2 f_0$ have *full* dimension.

Lemma 2.3. *Let $f \in \{f_{\mathrm{com}}, f_0\}$. The space $\mathfrak{g}_2 f$ has full dimension. That is, its dimension equals $\sum_{i=1}^{d} w_iw_{i+1}(w_i - 1)(w_{i+1} - 1)$.*

Proof. Suppose $f = f_{\mathrm{com}}$. The other case being analogous, we only argue this case.

We analyze the monomials that appear in the different $A^{(i)}_{e,e'} f_{\mathrm{com}}$ and argue that a monomial that appears in some $A^{(i)}_{e,e'} f_{\mathrm{com}}$ can only appear in that specific $A^{(i)}_{e,e'} f_{\mathrm{com}}$. Indeed, each monomial corresponds to a valid path in which one edge e in layer i is changed to e'. Since e and e' share no vertex, from this edge sequence we can reconstruct i, e, and e' uniquely: e' is the edge that does not have any vertex in common with the rest of the edge sequence, i is its layer, and e is the unique edge that we can replace e' by in order to form a valid path. We conclude that the $A^{(i)}_{e,e'} f_{\mathrm{com}}$ have disjoint support and the lemma follows. □

To establish that $\dim \mathfrak{g}_1 f_{\mathrm{com}} = \dim \mathfrak{g}_1 f_0$, we introduce some notation.

For a connected directed graph $G = (V, E)$ we define a *flow* to be a labeling of the edge set E by complex numbers such that at every vertex the sum of the labels of the incoming edges equals the sum of the labels of the outgoing edges. It is easily seen that the set of flows forms a vector space F. We have

$$\dim F = |E| - |V| + 1, \tag{2.7.2}$$

see e.g. [39, Theorem 20.7].

Recall that E^i denotes the set of edges from V^i to V^{i+1}. Let $\mathscr{X} := E^1 \times \cdots \times E^d$ denote the direct product of the sets of edge lists. Each directed path of length d from layer 1 to $d+1$ is an element of $\mathscr{X}$, but $\mathscr{X}$ contains other edge sets as well. Define $E_i := \mathbb{C}^{E^i}$. Consider the following map φ from $\mathscr{X}$ to $E_1 \otimes \cdots \otimes E_d$,

$$\varphi(e_1, \ldots, e_d) = x_{e_1} \otimes \cdots \otimes x_{e_d} \in E_1 \otimes \cdots \otimes E_d$$

where (x_{ij}) is the standard basis of E_i. Note φ is a bijection between $\mathscr{X}$ and the standard basis of $E_1 \otimes \cdots \otimes E_d$.

An edge set in $\mathscr{X}$ is called a *valid path* if it forms a path that starts and ends at corresponding vertices (see Section 2.1). Let $\mathscr{P} \subseteq \mathscr{X}$ denote the set of valid paths.

Proposition 2.7.3. $\dim \mathfrak{g}_1 f_{\mathrm{com}} = \dim \mathfrak{g}_1 f_0 = \sum_{i=1}^{d} (w_{i-1} + w_{i+1} - 1)(w_i - 1) w_i$, *where* $w_0 := w_d$.

Proof. The proof works almost analogously for f_{com} and f_0, so we treat only the more natural case f_{com}. We show that $\mathfrak{g}_1 f_{\mathrm{com}}$ is isomorphic to a direct sum of vector spaces of flows on very simple digraphs. Fix $1 \le i \le d$. Fix distinct $1 \le a, b \le w_i$. For distinct edges $e, e' \in E^i$, let $\mathscr{P}_{e,e'} \subseteq \mathscr{X}$ be the set of edge sets containing e' that are not valid paths, but that become valid paths by removing e' and adding e. Let $\mathscr{P}^i_{a,b} \subseteq \mathscr{X}$ be the set of edge sets that are not valid paths, but that become valid paths by switching the end point of the $(i-1)$-th edge to v^i_b and that also become valid paths by switching the start point of the i-th edge to v^i_a (if $i - 1 = 0$, then interpret $i - 1 := d$). Pictorially, this means that elements in $\mathscr{P}^i_{a,b}$ are almost valid paths, but there is a discontinuity at layer i, where the path jumps from vertex v^i_a to vertex v^i_b. We have

$$A^{(i)}_{e,e'} f_{\mathrm{com}} = \sum_{p \in \mathscr{P}_{e,e'}} \varphi(p).$$

The vectors $\{A^{(i)}_{e,e'} f_{\mathrm{com}} \mid 1 \le i \le d, e, e' \in E^i, |e \cap e'| = 1\}$ are not linearly independent, because for $a \neq b$ we have

$$\sum_{\substack{e \text{ and } e' \text{ have the same start point} \\ e' \text{ ends at the } a\text{-th vertex} \\ e \text{ ends at the } b\text{-th vertex}}} A^{(i-1)}_{e,e'} f_{\mathrm{com}} = \sum_{p \in \mathscr{P}^i_{a,b}} \varphi(p) = \sum_{\substack{h \text{ and } h' \text{ have the same end point} \\ h \text{ starts at the } a\text{-th vertex} \\ h' \text{ starts at the } b\text{-th vertex}}} A^{(i)}_{h,h'} f_{\mathrm{com}}. \tag{2.7.4}$$

Define

$$\begin{aligned} T_{a,b,i} \;:=\; & \operatorname{linspan}\left\{ A^{(i-1)}_{e,e'} f_{\mathrm{com}} \;\middle|\; \begin{array}{c} e \text{ and } e' \text{ have the same start point} \\ e' \text{ ends at the } a\text{-th vertex} \\ e \text{ ends at the } b\text{-th vertex} \end{array} \right\} \\ + \; & \operatorname{linspan}\left\{ A^{(i)}_{h,h'} f_{\mathrm{com}} \;\middle|\; \begin{array}{c} h \text{ and } h' \text{ have the same end point} \\ h \text{ starts at the } a\text{-th vertex} \\ h' \text{ starts at the } b\text{-th vertex} \end{array} \right\}. \end{aligned}$$

The support of $T_{a,b,i}$ and $T_{\tilde{a},\tilde{b},\tilde{i}}$ are disjoint, provided $(a,b,i) \neq (\tilde{a},\tilde{b},\tilde{i})$. Hence

$$\mathfrak{g}_1 f_{\mathrm{com}} = \bigoplus_{\substack{1 \le i \le d \\ 1 \le a,b \le w_i \\ a \neq b}} T_{a,b,i}$$

It remains to prove that the dimension of $T_{a,b,i}$ is $w_{i-1} + w_{i+1} - 1$, because then

$$\dim \mathfrak{g}_1 f_{\mathrm{com}} = \sum_{\substack{1 \le i \le d \\ 1 \le a,b \le w_i \\ a \neq b}} (w_{i-1} + w_{i+1} - 1) = \sum_{i=1}^{d} (w_{i-1} + w_{i+1} - 1)(w_i - 1) w_i.$$

Note that $T_{a,b,i}$ is defined as the linear span of $w_{i-1} + w_{i+1}$ many vectors, but (2.7.4) shows that these are not linearly independent. We prove that (2.7.4) is the only equality by showing that $T_{a,b,i}$ is isomorphic to a flow vector space. We define a multigraph with two vertices: $\odot$ and $\circledast$. We have w_{i+1} many edges from $\odot$ to $\circledast$, and we have w_{i-1} many edges from $\circledast$ to $\odot$. We denote by $\circledast \xrightarrow{k} \odot$ the k-th edge from $\circledast$ to $\odot$. Let $F_{a,b,i}$ denote the vector space of flows on this graph. Its dimension is $w_{i-1} + w_{i+1} - 1$, see (2.7.2). We define $\varrho : E^1 \otimes \cdots \otimes E^d \to F_{a,b,i}$ on rank 1 tensors via

$$\varrho(x_{e_1} \otimes \cdots \otimes x_{e_d})(\circledast \xrightarrow{k} \odot) = \begin{cases} 1 & \text{if } e_{i-1} \text{ starts at } k \text{ in layer } i-1 \text{ and ends at } a \text{ in layer } i, \\ 0 & \text{otherwise.} \end{cases}$$

$$\varrho(x_{e_1} \otimes \cdots \otimes x_{e_d})(\odot \xrightarrow{l} \circledast) = \begin{cases} 1 & \text{if } e_i \text{ starts at } b \text{ in layer } i \text{ and ends at } l \text{ in layer } i+1, \\ 0 & \text{otherwise.} \end{cases}$$

Using (2.7.4) it is readily verified that ϱ maps $T_{a,b,i}$ to $F_{a,b,i}$. It remains to show that $\varrho : T_{a,b,i} \to F_{a,b,i}$ is surjective. Let $\alpha := |\mathscr{P}^i_{a,b}|$. We observe that

$$\varrho(A^{(i-1)}_{e,e'} f_{\mathrm{com}})(\circledast \xrightarrow{k} \odot) = \begin{cases} \alpha/w_{i-1} & \text{if } e \text{ and } e' \text{ both start at the } k\text{-th vertex} \\ 0 & \text{if } e \text{ and } e' \text{ both start at the same vertex, but not at the } k\text{-th} \end{cases}$$

$$\varrho(A^{(i-1)}_{e,e'} f_{\mathrm{com}})(\odot \xrightarrow{l} \circledast) = \alpha/(w_{i-1} w_{i+1})$$

$$\varrho(A^{(i)}_{h,h'} f_{\mathrm{com}})(\odot \xrightarrow{l} \circledast) = \begin{cases} \alpha/w_{i+1} & \text{if } h \text{ and } h' \text{ both end at the } l\text{-th vertex} \\ 0 & \text{if } h \text{ and } h' \text{ both end at the same vertex, but not at the } l\text{-th} \end{cases}$$

$$\varrho(A^{(i)}_{h,h'} f_{\mathrm{com}})(\circledast \xrightarrow{k} \odot) = \alpha/(w_{i-1} w_{i+1})$$

Let $\Xi := \sum A_{e,e'}^{(i-1)} f_{\mathrm{com}}$. Then $\forall k : \varrho(\Xi)(\circledast \overset{k}{\to} \odot) = \alpha/w_{i-1}$ and $\forall l : \varrho(\Xi)(\odot \overset{l}{\to} \circledast) = \alpha$. Therefore, for e, e' starting at the k_0-th vertex and h, h' ending at the l_0-th vertex we have that

$$\varrho\Big(w_{i-1}w_{i+1}\varrho(A_{e,e'}^{(i-1)} f_{\mathrm{com}}) + w_{i-1}w_{i+1}\varrho(A_{h,h'}^{i} f_{\mathrm{com}}) - \Xi\Big)$$

is nonzero only on exactly two edges: $\circledast \overset{k_0}{\to} \odot$ and $\odot \overset{l_0}{\to} \circledast$. Cycles form a generating set of the vector space $F_{a,b,i}$, which finishes the proof of the surjectivity of ϱ. □

2.8 Flows on ABPs

We now proceed to the analysis of $\mathfrak{g}_0 f_{\mathrm{com}}$ and $\mathfrak{g}_0 f_0$. The connection to flow vector spaces will be even more prevalent than in Proposition 2.7.3. The main result of this section is $\dim \mathfrak{g}_0 f_{\mathrm{com}} > \dim \mathfrak{g}_0 f_0$ (Theorem 2.4 and Theorem 2.5), which implies that f_{com} and f_0 have different orbit dimensions. We thereby conclude that $f_0 \notin \mathsf{G} f_{\mathrm{com}}$.

To each edge e we assign its *path tensor* $\psi(e)$ by summing tensors over all valid paths passing through e,

$$\psi(e) := \sum_{p \in \mathscr{P} \text{ with } e \in p} \varphi(p) \in E_1 \otimes \cdots \otimes E_d.$$

By linear continuation this gives a linear map $\psi : \mathbb{C}^E \to E_1 \otimes \cdots \otimes E_d$.

Observe that $\psi(e) = A_{e,e}^{(i)} f_{\mathrm{com}}$. Let $\mathscr{T}$ denote the linear span of all $\psi(e)$, $e \in E$. In other words, $\mathscr{T} = \mathfrak{g}_0 f_{\mathrm{com}}$.

Let $\mathscr{P}' \subseteq \mathscr{P} \subseteq \mathscr{X}$ be the set of valid paths that contain exactly one parity preserving edge. To each edge e we assign its *parity path tensor* $\psi'(e)$ by summing tensors over paths in $\mathscr{P}'$,

$$\psi'(e) := \sum_{p \in \mathscr{P}' \text{ with } e \in p} \varphi(p) \in E_1 \otimes \cdots \otimes E_d.$$

By linear continuation this gives a linear map $\psi' : \mathbb{C}^E \to E_1 \otimes \cdots \otimes E_d$. Observe that $\psi'(e) = A_{e,e}^{(i)} f_0$. Let $\mathscr{T}'$ denote the linear span of all $\psi'(e)$, $e \in E$. In other words, $\mathscr{T}' = \mathfrak{g}_0 f_0$.

We will establish the following bounds on the dimensions of $\mathscr{T}$ and $\mathscr{T}'$.

Theorem 2.4. $\dim \mathscr{T} = |E| - \sum_{i=1}^{d} w_i + 1$.

Theorem 2.5. $\dim \mathscr{T}' \le |E| - \sum_{i=1}^{d} w_i$.

The rest of this section is dedicated to the proofs of Theorem 2.4 and Theorem 2.5 by showing that $\mathscr{T}$ is isomorphic to the vector space of flows "on the ABP", while the parity constraints lead to a smaller dimension of $\mathscr{T}'$.

From an ABP Γ we construct a digraph $\tilde{\Gamma}$ by identifying corresponding vertices from the first and the last layer in V and calling the resulting vertex set $\tilde{V}$. Note $|\tilde{V}| = \sum_{i=1}^{d} w_i$. The directed graphs Γ and $\tilde{\Gamma}$ have the same edge set. The resulting directed graph is called $\tilde{\Gamma} = (\tilde{V}, E)$. Let F denote the vector space of flows on $\tilde{\Gamma}$. Note that by (2.7.2) we

have $\dim F = |E| - |\tilde{V}| + 1$. All directed cycles in $\tilde{\Gamma}$ have a length that is a multiple of d. In particular, all cycles of length exactly d are in one-to-one correspondence with valid paths in Γ_{com}. For an edge $e \in E$, let $\chi(e) \in \mathbb{C}^E$ denote the characteristic function of e, i.e., the function whose value is 1 on e and 0 everywhere else.

We now prove Theorem 2.4 by establishing a matching upper (Lemma 2.4) and lower bound (Lemma 2.5) of $|E| - |\tilde{V}| + 1 = \dim F$ on $\dim \mathscr{T}$.

The upper bound

Lemma 2.4. $\dim \mathscr{T} \leq |E| - |\tilde{V}| + 1$.

Proof. For $v \in \tilde{V}$, let $\mathsf{in}(v) \subseteq E$ denote the set of incoming edges incident to v and $\mathsf{out}(v) \subseteq E$ denote the set of outgoing edges incident to v. For each $v \in \tilde{V}$, define the row vector

$$r_v = \sum_{e \in \mathsf{in}(v)} \chi(e) - \sum_{e \in \mathsf{out}(v)} \chi(e).$$

These vectors are the rows of the signed incidence matrix of $\tilde{\Gamma}$, and since $\tilde{\Gamma}$ is connected, they span a space of dimension $|\tilde{V}| - 1$ ([39, Ex. 1.5.6]). Now observe that for all $v \in \tilde{V}$,

$$\sum_{e \in \mathsf{in}(v)} \psi(e) = \sum_{e \in \mathsf{out}(v)} \psi(e).$$

Since ψ is linear, this is equivalent to

$$\psi\left(\sum_{e \in \mathsf{in}(v)} \chi(e) - \sum_{e \in \mathsf{out}(v)} \chi(e) \right) = 0.$$

Hence each r_v is in the kernel of ψ, and hence $\dim \operatorname{Ker} \psi \geq |\tilde{V}| - 1$. Using (2.7.2), we obtain $\dim \mathscr{T} = \dim \operatorname{Im} \psi = |E| - \dim \operatorname{Ker} \psi \leq |E| - |\tilde{V}| + 1 = \dim F$. □

The lower bound

To obtain the lower bound, we define a linear map $\varrho : E_1 \otimes \cdots \otimes E_d \to \mathbb{C}^E$ such that the image of ϱ restricted to $\mathscr{T}$ equals F. This will imply that $\dim \mathscr{T} \geq \dim F$, thereby achieving the required lower bound.

We define the linear map ϱ on standard basis elements $x_{e_1} \otimes \cdots \otimes x_{e_d}$ as follows,

$$\varrho(x_{e_1} \otimes \cdots \otimes x_{e_d}) := \chi(e_1) + \cdots + \chi(e_d),$$

and then extend it to the domain $E_1 \otimes \cdots \otimes E_d$ via linear continuation.

Lemma 2.5. *Let $\varrho|_{\mathscr{T}}$ denote the restriction of ϱ to the linear subspace $\mathscr{T}$. Then,* $\operatorname{Im} \varrho|_{\mathscr{T}} = F$. *In particular,* $\dim \mathscr{T} \geq \dim F = |E| - |\tilde{V}| + 1$.

Proof. To prove equality it suffices to show $\operatorname{Im} \varrho|_{\mathscr{T}} \subseteq F$ and $F \subseteq \operatorname{Im} \varrho|_{\mathscr{T}}$.

The first containment is easy to see. For an edge e, consider the image of $\psi(e)$ under the map ϱ,

$$\varrho(\psi(e)) = \sum_{e \in p \in \mathscr{P}} \sum_{e' \in p} \chi(e').$$

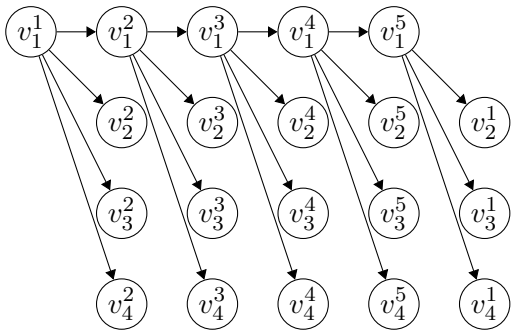

Figure 2.1: The spanning tree construction for width 4 and $d = 5$.

Observe that for a path $p \in \mathscr{P}$, $\sum_{e' \in p} \chi(e')$ is a flow on $\tilde{\Gamma}$ and hence it belongs to F. Thus, we have $\varrho(\psi(e)) \in F$. Since $\mathscr{T}$ is spanned by $\psi(e)$, for $e \in E$, we obtain that $\operatorname{Im} \varrho|_{\mathscr{T}} \subseteq F$.

To establish the second containment it suffices to show that the image of $\mathscr{T}$ under the map ϱ contains a basis of F. We identify a specific basis for F in Claim 2.8.1 and prove that it is contained in $\operatorname{Im} \varrho|_{\mathscr{T}}$ in Claim 2.8.2 to complete the argument. □

We identify directed cycles with their characteristic flows, i.e., flows that have value 1 on the cycle's edges and 0 everywhere else. We also identify directed cycles that use edges in any direction with their characteristic flow: the characteristic flow is defined to take the value 1 on an edge e if e is traversed in the direction of e, and value -1 on e if e is traversed against its direction.

From the theory of flows we know that for every (undirected) spanning tree T of $\tilde{\Gamma}$, the vector space $F \in \mathbb{C}^E$ has a basis given by the characteristic flows of cycles that only use edges from T and exactly one additional edge (for example, see [39, Theorem 20.8]). Thus, the cycle flows corresponding to the elements not in the spanning tree form a basis of F.

2.8.1 Claim. *F is spanned by the set of directed cycles in $\tilde{\Gamma}$ of length exactly d.*

Proof. We construct a spanning tree τ as follows, which will be a tree whose edges are all directed away from its root. Informally, the tree is given by the following subgraph, we make the first vertex in V^1 as root, and include all the outgoing edges incident to it. We then move to the first vertex in V^2 and include all the outgoing edges incident to it. We continue in this way until we reach V^d. Upon reaching the first vertex in V^d we include all but one outgoing edges incident to it. The one that is an incoming edge to the root is not included. Figure 2.1 illustrates the construction. We now formally define this.

Let $v_1^i \in V^i$ denote the first vertex in the layer i, $1 \le i \le d$. Further recall $\mathsf{in}(v) \subseteq E$ and $\mathsf{out}(v) \subseteq E$ denote the set of incoming and outgoing edges, respectively, incident to v. Define the edge set

$$\tau := \left(\bigcup_{i=1}^{d} \mathsf{out}(v_1^i) \right) \setminus \{(v_1^d, v_1^1)\},$$

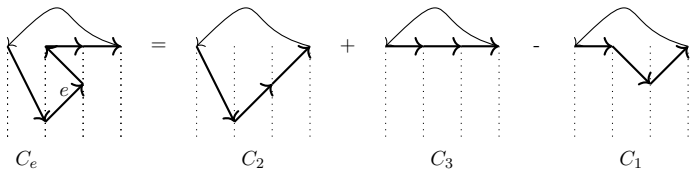

Figure 2.2: Decomposing a cycle of length $d+2$ as a linear combination of cycles of length d. The figure is an illustration when $d = 3$. The dotted layers in each cycle from the left are V^3, V^1, V^2, and V^3 again.

which is a spanning tree in $\tilde{\Gamma}$. We know that every edge not in the tree when added to the tree gives a unique undirected cycle. We now show that the characteristic flows of these undirected cycles can be expressed as a linear combination of the characteristic flows of directed cycles of length d. For $e \in E \setminus \tau$, let c_e denote the characteristic flow of the unique undirected cycle that uses e in its correct direction and only edges of τ. We argue depending on which layer the edge e belongs to.

- Suppose $e \in E^1 \setminus \tau$.
  - If e is incident to v_1^2, the first vertex in V^2, then the inclusion of e creates a directed cycle of length d. Hence, c_e equals the characteristic flow of this directed cycle.
  - Otherwise, the inclusion of e creates an undirected cycle of length $d+2$. If $e = (v_{j_1}^1, v_{j_2}^2)$ for some $j_1 \in [2, w_1]$ and $j_2 \in [2, w_2]$, then the cycle c_e is given as follows:
    $$v_1^d - v_{j_1}^1 - v_{j_2}^2 - v_1^1 - v_1^2 - \cdots - v_1^{d-1} - v_1^d.$$
    Consider the following two directed cycles:
    $$C_1 : v_1^1 - v_{j_2}^2 - \cdots - v_1^d - v_1^1 \text{ and}$$
    $$C_2 : v_{j_1}^1 - v_{j_2}^2 - \cdots - v_1^d - v_{j_1}^1,$$
    such that the part $v_{j_2}^2 - \cdots - v_1^d$ between $v_{j_2}^2$ and v_1^d in the two cycles is the same. Let us denote the characteristic flow of a cycle C by $\chi(C)$. We now observe that $\chi(C_2) - \chi(C_1)$ equals the characteristic flow of the undirected cycle $v_{j_1}^1 - v_{j_2}^2 - v_1^1 - v_1^d - v_{j_1}^1$. This is because the common part in C_1 and C_2 cancels out. To $\chi(C_2) - \chi(C_1)$ we add the characteristic flow of the directed cycle,
    $$C_3 : v_1^1 - v_1^2 - v_1^3 - \cdots - v_1^{d-1} - v_1^d - v_1^1.$$
    It is now easily seen that $\chi(C_2) - \chi(C_1) + \chi(C_3)$ equals the characteristic flow of the cycle c_e (see Figure 2.2 for an illustration).
- Suppose $e \in E^d \setminus \tau$.
  - If e is incident to v_1^1, the first vertex in V^1, then as before the inclusion of e creates a directed cycle of length d. Hence, c_e equals the characteristic flow

of this directed cycle.

– Otherwise, the inclusion of e creates an undirected cycle of length 4. If $e = (v^d_{j_1}, v^1_{j_2})$ for some $j_1 \in [2, w_d]$ and $j_2 \in [2, w_1]$, then the cycle c_e is given as follows:

$$v^d_{j_1} - v^1_{j_2} - v^d_1 - v^{d-1}_1 - v^d_{j_1}.$$

Consider the following two directed cycles:

$$C_4 : v^1_{j_2} - \cdots - v^{d-1}_1 - v^d_1 - v^1_{j_2} \text{ and}$$
$$C_5 : v^1_{j_2} - \cdots - v^{d-1}_1 - v^d_{j_1} - v^1_{j_2},$$

such that the part $v^1_{j_2} - \cdots - v^{d-1}_1$ between $v^1_{j_2}$ and v^{d-1}_1 in the two cycles is the same. We now claim that $\chi(C_5) - \chi(C_4)$ equals the characteristic flow of c_e. This is because the common part in C_4 and C_5 cancels out.

- Otherwise $e \in E^i \setminus \tau$ for some $i \in \{2, \ldots, d-1\}$. In such a case inclusion of e creates an undirected cycle of length 4. We can again argue exactly like in the previous case, and so we omit the argument here. □

We now prove that the generating set given by the directed cycles of length d is contained in the image of $\mathscr{T}$ under the map ϱ.

2.8.2 Claim. $\mathrm{Im}(\varrho|_{\mathscr{T}})$ contains the characteristic flow of each directed cycle of length d.

Proof. Let $\{e_1, e_2, \ldots, e_d\} \subseteq E$ be a directed cycle of length d, where each e_i points from a vertex in V^i to a vertex in V^{i+1}. Let $\{e^{(j)}_i\}$ denote the set of edges that start at the same vertex as e_i, but for which $e^{(j)}_i \neq e_i$. Thus $|\{e^{(j)}_i\}| = |V^{i+1}| - 1$. Let

$$\bar{\psi}(e) := \frac{1}{|\{p \in \mathscr{P} \text{ with } e \in p\}|}\psi(e),$$

so that $\varrho(\bar{\psi}(e))$ is a flow with value 1 on the edge e. It is instructive to have a look at the left side of Figure 2.3, where $\varrho(\bar{\psi}(e_1))$ is depicted. Subtracting $\frac{1}{w_3}\sum_{j=1}^{w_3-1}\varrho(\bar{\psi}(e^{(j)}_2))$ and adding $\frac{w_3-1}{w_3}\varrho(\bar{\psi}(e_2))$ reduces the support significantly and brings us one step closer to the cycle, see the right side of Figure 2.3. We iterate this process until only the cycle is left. Formally:

$$\begin{aligned}
\chi(e_1, \ldots, e_d) &= \varrho(\bar{\psi}(e_1)) \\
&+ \tfrac{w_3-1}{w_3}\varrho(\bar{\psi}(e_2)) - \tfrac{1}{w_3}\sum_{j=1}^{w_3-1}\varrho(\bar{\psi}(e^{(j)}_2)) \\
&+ \cdots \\
&+ \tfrac{w_d-1}{w_d}\varrho(\bar{\psi}(e_{d-1})) - \tfrac{1}{w_d}\sum_{j=1}^{w_d-1}\varrho(\bar{\psi}(e^{(j)}_{d-1})).
\end{aligned}$$

□

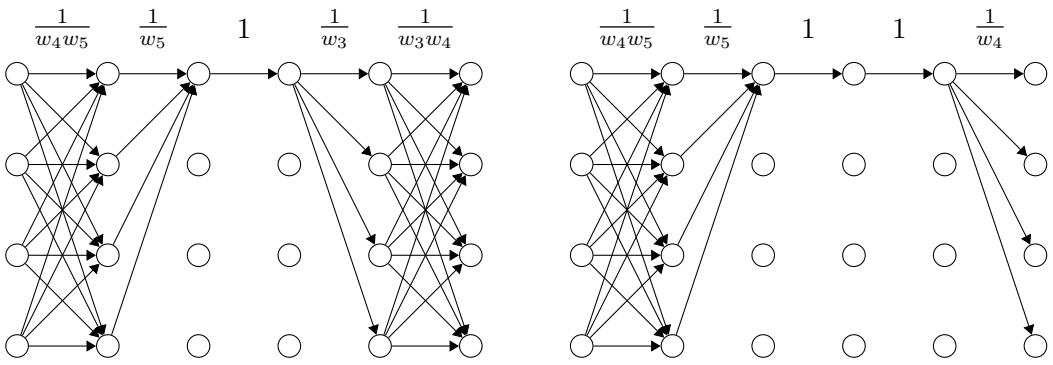

Figure 2.3: On the left: $\varrho(\bar{\psi}(e_1))$. On the right: $\varrho(\bar{\psi}(e_1)) - \frac{1}{w_3}\sum_{j=1}^{w_3-1}\varrho(\bar{\psi}(e_2^{(j)})) + \frac{w_3-1}{w_3}\varrho(\bar{\psi}(e_2))$. This is the case $d = 5$ and format $(4,4,4,4,4)$. Edges that are not drawn carry 0 flow. All edges in the same layer carry either 0 flow or the value that is depicted above the edge layer. For the purposes of illustation, e_1 is the top edge in the *center*. Here we assume that each e_i points from the first vertex in V^i to the first vertex in V^{i+1}.

The stronger upper bound via parities

We now proceed to upper bound $\dim \mathscr{T}'$ (Theorem 2.5). The proof is analogous to the proof of Lemma 2.4.

Theorem 2.6 (Restatement of Theorem 2.5)**.** $\dim \mathscr{T}' \le |E| - |\tilde{V}|$.

Proof. As in the proof of Lemma 2.4, for $v \in \tilde{V}$, we have

$$\sum_{e \in \mathsf{in}(v)} \psi'(e) = \sum_{e \in \mathsf{out}(v)} \psi'(e).$$

Furthermore, we have the following additional constraint on ψ',

$$(d-1) \sum_{e \text{ parity preserving}} \psi'(e) = \sum_{e \text{ parity changing}} \psi'(e).$$

By the linearity of ψ', we have

$$\psi'\left((d-1) \sum_{e \text{ parity preserving}} \chi(e) - \sum_{e \text{ parity changing}} \chi(e)\right) = 0.$$

Therefore, the kernel of ψ' is spanned by the vectors $(\sum_{e \in \mathsf{in}(v)} \chi(e) - \sum_{e \in \mathsf{out}(v)} \chi(e))$, for $v \in \tilde{V}$, and an additional vector $((d-1)\sum_{e \text{ parity preserving}} \chi(e) - \sum_{e \text{ parity changing}} \chi(e))$.

We now claim that the new vector is linearly independent from the earlier set of vectors. We prove the claim by constructing a vector in $\mathbb{C}^E$ that is orthogonal to the earlier set of vectors but is non-orthogonal to the additional vector. One such vector is given by the characteristic flow of the directed cycle $v_1^1 - v_1^2 - v_1^3 - \cdots - v_1^{d-1} - v_1^d - v_1^1$.

Thus, it follows that $\dim \mathrm{Ker}\, \psi' \ge |\tilde{V}|$, and hence $\dim \mathscr{T}' \le |E| - |\tilde{V}|$. □

CHAPTER 3

Separation between VQP and $\overline{\text{VNP}}$

In this chapter, we see an application of trying to capture an algebraic complexity class as an algebraic variety. Using this approach, Bürgisser [44] achieved a separation between the two well known complexity classes VQP and VNP. In this chapter, we observe that the very nature of the technique allows us to separate the complexity classes VQP and $\overline{\text{VNP}}$.

Bürgisser in his monograph [44] defined the complexity class VQP as the class of polynomials with quasi-polynomially bounded straight-line programs, and established its relation to the classes VP and VNP (defined in Section 3.1). He showed that the determinant polynomial is VQP-complete with respect to the so-called *qp*-projections [44, Corollary 2.29]. He strengthened Valiant's hypothesis of $\text{VNP} \not\subseteq \text{VP}$ to $\text{VNP} \not\subseteq \text{VQP}$ and called it *Valiant's extended hypothesis* [44, Section 2.5]. He further showed that VP is strictly contained in VQP as one would intuitively expect [44, Section 8.2]. Finally, he also showed that VQP is not contained in VNP [44, Proposition 8.5 and Corollary 8.9]. In this article, we observe that his proof is stronger and actually shows that VQP is not contained in $\overline{\text{VNP}}$ either, where $\overline{\text{VNP}}$ is the closure of the complexity class VNP.

3.1 Set-up and results

In this section, we compare the complexity classes VQP and $\overline{\text{VNP}}$. Valiant in his seminal paper [179] defined the complexity classes that are now called as VP and VNP, and the central question of algebraic complexity is to understand whether the two complexity classes are indeed different as sets (Valiant's hypothesis). Bürgisser [44] defined the complexity class VQP and related it to the complexity classes VP and VNP. We proceed to define the above three classes for establishing the context. For an exhaustive treatment of the classes, we refer the readers to Bürgisser's monograph [44] from where we are lifting the definitions. We first need to define so-called p-families.

Definition 3.1. *A sequence $f = (f_n)$ of multivariate polynomials over a field k is called a p-family (over k) iff the number of variables as well as the degree of f_n are bounded by polynomial functions in n.*

We now need to define the model of computation and the notion of complexity in order to define the complexity classes of interest.

Definition 3.2. *A* straight-line program Γ *(expecting m inputs) represents a sequence $(\Gamma_1, \ldots, \Gamma_r)$ of instructions $\Gamma_\rho = (\omega_\rho; i_\rho, j_\rho)$ with operation symbols $\omega_\rho \in \{+, -, *\}$ and the address i_ρ, j_ρ which are integers satisfying $-m < i_\rho, j_\rho < \rho$. We call r the size of Γ.*

So, essentially, in a straight-line program, we either perform addition or subtraction or multiplication on the inputs or the previously computed elements. The size of the straight-line program naturally induces a size complexity measure on polynomials as follows:

Definition 3.3. *The* complexity $L(f)$ *of a polynomial* $f \in \mathbb{F}[x_1, \ldots, x_n]$ *is the minimal size of a straight-line program computing* f *from variables* $x_1 \ldots, x_n$*, and constants in* $\mathbb{F}$*.*

We are now all set to define the above discussed complexity classes.

Definition 3.4. *A p-family* $f = (f_n)$ *is said to be p-computable iff the complexity* $L(f_n)$ *is a polynomially bounded function of* n*.* $\mathsf{VP}_{\mathbb{F}}$ *consists of all p-computable families over the field* $\mathbb{F}$*.*

Definition 3.5. *A p-family* $f = (f_n)$ *is said to be* p-definable *iff there exists a p-*computable *family* $g = (g_n)$*,* $g_n \in \mathbb{F}[x_1, \ldots, x_{u(n)}]$*, such that for all* n

$$f_n(x_1, \ldots, x_{v(n)}) = \sum_{e \in \{0,1\}^{u(n)-v(n)}} g_n(x_1, \ldots, x_{v(n)}, e_1, \ldots, e_{u(n)-v(n)}).$$

The set of p-definable families over $\mathbb{F}$ *forms the complexity class* $\mathsf{VNP}_{\mathbb{F}}$*.*

Definition 3.6. *A p-family* $f = (f_n)$ *is said to be* qp-computable *iff the complexity* $L(f_n)$ *is a quasi-polynomially bounded function*[1] *of* n*. The complexity class* $\mathsf{VQP}_{\mathbb{F}}$ *consists of all qp-computable families over* $\mathbb{F}$*.*

In the above three definitions, if the underlying field is clear from the context, we can drop the subscript $\mathbb{F}$ and simply represent the classes as $\mathsf{VP}, \mathsf{VNP}$ and VQP respectively. In what follows, the underlying field is always assumed to be $\mathbb{Q}$, the field of rational numbers.

In [44], Bürgisser showed the completeness of the determinant polynomial for VQP under *qp*-projections and strengthened Valiant's hypothesis of $\mathsf{VNP} \notin \mathsf{VP}$ to $\mathsf{VNP} \notin \mathsf{VQP}$ and called it *Valiant's extended hypothesis* [44, Section 2.5]. He also established that $\mathsf{VP} \subsetneq \mathsf{VQP}$ and went on to show that $\mathsf{VQP} \notin \mathsf{VNP}$ [44, Proposition 8.5 and Corollary 8.9]. The main observation of this section is that his proof is stronger and is sufficient to conclude that VQP is not contained in the closure of VNP either, where the closure is in the sense as mentioned in Section 2.1.

In fact, Bürgisser in his monograph [44] also gives a set of conditions which if the coefficients of a polynomial sequence satisfies, then that polynomial sequence cannot be in VNP [44, Theorem 8.1]. His theorem and the proof is inspired by Heintz and Sieveking [105]. The second observation of this section is that this proof is even stronger and actually those conditions are sufficient to show that the given polynomial sequence is not contained in $\overline{\mathsf{VNP}}$ either.

We now discuss both observations.

[1] a function $s(n)$ is quasi-polynomially bounded if it is bounded by $2^{O((\log n)^c)}$ for some $c > 0$.

3.2 The separation of VQP and $\overline{\text{VNP}}$

We first show that there is a $\log n$ variate polynomial of degree $(n-1)\log n$ which is in VQP but not in $\overline{\text{VNP}}$. In this exposition, for the sake of better readability, we do not present the Bürgisser's statements in full generality since it is not essential for the theorem that we want to show here. Moreover, the less general version that we present here contains all the ideas for the theorem statements and their proofs.

Theorem 3.1. *Let* $N_n := \{0,\dots,n-1\}^{\log n}$ *and* $f_n := \sum_{\mu\in N_n} 2^{2^{j(\mu)}} X_1^{\mu_1}\cdots X_{\log n}^{\mu_{\log n}}$, *where* $j(\mu) := \sum_{j=1}^{\log n} \mu_j n^{j-1}$. *Then* $f_n \in \text{VQP}$, *but* $f_n \notin \overline{\text{VNP}}$, *and hence* $\text{VQP} \notin \overline{\text{VNP}}$.

The theorem consists of two parts. The containment in VQP follows immediately from the fact that the total number of monomials in f_n is $n^{\log n}$. For the other part, we closely follow Bürgisser's lower bound proof [44, Proposition 8.5] against VNP here, making transparent the fact that the proof works also against $\overline{\text{VNP}}$. His proof techniques were borrowed from Strassen ([173]). The idea is to use the universal representation for polynomial sequences in VNP, so that we get a hold on how the coefficients of the polynomials look like. Using that, we establish polynomials H_n that vanish on all the polynomial sequences in VNP (in other words, H_n is in the vanishing ideal of sequences in VNP), but do not vanish on f_n (because the growth rate of its coefficients is too high), hence giving the separation. Since the vanishing ideal of a set characterizes its closure, we get the stronger separation, i.e., f_n does not belong to the closure of VNP, namely, $\overline{\text{VNP}}$.

Proof of Theorem 3.1. As stated above, the proof works in three stages: first, assuming the contrary and writing f_n using the universal representation for the polynomial sequences in VNP, then giving polynomials H_n of special forms in the vanishing ideal of polynomial sequences in VNP, and finally showing that H_n cannot vanish on our sequence f_n, hence arriving at a contradiction.

Assuming $(f_n) \in \text{VNP}$ implies the existence of a family $(g_n) \in \text{VP}$, with $L(g_n)$ bounded by a polynomial $r(n)$, and a polynomial $u(n)$ such that

$$f_n(X_1,\dots,X_{\log n}) = \sum_{e\in\{0,1\}^{u(n)-\log n}} g_n(X_1,\dots,X_{\log n},e_1,\dots,e_{u(n)-\log n}).$$

Next, we use the universal representation theorem [173, 163] as stated in Bürgisser's monograph ([44, Proposition 8.3]; for a proof see [46, Proposition 9.11]) for size $r(n)$ straight-line program to get that there exist polynomials $G_\nu^{(n)} \in \mathbb{Z}[Y_1,\dots,Y_{q(n)}]$, with $q(n)$ being a polynomial in n (more precisely, it is a polynomial in $r(n)$ and $u(n)$) which for $|\nu| \le \deg g_n = n^{O(1)}$, guarantee that $\deg G_\nu = n^{O(1)}, \log \mathsf{wt}(G_\nu)^{(n)} = 2^{n^{O(1)}}$, and also guarantee the existence of some $\zeta \in \overline{\mathbb{Q}}^{q(n)}$, such that

$$g_n = \sum_\nu G_\nu^{(n)}(\zeta) X_1^{\nu_1},\cdots,X_{u(n)}^{\nu_{u(n)}},$$

where weight of a polynomial f, $\mathsf{wt}(f)$ refers to the sum of the absolute values of its coefficients.

Now, taking exponential sum yields that

$$f_n = \sum_{\mu \in N_n} F_\mu^{(n)}(\zeta) X_1^{\mu_1} \cdots X_{\log n}^{\mu_{\log n}},$$

where the polynomials $F_\mu^{(n)}$ are obtained as a sum of at most $2^{u(n)}$ polynomials $G_\nu^{(n)}$. Thus, we now have a good hold on $F_\mu^{(n)}$, that is, $\deg F_\mu^{(n)} \le \alpha(n)$ and $\log \mathsf{wt}(F_\mu^{(n)}) \le 2^{\beta(n)}$, where both $\alpha(n)$ and $\beta(n)$ are polynomially bounded functions of n.

Thus, for f_n to be in VNP, the coefficients of f_n should be in the image of the polynomial map $F_\mu^n : \overline{\mathbb{Q}}^{q(n)} \to \overline{\mathbb{Q}}^{n^{\log n}}$. In other words, we must have some $\zeta \in \overline{\mathbb{Q}}^{q(n)}$, such that for all $\mu \in N_n$, we have $F_\mu^n(\zeta) = 2^{2^{j(\mu)}}$, where $j(\mu) := \sum_{j=1}^{\log n} \mu_j n^{j-1}$. Since F_μ^n takes all the values from 2^{2^0} to $2^{2^{n^{\log n}-1}}$, we have a subset of indices $\tilde{N}_n \subseteq N_n$ of size $s(n) := \lfloor |N_n|/n \rfloor = \lfloor n^{\log n}/n \rfloor$, such that for $\sigma \in \{0, 1, \ldots, s(n)-1\}$ and a bijection $\delta : \{0, 1, \ldots, s(n)-1\} \to \tilde{N}_n$ with $\sigma \mapsto \delta(\sigma)$, we have $F_{\delta(\sigma)}^n = 2^{2^{\sigma n+1}}$.

Now we can apply Lemma 9.28 from [46] which asserts that there will be polynomials of low height (ht) (the maximum of the absolute value of the coefficients) on which these coefficients shall vanish. More precisely, there exists non-zero forms $H_n \in \mathbb{Z}[Y_\mu \mid \mu \in \tilde{N}_n]$ with $\mathsf{ht}(H_n) \le 3$, $\deg H_n \le D(n)$, and such that $H_n(F_\mu^n \mid \mu \in N_n) = 0$, given that $D(n)^{s(n)-q(n)-2} > \alpha(n)^{q(n)} s(n)^{s(n)} 2^{\beta(n)}$.

It can be seen that $D(n) = 2^n - 1$ satisfies the above inequality, since $\alpha(n), \beta(n)$ and $q(n)$ are polynomially bounded and 2^n grows much faster than $s(n) = \lfloor n^{\log n}/n \rfloor$. This allows us to write $H_n = \sum_e \lambda_e \prod_{\mu \in \tilde{N}_n} Y_\mu^{e_\mu}$, where the absolute values of λ_e are bounded by 3. Since H_n vanishes on the subset of coefficients of f_n i.e it vanishes on $F_{\delta(\sigma)}^n = 2^{2^{\sigma n+1}}$ with $\sigma \in \{0, 1, \ldots, s(n)-1\}$, we have

$$0 = H_n(F_\mu^n \mid \mu \in \tilde{N}_n) = \sum_e \lambda_e \prod_{\sigma=0}^{s(n)-1} 2^{e_{\delta(\sigma)} 2^{\sigma n+1}} = \sum_e \lambda_e \cdot 4^{\sum_\sigma e_{\delta(\sigma)} (2^n)^\sigma}.$$

The last sum is essentially a 4-adic integer, since $|\lambda_e| \le 3$, and secondly, all the exponents of 4, that is, $\sum_\sigma e_{\delta(\sigma)} (2^n)^\sigma$ are all distinct, as they can be seen as 2^n-adic representation since $e_{\delta(\sigma)} < 2^n$. Thus λ_e has to be zero for all e. Hence H_n must be identically zero, which is a contradiction. □

3.2.1 A criterion for non-membership in $\overline{\mathsf{VNP}}$

In this section, we discuss a criterion Bürgisser presented in his monograph [44] based on a proof due to Heintz and Sieveking which gives a set of conditions that puts a p-family out of VNP. We observe that those conditions if satisfied, in fact, put a given p-family out of $\overline{\mathsf{VNP}}$ as well.

Theorem 3.2. *Let (p_n) be a sequence of polynomials over $\overline{\mathbb{Q}}$ and let $N(n)$ denote the degree of the field extension generated by the coefficients of p_n over $\mathbb{Q}$. Further suppose the following holds:*

(1) The map $n \mapsto \lceil \log N(n) \rceil$ is not p-bounded.

(2) For all n, there is a system G_n of rational polynomials of degree at most $D(n)$ with

finite zeroset, containing the coefficient system of f_n, and such that $n \mapsto \lceil \log D(n) \rceil$ is p-bounded.

Then the family $(p_n) \notin \overline{\mathsf{VNP}}$.

Thus the above theorem shows that certain p-families with algebraic coefficients of high degree are not contained in $\overline{\mathsf{VNP}}$.

For a proof of the theorem, we refer the readers to [44, Theorem 8.1]. We point out that the proof in its original form already works. In his proof, he wanted to conclude that $f_n \notin \mathsf{VNP}$. However, along the way, he arrives at a contradiction to the assertion that f_n is contained in the Zariski closure of VNP, which is exactly what is now known as $\overline{\mathsf{VNP}}$. During the time of the original proof, the complexity class $\overline{\mathsf{VNP}}$ was not defined.

We now give a simple example from [44] to illustrate the theorem.

Example 3.1. *Consider the following multivariate family defined as*

$$p_n = \sum_{e \in \{0,1\}^n \setminus 0} \sqrt{p_{j(e)}} X^e,$$

where $j(e) = \sum_{s=1}^{n} e_s 2^{s-1}$ and p_j refers to the j-th prime number. Then using the above Theorem 3.2, we can conclude that $p_n \notin \overline{\mathsf{VNP}}$. This is because the degree of field extension $N(n) = [\mathbb{Q}(\sqrt{p_j} \mid 1 \leq j \leq 2^n) : \mathbb{Q}] = 2^{2^n - 1}$ (see for e.g. [46, Lemma 9.20]), hence condition 1 above is satisfied. Condition 2 is also satisfied because the coefficients are the roots of the system $G_n = \{Z_j^2 - p_j \mid 1 \leq j < 2^n\}$, with $D(n) = 2$.

PART II

Easy varieties in algebraic complexity theory

This part is the result of close collaboration with Vishwas Bhargava, Markus Bläser, and Gorav Jindal. Chapter 4 is based on the article titled *Polynomial Identity Testing with Optimal Randomness*, with Markus Bläser, that appeared in *RANDOM,* 2020 [34]. Chapter 5 is based on the article titled *A Deterministic PTAS for the Algebraic Rank of Bounded Degree Polynomials*, with Vishwas Bhargava, Markus Bläser, and Gorav Jindal that appeared in *ACM-SIAM Symposium on Discrete Algorithms,* 2019 [23].

CHAPTER 4

Membership in the zero variety: Polynomial Identity Testing

In this chapter, we discuss one of the central problems in complexity theory, the polynomial identity testing problem. We are given an n-variate, degree d polynomial $P(x_1, \ldots, x_n)$ via blackbox access to its evaluations, and we have to decide if the polynomial is an identically zero polynomial, that is, when the polynomial is represented in the standard monomial-representation, are all its coefficients identically zero. This can be formulated as a problem of membership testing in the trivial variety, that is, the zero variety – the variety consisting of a single point, the origin. We are given a point $p \in \mathbb{F}^{\binom{n+d}{d}}$ implicitly, and we have to decide if the point is the origin. The point is given via a blackbox access to the evaluations of the n-variate, degree d polynomial $P(x_1, \ldots, x_n)$, whose coefficient vector precisely corresponds to the point p. Randomized polynomial time algorithms are known for this problem. The goal is to minimize the number of random bits needed. This is what we investigate in this chapter.

We give a randomized polynomial time algorithm for polynomial identity testing for the class of n-variate poynomials of degree bounded by d over a field $\mathbb{F}$, in the blackbox setting described above.

Our algorithm works for every field $\mathbb{F}$ with $|\mathbb{F}| \geq d + 1$, and uses only $d \log n + \log(1/\varepsilon) + O(d \log \log n)$ random bits to achieve a success probability $1 - \varepsilon$ for some $\varepsilon > 0$. In the low degree regime that is $d \ll n$, it hits the information theoretic lower bound and differs from it only in the lower order terms. Previous best known algorithms achieve the number of random bits (Guruswami and Xing [99] and Bshouty [42]) that are constant factor away from our bound. Like Bshouty, we use Sidon sets for our algorithm. However, we use a new construction of Sidon sets to achieve the improved bound.

We also collect two simple constructions of hitting sets with information theoretically optimal size against the class of n-variate, degree d polynomials. Our contribution is that we give new, very simple proofs for both the constructions.

4.1 Set-up and results

We investigate algorithms for the problem of Polynomial Identity testing (PIT). Given a polynomial in some implicit representation, it asks whether the polynomial is identically zero or not. It is a fundamental problem in algorithms and complexity theory. It has found applications in algorithm design, for example in algorithms for perfect matching in graphs [54, 133, 144], for primality testing [4, 5, 6], for equivalence testing of read once

branching programs [36], and for multi-set equality testing [37], and also in complexity theory, for example, in establishing some major results related to interactive proofs and probabilistically-checkable proofs [137, 18, 17, 16, 166]. In fact, it has also been discovered that a deterministic polynomial time algorithm for polynomial identity testing is intimately connected with complexity theoretic lower bounds [117, 3].

In order to formalize the algorithmic problem of polynomial identity testing, it is important to specify the representation in which the polynomial is given. One possibility is that the polynomial is given as a blackbox, which means that the algorithm is restricted to using the given representation of the given polynomial only as an oracle. That is, the algorithm is only allowed to query the values of the polynomial at points of its choice. Apart from that, the algorithm only knows that the given polynomial comes from some particular class of polynomials. The other possibility is that the algorithm is also allowed to look into the representation. In this case, if the polynomial is given as a list of coefficients, the problem becomes trivial. The problem remains interesting in the case when the polynomial is given in some succinct representation, for example, either as a determinant of a given symbolic matrix, as an algebraic branching program, or more generally, as some arithmetic circuit.

It is known that randomness is necessary for a polynomial time blackbox PIT algorithm (see for example [132]). The challenge thus in this case is to find polynomial time algorithms that use optimal amount of randomness. Randomness is not known to be essential in the setting when the polynomial is given as an arithmetic circuit. In fact, it is popularly believed that there do exist polynomial time algorithms for this version of PIT which do not use randomness. More generally, it is believed that in the regime of efficient computation, randomization is not essential, that is, the complexity classes P and BPP are equal (see [112]). In this case, the challenge is to come up with a deterministic algorithm. A lot of progress has happened over the years towards both the challenges [56, 132, 121, 29, 27, 97, 96, 74, 73, 71, 14, 28, 4, 125, 38, 136, 99], however the problems are still far from the complete solution. For a history on the progress on polynomial identity testing, we refer the readers to [170, 159, 160].

In this work, we are interested in blackbox polynomial identity testing. We will focus our attention to the case when the underlying field is a finite field. More precisely, we are interested in the following computational problem.

Problem 4.1. *Let $(\mathbb{F}_q, n, d)$ denote the class of multivariate polynomials over $\mathbb{F}_q$ in n variables with degree bounded by d [1] with $q \geq d+1$. Given a polynomial $p \in (\mathbb{F}_q, n, d)$ as a blackbox and a parameter $\varepsilon > 0$, decide whether p is an identically zero polynomial in randomized $poly(n, d)$ time with success probability $1 - \varepsilon$.*

We are interested in algorithms for Problem 4.1 which minimize the number of random bits needed to solve it. In the next subsection we discuss some previous works on the problem that are relevant to this chapter. While mentioning these works, we will assume the error bound ε to be some inverse polynomial in (nd), and we will focus only on algorithms that run in $poly(n, d)$ time under this assumption.

[1] in this chapter, unless stated otherwise, degree always refers to the total degree

4.1.1 Previous works on Problem 4.1

A lot of randomized algorithms are known for PIT in the blackbox setting. The first one is the algorithm due to Schwartz-Zippel-DeMillo-Lipton [2] [164, 184, 61]. It uses $\sum_{i=1}^{n} \log(d_i + 1) + n \log n + 1$ random bits, where d_i refers to the degree of the given polynomial with respect to the variable x_i. Then came the algorithm by Lewin and Vadhan [132] which used $\sum_{i=1}^{n} \lceil \log d_i \rceil$ random bits. Using the Kronecker substitution, Agrawal and Biswas [4] gave a test with $\lceil \sum_{i=1}^{n} \log(d_i + 1) \rceil$ random bits, while Bläser Hardt and Steurer [29] extended their Kronecker substitution based test to work for asymptotically smaller fields by using $\sum_{i=1}^{n} \log(d_i + 1) + \tilde{O}(\sqrt{\sum_{i=1}^{n} \log(d_i + 1)})$ random bits.

These works achieve optimal number of random bits in the regime where individual degrees of $x_1, \ldots, x_n$ are bounded by $d_1, \ldots, d_n$ respectively. In that regime, a simple dimension argument shows a lower bound of $\log(\prod_{i=1}^{n}(d_i+1)) - \log T(n, d_1, \ldots, d_n)$, where $T(n, d_1, \ldots, d_n)$ denotes the number of queries made to the blackbox [132, Theorem 7.1]. Thus, when $T(n, d_1, \ldots, d_n)$ is $poly(n)$ bounded, we get a $(1 - o(1)) \sum_{i=1}^{n} \log(d_i + 1)$ lower bound on the number of random bits needed. However, when we are in the setting as given in Problem 4.1, that is, when only a bound on the total degree is given, the number of random bits used by these methods are asymptotically similar to that of Schwartz-Zippel-DeMillo-Lipton, i.e., $\Omega(n \log d)$, which is far from optimal in the regime where $d \ll n$. In this regime, again using a simple dimension argument (see [132, Theorem 7.1] and Lemma 4.3), we have the following lower bound:

Fact 4.1. *Any blackbox identity testing algorithm against* $(\mathbb{F}_q, n, d)$, $q \geq d + 1$ *which makes* $T(n, d)$ *queries to the blackbox and succeeds with probability* $1 - \varepsilon$ *uses at least* $\log(\binom{n+d}{d}) + \log(1/\varepsilon) - \log T(n, d)$ *random bits.*

Applying Stirling's approximation on $\binom{n+d}{d}$ in the above when $d = o(n)$ gives $\log(\binom{n+d}{d}) = (1 + o(1))d \log(\frac{n+d}{d}) = d \log n + o(d \log n)$ [1]. Plugging this in above, with $T(n, d) = (nd)^{O(1)}$, we get the lower bound of $d \log n + \log(1/\varepsilon) + o(d \log n)$.[3]

Moving on to the previous works when $d \ll n$, several algorithms are known that actually do achieve the $O(d \log n)$ random bits. For instance, Klivans and Spielman [121], Bogdanov [38], Shpilka and Volkovich [169], Lu [136], Guruswami and Xing [99] and finally Bshouty [42] (also see Cohen and Ta-Shma [58]). However, except for [99] and [42], all of them require the field size to be superlinear in d/ε as a pre-condition for the algorithm. Moreover, in all of these algorithms including the ones in Bshouty [42] and Guruswami and Xing [99], the number of random bits used is $\geq 2d \log n$.

4.1.2 Results and methods

From the above, we can see that in the low-degree regime, the number of random bits needed by all the previously discovered algorithms, is away from the information theoretically optimal bound at least by a constant multiplicative factor. We take up the

[2] In their paper [61], DeMillo and Lipton work with the total degree. However, implicitly, the analysis of their algorithm only assumes the individual degrees to be bounded by d.

[3] this is what we refer to as the information theoretic lower bound in this article

challenge and solve it. We give an algorithm that matches the information theoretic lower bound differing from it only in the lower order terms.

More precisely, we show the following:

Theorem 4.1. *Given a polynomial $f \in (\mathbb{F}_q, n, d)$ with $q \geq d+1$ as a blackbox, and a parameter $\varepsilon > 0$, there exists a randomized poly(n,d) time algorithm which uses $d\log n + \log(1/\varepsilon) + O(d\log\log n)$ random bits and outputs whether f is an identically zero polynomial with success probability $1-\varepsilon$.*

Starting point of our algorithm is an algorithm given in Bshouty [42]. He used the so-called Sidon sets (discussed in Section 4.2.1) for polynomial identity testing by using them to reduce the problem to the univariate setting while preserving the nonzeroness. He then used the obvious randomized algorithm for the obtained univariate polynomial. This, however, requires the field-size to be large. He gets around this problem by inventing the concepts of testers (discussed in Section 4.2.2). Informally, testers take a point α from a field $\mathbb{F}$ and map it to a bunch of points in a smaller subfield of $\mathbb{F}$, while maintaining the property that if $f(\alpha) \neq 0$, then f will evaluate to a nonzero value on at least one of the points given by the tester.

He used two constructions for Sidon sets for this purpose. One of them is not known to be polynomial time constructible, while the other, which is polynomial time constructible is factor 2 away from the information theoretic lower bound. To overcome this, we use a new, elementary construction of Sidon sets that is mentioned in Timothy Gowers' weblog [89] (presented in Section 4.2.1).

Our second contribution is aesthetic in nature. We first remind the readers that a hitting set against a class $\mathcal{P} \subseteq \mathbb{F}[x_1,\ldots,x_n]$ is a set of point $\mathcal{H} \subseteq \mathbb{F}^n$ such that no nonzero polynomial in $\mathcal{P}$ evaluates to zero on all the points in $\mathcal{H}$. We present two simple constructions of information theoretically optimal hitting sets (i.e. of size $\binom{n+d}{d}$) against $(\mathbb{F}, n, d)$ with $|\mathbb{F}| \geq d+1$ that are, at least implicitly, present in the literature. We extract them out and give very simple and neat proofs for both. The first construction (presented in Section 4.3.1) is essentially the set of exponent vectors of all the monomials spanning $(\mathbb{F}, n, d)$. This works when $\{0,1,\ldots,d\} \subseteq \mathbb{F}$. The second construction (presented in Section 4.3.2) says that taking all the intersection points of n-sized subsets of a set of $n+d$ hyperplanes in general position also forms a hitting set against $(\mathbb{F}, n, d)$ of optimal size.

In the rest of the chapter, $(\mathbb{F}, n, d)$ (resp. $(\mathbb{F}_q, n, d)$) denotes the class of n-variate polynomials with degree bounded by d over $\mathbb{F}$ (resp. $\mathbb{F}_q$). For a natural number $d \in \mathbb{N}$, $[d]$ denotes the set $\{1,\ldots,d\}$.

4.2 Polynomial Identity Testing with optimal randomness

In this section, we present our main result. We first describe the main component of the proof, that is, the construction of Sidon sets in Section 4.2.1, and then describe the way to reduce the field size in Section 4.2.2. We finally give our algorithm and the proof for our main theorem in Section 4.2.3.

4.2.1 Sidon Sets

A set $\mathcal{S} := \{s_1, s_2, \ldots, s_n\} \subset \mathbb{Z}_{\geq 0}$ is said to be a Sidon B_d set if every element in the set $d\mathcal{S} := \{s_{i_1} + s_{i_2} + \ldots + s_{i_d} \mid \forall k \in [d], s_{i_k} \in \mathcal{S}\}$ are distinct up to rearrangements of the summands. We also have a stronger notion: we call $\mathcal{S}$ to be Sidon $B_{\leq d}$ set if the sums $\{s_{i_1} + s_{i_2} + \ldots + s_{i_r},\ r \leq d \mid \forall k \in [r], s_{i_k} \in \mathcal{S}\}$ are distinct up to rearrangements of the summands. For our purposes, the stronger notion of Sidon $B_{\leq d}$ set when $d \ll n$ will be useful. We are interested in constructions that minimize the size of the maximum element of $\mathcal{S}$ and are $poly(n, d)$ time constructible.

Sidon sets and its variants have a long history in mathematics and several explicit constructions are known. We refer the readers to a survey by Kevin O'Bryant [148].

In complexity theory, explicit Sidon set constructions have also been used, for example, by Bshouty for constructions of hitting sets for black box polynomial identity testing [42], and by Kumar and Volk for matrix factorization lower bounds [127].

Bshouty uses two constructions for polynomial identity testing. The first construction uses discrete log and is not known to be poly time constructible [42, Lemma 59]. The second construction is poly time constructible, but the value of the maximum element is $(2nd)^{2d}$ [42, Lemma 60]. This $2d$ in the exponent makes this construction suboptimal for our purposes because the resulting randomized PIT algorithm will have $\geq 2d \log n$ random bits, which is factor 2 away from the information theoretic bound in low degree regime which is the regime of interest in this chapter.

This motivated us to look for constructions that are both polynomial time constructible and also give rises to PIT algorithm with optimal randomness. That is when we stumbled across the weblog of Timothy Gowers about the so-called dense Sidon sets [89] where he describes the idea of a construction by Imre Z. Ruzsa [156] that scales up for our purposes, too.

In its core, the construction is based on the fundamental theorem of arithmetic. Informally, when we take a set of primes and consider two different multi-subsets of them. Then the product of elements will be different for the two multi-subsets. Now taking logarithm of products convert them to sums. These simple facts along with the mean value theorem constitute the ingredients of the proof of the construction. We give the construction now.

Theorem 4.2. *For every n, d, there exists a $poly(n, d)$ time constructible Sidon $B_{\leq d}$ set $S_{n,d} := \{b_1, \ldots, b_n\} \subset \mathbb{N}$ with $b_1 < b_2 < \cdots < b_n$, with $b_n \leq \lceil (d+1) \cdot (2n \log n)^d \cdot \log(2n \log n) \rceil$.*

Proof. We take the first n primes $p_1, \ldots, p_n$. By prime number theorem, we know that $p_n < n(\log n + \log\log n) < 2n \log n$. Let $I, J \subseteq \{1, \ldots, n\}$ be multisets, where $|I|, |J| \leq d$, $I \neq J$. By the fundamental theorem of arithmetic, we have $\prod_{i \in I} p_i \neq \prod_{j \in J} p_j$. Without loss of generality, we can assume that $\prod_{i \in I} p_i < \prod_{j \in J} p_j$, that is, $\prod_{i \in I} p_i \leq \prod_{j \in J} p_j + 1$. Now applying the mean value theorem on the function $f(x) = \log x$ in the interval $[a, b]$ with $a := \prod_{i \in I} p_i$ and $b := \prod_{j \in J} p_j$, we get that

$$\sum_{j \in J} \ell_j - \sum_{i \in I} \ell_i = \frac{1}{c} \left(\prod_{j \in J} p_j - \prod_{i \in I} p_i \right), \text{ for some } c \in (a, b), \text{where } \ell_k := \log p_k.$$

The numerator in the RHS of the equation is at least 1, while the denominator is upper bounded by $b = \prod_{j\in J} p_j$. Thus, we have

$$\sum_{j\in J} \ell_j - \sum_{i\in I} \ell_i \geq \frac{1}{\prod_{j\in J} p_j}. \tag{4.2.1}$$

Thus, if we choose the set to be set of logarithm of the first n primes, we do get, that for distinct multi-subsets of size at most d, the sum of elements are also distinct. However, clearly, the elements and their differences will not be all integers. But the above calculation is suggestive of what the set should be. Note that in Equation (4.2.1), the denominator of the RHS, that is, $\prod_{j\in J} p_j$ is upper bounded by $(2n\log n)^d$, Thus,

$$\sum_{j\in J}(d+1)\cdot \ell_j \cdot (2n\log n)^d - \sum_{i\in I}(d+1)\cdot \ell_i \cdot (2n\log n)^d \geq d+1$$

Now, if we consider the set $\mathcal{S}_{n,d}$ of size n with elements being positive integers $b_k := \lceil (d+1)\cdot \ell_k \cdot (2n\log n)^d \rceil$ of size n, we have that $\sum_{j\in J} b_j - \sum_{i\in I} b_i > 0$. Thus, $\mathcal{S}_{n,d}$ is a Sidon $B_{\leq d}$ set.

It only remains to argue that the construction can be done in $poly(n,d)$ time. We need to show that all the $b_k = \lceil (d+1)\cdot \log p_k \cdot (2n\log n)^d \rceil$ are $poly(n,d)$ time constructible, It is known that the first n primes are easily constructible, for example, by using Sieve of Eratosthenes which takes $O(n\log\log n)$ time. The other functions like log and powering function are also known to be efficiently computable to the desired precision. □

We now present the concept useful for transferring a polynomial identity testing algorithm over a large field to an algorithm for a small subfield of it.

4.2.2 Testers

The notion of testers is also crucial for our algorithm. They were introduced by Bshouty in [42]. He also used it for several applications including in the setting of blackbox PIT. We will be using it in the same fashion as he did i.e. to reduce the field size of the blackbox PIT set that we would be using for the algorithm. We present the definition of testers restricted to the setting that we need. He defined it for a more general setting.

Definition 4.1. *Let $\mathbb{F}_q$ be a finite field with q elements and let $\mathbb{F}_{q^{t_1}}$ and $\mathbb{F}_{q^{t_2}}$ be two field extensions of $\mathbb{F}_q$ viewed as $\mathbb{F}_q$-algebras with $t_1 \geq t_2$, and let $\mathcal{P} \subseteq \mathbb{F}_q[x_1,\ldots,x_n]$ be a class of multivariate polynomials. Let $L = \{\ell_1,\ldots,\ell_\nu\}$ be a set of maps $\mathbb{F}^n_{q^{t_1}} \to \mathbb{F}^n_{q^{t_2}}$. For $f \in \mathcal{P}$, we denote by fL the map $\mathbb{F}^n_{q^{t_1}} \to \mathbb{F}^\nu_{q^{t_2}}$ defined as: for $\boldsymbol{a} \in \mathbb{F}^n_{q^{t_1}}$, $(fL)(\boldsymbol{a}) = (f(\ell_1(\boldsymbol{a})),\ldots,f(\ell_\nu(\boldsymbol{a})))$. We say that L is an $(\mathcal{P},\mathbb{F}_{q^{t_1}},\mathbb{F}_{q^{t_2}})$-tester if for every $\boldsymbol{a} \in \mathbb{F}^n_{q^{t_1}}$ and $f \in \mathcal{P}$ we have*

$$(fL)(\boldsymbol{a}) = \boldsymbol{0} \implies f(\boldsymbol{a}) = 0.$$

The size of the tester L is defined as $|L| = \nu$, the number of maps constituting L.

So, essentially, a tester L for the class of polynomials $\mathcal{P}$ is a set of maps from a field to its subfield such that for every point on which a polynomial $f \in \mathcal{P}$ evaluates to a nonzero value, the tester gives a set of points in the subfield such that the polynomial evaluates to a nonzero value on at least one of the points given by the tester.

Hence a tester is very useful for reducing a blacbox PIT set over a bigger field to a blackbox PIT set over a smaller field while incurring a blowup by the size of the tester. Bshouty [42] also gave many constructions of testers against several classes of multivariate polynomials which helped him achieve constructions of hitting sets which are optimal with respect to the field size and the size of hitting sets.

The tester that is relevant to our purposes which we will be using as a blackbox has the following property. For a proof we refer the readers to [42].

Lemma 4.1 ([42], Theorem 40)**.** *Let $\mathcal{P} := (\mathbb{F}_q, n, d) \subseteq \mathbb{F}_q[x_1, \ldots, x_n]$ denote the class of n-variate, degree d polynomials over $\mathbb{F}_q$, with $q \geq d+1$. Then, for every n, d, t, there exists a $(\mathcal{P}, \mathbb{F}_{q^t}, \mathbb{F}_q)$-tester L of size $O(d^5 t)$ that can be constructed in $poly(n, d, t)$ time.*

The above lemma clearly suggests a strategy for construction of blackbox PIT sets: first design a blackbox PIT set over a large extension field and then reduce the field size to $d+1$ using the tester promised by the above lemma.

4.2.3 The algorithm: Proof of Theorem 4.1

In this section, we present our randomized algorithm for polynomial identity testing and prove Theorem 4.1.

Before we prove the theorem, we state a simple lemma about univariate polynomials that we will need in the proof.

Lemma 4.2. *Let $f \in \mathbb{F}_q[x]$ be a nonzero univariate polynomial whose degree is bounded by d. Let $\mathbb{F}_{q^t}$ be an extension field of $\mathbb{F}_q$ such that $|\mathbb{F}_{q^t}| \geq d/\varepsilon$ and $\boldsymbol{a}$ is sampled uniformly at random from $\mathbb{F}_{q^t}$, then $f(\boldsymbol{a}) \neq 0$ with probability $1-\varepsilon$.*

Lemma 4.2 follows from the folklore theorem that a univariate polynomial of degree d over a field $\mathbb{F}_q$ has at most d roots in any field extension $\mathbb{F}_{q^t}$ of $\mathbb{F}_q$.

We are now ready to prove Theorem 4.1.

Theorem 4.3 (Theorem 1 restated)**.** *Given a polynomial $f \in (\mathbb{F}_q, n, d)$ with $q \geq d+1$ as a blackbox or as a $poly(n, d)$-sized arithmetic circuit, and $\varepsilon > 0$, there exists a randomized $poly(n, d)$ time algorithm which uses $d \log n + \log(1/\varepsilon) + O(d \log\log n)$ random bits and succeeds with probability $1-\varepsilon$.*

Proof. Suppose we are given a polynomial $f \in (\mathbb{F}_q, n, d)$ as a blackbox. To test whether the given polynomial is an identically zero polynomial or not, our algorithm works as follows:

Step 1. Construct Sidon set: Given n, d, we construct a Sidon $B_{\leq d}$ set $\mathcal{S}_{n,d} = \{b_1, \ldots, b_n\}$ using the construction in Theorem 4.2.

Step 2. Pick a random point from large field: We pick a random point α from the field $\mathbb{F}_{q^t}$ with $t = \lceil \log_q((b_n d)/\varepsilon) \rceil$.

Step 3. Construct the tester: Next we construct a $(\mathcal{P}, \mathbb{F}_{q^t}, \mathbb{F}_q)$-tester, $L = \{\ell_1, \ldots, \ell_\nu\}$, for $\mathcal{P} = (\mathbb{F}_q, n, d)$ and $t = \lceil \log_q((b_n d)/\varepsilon) \rceil$ using the construction promised by Lemma 4.1.

Step 4. Reduce the field size by testers: We then apply the tester L on $(\alpha^{b_1}, \ldots, \alpha^{b_n}) \in \mathbb{F}_{q^t}^n$ to get the set of points $\ell_1(\alpha^{b_1}, \ldots, \alpha^{b_n}), \ldots, \ell_\nu(\alpha^{b_1}, \ldots, \alpha^{b_n}) \in \mathbb{F}_q^n$.

Step 5. Evaluate: We evaluate f on $\ell_1(\alpha^{b_1}, \ldots, \alpha^{b_n}), \ldots, \ell_\nu(\alpha^{b_1}, \ldots, \alpha^{b_n}) \in \mathbb{F}_q^n$. If f evaluates to zero on $\ell_k(\alpha^{b_1}, \ldots, \alpha^{b_n})$, for every $k \in 1, \ldots, \nu$, we output that f is an identically zero polynomial. Otherwise we output that f is not a zero polynomial.

We now show the correctness of the above algorithm. The Sidon $B_{\leq d}$ set $\mathcal{S}_{n,d} = \{b_1, \ldots, b_n\}$, $b_1 < b_2 < \cdots < b_n$ and $b_n = \lceil (d+1) \cdot (2n \log n)^d \cdot \log(2n \log n) \rceil$ from Theorem 4.2 is used to reduce the problem to the univariate case. It is also $poly(n, d)$ time constructible. By the definition of Sidon $B_{\leq d}$ set in Section 4.2.1, it follows that for distinct multi-subsets of $\mathcal{S}_{n,d}$, the sum of elements will also be distinct. Thus, the map $(x_1, x_2, \ldots, x_n) \mapsto (x^{b_1}, x^{b_2}, \ldots, x^{b_n})$ maps the monomials of the degree at most d in the variables $x_1, \ldots, x_n$ to distinct univariate monomials in x. In particular, every nonzero polynomial $f \in (\mathbb{F}_q, n, d)$ maps to a nonzero polynomial $g \in (\mathbb{F}_q, 1, b_n d)$. Thus, g is a polynomial of degree bounded by $b_n d$.

Now, by Lemma 4.2, on a randomly chosen point α from the extension field $\mathbb{F}_{q^t}$ with $|\mathbb{F}_{q^t}| \geq (b_n d)/\varepsilon$, g will evaluate to a nonzero value with probability $\geq 1 - \varepsilon$. Hence, f will evaluate to a non-zero value on $(\alpha^{b_1}, \ldots, \alpha^{b_n})$ with probability $\geq 1 - \varepsilon$. The number of random bits needed is $\log((b_n d)/\varepsilon) = \log b_n + \log d + \log(1/\varepsilon) = d \log n + O(d \log \log n) + \log(1/\varepsilon)$ as claimed.

Finally we use an $((\mathbb{F}_q, n, d), \mathbb{F}_{q^t}, \mathbb{F}_q)$-tester from Lemma 4.1 on $(\alpha^{b_1}, \ldots, \alpha^{b_n}) \in \mathbb{F}_{q^t}^n$ to get the set of points $\ell_1(\alpha^{b_1}, \ldots, \alpha^{b_n}), \ldots, \ell_\nu(\alpha^{b_1}, \ldots, \alpha^{b_n}) \in \mathbb{F}_q^n$. By Lemma 4.1, the number of points $\nu = O(d^5 t) = O(d^5(\log(b_n d) + \log(1/\varepsilon))) = O(d^6(\log n + \log 1/\varepsilon))$ and time to construct the tester is $poly(n, d, t) = poly(n, d, \log 1/\varepsilon)$. By the definition of testers from Definition 4.1, for a nonzero polynomial $f \in (\mathbb{F}_q, n, d)$, if $f(\alpha^{b_1}, \ldots, \alpha^{b_n}) \neq 0$, then at least one of $f(\ell_1(\alpha^{b_1}, \ldots, \alpha^{b_n})), \ldots, f(\ell_\nu(\alpha^{b_1}, \ldots, \alpha^{b_n}))$ also evaluates to a nonzero value. Thus, we get the desired result. □

Remark 4.1. *When $d = o(\log n)$, we can get an improvement on the number of random bits from $d \log n + d \log \log n +$ lower order terms (as in Theorem 4.1) to $d \log n + d \log d +$ lower order terms. This can be achieved by adapting an idea due to Goldwasser-Grossman [87, Lemma 8] that they used to construct weight assignments to get unique min-weight perfect matching. Their idea suggests a map $(x_1, x_2, \ldots, x_n) \mapsto (x^{b_1}, x^{b_2}, \ldots, x^{b_n})$, where $b_i = [i]_p + (pd)[i^2]_p + (pd)^2[i^3]_p + \cdots + (pd)^d[i^{d+1}]_p$, where $[x]_p$ denotes the number between 0 and $p-1$ which is equal to x modulo p, and p is a prime number greater than n. This map can replace the map via Sidon Sets in the Step 1 from the above proof, while the rest of the algorithm and the proof remains the same. As in the above proof of Theorem 4.1, the number of random bits needed equals $\log b_n + \log d + \log(\frac{1}{\varepsilon})$ which becomes $d \log n + d \log d +$ lower order terms. For the correctness of this map, we refer the reader to the proof of* Lemma 8 *in* [87].

Remark 4.2. *Our algorithm can be easily adapted to fields of zero characteristic as well. First note that directly evaluating f on $(\alpha^{b_1}, \ldots, \alpha^{b_n})$ will not work as the input numbers will become huge. For this reason, for a polynomial f which is non-zero over a field $\mathbb{F}$ of characteristic zero, we consider it over some field $\mathbb{F}_q$ of characteristic p with $q \geq d + 1$.*

Now, there are only two possibilities — either f is not identically zero over $\mathbb{F}_q$ as well, or there exists a natural number ℓ such that $f = p^\ell f'$, where f' in not identically zero over $\mathbb{F}_q$. In the first case, we can simply use the algorithm for $\mathbb{F}_q$, and the analysis also remains the same. In the second case, we can still use the same algorithm as that for $\mathbb{F}_q$ since the algorithm correctly detects the non-zeroness of f' with high probability. The fact that all the coefficients are divisible by p^ℓ will not be a problem since the blackbox is over the original field $\mathbb{F}$.

4.3 Optimal Hitting sets

In this section, we describe a few optimal hitting sets, i.e. the ones that exactly matches with the lower bound against the class of n-variate polynomials with total degree bounded by d.

We first begin by stating the straight-forward folklore lower bound.

Lemma 4.3. *For any hitting set $\mathcal{H}$ against the class of n-variate polynomials with total degree bounded by d over a field $\mathbb{F}$, we have $|\mathcal{H}| \geq \binom{n+d}{d}$.*

This follows by the fact that the set of all n-variate polynomials with total degree bounded by d over a field $\mathbb{F}$ forms a vector space of dimension $\binom{n+d}{d}$. This is true because the number of monomials supported on n-variables with total degree bounded by d is $\binom{n+d}{d}$, and they form the basis for the vector space as they are all $\mathbb{F}$-linearly independent, and all polynomials in the set can be represented as an $\mathbb{F}$-linear combination of them. Thus, in the worst case, one needs to query f on at least $\binom{n+d}{d}$ points.

We now consider a very popular construction which is suboptimal for our purposes.

Construction 0 – Schwartz-Zippel-DeMillo-Lipton lemma: As a consequence of the Lemma [164, 184, 61], for $(\mathbb{F}, n, d)$, one gets the hitting set $\mathcal{H}_0 := S^n$ where $S \subseteq \mathbb{F}$, with $|S| = d + 1$, which is the grid of size $(d+1)^n$. We point out that this is optimal when we are considering the set of n-variate polynomials with *individual degrees* of each variable bounded by d, by a similar argument as in Lemma 4.3. However, this is not optimal for $(\mathbb{F}, n, d)$ where we bound the *total degree*. Especially when $d \ll n$, the gap is huge.

Thus, it makes sense to investigate optimal hitting sets for $(\mathbb{F}, n, d)$. In what follows, we present two such constructions for the hitting set. They were both, at least implicitly, already present in the literature. We also believe that other constructions can be found without much effort. However our predilection towards these constructions and our new proofs is purely aesthetic.

4.3.1 Construction 1: The set of exponent vectors

The lower bound tells that any hitting set $\mathcal{H}$ should have size at least $\binom{n+d}{d}$. Now this $\binom{n+d}{d}$ comes from the number of monomials in n variables of degree at most d. Very interestingly, when $\{0, 1, \ldots, d\} \subseteq \mathbb{F}$, these monomials also suggest a set of points of size $\binom{n+d}{d}$ that can be seen as a potential hitting set: simply collect all the exponent vectors

of all the monomials in a set, viewing them as points in $\mathbb{F}^n$, that is, the suggested set is $\mathcal{H}_1 := \{(x_1, \ldots, x_n) \in \{0, 1, \ldots, d\}^n \mid x_1 + x_2 + \cdots + x_n \leq d\}$. The above construction obviously requires that $\mathbb{F}^n$ indeed contains $\mathcal{H}_1$ as a subset. Surprisingly to some, and beautifully to the authors, this indeed works. This was shown by Bshouty [42, Lemma 77] using the combinatorial Nullstellensatz [12].

In this work, we give a direct inductive proof which we found with Mrinal Kumar. It bypasses the combinatorial Nullstellensatz and flows along the lines of the proof of the Schwartz-Zippel-DeMillo-Lipton lemma. We are surprised that we did not find this proof somewhere in the literature.

Theorem 4.4. *If $\{0, 1, \ldots, d\} \subseteq \mathbb{F}$, then the set $\mathcal{H}_1 := \{(x_1, \ldots, x_n) \in \{0, 1, \ldots, d\}^n \mid x_1 + x_2 + \cdots + x_n \leq d\}$, is a hitting set for $(\mathbb{F}, n, d)$.*

Proof. Consider the integral grid $\mathcal{G} := \{0, 1, \ldots d\}^n$ with $|\mathcal{G}| = (d+1)^n$. Now the statement of the theorem can be rephrased as: for every nonzero polynomial $f \in (\mathbb{F}, n, d)$, there exists a point $g \in \mathcal{G}$ with its ℓ_1-norm $\|g\|_1 \leq d$, such that $f(g) \neq 0$. We use this as our induction hypothesis and prove it by induction on the number of variables n.

For $n = 1$, that is the univariate case, this holds true because every degree d polynomial has at most d zeros. Suppose the hypothesis holds for $n-1$. For the inductive step, write a nonzero $f \in (\mathbb{F}, n, d)$ as a univariate in x_n as $f = \sum_{i=1}^{d'} P_i(x_1, \ldots, x_{n-1}) x_n^i$, where d' is the maximum degree of f in x_n and $P_i(x_1, \ldots, x_{n-1})$ are the coefficients coming from $\mathbb{F}[x_1, \ldots, x_{n-1}]$. Now consider $P_{d'}(x_1, \ldots, x_{n-1})$ which is the coefficient of the highest degree term in x_n. If f is a nonzero polynomial, then so is $P_{d'}(x_1, \ldots, x_{n-1})$. Also, $\deg(P_{d'})$ is bounded by $d - d'$. By the induction hypothesis, there is a point s' in the grid $\mathcal{G}' := \{0, 1, \ldots, d - d'\}^{n-1}$ with $\|s'\|_1 \leq d - d'$ such that $P_{d'}(s') \neq 0$. Now we fix $x_1, \ldots, x_{n-1}$ to the values as given by s'. Now $P_{d'}$ restricted to the assignment s' is a univariate polynomial in x_n of degree d'. Thus, setting x_n to a value in $\{0, 1, \ldots d'\}$ gives a point on which f evaluates to nonzero. The ℓ_1 norm of the point is at most $(d - d') + d' = d$. □

We now give another construction which we find beautiful.

4.3.2 Construction 2: Intersection of hyperplanes in general position

The construction is as follows: In the projective space $\mathbb{P}^n(\mathbb{F})$ over a field $\mathbb{F}$, with $|\mathbb{F}| \geq d+1$, take a collection $\mathcal{C}$ of $n + d$ hyperplanes in general position, i.e., every size n subsets of $\mathcal{C}$ intersect in a point, whereas no size $n + 1$ subset of $\mathcal{C}$ intersect. Now all intersection points of n-sized subset of $\mathcal{C}$ gives the desired hitting set.

We now mention a standard explicit family of hyperplanes in general position.

Example 4.1. *Take hyperplanes $H_1, \ldots, H_{n+d}$ in the projective space $\mathbb{P}^n$ where H_i is given by the equation $t_i^1 x_1 + t_i^2 x_2 + \cdots + t_i^n x_n + x_{n+1} = 0$, where $t_i \in \mathbb{F}$. Then, $H_1, \ldots, H_{n+d}$ are hyperplanes in general position.*

Itaï Ben Yaacov [182] considers hyperplanes in general position and gives an algebraic proof of a generalized Vandermonde Identity in higher dimension. What his identity

implies is that taking all intersection points of n-sized subsets of $n+d$ hyperplanes in general position gives a hitting set for $(\mathbb{F}, n, d)$.

In order to state his result, we need some notations. Let $M_{p\times q}(\mathbb{F})$ denote the set of all $p \times q$ matrices over $\mathbb{F}$. He defines the following three maps. It useful to think that the $(n+1) \times m$ matrix Q is denoting the family of m hyperplanes, say $\mathcal{H}_m$ in $\mathbb{P}^n$, i.e. the entries of each column correspond to the coefficients of a hyperplane.

(1) $\mu : M_{(n+1)\times m}(\mathbb{F}) \to M_{(n+1)\times\binom{m}{n}}(\mathbb{F})$ sends an $(n+1) \times m$ matrix Q to a matrix P whose entries are all the minors of Q of order n. Note that a lexicographic ordering on the chosen sequence of rows and columns of Q induces an ordering on the minors as well. By Cramer's rule from linear algebra, P is precisely the matrix of intersection points of all n-sized subsets of $\mathcal{H}_m$, where each column of Q has the coordinates of an intersection point as its entries.

(2) $\delta : M_{(n+1)\times m}(\mathbb{F}) \to \mathbb{F}$ sends an $(n+1)\times m$ matrix Q to the product of all its minors of order $n+1$.

(3) $\nu_r : M_{(n+1)\times m}(\mathbb{F}) \to M_{\binom{n+r}{n}\times m}(\mathbb{F})$ applies the Veronese map on each column i.e. for each column, it applies all the n-variate degree r monomials on the entries of the column. Assume an ordering (say, inverse lexicographical ordering) on the monomials.

We are now ready to quote the generalized Vandermonde identity in higher dimension.

Theorem 4.5 ([182], Theorem 1.4)**.** *Let R be a commutative ring. $n \le m \in \mathbb{N}$, and let Q be a $(n+1)\times m$ matrix over R. Then $\nu_{m-n}\mu Q$ is a square matrix of order $\binom{m}{n}$, and the following Vandermonde identity of order m in dimension n holds:*

$$\det(\nu_{m-n}\mu Q) = (\delta Q)^n$$

Then the above theorem with $m = n+d$, implies that if $(\delta Q) \neq 0$, which is the algebraic condition for the $n+d$ hyperplanes to be in general position, then $\det(\nu_{n+d-n}\mu Q) \neq 0$. Now $\nu_{m-n}\mu Q$ is the matrix with the Veronese map applied on the intersection points of n-sized subsets of m hyperplanes. Normalizing the coordinates by the last coordinate gives us the points in the affine setting with the Veronese maps essentially applying all the monomials of degree at most d on the points. Thus, $\det(\nu_d\mu Q) \neq 0$ means that the set of intersection points form a hitting set against $(\mathbb{F}, n, d)$.

We now present a direct, simple, geometric proof of the construction which we found with Raimund Seidel.

Theorem 4.6. *Let $H_1, \ldots, H_{n+d}$ be hyperplanes in general position. If $f \in (\mathbb{F}, n, d)$ vanishes on all the points obtained by intersecting all n-sized subsets of $\{H_1, \ldots, H_{n+d}\}$, then f is an identically zero polynomial.*

Proof. We prove the above statement by induction on the number of variables n. The base case $n = 1$ is the univariate case and the hyperplanes become $d+1$ points, and the statement of the lemma reduces to f vanishing on $d+1$ points. Thus, the statement

holds in this case because a degree d univariate polynomial that vanishes on $d+1$ points is an identically zero polynomial.

Suppose that the statement holds for the number of variables up to $n-1$, and we assume an $f \in (\mathbb{F}, n, d)$ that vanishes on all the intersection points of n-sized subsets of $\{H_1, \ldots, H_{n+d}\}$. The assumption, in particular, implies that f restricted to the hyperplane H_1 vanishes on all the intersection points of $(n-1)$-sized subsets of $\{H_2, \ldots, H_{n+d}\}$. However, note that f restricted to H_1 reduces to an $(n-1)$-variate case and hence we can apply the induction hypothesis and conclude that f restricted to H_1 is identically zero. Doing the same for all the hyperplanes, we get that f restricted to all the hyperplanes $H_1, \ldots, H_{n+d}$ is identically zero. It remains to conclude that f is indeed identically zero. For this, restrict f to a generic line ℓ. Note that $H_1, \ldots, H_{n+d}$ all intersect ℓ at distinct points. Thus, f restricted to ℓ is a univariate which vanishes on $n+d$ points, hence f restricted to a generic ℓ is identically zero. Hence f is identically zero. □

Note that an explicit construction corresponding to Theorem 4.6 can be obtained using the family given in Example 4.1.

CHAPTER 5

The variety of bounded rank symbolic matrices

In this chapter, we consider the rank of symbolic matrices. It is a well known fact that the rank of an $n \times n$ matrix M, $\mathrm{rk}(M) \leq r$ if and only if, all the $(r+1) \times (r+1)$ minors of M are zero. A minor is nothing but a determinant, which is a polynomial in the entries of the matrix. Thus, we get that the set $\mathcal{M}_r := \{M \in \mathbb{F}^{n \times n} \mid \mathrm{rk}(M) \leq r\}$, that is, the set of matrices of rank at most r, is the set of common solutions to a system of polynomial equations, and is hence a variety. Hence, testing if a given matrix M has rank at most r, is an instance of variety membership testing problem. In this chapter, we discuss the problem of finding the rank of a symbolic matrix M, which asks for the largest r such that $M \in \mathcal{M}_r$. For this problem, randomized polynomial time algorithms are known, and the challenge is to come up with a deterministic polynomial time algorithm. We give a a deterministic polynomial time approximation scheme (PTAS) for computing the rank of symbolic matrices when the entries of matrices are polynomials whose degrees are bounded by a constant.

More specifically, consider a matrix $Q(x_1, x_2, \ldots, x_m) = (q_{ij})_{n \times n}$ of size $n \times n$, the entries q_{ij} of which are polynomials of degrees bounded by some constant d in the variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. We want to compute the rank of Q over the rational function field $\mathbb{F}(x_1, x_2, \ldots, x_m)$.

We give an algorithm that takes as input a matrix Q as above over a field $\mathbb{F}$ with $|\mathbb{F}| > nd$ and a constant $0 < \varepsilon < 1$, and computes an assignment $(\lambda_1, \lambda_2, \ldots, \lambda_m) \in \mathbb{F}^m$ such that,

$$\mathrm{rk}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) \geq (1 - \varepsilon)\,\mathrm{rk}(Q(x_1, \ldots, x_m)).$$

Our key contribution is a new technique which allows us to achieve the higher degree generalization of the results by Bläser, Jindal, and Pandey [33] who gave a deterministic PTAS for computing the rank of a matrix with homogeneous linear entries. It is known that a deterministic algorithm for exactly computing the rank in the linear case is already equivalent to the celebrated Polynomial Identity Testing (PIT) problem which itself would imply circuit complexity lower bounds (Kabanets and Impagliazzo [117]).

Such a higher degree generalization is already known to a much stronger extent in the non-commutative world, where the more general case in which the entries of the matrix are given by poly-sized formulas reduces to the case where the entries are given by linear polynomials using Higman's trick, and in the latter case, one can also compute the exact rank in polynomial time (Garg, Gurvits, Oliviera, Wigderson [84], Ivanyos, Qiao, Subrahmanyam [114]). Higman's trick only preserves the co-rank, hence it cannot be

used to reduce the problem of rank approximation to the case when the matrix entries are linear polynomials. Thus the result in this chapter can also be seen as a step towards bridging the knowledge gap between the non-commutative world and the commutative world.

5.1 Set-up and results

This chapter is a result of an exploration of three related fundamental themes in algebra from a computational perspective - *polynomial identities*, *algebraic dependence of polynomials* and *rank of symbolic matrices*. These are already known to be crucial in several aspects of the theory of computation. We now give a brief overview of these three themes[1], the corresponding naturally arising computational problems, the interrelations among them and their applications to theoretical computer science.

5.1.1 Polynomial identity testing

Polynomial identities are useful and ubiquitous in mathematics and computer science. Simple examples include the *two square identity*: $(a^2+b^2)(x^2+y^2) = (ax-by)^2+(bx+ay)^2$, which can be used to show that the distance is invariant under a rotation of axes. Similarly the identity $\sum_{1\le i<j\le 4}(x_i+x_j)^4+(x_i-x_j)^2 = 6(x_1^2+x_2^2+x_3^2+x_4^2)^2$ was used by Liouville to show that every positive integer is a sum of at most 53 fourth powers of integers. More recently, even an identity as simple as $2xy = (x+y)^2-x^2-y^2$ was crucial in demonstrating the power of an approximative model of computation in algebraic complexity theory [41, 123]. Having an awareness of the pervasiveness of polynomial identities in mathematics, perhaps one would not be surprised to discover the extent of applications of the computational problem of testing if a given compact representation of a polynomial (arithmetic circuit) is indeed computing the identically zero polynomial. The *Polynomial Identity Testing* (PIT) question asks that given an arithmetic circuit C computing a polynomial $f \in \mathbb{F}[x_1,\ldots,x_m]$, test if f is the zero polynomial. The PIT problem captures several problems in algebra and combinatorics. For example, its special case captures the *perfect matching* problem via the *Tutte Matrix* [178, 133]. The breakthrough *primality testing* algorithm by Agrawal, Kayal and Saxena [5, 6] also reduced the primality testing problem to a special case of the PIT problem. PIT was also crucial in Shamir's proof of $\mathsf{IP} = \mathsf{PSPACE}$ [166]. Kabanets and Impagliazzo showed that a deterministic polynomial time algorithm would imply circuit complexity lower bounds, i.e., either $\mathsf{NEXP} \not\subset \mathsf{P/Poly}$ or the permanent does not have polynomial sized arithmetic circuits [117].

5.1.2 Algebraic dependence of polynomials

Algebraic dependence is a fundamental concept in algebra that captures polynomial relationships among polynomials. Polynomials $f_1,\ldots,f_m \in \mathbb{F}[x_1,\ldots,x_n]$ are called *algebraically dependent* over a field $\mathbb{F}$ if there exists a non-zero polynomial $A(y_1,\ldots,y_m) \in \mathbb{F}[y_1,\ldots,y_m]$ such that $A(f_1,\ldots,f_m) = 0$. Such an A is called an *annihilating polynomial*

[1] Even though we discussed Polynomial identity testing in the previous chapter in detail, we describe it here briefly again, to keep the exposition of this chapter self contained.

of $f_1, \ldots, f_m$. If no such nonzero polynomial A exists, then the given polynomials are called *algebraically independent* over $\mathbb{F}$.

For example, $f_1 = (x+y)^2$ and $f_2 = (x+y)^3$ are algebraically dependent over any field, as $y_1^3 - y_2^2$ is an annihilating polynomial. Polynomials $x+y$ and x^p+y^p are dependent over $\mathbb{F}_p$, but independent over $\mathbb{Q}$. The monomials $x_1,\ x_2,\ \ldots,\ x_n$ are an example of algebraically independent polynomials over any field.

Algebraic dependence can be viewed as a generalization of linear dependence as the former captures algebraic relationships of any degree, whereas the latter captures linear relationships. Algebraic dependence shares a few combinatorial properties (known as matroid properties, see [150]) with linear dependence. For example, if a set of polynomials is algebraically independent then any subset of them is algebraically independent.

The *algebraic rank*, also known as *transcendence degree*, of a set of polynomials is defined as the maximal number of algebraically independent polynomials and it is well defined thanks to the matroid properties. The concepts of rank and basis in linear algebra have analogs here as algebraic rank and *transcendence basis*, respectively.

Algebraic rank is a generalization of several natural problems in algebra and combinatorics, for example, computing the size of the *maximum matching* in general graphs is also a special case of the problem. More generally, it is a generalization of the celebrated *Polynomial Identity Testing* (PIT) problem, which itself captures several problems including the *Primality Testing* problem. The notion of algebraic rank has been used to make progress on PIT by helping in the hitting set construction for $\overline{\mathsf{VP}}$ [95] and by being crucial in solving several special cases of the PIT problem [7, 126], and very recently has also been used in the famous *Integer Factorization* problem [8]. It has also been used to construct explicit extractors, condensers and dispersers for polynomial sources [65], crucial in the area of pseudorandom generators. It was also important in proving the best known general formula lower bounds for determinant [118], and more recently for proving strong lower bounds for restricted class of arithmetic circuits [7, 126, 151].

5.1.3 Rank of symbolic matrices

Symbolic matrices, i.e., matrices with polynomial entries are ubiquitous objects in mathematics. Edmonds [66] was the first one who explicitly stated the problem of computing the rank of a symbolic matrix when the entries are linear forms. Some applications of symbolic rank computation in computer science include the algebraic algorithm for *maximum matching* problem for bipartite and general graphs [133, 144, 68]. Even the *linear matroid parity problem* and the *linear matroid intersection problem* are special cases of the commutative rank problem of symbolic matrices with linear forms as entries (see [167, 77, 149, 101, 98]). Owing to the works of Valiant and Mahajan-Vinay, we know that the rank computation of symbolic matrices (the decision version) with linear entries is equivalent to *Polynomial Identity Testing* (PIT) for Algebraic Branching Programs (ABP) [179, 138] which became a central problem in complexity theory after the results of Kabanets and Impagliazzo [117] showing that a deterministic algorithm for PIT would imply circuit complexity lower bounds. When the entries are allowed to be higher degree polynomials, too, the symbolic matrix rank computation also captures the computation

of the rank of the Jacobian matrix, which in turn captures the algebraic rank problem over fields of zero characteristic via the classical Jacobian criterion. It also captures the rank of the Hessian matrix which like the Jacobian matrix is pervasive in mathematics and physics. For example, using the Katz's dimension formula, the rank of the Hessian matrix corresponds to the dimension of the dual varieties of hypersurfaces, which is useful in studying the determinantal complexity in the Geometric Complexity Theory (see e.g. [104, 131]).

5.1.4 A tale of three computational problems

In this section, we give an account of the three main computational problems relevant to the chapter, each inspired by one of themes discussed in the above subsections. In order to discuss them more precisely, we have to specify the representation of the input polynomials. An *arithmetic circuit* is a directed acyclic graph consisting of addition (+) and multiplication (×) gates as nodes, takes variables $x_1, \ldots, x_n$ and field constants as input (leaves), and outputs a polynomial $f(x_1, \ldots, x_n)$. This is a succinct representation of multivariate polynomials, as polynomials of high degree (or having many monomials) can be represented by small circuits.

Problem 5.1 (**PIT**). *Given an arithmetic circuit computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$, test if f is identically zero.*

Problem 5.2 (**AlgRank**). *Given arithmetic circuits $C_1, \ldots, C_n$ computing polynomials $f_1, \ldots, f_n \in \mathbb{F}[x_1, \ldots, x_m]$, compute* $\mathrm{algRank}(\{f_1, \ldots, f_n\})$.

Problem 5.3 (**RANK**). *Given an $n \times n$ matrix $Q(x_1, x_2, \ldots, x_m) = (q_{ij})_{n\times n}$ whose entries are given by arithmetic circuits C_{ij} computing polynomials $q_{ij} \in \mathbb{F}[x_1, \ldots, x_m]$, compute the rank of Q over $\mathbb{F}(x_1, \ldots, x_m)$.*

Connections among the three problems

It is clear that **PIT** reduces to the decision version of **RANK**. In fact, it is known that the case of **RANK**, where the entries are just *linear forms*, is strong enough to capture **PIT** for *algebraic branching programs* (ABP) [179, 138]. It is also easy to see that **PIT** reduces to the decision version of **AlgRank**: we can just give our input instance to the **AlgRank** problem and ask whether the algebraic rank is 0 or 1. It might happen that the circuit was computing a non-zero constant polynomial, in this case the algebraic rank will not be able to certify the non-zeroness, because the algebraic rank is still 0 in this case. However, this is an easy case anyway, because we can test these cases beforehand simply by evaluating the circuit on the point $(0, \ldots, 0)$ and checking if the circuit evaluates to 0. It might be the case that the value at $(0, \ldots, 0)$ is too large to compute, since in the most general setting, the formal degree of the circuit can be exponential. In this case, we can alternatively check whether $x_1 \cdot f$ has algebraic rank 0 or 1, where f is the polynomial computed by the circuit.

Over fields of characteristic zero, the problem **AlgRank** reduces to the problem **RANK** via the classical Jacobian criterion:

Definition 5.1 (Jacobian)**.** *The* Jacobian *of polynomials* $\mathbf{f} = \{f_1, \cdots, f_n\} \subset \mathbb{F}[x_1, \cdots, x_m]$ *is the matrix* $\mathcal{J}_{\mathbf{x}}(\mathbf{f}) = (\partial_{x_j} f_i)_{m \times n}$*, where* $\partial_{x_j} f_i := \partial f_i / \partial x_j$*.*

We state the classical Jacobian criterion [115, 151].

Lemma 5.1 (Jacobian criterion)**.** *Let* $\mathbf{f} \subset \mathbb{F}[\mathbf{x}]$ *be a finite set of polynomials of degree at most* d*, and* $\text{algRank}_{\mathbb{F}} \mathbf{f} \leq r$*. If* $\text{char}(\mathbb{F}) = 0$*, or* $\text{char}(\mathbb{F}) > d^r$*, then* $\text{algRank}_{\mathbb{F}} \mathbf{f} = \text{rank}_{\mathbb{F}(\mathbf{x})} \mathcal{J}_{\mathbf{x}}(\mathbf{f})$*.*

Thus, in order to solve **AlgRank** for a set of polynomials $\mathbf{f}$, it suffices to solve **RANK** for the matrix where the entries are the first order partial derivatives of the elements in $\mathbf{f}$. Now, we know that for a given arithmetic circuit C of size s computing a polynomial f, there exists an arithmetic circuit C' of size $O(s)$ computing all the first order partial derivatives of f ([20], see also [170, Section 2.3]). Thus having a deterministic poly-time algorithm for computing the rank of a matrix with entries given by arithmetic circuits, we can solve the **AlgRank** problem in deterministic polynomial time when the input is given as the arithmetic circuits of the set of polynomials whose algebraic rank we want to compute.

Similarly, if the input to our **AlgRank** problem is a set of polynomials with bounded degrees (say, with an upper bound of d), the Jacobian matrix will have entries which are polynomials with degrees bounded by $d - 1$. So, in order to solve the bounded degree version of the **AlgRank** problem, it suffices to solve the bounded degree version of the **RANK** problem.

Thus, over fields of characteristic 0, it suffices to solve **RANK** in order to solve **PIT** and **AlgRank**. In order to give a PTAS for the **AlgRank** problem for bounded degree polynomials, we give a PTAS for the **RANK** problem where entries of the matrix are bounded degree polynomials. We remind the reader that the decision version of the **RANK** problem in the bounded degree case still gives an unbounded degree PIT instance (simply recall that it is already true when the entries were linear forms). In fact, for the **RANK** and the **AlgRank** problem, we need such restrictions, else we will have to solve the general PIT problem beforehand.

The current status of the three problems

All of the three problems can be solved in randomized polynomial time thanks to the Schwarz-Zippel lemma [164, 184, 61]. For **RANK**, we just need to evaluate our polynomials at a random point to obtain a matrix of field elements, and the lemma guarantees that with high probability the rank of the obtained matrix over the base field $\mathbb{F}$ would be the same as the rank of the original matrix (over the function field $\mathbb{F}(x_1, \ldots, x_m)$). And as we already pointed out in Section 5.1.4, **RANK** is the most general of the three problems, all the three problems can be solved in randomized polynomial time. However, a deterministic algorithm has remained elusive for all of the three problems. For the simplest problem among the three, i.e., the **PIT** problem, we know deterministic polynomial time algorithms only in special cases. In fact, there has been a plethora of works derandomizing special cases in polynomial or quasipolynomial time. For example, when the input is given in the sparse representation, a deterministic polynomial time algorithm is known [121]. Similarly, if the input is a diagonal depth 3 circuit, we know

a deterministic polynomial time algorithm [158]. We refer the reader to the excellent surveys on the problem by Saxena [159, 160] and Shpilka and Yehudayoff [170] for a detailed account of the progress and techniques involved in derandomizing the PIT problem. Derandomizing the **RANK** problem in its simplest case, i.e., when the entries are just *linear forms* has already proven to be very challenging. It is equivalent to solving PIT problem for Algebraic Branching Programs (ABPs). Only for very restricted classes of ABPs (the so called ROABP model and its variants), we know how to derandomize PIT ([155, 74, 72, 96]). Recently, [33] gave a derandomization for approximately computing the rank, i.e., they gave a deterministic PTAS for the **RANK** problem, when the entries are linear forms.

RANK in the non-commutative world

We point out that in the non-commutative world, several computational problems are better understood as compared to the commutative world. For example, PIT for non-commutative formulas is known to be in polynomial time [155]. Moreover, exponential lower bounds are known against non-commutative formulas and algebraic branching programs [147]. The same is true for the **RANK** problem in the non-commutative world. Here, Garg, Gurvits, Oliviera, and Wigderson [84] and Ivanyos, Qiao, and Subrahmanyam [114], gave a deterministic polynomial time algorithm for the **RANK** problem when the entries of the matrices are linear forms. In fact, they also solved the RANK problem when the entries are given by formulas, because in the non-commutative world, the case in which the entries are given by formulas reduces to the case in which the entries are given by linear forms using Higman's trick ([106], see [83, Appendix A.1]). One would be tempted to use the same trick for the commutative rank and then use the deterministic PTAS for linear forms case given by [33] to have a deterministic PTAS for the case in which the entries are given by formulas. Unfortunately, this trick does only preserves the co-rank. Hence, it is not useful for computing an approximation of the rank in the general **RANK** problem, since it enlarges the size of the matrix. Another interesting fact is that in the case when the entries are linear forms, we know that the non-commutative rank (see [83] for a definition) is at most twice the commutative rank [75]. Thus, an algorithm for the non-commutative rank gives a 1/2-approximation for the commutative rank when the entries are linear forms. Here, one would be tempted to claim that even when the entries are given by formulas, we get a 1/2-approximation for the commutative rank using the known exact algorithms for the non-commutative rank. This also does not work unfortunately. The following very simple example denies any such possibilities when entries compute higher degree polynomials.

Let $f = xy - yx$. Consider, the following 1×1 matrix Q,

$$Q = \begin{bmatrix} f \end{bmatrix}.$$

Notice that the non-commutative rank of Q is 1, but the commutative rank is 0. This gap can be made arbitrarily large by simply taking a diagonal matrix with all the diagonal entries being $xy - yx$. Thus, in general, we cannot approximate the commutative rank with non-commutative rank.

Thus, there is a huge knowledge gap that we are observing between the commuta-

tive world and the non-commutative world with respect to the **RANK** problem. On the one hand, we have polynomial time algorithms for exact rank computation in the non-commutative world even when the entries are given by formulas, whereas in the commutative case, all we have is a deterministic PTAS, that only works in the case when the entries of the matrix are *linear forms*. No deterministic PTAS was known even when the entries of the matrix are given by *quadratic forms*. In this chapter, we solve precisely a more general version of this, i.e., we give a deterministic PTAS in the case when the entries are given by polynomials whose degrees are bounded by an arbitrary constant, hence taking another step towards bridging this knowledge gap between the two worlds.

5.1.5 Results

In this chapter, we give the first deterministic polynomial time approximation scheme (PTAS) for the **RANK** problem under the restriction that the entries of the matrix are bounded degree polynomials. We give a new technique which allows us to achieve generalizations to higher degrees of the results of [33], who gave a PTAS for the **RANK** problem when the entries were linear forms.

We need to formalize the setup of the problem and fix some notations to formally state our main result.

Consider a matrix $Q(x_1, x_2, \ldots, x_m) = (q_{ij})_{n\times n}$ of size $n \times n$, the entries q_{ij} of which are polynomials of degrees bounded by some constant d in the variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. We want to compute the rank of Q over the rational function field $\mathbb{F}(x_1, x_2, \ldots, x_m)$. In fact, it suffices to consider the case when the entries are homogeneous forms of degree d (see Section 5.6).

To this end, we define the following problem.

Problem 5.4. *Given a matrix $Q(x_1, x_2, \ldots, x_m) = (q_{ij})_{n\times n}$ of size $n \times n$, the entries q_{ij} of which are homogeneous forms of constant degree d, compute the rank of Q over $\mathbb{F}(x_1, x_2, \ldots, x_m)$.*

Since the degrees of the polynomials in the entries are bounded by a constant d, we can assume that they are given explicitly as the list of coefficients.

As stated above in Section 5.1.4, this problem has a very simple randomized algorithm. But we want deterministic algorithms to compute the rank of Q. We know that finding deterministic algorithms for Problem 5.4 is hard. Thus in this chapter, we consider whether one can approximate $\operatorname{rank}(Q)$ deterministically. Following is the main contribution of this chapter.

Theorem 5.1 (PTAS for RANK)**.** *Given Q as in Problem 5.4 over a field $\mathbb{F}$ with $|\mathbb{F}| > nd$ and a constant $0 < \varepsilon < 1$, there exists a deterministic algorithm which computes an assignment $(\lambda_1, \lambda_2, \ldots, \lambda_m) \in \mathbb{F}^m$ such that,*

$$\operatorname{rk}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) \geq (1 - \varepsilon) \operatorname{rk}(Q(x_1, \ldots, x_m)).$$

Clearly, the above running time is polynomial when d is a constant.

Now we see that it suffices to solve Problem 5.4 for ε being a constant, as this implies the general case via tensoring. That is, for an $n \times n$ matrix Q of Problem 5.4, we can tensor Q with itself to get an $n^2 \times n^2$ size matrix, where $\text{rank}(Q)^2 = \text{rank}(Q \otimes Q)$. Also, if Q has degree d entries, then $Q \otimes Q$ has degree $2d$ entries. Thus, if we tensor k times we get a matrix of size $n^k \times n^k$ with entries of degree dk and rank $(\text{rank}(Q))^k$. If we get a $(1-\varepsilon)$ approximation to this rank, then taking the k^{th} root of this value is a $(1-\varepsilon)^{\frac{1}{k}}$-approximation to the original rank. As this is approximately $(1-\frac{\varepsilon}{k})$, it follows that we can get a pretty good approximation this way. By this method, we get a trade-off between the degree and the approximation parameter. That is, if an algorithm finds a $1-\varepsilon$ approximation of the rank when the entries are degree dk polynomials, then this algorithm can be used to find a $(1-\frac{\varepsilon}{k})$ approximation of the rank when the entries are degree d polynomials. This reduction also shows that at least a linear dependence on d in the exponent is essentially required for this problem, even for $\varepsilon = O(1)$, as otherwise via tensoring we can solve PIT non-trivially fast. We remark that our algorithm directly tackles the problem of $1-\varepsilon$ rank approximation without using this tensoring idea.

Since we have already established in the previous section that **AlgRank** reduces to **RANK** using the Jacobian criterion, it is obvious that the deterministic PTAS for **AlgRank** under the restriction that the input polynomials are of bounded degree is an easy consequence of the above stated result.

Theorem 5.2 (PTAS for AlgRank)**.** *Given a set $\boldsymbol{f} := \{f_1, \ldots, f_n\} \subset \mathbb{F}[x_1, \ldots, x_m]$ of polynomials of degrees bounded by a constant d with $\text{char}(\mathbb{F}) = 0$, and a rational number $\varepsilon > 0$, there is a deterministic algorithm that outputs a number r, such that $r \geq (1-\varepsilon) \cdot \text{algRank}(\boldsymbol{f})$. The algorithms runs in time $O\left((nmd)^{O\left(\frac{d^2}{\varepsilon}\right)} \cdot M(n)\right)$, where $M(n)$ is the time required to compute the rank of an $n \times n$ matrix over $\mathbb{F}$.*

Again, the above algorithm is a polynomial time algorithm when d is a constant.

5.1.6 Comparison with the techniques of [33]

The PTAS for the linear case in [33] greedily increased the rank starting with the zero matrix, and the proof of correctness of the algorithm rested on the guarantee that when we are unable to increase the rank greedily, we are already done, i.e., the current matrix already has the desired approximation of the rank. The main component of the proof of this guarantee was a refined analysis of the so-called Wong sequence which are defined for matrix spaces and a matrix with entries as linear forms can be interpreted as a matrix space. [33] introduced a novel notion of Wong index. It was shown in [33] that if the Wong index of a matrix is "high" then this matrix is a good approximation of the commutative rank. If the Wong index of a matrix is "low" then it was shown in [33] that one can find a matrix of higher rank efficiently.

The limitation of the techniques in [33] is that the Wong sequences are defined and studied only in the case of matrix spaces and a correspondence between the matrix spaces and matrix with higher degree (non-linear) polynomials does not exist. So, for the higher degree case, it is not clear how to define a notion of a Wong sequence and hence the techniques of [33] do not generalize. Thus, one needs to find a new technique

to deal with the higher degree case. This is precisely what we do in this chapter. Our starting point is an alternative analysis for the [33]'s algorithm for the linear case that appeared in [116], where a new way to analyze the low degree components of the minors of the matrix obtained in the greedy step is given (see Section 5.3 for the main proof ideas). In this chapter, we build upon the analysis in [116] and show that the analysis can be extended to work when the entries are higher degree polynomials as well. This allows us to use the same algorithm strategy as in [33] for higher degree forms as well. It can be shown that the Wong index of [33] corresponds to the degree of the least degree monomial of a suitable minor.

5.1.7 Organization of the chapter

In the next section, we define some notations and recall some linear-algebraic tools that will be useful for us. In Section 5.3, we discuss our main idea and give an overview of the proof strategy. Section 5.4 contains the technical details of the proof. We present our commutative rank algorithm in Section 5.5. Finally, Section 5.6 contains the reduction of the arbitrary case to the homogeneous case.

5.2 Preliminaries

In the following, we present some of the definitions and tools which are used frequently in this chapter. When we speak of a matrix polynomial, we mean a matrix with polynomials as entries.

(1) For an $r \times r$ matrix $A \in \mathbb{F}^{r\times r}$, we use $A_{\widehat{ij}}$ to denote the sub-matrix of A obtained by removing the i^{th} column and the j^{th} row.

(2) I_r is used to denote the $r \times r$ identity matrix.

(3) For a polynomial f, $\hom_k(f)$ denotes the homogeneous degree k part of f.

(4) We also use the same notation $\hom_k(M)$ to denote the homogeneous degree k part of a matrix polynomial M. Note that $\hom_k(M)$ is also a matrix polynomial.

(5) For a polynomial f, $\operatorname{ord}(f)$ is used to denote the degree of the least degree monomial in f. We use the same notation $\operatorname{ord}(M)$ for matrix polynomials M also, where $\operatorname{ord}(M)$ is defined as the degree of the least degree monomial in M. Notice that $\operatorname{ord}(f)$ and $\operatorname{ord}(M)$ are just natural numbers.

Definition 5.2 (Characteristic Polynomial)**.** *For an $r \times r$ matrix A, its* characteristic polynomial $p_A(t)$ *is defined as:*

$$p_A(t) := \det(tI - A) = p_0 t^r + p_1 t^{r-1} + \cdots + p_r.$$

Note that in Definition 5.2, $p_0 = 1$ is always true.

Fact 5.1. *Over all fields, for any $r \times r$ matrix A,* $\det(A) = (-1)^r p_A(0) = (-1)^r p_r$.

Definition 5.3 (Adjoint)**.** *For an $r \times r$ matrix A, the* adjoint $\operatorname{adj}(A)$ *is also an $r \times r$ matrix whose $(i,j)^{th}$ entry is* $(-1)^{i+j}\det(A_{\widehat{ij}})$.

Theorem 5.3. *For a square $r \times r$ matrix L, define $q_L(t) := \frac{p_L(t)-p_L(0)}{t}$. Over all fields, we have the following equality for* $\mathrm{adj}(L)$*:*

$$\mathrm{adj}(L) = (-1)^{r+1} q_L(L). \tag{5.2.1}$$

Proof. Here we only prove this claim for $\mathbb{F} = \mathbb{C}$ but it is true for all fields. We use the following facts:

(1) If L is non-singular, then $\mathrm{adj}(L) = \det(L)L^{-1}$.

(2) The Cayley-Hamilton theorem, which states that for any L, $p_L(L) = 0$.

(3) The set GL_r of non-singular matrices is dense (under the Euclidean topology) in the set $\mathbb{F}^{r\times r}$ of all the matrices.

We first prove the claim when L is non-singular. Let $p_L(t) = p_0 t^r + p_1 t^{r-1} + \cdots + p_r$. By using the Cayley-Hamilton theorem, we have the following equality:

$$p_0 L^r + p_1 L^{r-1} + \cdots + p_r = 0. \tag{5.2.2}$$

Since L is non-singular, we multiply by L^{-1} on both the sides of Equation (5.2.2). Thus

$$p_0 L^{r-1} + p_1 L^{r-2} + \ldots p_{r-1} = -p_r L^{-1} = (-1)^{r+1}\det(L)L^{-1} = (-1)^{r+1}\,\mathrm{adj}(L). \tag{5.2.3}$$

Note that $q_L(L) = p_0 L^{r-1} + p_1 L^{r-2} + \cdots + p_{r-1}$. Therefore Equation (5.2.3) implies $\mathrm{adj}(L) = (-1)^{r+1} q_L(L)$.

Now notice that Equation (5.2.1) is an equation where entries of the matrices on both sides are polynomials in the entries of L. Now the claim follows using the denseness of GL_r in $\mathbb{F}^{r\times r}$. □

Theorem 5.4. *For a square $r \times r$ matrix L with $p_L(t) = p_0 t^r + p_1 t^{r-1} + \cdots + p_r$, the following equality holds over any field:*

$$\mathrm{adj}(I + L) = \sum_{i=0}^{r-1} (-1)^i p_i \cdot \left(\sum_{j=0}^{r-i-1} (-L)^j \right).$$

Proof. First we compute the characteristic polynomial p_{I+L} of $I + L$. We have:

$$\begin{aligned} p_{I+L}(t) &= \det(tI - (I + L)) \\ &= \det((t-1)I - L)) \\ &= p_L(t-1). \end{aligned}$$

Thus we have,

$$\begin{aligned} q_{I+L}(t) &:= \frac{p_{I+L}(t) - p_{I+L}(0)}{t} \\ &= \frac{p_L(t-1) - p_L(-1)}{t} \\ &= \sum_{i=0}^{r} \frac{p_{r-i} \cdot ((t-1)^i - (-1)^i)}{t} \\ &= \sum_{i=1}^{r} p_{r-i} \cdot \left(\sum_{j=0}^{i-1} (-1)^j (t-1)^{i-j-1} \right). \end{aligned}$$

Therefore,

$$\begin{aligned} \operatorname{adj}(I+L) &= (-1)^{r+1} q_{I+L}(I+L) \\ &= (-1)^{r+1} \sum_{i=1}^{r} p_{r-i} \cdot \left(\sum_{j=0}^{i-1} (-1)^j (L)^{i-j-1} \right) \\ &= p_0(I - L + \cdots + (-L)^{r-1}) - p_1(I - L + \ldots (-L)^{r-2}) + \cdots + (-1)^{r-1} p_{r-1}(I) \\ &= \sum_{i=0}^{r-1} (-1)^i p_i \cdot \left(\sum_{j=0}^{r-i-1} (-L)^j \right). \end{aligned}$$

□

Next come some easy facts from linear algebra.

Fact 5.2. *Let $\mathbb{F}$ be any field. If A is an $n \times n$ matrix of rank r over $\mathbb{F}$, then there exist two $n \times n$ non-singular matrices $P, R \in \mathbb{F}^{n \times n}$ such that:*

$$PAR = \begin{array}{c} \overbrace{}^{r \text{ columns}} \\ \left(\begin{array}{cc} I_r & 0 \\ 0 & 0 \end{array} \right) \begin{array}{l} \} \, r \text{ rows} \\ \} \, n-r \text{ rows} \end{array} \\ \underbrace{}_{n-r \text{ columns}} \end{array} \tag{5.2.4}$$

Fact 5.3. *Let $\mathbb{F}$ be any field and let M be a matrix of the following form over $\mathbb{F}$:*

$$M = \begin{array}{c} \overbrace{}^{r \text{ columns}} \\ \left(\begin{array}{cc} L & B \\ A & C \end{array} \right) \begin{array}{l} \} \, r \text{ rows} \\ \} \, n-r \text{ rows} \end{array} \\ \underbrace{}_{n-r \text{ columns}} \end{array} \tag{5.2.5}$$

Also, let $\operatorname{rank}(\begin{bmatrix} A & C \end{bmatrix}) = a$ *and* $\operatorname{rank}\left(\begin{bmatrix} B \\ C \end{bmatrix} \right) = b$. *Then* $\operatorname{rank}(M) \leq r + \min(a, b)$.

Lemma 5.2. *If $|\mathbb{F}| > nd$, then we can construct a hitting set $H_{m,d,\ell}$ of size $O((m(d+1))^\ell)$ for the set $F_{m,d,\ell}$ of polynomials defined by:*

$$F_{m,d,\ell} := \{ f \in \mathbb{F}[x_1, \ldots, x_m] \mid \deg(f) \leq d, \operatorname{ord}(f) \leq \ell \}.$$

Proof. Let $f \in F_{m,d,\ell}$. Since $\mathrm{ord}(f) \leq \ell$, there exists a non-zero monomial $x_{i_1} \cdot x_{i_2} \cdot \cdots \cdot x_{i_\ell}$ of f. The variables x_{i_j} need not be distinct here. We first do a brute force search for these ℓ variables by making all the other $m - \ell$ variables zero. This can be done using $\binom{m}{\ell} = O(m^\ell)$ assignments. Now we are left with a polynomial f' of degree d in at most ℓ variables. By using Schwartz-Zippel lemma [184, 164], we can find a non-zero assignment of f' using $(d+1)^\ell$ assignments. Thus there exists a hitting set of size $O(m^\ell \cdot (d+1)^\ell) = O((m(d+1))^\ell)$. □

5.3 Main proof ideas

Here we explain the main idea used in devising the desired algorithm claimed in Theorem 5.1. Since Theorem 5.1 is essentially a generalization of [33], a direct approach seems to be converting a matrix of degree d forms to a matrix of linear forms (using the equivalence of ABPs and determinants, see [138]). However, such a direct approach (although it preserves non-zeroness) gives no information about the rank of the matrix.

So, instead of directly reducing an instance of Problem 5.4 to an instance of linear forms case as in [33], we follow the high level approach of [33] of greedily increasing the rank of Q starting with the zero matrix. Suppose we have found $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that $\mathrm{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) = r$. We want to find an assignment of the form $(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)$ such that $\mathrm{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)) > r$. This step of finding a matrix of bigger rank is called a rank increasing step. Under this transformation $(x_i \to x_i + \lambda_i)$, we have the following equality:

$$Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) = Q(\lambda_1, \lambda_2, \ldots, \lambda_m) + Q_d(x_1, x_2, \ldots, x_m). \quad (5.3.1)$$

Here $Q_d(x_1, x_2, \ldots, x_m)$ is some matrix whose entries are polynomials of degree at most d. By using Fact 5.2, we know that there exists non-singular matrices $P, R \in \mathbb{F}^{n \times n}$ such that:

$$P \cdot Q(\lambda_1, \lambda_2, \ldots, \lambda_m) \cdot R = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

Now consider the following equation:

$$\begin{aligned} &P \cdot Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) \cdot R \\ &\quad = P \cdot Q(\lambda_1, \lambda_2, \ldots, \lambda_m) \cdot R + P \cdot Q_d(x_1, x_2, \ldots, x_m) \cdot R. \end{aligned} \quad (5.3.2)$$

Since P, R are non-singular, we know that

$$\mathrm{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)) = \mathrm{rank}(P \cdot Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) \cdot R).$$

Thus it is enough to find an assignment to the variables $x_1, \ldots, x_m$ such that

$$\mathrm{rank}(P \cdot Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) \cdot R) > r.$$

The following Lemma 5.3 is easy to verify.

Lemma 5.3. *For any* $(\lambda_1, \lambda_2, \ldots, \lambda_m) \in \mathbb{F}^m$,

$$\begin{aligned} &\mathrm{rank}(Q(x_1, x_2, \ldots, x_m)) = \mathrm{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)) \\ &\quad = \mathrm{rank}(P \cdot Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) \cdot R). \end{aligned}$$

Proof. We assume that $|\mathbb{F}| > dn$. Now suppose that $s = \max\{\text{rank}(Q(\lambda_1, \ldots, \lambda_m)) \mid (\lambda_1, \ldots, \lambda_m) \in \mathbb{F}^m\}$ and $r = \text{rank}(Q(x_1, x_2, \ldots, x_m))$. We want to show that $s = r$. We know that there exists a non-zero $r \times r$ minor M_r of $Q(x_1, x_2, \ldots, x_m)$. Notice that M_r is a polynomial of degree at most $rd \leq nd$. Thus by the Schwartz-Zippel lemma [184, 164], there exists $(\lambda_1, \lambda_2, \ldots, \lambda_m) \in \mathbb{F}^m$ such that $M_r(\lambda_1, \lambda_2, \ldots, \lambda_m) \neq 0$. Therefore $s \geq r$. The other direction is trivial.

This also implies that

$$\text{rank}(Q(x_1, x_2, \ldots, x_m)) = \text{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)) = r.$$

This is because there is a bijection from $\mathbb{F}^m$ to $\mathbb{F}^m$ given by $x_i \mapsto x_i + \lambda_i$. This last equality is trivial because we only multiply by non-singular matrices P and R. □

By using Lemma 5.3, we can omit P, R in the discussion of our rank increasing step. Thus we can assume that:

$$Q(\lambda_1, \lambda_2, \ldots, \lambda_m) = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}.$$

We want to ensure that at least one of the following two scenarios happens.

(1) We can "easily" find an assignment of the form $(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)$ such that $\text{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)) > r$. This is our rank increasing step. "Easily" here means in time $O((nmd)^{O(\frac{d}{\varepsilon})})$ deterministically.

(2) $r \geq (1 - \varepsilon) \cdot \text{rank}(Q(x_1, x_2, \ldots, x_m))$, i.e., we are already done.

We decompose $Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)$ as:

$$Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m) = \begin{bmatrix} I_r + L & B \\ A & C \end{bmatrix}. \tag{5.3.3}$$

We write $L = L_1 + \cdots + L_d$, where L_i is a matrix whose entries are homogeneous polynomials of degree i. Similarly we decompose A, B, C into A_i, B_i, C_i. In other words, we have $L_s = \text{hom}_s(L)$, $A_s = \text{hom}_s(A)$, $B_s = \text{hom}_s(B)$, and $C_s = \text{hom}_s(C)$.

We now describe when the first of the two scenarios described above happens. When is the condition "$\text{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)) > r$" true? It happens when there exists a non-zero $(r + 1) \times (r + 1)$ minor of $Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)$. Consider a sub-matrix $M_{k,\ell}$ of $Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)$ of size $(r + 1) \times (r + 1)$ obtained by taking $I_r + L$, the k^{th} row of A, the ℓ^{th} column of B, and also the $(k, \ell)^{\text{th}}$entry of C. Thus $M_{k,\ell}$ looks like below:

$$M_{k,\ell} = \begin{pmatrix} 1 + l_{11} & l_{12} & \ldots & l_{1r} & b_1 \\ l_{12} & l_{22} & \ldots & l_{2r} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_{r1} & l_{r2} & \ldots & 1 + l_{rr} & b_r \\ a_1 & a_2 & \ldots & a_r & c \end{pmatrix}. \tag{5.3.4}$$

Here l_{ij} is the $(i, j)^{\text{th}}$ entry of L. To ensure $\text{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots., x_m + \lambda_m)) > r$, we want to find an assignment to the x_i's such that $\exists k, \ell \in [n - r]$ satisfying $\det(M_{k,\ell}) \neq 0$.

How to find an assignment to the x_i's such that $\det(M_{k,\ell}) \neq 0$? Note that $\det(M_{k,\ell})$ is a polynomial of degree at most $(r+1)d$ in the variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. Suppose $\det(M_{k,\ell})$ has a non-zero monomial of some constant degree s then we can easily (see Lemma 5.2) find an assignment to the x_i's such that $\det(M_{k,\ell}) \neq 0$. To check if $\det(M_{k,\ell})$ has a non-zero monomial of degree s, we just need to analyze $\hom_s(\det(M_{k,\ell}))$. This is our overall strategy. Therefore the scenarios described above can be reformulated as below.

(1) For an appropriately chosen s (depending upon d and ε), $\exists k, \ell \in [n-r]$ such that $\det(M_{k,\ell})$ has a non-zero monomial of degree at most s. In this case, we can "easily" find an assignment to the x_i's such that $\det(M_{k,\ell}) \neq 0$. This ensures that $Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) > r$. This is our rank increasing step.

(2) $\forall k, \ell \in [n-r]$, $\det(M_{k,\ell})$ has no non-zero monomials of degree at most s. In this case, we show that $r \geq (1-\varepsilon) \cdot \operatorname{rank}(Q(x_1, x_2, \ldots, x_m))$.

To analyze $\hom_s(\det(M_{k,\ell}))$, it is useful to find a compact expression for $\det(M_{k,\ell})$. We now give such a compact expression for $\det(M_{k,\ell})$. In whatever follows, we use the symbol $\mathbf{a}$ to denote the row vector $\begin{bmatrix} a_1 & a_2 & \ldots & a_r \end{bmatrix}$, symbol $\mathbf{b}$ to denote the column vector $\begin{bmatrix} b_1 & b_2 & \ldots & b_r \end{bmatrix}^t$ and

$$p_L(t) := p_0 t^r + p_1 t^{r-1} + \cdots + p_r.$$

Lemma 5.4. *Let $M_{k,\ell}$ be as in Equation (5.3.4). Then we have the following equality:*

$$\det(M_{k,\ell}) = -\mathbf{a} \cdot (\operatorname{adj}(I_r + L)) \cdot \mathbf{b} + c \cdot (\det(I_r + L)).$$

Proof. By Laplace expansion, we know that the following equality holds for $\det(M_{k,\ell})$:

$$\begin{aligned}
&\sum_{1 \leq i \leq r} (-1)^{i+r} a_i \cdot \left(\sum_{1 \leq j \leq r} (-1)^{j+r-1} \cdot b_j \cdot \det((I_r + L)_{\widehat{ij}}) \right) + c \cdot (\det(I_r + L)) \\
&= - \sum_{1 \leq i,j \leq r} a_i b_j (-1)^{i+j} \det((I_r + L)_{\widehat{ij}}) + c \cdot (\det(I_r + L)) \\
&= - \sum_{1 \leq i,j \leq r} a_i b_j (\operatorname{adj}(I_r + L))_{ij} + c \cdot (\det(I_r + L)) \\
&= -\mathbf{a} \cdot \operatorname{adj}(I_r + L) \cdot \mathbf{b} + c \cdot (\det(I_r + L)) \\
&= \det(M_{k,\ell}).
\end{aligned}$$

□

Lemma 5.5. *Let $M_{k,\ell}$ be as in Equation (5.3.4). Then the following equality holds for $\det(M_{k,\ell})$:*

$$\det(M_{k,\ell}) = -\mathbf{a} \cdot \left(\sum_{i=0}^{r-1} (-1)^i p_i \cdot \left(\sum_{j=0}^{r-i-1} (-L)^j \right) \right) \cdot \mathbf{b} + c \cdot (p_0 - p_1 + \cdots + (-1)^r p_r). \quad (5.3.5)$$

Proof. By using Fact 5.1 we know that for any matrix A of size $r \times r$, $\det(A) = (-1)^r p_A(0)$. Now observe that $p_{I_r + L}(t) = p_L(t-1)$. Thus

$$\det(I_r + L) = (-1)^r p_L(-1) = (p_0 - p_1 + \cdots + (-1)^r p_r).$$

Now the claim follows by using Lemma 5.4 and Theorem 5.4. □

Corollary 5.1. *If M is the $(n-r)\times(n-r)$ matrix polynomial having the polynomial $\det(M_{u,v})$ as its $(u,v)^{th}$-entry for all $1 \le u,v \le n-r$, then the following equality holds for M:*

$$M = -A \cdot \left(\sum_{i=0}^{r-1} (-1)^i p_i \cdot \left(\sum_{j=0}^{r-i-1} (-L)^j \right) \right) \cdot B + (p_0 - p_1 + \cdots + (-1)^r p_r) \cdot C. \quad (5.3.6)$$

Proof. It immediately follows from Lemma 5.5. □

By using Corollary 5.1, it is easy to observe the following Lemma 5.6.

Lemma 5.6. *There are $k,\ell \in [n-r]$ such that $\hom_s(\det(M_{k,\ell})) \neq 0$ if and only if $\hom_s(M) \neq 0$.*

5.4 The proof: analyzing the degree

In this section, we formally describe the idea described sketched in Section 5.3. Here we want to analyze the homogeneous degree s component $\hom_s(M)$ of M in Corollary 5.1. Recall that $p_L(t) = p_0 t^r + p_1 t^{r-1} + \cdots + p_r$. In Corollary 5.1, the coefficient of p_i is the sum of powers of $(-L)$ up to $r-i-1$. Thus, if we only want to analyze the degree s component $\hom_s(M)$ of M for some $s < \frac{r}{2}$, then we only need to consider p_i and $(-L)^i$ for $i < \frac{r}{2}$. To this end, we use the following notations in this section:

$$T := \sum_{j=0}^{\lceil \frac{r}{2} \rceil} (-L)^j, \qquad f := -p_1 + \cdots + (-1)^{\lceil \frac{r}{2} \rceil} p_{\lceil \frac{r}{2} \rceil}.$$

Theorem 5.5. *Suppose $s \in \mathbb{N}$ is such that the condition $1 \le s \le \lfloor \frac{r}{2} \rfloor - 1$ holds. Then we have that:*

$$\begin{aligned} \hom_s(M) &= -\hom_s((ATB - C)\cdot(p_0 + f)) \\ &= -\hom_s((ATB - C)\cdot(1 + f)). \end{aligned}$$

Proof. We use the fact that for $0 \le k \le r$, we have $\mathrm{ord}(p_k) \ge k$ and $\mathrm{ord}(L^k) \ge k$. Thus to obtain the homogeneous degree s part in Corollary 5.1, it is enough to consider the p_i and L^i with $i \le s$. Using $1 \le s \le \lfloor \frac{r}{2} \rfloor - 1$, we obtain that $r-1-s \ge r - \lfloor \frac{r}{2} \rfloor = \lceil \frac{r}{2} \rceil$. Therefore the claimed equality follows. □

By using Theorem 5.5, we see that $\hom_1(M) = C_1$ and $\hom_2(M) = C_2 + C_1 \hom_1(f) - A_1 B_1$. Extending this argument, we observe the following equality.

$$-\hom_s(M) = \hom_s(ATB - C) + \sum_{i=1}^{s-1} \hom_i(f) \cdot \hom_{s-i}(ATB - C). \quad (5.4.1)$$

With the aid of Equation (5.4.1), it is easy to prove the following Theorem 5.6.

Theorem 5.6. *Suppose* $s \in \mathbb{N}$ *is such that the condition* $1 \leq s \leq \lfloor \frac{r}{2} \rfloor - 1$ *holds. If* $\hom_\ell(M) = 0$ *for all* $\ell \in [s]$*, then* $\hom_\ell(ATB - C) = 0$ *for all* $\ell \in [s]$*.*

Proof. We prove it by induction on ℓ. For the base case $\ell = 1$, we have $\hom_1(M) = -\hom_1(ATB - C)$. For the induction step, consider for $\ell+1 \leq s$. By using Equation (5.4.1), we have that

$$-\hom_{\ell+1}(M) \;=\; \hom_{\ell+1}(ATB - C) + \sum_{i=1}^{\ell} \hom_i(f) \cdot \hom_{\ell+1-i}(ATB - C). \quad (5.4.2)$$

By induction hypothesis, we have $\hom_k(ATB - C) = 0$ for all $k \in [\ell]$. Therefore we obtain that:

$$\sum_{i=1}^{\ell} \hom_i(f) \cdot \hom_{\ell+1-i}(ATB - C) = 0.$$

Thus $-\hom_{\ell+1}(M) = \hom_{\ell+1}(ATB - C) = 0$. □

Let us now further reformulate the two scenarios we described above.

(1) If $\hom_\ell(M) \neq 0$ for some $\ell \in [s]$ then we can implement our rank increasing step due to Lemma 5.6.

(2) If $\hom_\ell(M) = 0$ for all $\ell \in [s]$ then Theorem 5.6 gives us a set of conditions on matrices A, B, C, T. We will show that these conditions on A, B, C, T can be used to bound $\operatorname{rank}(Q(x_1, x_2, \ldots, x_m))$.

The rest of this section analyzes the condition "$\forall \ell \in [s] : \hom_\ell(ATB - C) = 0$" and gives an upper bound on $\operatorname{rank}(Q(x_1, x_2, \ldots, x_m))$ in terms of r. In whatever follows, we use

$$\mathcal{A} := \begin{bmatrix} A_1 & A_2 & \ldots & A_d \end{bmatrix}$$

to denote the $(n - r) \times rd$ matrix and

$$\mathcal{B} := \begin{bmatrix} B_1^t & B_2^t & \ldots & B_d^t \end{bmatrix}^t$$

to denote the $rd \times (n - r)$ matrix. To simplify the notation, define R_s to be the $rd \times rd$ block matrix (composed of d^2 blocks of size $r \times r$) whose $(i, j)^{\text{th}}$ block is $\hom_{s-(i+j)}(T)$. (We here use the convention that $\hom_\ell(T) = 0$ if $\ell < 0$.)

5.4.1 Analyzing the degree $s \leq d$.

Lemma 5.7. *For all* $s \geq 1$*,* $\hom_s(ATB) = \mathcal{A} \cdot R_s \cdot \mathcal{B}$*.*

Proof. We have $\hom_s(ATB) = \sum_{i=1}^{d} \sum_{j=1}^{d} A_i \hom_{s-(i+j)}(T) B_j = \mathcal{A} \cdot R_s \cdot \mathcal{B}$. □

Corollary 5.2. *Suppose* $r \in \mathbb{N}$ *is such that the condition* $1 \leq d \leq \lfloor \frac{r}{2} \rfloor - 1$ *holds. If* $\hom_s(M) = 0$ *for all* $s \in [d]$ *then* $C_s = \mathcal{A} \cdot R_s \cdot \mathcal{B}$ *for all* $s \in [d]$*.*

Proof. It immediately follows from Theorem 5.6 and Lemma 5.7. □

In whatever follows, we use the notations:

$$M_1 := \begin{bmatrix} A & C \end{bmatrix}_{(n-r)\times n} \quad \text{and} \quad M_2 := \begin{bmatrix} B \\ C \end{bmatrix}_{n\times(n-r)}$$

Lemma 5.8. *Suppose $r \in \mathbb{N}$ is such that the condition $1 \le d \le \lfloor \frac{r}{2} \rfloor - 1$ holds. If $\hom_s(M) = 0$ for all $s \in [d]$ then the following inequalities are true:*

$$\operatorname{rank}(M_1) \le \operatorname{rank}(\mathcal{A}), \qquad \operatorname{rank}(M_2) \le \operatorname{rank}(\mathcal{B}).$$

Proof. By using Corollary 5.2, we have the following equality:

$$C = \sum_{i=1}^{d} C_i = \mathcal{A} \cdot (\sum_{i=1}^{d} R_i) \cdot \mathcal{B} \tag{5.4.3}$$

Let N_1 be the $rd \times n$ matrix whose first r columns form the matrix $(\begin{bmatrix} I_r & I_r & \dots & I_r \end{bmatrix}^t)_{r\times rd}$ and whose last $n-r$ columns are the matrix $(\sum_{i=1}^{d} R_i) \cdot \mathcal{B}$. Now the following Equation (5.4.4) follows from Equation (5.4.3):

$$M_1 = \begin{bmatrix} A & C \end{bmatrix}_{(n-r)\times n} = \mathcal{A} \cdot N_1. \tag{5.4.4}$$

Thus $\operatorname{rank}(M_1) \le \operatorname{rank}(\mathcal{A})$. Let N_2 be the $n \times rd$ matrix whose first r rows form the matrix $\begin{bmatrix} I_r & I_r & \dots & I_r \end{bmatrix}_{r\times rd}$ and last $n-r$ rows are the matrix $\mathcal{A} \cdot (\sum_{i=1}^{d} R_i)$. The following equality Equation (5.4.5) follows from Equation (5.4.3):

$$M_2 = \begin{bmatrix} B \\ C \end{bmatrix}_{n\times(n-r)} = N_2 \cdot \mathcal{B}. \tag{5.4.5}$$

Thus $\operatorname{rank}(M_2) \le \operatorname{rank}(\mathcal{B})$. □

5.4.2 Analyzing the higher degrees.

Lemma 5.9. *Suppose $s \in \mathbb{N}$ is such that the condition $1 \le s \le \lceil \frac{r}{2} \rceil$ is true. Then we have $\hom_s(T) = -\sum_{i=1}^{d-1} L_i \hom_{s-i}(T)$.*

Proof. Since $1 \le s$, we can safely ignore the term I in the summation in the definition of T, since it has degree 0. Since $s \le \lceil \frac{r}{2} \rceil$, we can also add the term $(-L)^{\lceil \frac{r}{2} \rceil + 1}$, since it will not contribute to $\hom_s(T)$ either. Therefore, we have

$$\begin{aligned} \hom_s(T) &= \hom_s\big(-L(I - L + \ldots \cdots + (-L)^{\lceil \frac{r}{2} \rceil - 1} + (-L)^{\lceil \frac{r}{2} \rceil}) + (-L)^{\lceil \frac{r}{2} \rceil + 1}\big) \\ &= \hom_s(-LT). \end{aligned}$$

Now the claim follows. □

Lemma 5.10. *If $s \in \mathbb{N}$ is such that the condition $d + 2 \le s \le \lceil \frac{r}{2} \rceil + 2$ holds, then we have $R_s = ER_{s-1}$, where*

$$E := \begin{bmatrix} -L_1 & -L_2 & \dots & -L_d \\ I_r & 0 & \dots & 0 \\ 0 & \dots & \ddots & \vdots \\ 0 & 0 & I_r & 0 \end{bmatrix}_{rd\times rd}$$

.

Proof. It immediately follows from Lemma 5.9. □

Theorem 5.7. *Suppose $s \in \mathbb{N}$ is such that the condition $d+1 \leq s \leq \lfloor \frac{r}{2} \rfloor - 1$ holds. If $\hom_i(M) = 0$ for all $i \in [s]$ then $\mathcal{A} \cdot R_{d+1} \cdot \mathcal{B} = \mathcal{A} \cdot ER_{d+1} \cdot \mathcal{B} = \ldots = \mathcal{A} \cdot E^{s-d-1} \cdot R_{d+1} \cdot \mathcal{B} = 0$.*

Proof. By using Theorem 5.6, we know that $\hom_\ell(ATB - C) = 0$ for all $\ell \in [s]$. Since $\deg(C) \leq d$, we get that $\forall i \in \{d+1, \ldots, s\}$, $\hom_i(ATB) = 0 = \mathcal{A} \cdot R_i \cdot \mathcal{B}$. Now the theorem follows by using the recursive formulation of R_i proved in Lemma 5.10. □

Notice that the $r \times r$ matrix L_d is non-singular because there is an assignment $\lambda_1, \lambda_2, \ldots, \lambda_m$ to the variables of L_d which makes $L_d(\lambda_1, \lambda_2, \ldots, \lambda_m) = I_r$. (Since Q is homogeneous of degree d, L_d equals the upper-right $r \times r$-submatrix of Q.) Therefore L_d as a matrix with polynomial entries is also non-singular. This implies that E is also non-singular because L_d is.

Lemma 5.11 (Lemma 5.3 in [33])**.** *Let $B \in \mathbb{F}^{n \times n}$ and*

$$B = \begin{array}{c} \\ r \text{ rows} \\ \\ n-r \text{ columns} \end{array} \overbrace{\begin{array}{c} \\ \left\{ \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} \right. \\ \\ \end{array}}^{r \text{ columns}} \begin{array}{l} \\ \left. \right\} n-r \text{ rows} \\ \\ \end{array} \tag{5.4.6}$$

Consider the following sequence of matrices $B_{22}, B_{21}B_{12}, B_{21}B_{11}B_{12}, \ldots, B_{21}B_{11}^j B_{12} \ldots$ If the first $k \geq 1$ elements in this sequence are equal to the zero matrix and B_{11} is non-singular, then $\operatorname{rank}(B_{12}) \leq \frac{r}{k}$ or $\operatorname{rank}(B_{21}) \leq \frac{r}{k}$.

Theorem 5.8. *If the conditions in Theorem 5.7 are true, then $\operatorname{rank}(M_1) \leq \frac{dr}{s-d+1}$ or $\operatorname{rank}(M_2) \leq \frac{dr}{s-d+1}$.*

Proof. By using Theorem 5.7, we know that

$$\mathcal{A} \cdot R_{d+1} \cdot \mathcal{B} = \mathcal{A} \cdot ER_{d+1} \cdot \mathcal{B} = \cdots = \mathcal{A} \cdot E^{s-d-1} \cdot R_{d+1} \cdot \mathcal{B} = 0.$$

Consider the $(n + r(d-1)) \times (n + r(d-1))$ matrix $\mathcal{S}$ whose whose first rd rows and first rd columns form the matrix E. The last $n-r$ rows form the matrix $\mathcal{A}$ and the last $n-r$ columns form the matrix $R_{d+1}\mathcal{B}$ and the remaining entries are zero. Thus we have:

$$\mathcal{S} = \begin{bmatrix} E & R_{d+1}\mathcal{B} \\ \mathcal{A} & 0 \end{bmatrix}.$$

Now we apply Lemma 5.11 with $B_{11} = E$ and $B_{12} = R_{d+1}\mathcal{B}$ and $B_{21} = \mathcal{A}$. This implies that $\operatorname{rank}(\mathcal{A}) \leq \frac{dr}{s-d+1}$ or $\operatorname{rank}(R_{d+1}\mathcal{B}) \leq \frac{dr}{s-d+1}$. Note that R_{d+1} looks like below:

$$R_{d+1} = \begin{bmatrix} * & * & \ldots & * & I_r \\ * & \vdots & * & I_r & 0 \\ \vdots & * & I_r & 0 & \vdots \\ * & I_r & 0 & \ldots & 0 \\ I_r & 0 & \ldots & 0 & 0 \end{bmatrix}.$$

In particular, R_{d+1} is non-singular. Thus $\operatorname{rank}(R_{d+1}\mathcal{B}) = \operatorname{rank}(\mathcal{B})$. Now the claim follows from Lemma 5.8. □

Corollary 5.3. *If the conditions in Theorem 5.7 are true then we have:*

$$\operatorname{rank}(Q(x_1, x_2, \ldots, x_m)) \leq r\left(1 + \frac{d}{s-d+1}\right).$$

Proof. Recall that

$$Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m) = \begin{bmatrix} I_r + L & B \\ A & C \end{bmatrix}.$$

By using Fact 5.3 and Theorem 5.8, we obtain that $\operatorname{rank}(Q(x_1 + \lambda_1, x_2 + \lambda_2, \ldots, x_m + \lambda_m)) \leq r\left(1 + \frac{d}{s-d+1}\right)$. Now the claim follows by using Lemma 5.3. □

5.5 Final algorithm

Let us recall our strategy once again. We have shown above that at least one of the following conditions holds:

(1) If $d + 1 \leq s \leq \lfloor \frac{r}{2} \rfloor - 1$ and $\hom_\ell(M) \neq 0$ for some $\ell \in [s]$, then our rank increasing step succeeds.

(2) Otherwise, we have $\operatorname{rank}(Q(x_1, x_2, \ldots, x_m)) \leq r\left(1 + \frac{d}{s-d+1}\right)$ by Corollary 5.3.

Thus if we choose s large enough then our rank increasing step succeeds, otherwise r is already a good approximation of $\operatorname{rank}(Q(x_1, x_2, \ldots, x_m))$ by Corollary 5.3. This leads to the following Algorithm 5.1, which is a natural greedy algorithm and it tries to increase the current rank as long as it can.

Algorithm 5.1 Greedy algorithm for $(1-\varepsilon)$-approximating commutative rank

Input: A $n \times n$ matrix $Q(x_1, x_2, \ldots, x_m) = (q_{ij})_{n\times n}$ whose entries q_{ij} are homogeneous polynomials of degree d in the variables $\mathbf{x} = (x_1, x_2, \ldots, x_m)$. An approximation parameter $0 < \varepsilon < 1$.

Output: $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{F}$ such that $\operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) \geq (1-\varepsilon)\operatorname{rank}(Q(x_1, x_2, \ldots, x_m))$.

1: $\ell \leftarrow \lceil \frac{d}{\varepsilon} - 1 \rceil$
2: $\lambda \leftarrow (\lambda_1, \lambda_2, \ldots, \lambda_m)$ such that $\operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) \geq 2\ell + 2$
▷ This is just to satisfy the condition $d + 1 \leq s \leq \lfloor \frac{r}{2} \rfloor - 1$ assumed in Corollary 5.3.
3: **while** Rank is increasing **do**
4: Check if there exist $(\mu_1, \mu_2, \ldots, \mu_m) \in H_{m,nd,\ell}$ such that
$\operatorname{rank} Q(\mu_1 + \lambda_1, \mu_2 + \lambda_2, \ldots, \mu_m + \lambda_m) > \operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m))$.
5: **if** $\operatorname{rank}(Q(\mu_1 + \lambda_1, \mu_2 + \lambda_2, \ldots, \mu_m + \lambda_m)) > \operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m))$ **then**
6: $\lambda \leftarrow \lambda + \mu$
7: **end if**
8: **end while**
9: **return** λ

Theorem 5.9 (Theorem 5.1 restated)**.** *Algorithm 5.1 runs in time*

$$O((mnd)^{\frac{d}{\varepsilon}} + (\frac{md^2}{\varepsilon})^{\frac{2d^2}{\varepsilon}+2d}) \cdot n \cdot M(n))$$

and returns $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{F}$ *such that*

$$\operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m)) \geq (1-\varepsilon)\operatorname{rank}(Q(x_1, x_2, \ldots, x_m)).$$

Here $M(n)$ *is the time required to compute the rank of an* $n \times n$ *matrix over* $\mathbb{F}$.

Proof. Let $(\lambda_1, \lambda_2, \ldots, \lambda_m)$ be the assignment returned by Algorithm 5.1 and $r = \operatorname{rank}(Q(\lambda_1, \lambda_2, \ldots, \lambda_m))$. We have $\ell = \lceil \frac{d}{\varepsilon} - 1 \rceil$. We know that $\hom_i(M) = 0$ for $i \in [\ell]$, otherwise Line 4 would succeed in increasing the rank of $Q(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Here M is the matrix defined in Corollary 5.1. By using Corollary 5.3, we obtain that $\operatorname{rank}(Q) \leq r\left(1 + \frac{d}{\ell - d + 1}\right)$. Thus $r \geq \left(1 - \frac{d}{\ell+1}\right)\operatorname{rank}(Q)$. By using $\ell = \lceil \frac{d}{\varepsilon} - 1 \rceil$, we know that $\ell + 1 \geq \frac{d}{\varepsilon}$. Therefore $r \geq (1-\varepsilon)\operatorname{rank}(Q)$.

The desired running time can also be proved easily. By using Lemma 5.2 on $H_{m,d(2\ell+2),d(2\ell+2)}$, the running time of Line 2 is $O((md\ell)^{2d\ell+2d} \cdot M(n))$. The outer while loop runs at most n times, thus the total running time is at most n times the running time of one iteration. The running time of one iteration is bounded by $O((m^\ell(nd+1)^\ell M(n))$. Thus the claimed bound on total running time follows. □

5.6 A PTAS for general degree d polynomials

We have demonstrated a PTAS above for the rank of an $n \times n$ matrix $Q(x_1, x_2, \ldots, x_m)$ whose entries are homogeneous degree d polynomials in the variables $x_1, x_2, \ldots, x_m$. But entries being homogeneous polynomials is not a restriction. Here we show that even if the entries of Q are general degree d polynomials, we can still use our algorithm to approximate the rank of Q. For a polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_m]$ of degree at most d, the homogenization f^H of f is a homogeneous polynomial of degree d in $\mathbb{F}[x_1, x_2, \ldots, x_m, y]$. More specifically, f^H is defined as $f^H := \sum_{i=0}^{d} \hom_i(f) \cdot y^{d-i}$. We can extend this definition to matrix polynomials in the obvious way. More precisely, the homogenization $Q^H(x_1, x_2, \ldots, x_m, y)$ of a given matrix polynomial $Q(x_1, x_2, \ldots, x_m)$ is defined as $(Q^H)_{ij} := (Q_{ij})^H$. Thus to homogenize a matrix, we just homogenize all its entries.

Lemma 5.12. *If* $Q(x_1, x_2, \ldots, x_m)$ *is matrix with its entries being polynomials of degree at most* d *in the variables* $x_1, x_2, \ldots, x_m$ *and* $|\mathbb{F}| > dn + 1$ *then* $\operatorname{rank}(Q) = \operatorname{rank}(Q^H)$.

Proof. Clearly, $\operatorname{rank}(Q) \leq \operatorname{rank}(Q^H)$ because $Q^H(x_1, x_2, \ldots, x_m, 1) = Q(x_1, x_2, \ldots, x_m)$. Now suppose that $\operatorname{rank}(Q^H) = r$. Thus there exists a non-zero $r \times r$ minor M_r of Q^H. Notice that M_r is a homogeneous polynomial of degree at most $rd \leq nd$ in the variables $x_1, x_2, \ldots, x_m, y$. Thus by using the Schwartz-Zippel lemma [184, 164], there exist scalars $(\lambda_1, \lambda_2, \ldots, \lambda_m, \mu) \in \mathbb{F}^{m+1}$ with the property that $M_r(\lambda_1, \lambda_2, \ldots, \lambda_m, \mu) \neq 0$. Here μ can be assumed to be non-zero as $|\mathbb{F}| > dn + 1$. Since M_r is homogeneous, $\mu \neq 0$ and $M_r(\lambda_1, \lambda_2, \ldots, \lambda_m, \mu) \neq 0$, we get that $M_r\left(\frac{\lambda_1}{\mu}, \frac{\lambda_2}{\mu}, \ldots, \frac{\lambda_m}{\mu}, 1\right) \neq 0$. Thus M_r would be a non-zero minor in Q as well. Hence $\operatorname{rank}(Q) \geq \operatorname{rank}(Q^H)$. Therefore $\operatorname{rank}(Q) = \operatorname{rank}(Q^H)$. □

PART III

Hard varieties in algebraic complexity theory

This part is the result of close collaboration with Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, and Frank-Olaf Schreyer. It is based on an article titled *On the Orbit Closure Containment Problem and Slice Rank of Tensors* that appeared in the *ACM-SIAM Symposium on Discrete Algorithms,* 2021.

CHAPTER 6

Orbit closure containment problem and the minrank variety

In this chapter, we consider another special case of the variety membership testing problem, the orbit closure containment problem, where the variety of interest is the closure of a given group orbit. Orbit closure containment problem, for a given vector and a group orbit, asks if the vector is contained in the closure of the group orbit. Recently, many algorithmic problems related to orbit closures have proved to be quite useful in giving polynomial time algorithms for special cases of the polynomial identity testing problem and several non-convex optimization problems. Answering a question posed by Wigderson [181], we show that the algorithmic problem corresponding to the orbit closure containment problem for 3-tensors with $\mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ action is NP-hard. We show this by establishing a computational equivalence between the solvability of homogeneous quadratic equations and a homogeneous version of the matrix completion problem, while showing that the latter is an instance of the orbit closure containment problem.

6.1 Set-up and results

The problems related to group orbits have been ubiquitous in mathematics and computer science, both from the perspective of theory and practice. For a group G acting[1] on a vector space V, the orbit of a vector $v \in V$, denoted as Gv, is defined to be the set $\{gv \mid g \in G\}$. That is, the orbit Gv is the set of points that v gets mapped to on the action of G. The group problem that has received the widest attention in computer science is the orbit containment problem.

Problem 6.1. Orbit Containment: *For a group G acting on a vector space V, and given two elements $u, v \in V$ as inputs, decide if $u \in Gv$.*

Thus, it asks if a vector is in the orbit of another vector. This problem is quite general and captures many problems, for instance the graph isomorphism problem and the module isomorphism problem. We can see the graph isomorphism problem as an instance of the orbit containment problem as follows. Suppose we are given two graphs G_1 and G_2 on n vertices each, and we want to know if they are isomorphic to each other. This can be rephrased as whether the adjacency matrix of the graph G_1 is in the orbit of the adjacency matrix of the graph G_2, under the action of the permutation group S_n. Here S_n acts by permuting the rows and columns of the matrix, induced by the permutation of vertices of the graph corresponding to the matrix. Owing to its generality, the orbit

[1] When we say a group G acts on the ambient space S, we have a mapping $\cdot : G \times S \to S$ that satisfies the axioms $1 \cdot s = s$ and $(gh) \cdot s = g \cdot (h \cdot s)$ for all $s \in S$ and $g, h \in G$. Here gh is the group operation.

containment problem has been very important from the point of view of both algorithm design as well as complexity theory for decades. While the graph isomorphism problem remains one of the central algorithmic problem in graph theory, the module isomorphism problem has been very crucial in cryptography [152, 40, 113, 171, 22, 176].

From the perspective of topology, it is more natural to consider *orbit closures* instead. For a group G acting on a vector space V, the orbit closure of $v \in V$, denoted as $\overline{Gv}$, is defined to be the smallest closed subset of V which contains Gv. In the standard Euclidean topology, this translates to $\overline{Gv}$ being the smallest superset of Gv which contains the limit points of all convergent sequences comprising of elements of Gv. In the Zariski topology, this translates to $\overline{Gv}$ being the smallest superset of Gv which contains all the common zeros of the set of polynomials that vanish on all the elements of Gv. In most of the cases of interest, in particular, when the underlying field is $\mathbb{C}$, the definitions of $\overline{Gv}$ obtained by considering the above two topologies, that is, the Euclidean (or analytic) closure and the Zariski (or algebraic) closure coincide[2]. Thus we can ask the following weakening of the orbit containment problem.

Problem 6.2. Orbit Closure containment: *For a group G acting on a vector space V, and given two elements $u, v \in V$ as inputs, decide if $u \in \overline{Gv}$.*

This problem is quite general, too, and has appeared centrally in algorithmic and complexity theoretic problems related to algebra and combinatorial optimization, since it captures problems like border rank of tensors, the null cone problem, and the permanent versus determinant problem. As an example, to see the border rank problem as an orbit closure containment problem, let GL_n denote the group of all invertible $n \times n$ matrices. GL_n acts on $\mathbb{F}^n$ by the usual matrix-vector multiplication. $G_n := \mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ acts on rank-one tensors $u \otimes v \otimes w$ by $(A, B, C) \cdot u \otimes v \otimes w = Au \otimes Bv \otimes Cw$ and on arbitrary tensors by linear continuation. The orbit of a tensor t under G_n is the set $G_n t := \{g \cdot t \mid g \in G_n\}$ and its orbit closure is the closure $\overline{G_n t}$ in the Zariski topology. It is well known that the set of all tensors of border rank $\leq r$ can be written with the help of an orbit closure [46], namely $\overline{G_r e_r}$ where e_r is the so-called unit tensor in $\mathbb{F}^{r \times r \times r}$: A tensor $t \in \mathbb{F}^{n \times n \times n}$ has border rank $\leq r$ iff $\tilde{t} \in \overline{G_r e_r}$, where $\tilde{t}$ is an embedding of t into the larger ambient space $\mathbb{F}^{r \times r \times r}$.

The null cone problem is a special case of the orbit closure intersection problem where vector u is always the 0 vector. That is, we ask the following:

Problem 6.3. Null Cone: *For a group G acting on a vector space V, and given $v \in V$ as input, decide if $0 \in \overline{Gv}$.*

For an example set up of the null cone problem, let us think of a tensor $t \in \mathbb{F}^{n \times n \times m}$ as a set of m matrices $A_1, \ldots, A_m$ of size $n \times n$, stacked up on top of each other (also called slices). The group $\Gamma_n := \mathsf{SL}_n \times \mathsf{SL}_n$ acts on t by simultaneously multiplying each of the matrices from the left and the right. King [120] showed that the *noncommutative* rank of the matrix space given by $A_1, \ldots, A_m$ is maximal iff $0 \in \overline{\Gamma_n t}$. (All such tensors t are said to lie in the *null cone*.) Garg et al. [80] show how to decide the null-cone problem in this setting in polynomial time, hence giving a deterministic noncommutative identity testing algorithm. Ivanyos, Qiao, and Subrahmanyam [114], based on work by Derksen and Makam [62], give a different algorithm for this problem, which works over

[2]Unless stated otherwise, we assume the underline field to be $\mathbb{C}$ in this paper

arbitrary characteristic. (Unfortunately, we do not know whether something similar can be achieved in the commutative setting. More unfortunately, Makam and Wigderson proved recently that the commutative case cannot be written as a null-cone problem [140].)

Orbit closure containment problems have played a central role in algebraic complexity theory in the recent years. On the algorithmic side, orbit closure containment has been crucial in several advancements in the fast matrix multiplication algorithms due to the notion of border rank of tensors, see e.g. [26]. On the complexity theoretic side, the famous permanent versus determinant problem can also be phrased as an orbit closure containment problem. This is the starting point of the geometric complexity program initiated by Mulmuley and Sohoni [142]. While all the above mentioned problems remain far from being completely understood, the interest towards studying algorithmic problems related to orbit closures has seen a rise in the past few years. Thanks to the sequence of works connecting several areas of mathematics, combinatorial optimization, and complexity theory, many special instances of the orbit closure containment problem, in particular the null cone problem, have proved to be useful in giving polynomial time algorithms for special cases of the polynomial identity testing problem and several non-convex optimization problems. See [48, 82, 79, 47, 50, 9, 81, 49, 80] for details.

As a result, Wigderson in his invited talk in CCC'17 posed the orbit containment problem, the orbit closure containment problem and the null cone problem to the community [181].

While there has been a lot progress recently towards the null cone problem and we have efficient algorithms in many setups, most of the instances of the orbit closure containment problem is not understood from the algorithmic perspective. In particular, we neither know the NP-hardness nor a polynomial time algorithm for the tensor border rank problem. This is in contrast to the tensor rank problem, where we know the NP-completeness for 30 years now. Similarly, we currently do not know whether it is hard to test whether a polynomial lies in the orbit closure of the determinant, which is an algebraic variant of the so-called minimum circuit size problem. The main challenge for proving hardness or getting an algorithm for the problems related to orbit closure is that it is difficult to get a hold on how the closure will behave.

6.1.1 Results, methods and relation to previous works

We make progress on the above Problem 6.2, that is, the orbit closure containment problem, by showing its NP-hardness, while we observe an upper bound for Problem 6.1, that is. the orbit containment problem.

Orbit closure containment problem

Our first contribution is that we rule out the possibility of an efficient algorithm for the general case of the orbit closure containment problem under the assumption that $\mathsf{P} \neq \mathsf{NP}$, answering a question posed by Wigderson. We do so by showing that testing whether a 3-tensor t lies in the orbit closure of another 3-tensor t' under the group action $\mathrm{GL}_k \times \mathrm{GL}_m \times \mathrm{GL}_n$ is NP-hard.

Theorem 6.1. *Given two tensors t and t', deciding whether the orbit closure of t is contained in the orbit closure of t' (under the usual $\mathrm{GL}_k \times \mathrm{GL}_m \times \mathrm{GL}_n$ action) is* NP*-hard.*

We show this by defining a quantity called minrank (see Section 6.2 and Section 6.4) and proving that deciding whether the minrank is bounded by some given bound b can be phrased as an orbit closure containment problem. We then show that it is an NP-hard question (see Section 6.2.1) by showing that it is polynomial time equivalent to the solvability of homogeneous quadratic equations. Since the solvability of homogeneous quadratic equations is NP-hard, we get that the orbit closure containment is NP-hard as well.

This is in contrast to the recent results on the null cone problem, for which polynomial time algorithms have been discovered for several group actions. Since the null cone has equivalent characterizations via invariant theory, we have more tools there. On the other hand, for the orbit closure containment problem corresponding to a group action, no such characterization is available, and we need to understand the corresponding orbit closure better. Unfortunately, in most of the interesting settings, our understanding of the closure of the set is in quite limited. For instance, we do not understand the tensor border rank well, neither do we understand the closures of algebraic complexity classes. Thus, the main challenge is to find a set up where one has a good control over the orbit closure. In this work, we find one such set up.

The initial inspiration of the set up that we find is the NP-hardness of the completion rank and the border completion rank [31]. Let us briefly look at those notions.

We can phrase the *matrix completion problem* as a problem on tensors or on tuples of matrices. Many variants of matrix completion problem has been studied in the literature. In its most general form, we are given a tuple of $n \times n$ matrices $(A_1, A_2, \ldots, A_m)$. We can view $(A_1, A_2, \ldots, A_m)$ as a tensor in $\mathbb{F}^{n \times n \times m}$ with slices $A_1, \ldots, A_m$ of size $n \times n$, stacked up on top of each other. Then the matrix completion problem can be phrased as follows:

Problem 6.4. Matrix Completion: *Given a tensor t as a tuple of $n \times n$-matrices $(A_1, A_2, \ldots, A_m)$, and a number r, decide if there exist $\lambda_2, \ldots, \lambda_m \in K$ such that* $\mathrm{rk}(A_1 + \lambda_2 A_2 + \cdots + \lambda_m A_m) \leq r$.

Here rk denotes the usual matrix rank. The minimum achievable value of r above is called the *completion rank* of t. Matrix completion has many applications, for instance, in machine learning and network coding, we here just refer to [153, 102, 100], which contain relevant hardness results. When we consider minimization, the problem is NP-hard, even when the resulting matrix has rank 3 [153]. When we consider maximization, then the problem is NP-hard over finite fields [102]. Over large enough fields, there is a simple randomized polynomial time algorithm that simply works by plugging in random elements from a large enough set. The correctness of this algorithm follows from the well-known Schwartz-Zippel lemma.

In [31], it is shown that given t and a bound r, deciding whether the completion rank of t is bounded by r is NP-hard. Furthermore—and this is the interesting case here—even testing whether t is in the closure of the set of all tensors of completion rank $\leq r$ is NP-hard. The smallest r such that this is the case, is called the *border completion rank*.

It is shown in [31] that given t and a bound r, deciding whether the border completion rank of t is bounded by r is NP-hard. Thus, completion rank is one of the rare examples where we understand the border well. Thus the hope was to exploit this understanding.

However, the above result could not help us simply because the border completion rank problem cannot be phrased as an interesting orbit closure problem. We overcome this challenge by defining a homogeneous version of matrix completion problem, which we call as the *minrank* problem, where, in contrast to the completion rank, we allow any nontrivial linear combination of the slices.

Problem 6.5. MINRANK: *Given $A_1, \dots, A_k$ of the same size $m \times n$ and a number r, decide whether there exists a nonzero linear combination $x_1A_1 + \cdots + x_kA_k$ with rank at most r. The smallest r for which the answer is YES is called the* minrank *of $A_1, \dots, A_k$.*

Here again, instead of thinking of a tuple of matrices, we can also view $A_1, \dots, A_k$ as a tensor in $\mathbb{F}^{k\times m\times n}$ with $A_1, \dots, A_k$ being its slices. We will use both views in this paper. We show that the obtained homogeneous version of the problem can indeed be phrased as an orbit closure containment problem. For this, we first show that the set of matrix tuples (or tensor) with minrank at most r is a Zariski closed set by showing that the set can be viewed as a projective variety. Next, in order to show that we can phrase the minrank problem as an orbit closure problem, we give an explicit tensor $T_{k,n,r}$ such that every tensor (or matrix tuple) with minrank at most r lies inside the orbit closure of this tensor $T_{k,n,r}$. We now elaborate on the above.

For a tensor $T \in \mathbb{F}^{k\times m\times n}$ given as $e_1 \otimes A_1 + \cdots + e_k \otimes A_k$ (indicating that $A_1, A_2, \dots, A_k$ correspond to different slices of T) and a linear form $x \in (\mathbb{F}^k)^*$, we define the *contraction* Tx by $Tx := x(e_1)A_1 + \cdots + x(e_k)A_k$, where $x(e_i)$ denotes the i-th coordinate of x. That is, we form a linear combination of the slices. If we take the set of all (T, x) with $\mathrm{rk}(Tx) \le r$ and $x \neq 0$ and project on the first component, we get all tensors of minrank at most r. Since the set of all such (T, x) is invariant under scaling of T or x by nonzero factors, it also defines a *projective* variety, and the projection on the first component is a projective variety, too (see Section 6.4 for more details). So we are in the nice situation where the set of all tensors of minrank at most r is Zariski closed (Theorem 6.9). Thus we do not need an additional border complexity measure, i. e., minrank and border minrank coincide. This is different to the situation with completion rank and border completion rank or tensor rank and border rank. We denote the corresponding variety of all tensors $T \in U \otimes V \otimes W$ of minrank at most r by $\mathcal{M}_{U\otimes V\otimes W,r}$ or just $\mathcal{M}_r$ when the tensor space is clear from the context.

Next, we want to write the minrank varieties $\mathcal{M}_{U\otimes V\otimes W,r}$ as orbit closures. Note that we can always embed a tensor $T \in U \otimes V \otimes W$ into a larger ambient space $U \otimes L \otimes L$, where V and W are subspaces of L, by filling the new entries with zeros. (This process is called *padding*.) We then show (Corollary 6.3), that $\mathcal{M}_{U\otimes V\otimes W,r}$ is the $\mathrm{GL}(U)\times\mathrm{GL}(L)\times\mathrm{GL}(L)$-orbit closure of the tensor

$$T_{k,n,r} = e_1 \otimes \left(\textstyle\sum_{j=1}^{r} e_{1j} \otimes e_{1j}\right) + \sum_{i=2}^{k} e_i \otimes \left(\sum_{j=1}^{n} e_{ij} \otimes e_{ij}\right)$$

intersected with the ambient space $U \otimes V \otimes W$ (here $k = \dim U$, $n = \dim L$). This means that we can reduce the question whether a tensor has minrank at most r to the question whether it is contained in the orbit closure of $T_{k,n,r}$.

Now, one might hope that the proof of hardness of border completion rank in [31] can be adapted to the homogeneous setting. However, unfortunately, this NP-hardness proof breaks down in the homogeneous setting since the hard instance in this proof critically used the fact that we are in the affine setting, since A_1 was a matrix that had rank linear in the input size whereas all other matrices had the same, constant rank. Thus the hardness proofs do not work in the homogeneous setting, since all instances created in the proofs trivially have the same minrank. Thus, we need to do something completely different. We solve the problem by showing the equivalence of solvability of homogeneous quadratic equations and the minrank problem, hence establishing the NP-hardness of the minrank problem. In fact, it turns out that even deciding whether the minrank is ≤ 1 is already NP-hard. Thus, we get that the orbit closure containment problem in NP-hard as well. See Section 6.2.1 for details.

When the underlying field is the set of real numbers, and we are taking the Euclidean closure, then we can say something more about the orbit closure containment problem. In Section 6.3.1, we show the equivalence of the orbit closure containment problem and the existential theory over reals (see [161]) in this case.

Theorem 6.2. *The (Euclidean) orbit closure containment problem over the reals is polynomial-time equivalent to the existential theory over the reals.*

For the *tensor rank problem*, such an equivalence with the existential theory over reals was recently established by Shitov [168].

Orbit Containment Problem

We also show an upper bound for the algorithmic problem of the orbit containment problem in Section 6.5 by reducing it to the solvability of polynomial equations. Since the solvability of polynomial equations is known to be in the complexity class AM[3], assuming the generalized Riemann hypothesis (GRH), by a result of Koiran [122], we deduce that over the field of complex numbers, the orbit containment problem can be shown to be in the complexity class AM under the same assumption.

Theorem 6.3. *Over* $\mathbb{C}$, ORBIT CONTAINMENT PROBLEM $\in$ AM, *assuming the generalized Riemann hypothesis.*

6.1.2 Organization of the chapter

We give the algorithmic hardness of the minrank problem in Section 6.2. In Section 6.4, we show that the minrank problem is an instance of the orbit closure containment problem. Combining the above two, we conclude that the NP-hardness of the orbit closure containment problem in Corollary 6.2. We discuss the case when the underlying field is $\mathbb{R}$ in Section 6.3.1, showing an equivalence between the orbit closure containment problem and the existential theory over reals. We finally close with an algorithmic upper bound in the case of orbit containment problem in Section 6.5.

[3] AM refers to the complexity class containing the set of decision problems decidable in polynomial time by an Arthur-Merlin protocol with 2 messages. It is contained in the complexity class Π_2P, and is hence contained in the second level of polynomial hierarchy. See [15] for details.

6.2 Complexity of the minrank problem

We consider the following problem: given a tuple of matrices $A_1, \dots, A_k$ of the same size $m \times n$ and a number r, does there exist a nonzero linear combination $x_1A_1 + \cdots + x_kA_k$ with rank at most r? This is a homogeneous variant of the generalized matrix completion problem considered in [31], where instead of a linear combination we have an affine expression $A_0 + x_1A_1 + \cdots + x_kA_k$. A restricted variant of this problem was first considered in [53], where it is proven that the problem is NP-hard. The related problem of low rank matrix completion is widely studied in optimization.

Clearly, the answer depends on the field from which we take the coefficients of the linear combination. For example, the pair of matrices

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

does not have any nontrivial linear combinations of rank 1 over $\mathbb{R}$, but over $\mathbb{C}$ we have $\operatorname{rk}(A_1 + iA_2) = 1$. We will mostly work over algebraically closed fields such as $\mathbb{C}$, but many results are also true over other fields.

Let $\mathbb{F}$ be a field. Instead of talking about matrices $A_1, \dots, A_k \in \mathbb{F}^{m \times n}$, we can also phrase the homogeneous minrank problem in terms of a linear subspace $\langle A_1, \dots, A_k \rangle$, a matrix of linear forms $A \colon \mathbb{F}^k \to \mathbb{F}^{m \times n}$ where $A(x) = \sum_{i=1}^{k} x_i A_i$ or a tensor $T \in \mathbb{F}^k \otimes \mathbb{F}^m \otimes \mathbb{F}^n$ such that $T = \sum e_i \otimes A_i$. We will mainly use the tensor language.

Recall the definition of minrank.

Definition 6.1. *Let U, V, W be finite-dimensional vector spaces over some field $\mathbb{F}$. The minrank of a tensor $T \in U \otimes V \otimes W$ is the minimal number r such that there exists a nonzero $x \in U^*$ with $\operatorname{rk}(Tx) = r$.*

Let S be a finite or countable subset of $\mathbb{F}$.

Problem 6.6. ($\text{HMinRank}_{S,\mathbb{F}}$) *Given a tensor $T \in \mathbb{F}^{k \times m \times n}$ with all components in S and a number r, decide if the minrank of T is at most r.*

In Section 6.2.1 we will prove that this problem is NP-hard. Moreover, it is hard even when r is fixed to one.

Problem 6.7. ($\text{HMinRank1}_{S,\mathbb{F}}$) *Given a tensor $T \in \mathbb{F}^{k \times m \times n}$ with all components in S, decide if the minrank of T is at most 1.*

6.2.1 Equivalence of minrank and solvability of quadratic equations

In this section we prove NP-hardness of HMinRank by reducing it to the following problem:

Problem 6.8. ($\text{HQuad}_{S,\mathbb{F}}$) *Given a set of quadratic forms with coefficients from S, represented by lists of coefficients, determine if it has a nonzero common zero over $\mathbb{F}$.*

To implement the reduction, we need to perform linear algebra computations with elements of the field.

Definition 6.2. *An* effective field *is a finite or countable field $\mathbb{F}$ with a binary encoding of elements of $\mathbb{F}$ such that the following operations can be performed in time polynomial in the length of the encoding of arguments:*

- *multiplication and addition of two elements over $\mathbb{F}$,*
- *multiplication of an arbitrary number of matrices over $\mathbb{F}$,*
- *equality comparison of two elements of $\mathbb{F}$,*
- *division of two elements of $\mathbb{F}$ (if the denominator is zero, the algorithm should fail).*

Furthermore, we want that polynomial identity testing is in BPP*, that is, there is a* BPP*-machine that given an algebraic circuit computing a polynomial over $\mathbb{F}$, decides whether this polynomial is identically zero.*

In our paper, we usually deal with polynomials over uncountable fields like $\mathbb{C}$. In the algebraic complexity setting, this is no problem. However, when we want to compute with Turing machines, we have to restrict ourselves to appropriate subfields. This is modelled by effective fields. In particular, $\mathbb{Q}$ is effective and the natural effective subfield of $\mathbb{R}$ and $\mathbb{Q}+i\mathbb{Q}$ is a natural choice for $\mathbb{C}$. Finite fields are effective, when we drop the last condition about identity testing, which we only need in the second part of this section.

Efficient multiplication of several matrices implies that products and linear combinations of elements can also be computed in polynomial time. It also allows for various polynomial-time linear algebra procedures. In particular, we are interested in the following:

Theorem 6.4. *For an effective field K there is a polynomial time algorithm which, given a matrix A over K, computes a basis of* $\operatorname{Ker} A$.

Proof. Determinants of matrices over an effective field are computable in polynomial time, because determinant can be represented as an iterated matrix multiplication of polynomial size (see e. g. [111]). This allows computing the inverse of a nonsingular matrix. Also, we can find one of the maximal nonzero minors of a given nonzero matrix, by starting from any nonzero entry and trying to enlarge the minor by checking all rows and columns at each step. We can then compute the basis of the kernel by basic linear algebra. □

Hillar and Lim [107, Thm. 2.6] proved that HQuad is NP-hard over the fields $\mathbb{R}$ and $\mathbb{C}$. Their proof also works for any field of characteristic different from 3 containing cubic roots of unity. The NP-hardness for arbitrary fields was proven by Grenet, Koiran and Portier in [91]. We give another proof for arbitrary fields based on the idea of Hillar and Lim. Compared to [91], we describe a general construction for all fields instead of treating characteristic 2 as a special case, and only use coefficients from $\{-1, 0, 1\}$.

Theorem 6.5. $\text{HQuad}_{\{0,1,-1\},\mathbb{F}}$ *is* NP*-hard for any field $\mathbb{F}$.*

Proof. We reduce from graph 3-colorability.

Given a graph $G = (V, E)$, we will construct a system of quadratic homogeneous equation, solutions of which correspond to colorings of the graph. The set of variables consists of two variables x_v and y_v for each vertex $v \in V$ and one additional variable z. Consider a system of homogeneous quadratic equations which contains for each vertex v the three equations

$$x_v y_v = 0$$
$$x_v^2 - x_v z = 0$$
$$y_v^2 - y_v z = 0$$

and for each edge $(v, w) \in E$ the equation

$$x_v^2 + y_v^2 + x_w^2 + y_w^2 - x_v y_w - x_w y_w - z^2 = 0$$

If $z = 0$, then from vertex equations we deduce $x_v = y_v = 0$ for all $v \in V$. Therefore, a nontrivial solution must have nonzero z. We can scale it so that $z = 1$. When $z = 1$, the vertex equations give $(x_v, y_v) \in \{(0,0), (0,1), (1,0)\}$. Restricted to these values, the left-hand side of the edge equation has the following values:

v \ w	$(0,0)$	$(0,1)$	$(1,0)$
$(0,0)$	-1	0	0
$(0,1)$	0	1	0
$(1,0)$	0	0	1

That is, the edge equation forces the tuples (x_v, y_v) and (x_w, y_w) to be different. Thus, nontrivial solutions with $z = 1$ are in one-to-one correspondence with colorings of the graph G into three colors, given by the three possible solutions of the vertex equations. □

Theorem 6.6. *Let $\mathbb{F}$ be a field and K be an effective subfield of $\mathbb{F}$. Then* $\text{HMinRank1}_{K,\mathbb{F}}$ *is polynomial-time equivalent to* $\text{HQuad}_{K,\mathbb{F}}$.

Proof. To reduce from HMinRank1 to HQuad, note that the condition $\operatorname{rk}(Tx) \le 1$ can be expressed by homogeneous quadratic equations on x, namely, vanishing of 2×2 minors of the matrix of linear forms Tx.

Now we describe the reduction from HQuad to HMinRank1. Let k be a number of given quadratic forms and n be the number of variables. Each quadratic form $q(x) = \sum_{1 \le i \le j \le n} a_{ij} x_i x_j$ on $\mathbb{F}^n$ corresponds to a linear form $Q(X) = \sum_{1 \le i \le j \le n} a_{ij} x_{ij}$ on the space $\mathsf{Sym}^2 \mathbb{F}^n \subset \mathbb{F}^n \otimes \mathbb{F}^n$ of symmetric matrices, and a vector x is a zero of q if and only if $x \otimes x$ is a zero of Q. Therefore, a set of k linear forms on $\mathbb{F}^n$ corresponds to a linear map $L \colon \mathsf{Sym}^2 \mathbb{F}^n \to \mathbb{F}^k$ given by a matrix consisting from the coefficients of quadratic forms, and x is a common zero if and only if $x \otimes x$ is contained in $\operatorname{Ker} L$. Since all the coefficients lie in K, the map L is an extension of a linear map $\mathsf{Sym}^2 K^n \to K^k$, and its kernel has a basis consisting of vectors in $\mathsf{Sym}^2 K^n$, which, by Theorem 6.4, can be computed in polynomial time. Let $A_1, \dots, A_m$ be such basis and $T = \sum_{i=1}^m e_i \otimes A_i \in K^m \otimes K^n \otimes K^n$. Nontrivial common zeros $x \in \mathbb{F}^n$ of the original set of quadratic forms corresponds to rank 1 symmetric matrices $x \otimes x$ which can be presented as a nontrivial linear combination

$\sum_{i=1}^{m} y_i A_i$ with $y_i \in \mathbb{F}$ or, equivalently, as a contraction Ty with nonzero $y \in \mathbb{F}^m$. This is the resulting instance of HMinRank1 problem. □

Corollary 6.1. *Let $\mathbb{F}$ be a field and K be an effective subfield of $\mathbb{F}$. Then* $\text{HMinRank1}_{K,\mathbb{F}}$ *is* NP*-hard.*

The HMinRank problem is also hard in other regimes.

Theorem 6.7. *Let $\mathbb{F}$ be a field of characteristic 0 and K be an effective subfield of $\mathbb{F}$. Then* $\text{HMinRank}_{K,\mathbb{F}}$ *is* NP*-hard for $n \times (2n+1) \times (2n+1)$ tensors and $r = n+1$.*

Proof. The proof is based on a similar theorem for finite fields is sketched in [59, §3.3], which uses the NP-completeness of the minimum distance problem for linear codes proved in [180].

We reduce from a variant of the Subset Sum problem: given a set of $2n$ integers, and a number S, determine if a subset of these integers sum up to S. NP-completeness of this variant is noted in [78, SP13].

From the input $\{a_1, \ldots, a_{2n}\}$ of the Subset Sum problem, construct a $(n+1) \times (2n+1)$ matrix

$$A = \begin{bmatrix} 1 & 1 & \ldots & 1 & 0 \\ a_1 & a_2 & \ldots & a_{2n} & 0 \\ a_1^2 & a_2^2 & \ldots & a_{2n}^2 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1^{n-2} & a_2^{n-2} & \ldots & a_{2n}^{n-2} & 0 \\ a_1^{n-1} & a_2^{n-1} & \ldots & a_{2n}^{n-1} & 1 \\ a_1^n & a_2^n & \ldots & a_{2n}^n & S \end{bmatrix}$$

From the properties of Vandermonde determinants we see that any $(n+1) \times (n+1)$ minor is nonzero if it does not contain the last column. If a minor does contain the last column and columns $i_1, \ldots, i_n$, it vanishes if and only if $S = a_{i_1} + \cdots + a_{i_n}$ [180, Lem. 1].

Thus, the matrix A has rank $n+1$. Moreover, it has $n+1$ linearly dependent columns if and only if the original Subset Sum problem has a solution.

Let $b_1, \ldots, b_n$ be a basis of $\operatorname{Ker} A$. Since subsets of k linearly dependent columns corresponds to vectors in $\operatorname{Ker} A$ which have at most k nonzero coordinates, the original problem has a solution if and only if there is a nonzero linear combination of b_i with at most $n+1$ nonzero coordinates.

Let B_i be a $(2n+1) \times (2n+1)$ matrix constructed from b_i by placing its coordinates on the diagonal. The rank of a linear combination of B_i is equal to the number of nonzero coordinates in the corresponding linear combination of vectors b_i. Thus, the answer to the HMinRank problem for the $n \times (2n+1) \times (2n+1)$ tensor $\sum_{i=1}^{n} e_i \otimes B_i$ and $r = n+1$ determines the answer to the original problem. □

6.3 Complexity of the orbit closure containment problem

Thus, Theorem 6.6 proves that the homogeneous minrank problem is computationally equivalent to the solvability of homogeneous quadratic equations. This, combined with the fact that the minrank problem can be phrased as an orbit closure containment problem (Section 6.4), proves that the orbit closure containment problem ("$w \in \overline{Gv}$") is at least as hard as the solvability of homogeneous quadratic equations. In particular, the orbit closure containment problem is NP-hard.

Corollary 6.2 (Theorem 6.1 restated)**.** *Given two tensors t and t', deciding whether the orbit closure of t is contained in the orbit closure of t' (under the usual $\mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ action) is* NP*-hard.*

6.3.1 Orbit closure containment and existential theory over reals

Over the reals, we can say even more, when closure means Euclidean closure. Let ETR denote the problem of the existential theory over the reals, ETR is the set of true sentences of the form $\exists x_1, \ldots, x_n : \phi(x_1, \ldots, x_n)$, where ϕ is a quantifier-free Boolean formula over the signature 0, 1, $+$ $*$, $<$, $=$ interpreted in the intended way over the real numbers. w being in the orbit closure can be expressed by

$$\forall \varepsilon > 0 \, \exists g \in G : \det(g) \neq 0 \wedge \|w - gv\|_2^2 < \varepsilon.$$

Except for the first quantifier, this is a statement in ETR. By the results of Grigoriev and Vorobjov [92], see also [19, Thm 3.15], this universal quantifier can be removed and ε can be replaced by a double exponentially small constant, which can be expressed in ETR.

On the other hand, we can also reduce ETR to orbit (Euclidean) closure containment over the reals. By results of Schaefer [161, Cor. 3.10], Hilbert's Homogeneous Nullstellensatz $\mathrm{H}_2\mathrm{N}$ over the reals is equivalent for ETR. In Schaefer's construction all equations have degree two except for one, which has degree four. However, it is easy to see that the degree of this equation can be reduced to two, too, see [162]. Therefore, from our reduction in Theorem 6.6, it follows that orbit closure containment over the reals is computationally equivalent to ETR.

Theorem 6.8 (Theorem 6.2 restated)**.** *The (Euclidean) orbit closure containment problem over the reals (with coefficients computable by polynomial-size circuits) is polynomial-time equivalent to the existential theory over the reals* ETR*.*

Since the minrank problem can be phrased as an orbit closure containment problem when we have the action of $\mathrm{GL}_n \times \mathrm{GL}_n \times \mathrm{GL}_n$ on 3-tensors (as shown in Section 6.4), the above equivalence between orbit closure containment problem and ETR still holds if one restricts the orbit closure containment problem to the tensor action.

6.4 Minrank as an orbit closure containment problem

In this section, we show that over algebraically closed fields, the answer to the homogeneous minrank problem is determined by membership in a certain orbit closure.

We first show that the set of tensors with minrank at most r is Zariski closed.

Theorem 6.9. *Let U, V, W be vector spaces over an algebraically closed field $\mathbb{F}$. The set of all tensors $T \in U \otimes V \otimes W$ with minrank at most r is Zariski closed.*

Proof. Define an affine variety

$$\mathcal{X}_{U\otimes V\otimes W,r} = \{(T, x) \in (U \otimes V \otimes W) \times U^* \mid \operatorname{rk}(Tx) \leq r\}.$$

Since the condition $\operatorname{rk}(Tx) \leq r$ is scale-invariant with respect to both T and x, we can define the corresponding projective variety

$$\mathbb{P}\mathcal{X}_{U\otimes V\otimes W,r} = \{([T], [x]) \in \mathbb{P}(U \otimes V \otimes W) \times \mathbb{P}U^* \mid \operatorname{rk}(Tx) \leq r\} \subset \mathbb{P}(U \otimes V \otimes W) \times \mathbb{P}U^*$$

Let $\pi\colon \mathbb{P}(U \otimes V \otimes W) \times \mathbb{P}U^* \to \mathbb{P}(U \otimes V \otimes W)$ be the projection onto the first component of the product. Consider the image of $\mathbb{P}\mathcal{X}_{U\otimes V\otimes W,r}$ under π:

$$\pi\mathbb{P}\mathcal{X}_{U\otimes V\otimes W,r} = \{[T] \in \mathbb{P}(U \otimes V \otimes W) \mid \exists x \neq 0\colon \operatorname{rk}(Tx) \leq r\}.$$

As an image of a projective variety, it is a closed subvariety of $\mathbb{P}(U \otimes V \otimes W)$ (see e. g. [165, Thm. 1.10]). The affine cone over this subvariety is therefore also closed. This affine cone is exactly the set of tensors of minrank at most r. □

Definition 6.3. *We call the projective variety*

$$\mathbb{P}\mathcal{M}_{U\otimes V\otimes W,r} = \{[T] \in \mathbb{P}(U \otimes V \otimes W) \mid \exists x \neq 0\colon \operatorname{rk}(Tx) \leq r\}$$

the projective minrank variety*, and the corresponding affine cone*

$$\mathcal{M}_{U\otimes V\otimes W,r} = \{T \in U \otimes V \otimes W \mid \exists x \neq 0\colon \operatorname{rk}(Tx) \leq r\}$$

the affine minrank variety*, or just the* minrank variety*. We omit the index $U \otimes V \otimes W$ if it is clear from context.*

Some simple properties of minrank varieties follow directly from the definition:

Lemma 6.1. *Let V' and W' be subspaces of V and W respectively. Then*

$$\mathcal{M}_{U\otimes V'\otimes W',r} = \mathcal{M}_{U\otimes V\otimes W,r} \cap (U \otimes V' \otimes W').$$

Proof. Trivial. A tensor lies in $\mathcal{M}_{U\otimes V'\otimes W',r}$ iff it is an element of the space $U \otimes V' \otimes W'$ and has minrank at most r, i. e., lies in $\mathcal{M}_{U\otimes V\otimes W,r}$. □

Lemma 6.2. *Let $\dim U = k$, $\dim V = n$ and $\dim W > s = n(k-1) + r$. Then*

$$\mathcal{M}_{U\otimes V\otimes W,r} = \bigcup_{\substack{W'\subset W\\ \dim W'=s}} \mathcal{M}_{U\otimes V\otimes W',r}.$$

Proof. Let T be a tensor in $\mathcal{M}_{U\otimes V\otimes W,r}$ and x_1 be a nonzero vector in U^* such that $\operatorname{rk}(Tx_1) \leq r$. Choose $x_2, \ldots, x_k$ such that $\{x_i\}$ is a basis of U^* and set $A_i = Tx_i \in V \otimes W$. Since $\operatorname{rk} A_1 \leq r$, there exists a subspace $W_1 \subset W$ of dimension at most r such that $A_1 \in V \otimes W_1$. Analogously, for $i > 1$ we have $A_i \in V \otimes W_i$ for some subspace $W_i \subset W$ of dimension at most n. The sum W' of all W_i is a subspace of dimension at most s. We extend it to dimension s in arbitrary way if needed. The tensor T lies in $U \otimes V \otimes W'$ and, therefore, in $\mathcal{M}_{U\otimes V\otimes W',r}$. □

Lemma 6.3. *The variety $\mathcal{M}_{U\otimes V\otimes W,r}$ is invariant under the standard action of* $\mathrm{GL}(U)\times\mathrm{GL}(V)\times\mathrm{GL}(W)$ *on* $U\otimes V\otimes W$.

Proof. Straightforward. If $\mathrm{rk}(Tx)\le r$, then $(F\otimes G\otimes H)T\cdot(Fx)=(G\otimes H)(Tx)$ also has rank at most r (here Fx denotes the dual action of $GL(U)$ on U^*). □

6.4.1 Minrank varieties and orbit closures

The minrank varieties are related to orbit closures of some tensors. Let $L=(\mathbb{F}^n)^{\oplus(k-1)}\oplus\mathbb{F}^r$ be a vector space of dimension $s=n(k-1)+r$ decomposed into k summands of dimension n each, except the first one, which is of dimension r. Let L_i be the i-th summand and denote the standard basis of L_i by e_{ij}, $1\le j\le\dim L_i$. Let $U=\mathbb{F}^k$ be a k-dimensional space with a standard basis e_i. Define the tensor $T_{k,n,r}\in U\otimes L\otimes L$ as

$$T_{k,n,r}=e_1\otimes(\sum_{j=1}^{r}e_{1j}\otimes e_{1j})+\sum_{i=2}^{k}e_i\otimes(\sum_{j=1}^{n}e_{ij}\otimes e_{ij}),$$

that is, the first slice of $T_{k,n,r}$ consists of an $r\times r$ identity matrix at the top-left corner, with zero everywhere else. Whereas, for $i>1$, the i-th slice of $T_{k,n,r}$ is a block diagonal matrix, whose only nonzero block is the i-th block, which is an identity matrix of size $n\times n$.

The group $\mathrm{GL}_k\times\mathrm{GL}_s\times\mathrm{GL}_s$ acts in a usual way on $U\otimes L\otimes L$. The minrank variety $\mathcal{M}_r$ can be defined using the orbit closure of $T_{k,n,r}$:

Theorem 6.10. *Let V be an n-dimensional subspace of L. Then*

$$\mathcal{M}_{U\otimes V\otimes L,r}=\overline{(\mathrm{GL}_k\times\mathrm{GL}_s\times\mathrm{GL}_s)T_{k,n,r}}\cap(U\otimes V\otimes L).$$

Proof. We have $T_{k,n,r}\in\mathcal{M}_{U\otimes L\otimes L,r}$. Since the minrank variety is invariant, the entire orbit $(\mathrm{GL}_k\times\mathrm{GL}_s\times\mathrm{GL}_s)T_{k,n,r}$ lies in it. Since the minrank variety is Zariski closed, it also contains the orbit closure. By Lemma 6.1 we have $\overline{(\mathrm{GL}_k\times\mathrm{GL}_s\times\mathrm{GL}_s)T_{k,n,r}}\cap(U\otimes V\otimes L)\subset\mathcal{M}_{U\otimes V\otimes L,r}$.

Conversely, let $T\in\mathcal{M}_{U\otimes V\otimes L,r}$. We can write T as $\sum_{i=1}^{k}u_i\otimes A_i$ where $\{u_i\}$ is some basis of U and A_1 is a slice with $\mathrm{rk}(A_1)\le r$.

Since $\mathrm{rk}(A_1)\le r$, it can be presented as $(P_1\otimes Q_1)(\sum_{j=1}^{r}e_{1j}\otimes e_{1j})$ where $P_1\colon L_1\to V$ and $Q_1\colon L_1\to L$ are some linear maps. Analogously, for $i>1$ we have $\mathrm{rk}(A_i)\le\dim V=n$ and $A_i=(P_i\otimes Q_i)(\sum_{j=1}^{n}e_{ij}\otimes e_{ij})$ for some $P_i\colon L_i\to V$ and $Q_i\colon L_i\to L$. Let $P\colon L\to V$ and $Q\colon L\to L$ be the linear maps which are equal to P_i and Q_i respectively when restricted to L_i. Let $R\colon U\to U$ be the map sending each e_i to the corresponding u_i. Then $T=(R\otimes P\otimes Q)T_{k,n,r}$. The closure of $\mathrm{GL}(L)$ consists of all linear endomorphisms of L and thus contains P and Q. Therefore, T lies in the closure $\overline{(\mathrm{GL}_k\times\mathrm{GL}_s\times\mathrm{GL}_s)T_{k,n,r}}$. □

Corollary 6.3. *Let $\dim U=k$ and $\dim V=n$. Suppose V and W are subspaces of a vector space L of dimension $s=(k-1)n+r$. Then*

$$\mathcal{M}_{U\otimes V\otimes W,r}=\overline{(\mathrm{GL}(U)\times\mathrm{GL}(L)\times\mathrm{GL}(L))T_{k,n,r}}\cap(U\otimes V\otimes W).$$

6.5 Complexity of the orbit containment problem

The orbit containment problem ("$w \in Gv$") can be phrased as a polynomial systems of polynomial size by simply writing out the equations of gv for some generic g, and therefore can reduced to the problem Hilbert's Nullstellensatz HN. To ensure that $\det(g) \neq 0$, we can use a poly-size circuit for det to encode $z\det(g) = 1$ as a poly-size system of equations, where z is a new variable. Thus, we have the following theorem.

Theorem 6.11. *Let $\mathbb{F}$ be a field and K be an effective subfield. Then the orbit containment problem over $\mathbb{F}$ (with coefficients from K) is polynomial-time reducible to Hilbert's Nullstellensatz* HN *over $\mathbb{F}$ (with coefficients from K).*

By the results of Koiran [122], the above theorem implies that the orbit containment problem over the complex numbers is in AM assuming the generalized Riemann hypothesis (GRH), since Koiran's result also assumes GRH.

CHAPTER 7

Membership in the slice rank variety

In this chapter, we consider the notion of slice rank of tensors, which was recently introduced by Tao [174], and has subsequently been used for breakthroughs in several combinatorial problems like capsets, sunflower free sets, tri-colored sum-free sets, and progression-free sets. We show that the corresponding algorithmic problem, which can also be phrased as a problem about union of orbit closures, is also NP-hard, hence answering an open question by Bürgisser, Garg, Oliveira, Walter, and Wigderson [49]. We show this by using a connection between the slice rank and the size of a minimum vertex cover of a hypergraph revealed by Tao and Sawin.

7.1 Set-up and results

The notion of slice rank was first used implicitly by Croot, Lev, and Pach in their application of the so-called polynomial method in their breakthrough work on progression-free sets, also known as capsets [60]. Later Tao [174] gave a symmetrized formulation of this method and used slice rank explicitly. The term "slice rank", however, was first used by Blasiak et al. [35], who used the term for the notion that Tao introduced. They used this notion to extend the results on capsets and obtained some barrier results on the group-theoretic approach to the matrix multiplication, hence making slice rank quite important from the perspective of algorithm design. Tao and Sawin [175] explored the slice rank of tensors systematically. The methods based on slice rank have been very useful in advancement of several combinatorial problems like the sunflowers free sets, the tri-colored and multi-colored sum-free sets, the capsets and the progression-free problem, and multiplicative matching in nonabelian groups (see, for instance, [67, 146, 134, 157]). Finally, upper bounds on slice rank can be used to lower bound the matrix-multiplication exponent achievable by the so called universal method (which generalizes many known methods), and thus the computation of slice rank is interesting for analyzing the scope of the methods for finding fast matrix multiplication algorithms. See, for example, [11, Section 5].

We now describe the notion of slice rank and then the corresponding computational problem. For this, we consider the space $V_1 \otimes V_2 \otimes V_3$. It can also be written as $\bigotimes_{i=1}^{3} V_i$, and is generated by the decomposable (also called rank-one) tensors $v_1 \otimes v_2 \otimes v_3$, where $v_i \in V_i$. The usual tensor rank is the minimum number of decomposable tensors that is needed to write a given tensor as a sum of decomposable tensors. The slice rank is defined in a similar manner, however, the basic building blocks are not decomposable tensors but tensors that can be decomposed into a matrix and a single vector. More formally, consider the smaller tensor products $\bigotimes_{1 \le i \le 3 : i \ne j} V_i$ and the j-th tensor products

$\otimes_j : V_j \times \bigotimes_{1\le i\le 3: i\neq j} V_i \to \bigotimes_{i=1}^{3} V_i$ with its natural definition. Now the rank one functions are the elements of the form $v_j \otimes_j v_{\hat{j}}$ for some $v_j \in V_j$ and $v_{\hat{j}} \in \bigotimes_{1\le i\le 3: i\neq j} V_i$. The slice rank (or srk for short) of a tensor $T \in \bigotimes_{i=1}^{3} V_i$ is the smallest nonnegative integer r such that T can be expressed as a linear combination of r rank one functions. For its comparison with other notions of rank of tensors, like subrank and multi-slice rank, see [57, Section 5]. For its relation to the analytic rank and the partition rank, see [135].

The algorithmic problem corresponding to slice rank problem is the following.

Problem 7.1. Slice Rank of Tensors: *Given* $T \in \mathbb{F}^n \otimes \mathbb{F}^n \otimes \mathbb{F}^n$ *and a number* r*, decide if* $\mathrm{srk}(T) \le r$.

The notion of slice rank is closely related to the orbit closure containment problem. In particular, [35] established some interesting connections between the slice rank and the null cone problem. Bürgisser et al. [49] showed an equivalence between the *asymptotic* fullness of slice-rank and the null cone problem, and make some algorithmic progress towards it. In this work, we make the connection between slice rank and the orbit closure containment even more apparent, thanks to a formulation of slice rank by Tao and Sawin [175]. Bürgisser et al. [49, page 27] report that Sawin has an unpublished proof that *computing* the slice-rank of tensors of order three is NP-hard. However, they state that the decision version, that is, the above Problem 7.1 remains open, while expressing that it is plausible that this should be NP-hard as well.

7.1.1 Results and methods

Our main result result is the progress towards understanding Problem 7.1, the algorithm corresponding to slice rank problem. We rule out an efficient algorithm under the assumption that P ≠ NP by showing that the problem is NP-hard under polynomial time many-one reductions. (see Section 7.3).

Theorem 7.1. *Given a 3-tensor* T *and a positive integer* r*, determining if the slice rank of* T *is at most is* r*, is* NP*-hard.*

For this, we use a connection of the slice rank to the size of a minimum vertex cover of a hypergraph by Tao and Sawin [175]. They showed that for every 3-uniform, 3-partite hypergraph H, one can associate a tensor T_H, and if the edge set of the hypergraph forms an antichain, then the slice rank of the associated tensor T_H equals the size of the minimum vertex cover of the hypergraph H. To our best knowledge, the complexity of the decision version of the slice rank problem for order-three tensors has been open so far. Prahladh Harsha, Aditya Potukuchi, and Srikanth Srinivasan kindly sent us an unpublished manuscript, in which they prove that the order-four case is NP-hard. However, this one more tensor leg gives an additional degree of freedom, which easily allows to establish the antichain condition. Bürgisser et al. [49, page 27] report that Sawin has an unpublished proof that *computing* the slice-rank of tensors of order three is NP-hard. However, they also state that the decision version is open.

We show the NP-hardness of the slice rank problem for order-three tensors by showing that the 3-uniform, 3-partite hypergraph minimum vertex cover problem where the edge set forms an antichain is NP-hard. The corresponding hypergraph minimum vertex cover problem without the antichain restriction is known to be NP-hard [88] by reduction

from the usual 3-SAT problem. However, their reduction does not work if one wants to adapt it to the antichain restriction. We use a reduction from a restricted SAT-variant, the bounded-occurrence mixed SAT (bom-SAT) problem, in which there are 3-clauses and 2-clauses, and every variable occurs exactly thrice, once in a 3-clause and twice in 2-clauses. Because of the antichain restriction, our labelling of the gadget becomes very delicate and needs to be handled very carefully in the reduction (see Lemma 7.7).

Next, we phrase the slice rank problem in terms of orbit closures. More specifically, we show that testing whether a tensor $T \in \mathbb{F}^{n\times n\times n}$ has $\mathrm{srk}(T) \leq r$ is equivalent to testing if the tensor T is contained in a polynomially large union of orbit closures. Let (r_1, r_2, r_3) be such that $r_1+r_2+r_3 = r$. We first embed T in a larger subspace $U'\otimes V'\otimes W' \cong \mathbb{F}^{s_1}\otimes\mathbb{F}^{s_2}\otimes\mathbb{F}^{s_3}$ (this is called padding), where $s_1 = r_1+nr_2+nr_3$, $s_2 = nr_1+r_2+nr_3$ and $s_3 = nr_1+nr_2+r_3$, and define

$$S_{n,r_1,r_2,r_3} = \sum_{i=1}^{r_1}\sum_{j=1}^{n} e_i^1 \otimes e_{ij}^1 \otimes e_{ij}^1 + \sum_{i=1}^{r_2}\sum_{j=1}^{n} e_{ij}^2 \otimes e_i^2 \otimes e_{ij}^2 + \sum_{i=1}^{r_3}\sum_{j=1}^{n} e_{ij}^3 \otimes e_{ij}^3 \otimes e_i^3.$$

Intuitively, in the sum above, we have r_1 rank-one elements of the form $v_1 \otimes_1 v_{\hat{1}}$ with $v_1 \in V_1$ and $v_{\hat{1}} \in \bigotimes_{1\leq i\leq 3: i\neq 1} V_i$, r_2 elements of the form $v_2 \otimes_2 v_{\hat{2}}$, and r_3 elements of the form $v_3 \otimes_3 v_{\hat{3}}$. Now $\mathrm{srk}(T) \leq r$ becomes equivalent to testing whether T is in the orbit closure of S_{n,r_1,r_2,r_3} for some (r_1, r_2, r_3) with $r_1 + r_2 + r_3 = r$. Thus we show that the slice rank variety $\mathcal{SV}_{\mathbb{F}^n\otimes\mathbb{F}^n\otimes\mathbb{F}^n,r}$ is the union of orbit closure of S_{n,r_1,r_2,r_3} over all (r_1, r_2, r_3) with $r_1 + r_2 + r_3 = r$, intersected with the ambient space $\mathbb{F}^n \otimes \mathbb{F}^n \otimes \mathbb{F}^n$. It is worth noting that S_{n,r_1,r_2,r_3} is very similar to $T_{k,n,r}$ defined for the minrank (see Section 6.4 and Section 7.2 for details). Note that Tao showed that the set of all T with $\mathrm{srk}(T) \leq r$ is closed, so, similar to minrank, there is no need to define a notion of border slice rank either (see [175, Corollary 2]).

Next we go on to determine the stabilizer of S_{n,r_1,r_2,r_3}, i.e., the subgroup of $\mathrm{GL}(U') \times \mathrm{GL}(V') \times \mathrm{GL}(W')$ which fixes S_{n,r_1,r_2,r_3} (Theorem 7.3). We can also show that each S_{n,r_1,r_2,r_3} is almost characterized by its stabilizer, i.e., it is a direct sum of three tensors that are each characterized by their respective stabilizers (Theorem 7.4). This is an important property in the context of geometric complexity theory. Both the permanent and the determinant are characterized by their respective stabilizers as well.

7.2 Slice rank problem as a variety membership problem

For the necessary mathematical background, the reader is referred [165, 128, 130, 30].

The tensor used to show that the slice rank problem can be phrased as a problem about the union of orbit closures (see Definition 7.2) turns out to be very similar to the tensor $T_{k,n,r}$, which we used to show that the minrank problem is an instance of the orbit closure containment problem in Section 6.4. Thus, the exposition and the proofs in this section are very similar to that in Section 6.4.

Let us say we are given a 3-tensor $T \in U \otimes V \otimes W$, and we are interested in finding out if it has slice rank at most r, i.e., if $\mathrm{srk}(T) \leq r$.

In what follows, we phrase this problem geometrically and formulate it as membership

testing of T in a union of orbit closures of certain tensors.

Lemma 7.1. *([175, Corollary 2]) Let U, V, W be vector spaces over an algebraically closed field $\mathbb{F}$. The set of all tensors $T \in U \otimes V \otimes W$ with slice rank at most r is a Zariski closed set.*

In fact, they even showed that the set of all tensors $T \in U \otimes V \otimes W$ with slice rank at most r decomposed as (r_1, r_2, r_3) for a fixed tuple (r_1, r_2, r_3) with $r_1 + r_2 + r_3 = r$ is also Zariski closed.

Definition 7.1. *We call the the affine variety*

$$\mathcal{SV}_{U\otimes V\otimes W,r} = \{T \in U \otimes V \otimes W \mid \operatorname{srk}(T) \leq r\}$$

the affine slice rank variety *or simply the* slice rank variety*.*

When clear from the context, we drop the index $U \otimes V \otimes W$.

Lemma 7.2. *Let U, V, and W be subspaces of vector spaces U', V', and W', respectively. Then*

$$\mathcal{SV}_{U\otimes V\otimes W,r} = \mathcal{SV}_{U'\otimes V'\otimes W',r} \cap (U \otimes V \otimes W).$$

Proof. A tensor lies in $\mathcal{SV}_{U\otimes V\otimes W,r}$ iff it is an element of the space $U \otimes V \otimes W$ and has slice rank at most r, i.e., lies in $\mathcal{SV}_{U'\otimes V'\otimes W',r}$. □

Lemma 7.3. *The slice rank variety $\mathcal{SV}_{U\otimes V\otimes W,r}$ is invariant under the standard action of $GL(U) \times GL(V) \times GL(W)$ on $U \otimes V \otimes W$.*

Proof. If $\operatorname{srk}(T) \leq r$, we have $T = \sum_{i=1}^{r_1} u_{i,1} \otimes_1 T_{i,1} + \sum_{i=1}^{r_2} u_{i,2} \otimes_2 T_{i,2} + \sum_{i=1}^{r_3} u_{i,3} \otimes_3 T_{i,3}$ for some (r_1, r_2, r_3) such that $r_1 + r_2 + r_3 = r$, where $u_{i,1} \in U$, $u_{i,2} \in V$, $w_{i,3} \in W$, and $T_{i,1} \in V \otimes W$, $T_{i,2} \in U \otimes W$, $T_{i,3} \in U \otimes V$. Clearly when $A \otimes B \otimes C \in \mathrm{GL}(U) \times \mathrm{GL}(V) \times \mathrm{GL}(W)$ acts on T, the slice rank remains at most r. □

7.2.1 Slice rank varieties and orbit closures

For every tuple (r_1, r_2, r_3) of non-negative integers such that $r_1 + r_2 + r_3 = r$, we consider the vector spaces $U'_{(r_1,r_2,r_3)} = \mathbb{F}^{r_1} \oplus (\mathbb{F}^n)^{\oplus(r_2)} \oplus (\mathbb{F}^n)^{\oplus(r_3)}$, $V'_{(r_1,r_2,r_3)} = (\mathbb{F}^n)^{\oplus(r_1)} \oplus \mathbb{F}^{r_2} \oplus (\mathbb{F}^n)^{\oplus(r_3)}$, and $W'_{(r_1,r_2,r_3)} = (\mathbb{F}^n)^{\oplus(r_1)} \oplus (\mathbb{F}^n)^{\oplus(r_2)} \oplus \mathbb{F}^{r_3}$. We will drop the index (r_1, r_2, r_3) in the following.

U' has dimension $s_1(r_1, r_2, r_3) = r_1 + nr_2 + nr_3$, and is decomposed into $1 + r_2 + r_3$ summands, where one summand is of dimension r_1, while the other summands are of dimensions n each. Similarly, V' and W' have dimensions $s_2(r_1, r_2, r_3) = nr_1 + r_2 + nr_3$ and $s_3(r_1, r_2, r_3) = nr_1 + nr_2 + r_3$, respectively, and are decomposed analogously as U', into $r_1 + 1 + r_3$ summands and $r_1 + r_2 + 1$ summands respectively. We will denote $s_1(r_1, r_2, r_3)$, $s_2(r_1, r_2, r_3)$ and $s_3(r_1, r_2, r_3)$ simply by s_1, s_2, and s_3, respectively. Thus $U' \otimes V' \otimes W' \cong \mathbb{F}^{s_1} \otimes \mathbb{F}^{s_2} \otimes \mathbb{F}^{s_3}$.

Let us give names to the components: Let L^1 be $(\mathbb{F}^n)^{\oplus(r_1)}$ of dimension nr_1, L^2 be $(\mathbb{F}^n)^{\oplus(r_2)}$, and L^3 be $(\mathbb{F}^n)^{\oplus(r_3)}$, respectively, and we have vector spaces $\tilde{U} = \mathbb{F}^{r_1}$, $\tilde{V} = \mathbb{F}^{r_2}$

and $\tilde{W} = \mathbb{F}^{r_3}$ respectively. Let L_i^k be the i-th summand of $L^k, k \in \{1,2,3\}$ with standard basis e_{ij}^k, $j \in [n]$, and let e_i^1, e_i^2 and e_i^3 be the standard basis of $\tilde{U}$, $\tilde{V}$ and $\tilde{W}$. We have $U' = \tilde{U} \oplus L_1^2 \oplus \cdots \oplus L_{r_2}^2 \oplus L_1^3 \oplus \cdots \oplus L_{r_3}^3$ and similar decomposition for V' and W'.

Definition 7.2. *For (r_1, r_2, r_3), we define the* unit slice rank tensor $S_{n,r_1,r_2,r_3} \in (\tilde{U} \otimes L^1 \otimes L^1) \oplus (L^2 \otimes \tilde{V} \otimes L^2) \oplus (L^3 \otimes L^3 \otimes \tilde{W}) \subseteq U' \otimes V' \otimes W'$ *as*

$$S_{n,r_1,r_2,r_3} = \sum_{i=1}^{r_1} \sum_{j=1}^{n} e_i^1 \otimes e_{ij}^1 \otimes e_{ij}^1 + \sum_{i=1}^{r_2} \sum_{j=1}^{n} e_{ij}^2 \otimes e_i^2 \otimes e_{ij}^2 + \sum_{i=1}^{r_3} \sum_{j=1}^{n} e_{ij}^3 \otimes e_{ij}^3 \otimes e_i^3.$$

Along $\tilde{U}$ we have r_1 slices where each slice contains an $n \times n$ identity matrix each in disjoint blocks. Then along $\tilde{V}$, we have r_2 slices with $n \times n$ identity matrices in disjoint blocks. Finally, we have r_3 slices with $n \times n$ identity matrices in disjoint blocks along $\tilde{W}$. Thus S_{n,r_1,r_2,r_3} can be decomposed into three summands $S_{n.r_1} \in \tilde{U} \otimes L^1 \otimes L^1, S_{n,r_2} \in L^2 \otimes \tilde{V} \otimes L^2$ and $S_{n,r_3} \in L^3 \otimes L^3 \otimes \tilde{W}$ such that $S_{n,r_1,r_2,r_3} = S_{n,r_1} \oplus S_{n,r_2} \oplus S_{n,r_3}$. Notice the similarity with the minrank case. In particular, $T_{k,n,r}$ is almost like S_{n,r_1} – the only difference is that the first slice in $T_{k,n,r}$r is of different rank than the rest of its slices. As a consequence, in the spirit, the proof of the following Theorem 7.2 is very similar to the proof of Theorem 6.10.

The group $\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}$ acts on $U' \otimes V' \otimes W'$ in a natural way. The slice rank variety can be defined as the union of orbit closures of S_{n,r_1,r_2,r_3} under the action of $\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}$, where the union is taken over (r_1, r_2, r_3) such that $r_1 + r_2 + r_3 = r$.

Theorem 7.2. *Let U, V, and W be n-dim. subspaces of U', V', and W', respectively. Then*

$$\mathcal{SV}_{U \otimes V \otimes W, r} = \bigcup_{\substack{r_1,r_2,r_3 \\ r_1+r_2+r_3=r}} \overline{(\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}) S_{n,r_1,r_2,r_3}} \cap (U \otimes V \otimes W).$$

Note that each of the orbit closures is taken in a different ambient space, since each S_{n,r_1,r_2,r_3} lives in a different ambient space. But since we intersect each closure with $U \otimes V \otimes W$, this is fine.

Proof. First of all note that for every such (r_1, r_2, r_3), we have that $S_{n,r_1,r_2,r_3} \in \mathcal{SV}_{U' \otimes V' \otimes W', r}$, simply by the construction of S_{n,r_1,r_2,r_3}, where $U' \cong \mathbb{F}^{s_1}, V \cong \mathbb{F}^{s_2}, W' \cong \mathbb{F}^{s_3}$. Now since by Lemma 7.3, $\mathcal{SV}_{U' \otimes V' \otimes W', r}$ is invariant under the action of $\mathrm{GL}(U') \times \mathrm{GL}(V') \times \mathrm{GL}(W')$, we have that the entire orbit $(\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}) S_{n,r_1,r_2,r_3}$ lies in it. Also, from Lemma 7.1 (see [175, Corollary 2]), it follows that $(\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}) S_{n,r_1,r_2,r_3}$ is contained in a Zariski closed subset of $\mathcal{SV}_{U' \otimes V' \otimes W'}$ and hence the orbit closure $\overline{(\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}) S_{n,r_1,r_2,r_3}}$ also lies in $\mathcal{SV}_{U' \otimes V' \otimes W'}$. Now we apply Lemma 7.2 to get the desired inclusion.

For the other direction, let us assume $T \in \mathcal{SV}_{U \otimes V \otimes W, r}$. Since $\mathrm{srk}(T) \leq r$, we have that we have $T = \sum_{i=1}^{r_1} u_{i,1} \otimes_1 T_{i,1} + \sum_{i=1}^{r_2} u_{i,2} \otimes_2 T_{i,2} + \sum_{i=1}^{r_3} u_{i,3} \otimes_3 T_{i,3}$, for some (r_1, r_2, r_3) such that $r_1 + r_2 + r_3 = r$, where $u_{i,1} \in U$, $u_{i,2} \in V$, and $w_{i,3} \in W$ and $T_{i,1} \in V \otimes W$, $T_{i,2} \in U \otimes W$, and $T_{i,3} \in U \otimes V$. Since $\forall i \in [r_1]$, $\mathrm{rk}(T_{i,1}) \leq n$, we can write $T_{i,1}$ as $(Q_{i,1} \otimes R_{i,1})(\sum_{j=1}^{n} e_{i,j}^1 \otimes e_{i,j}^1)$ for linear maps $Q_{i,1} : L_i^1 \to V$ and $R_{i,1} : L_i^1 \to W$. Analogously,

$T_{i,2} = (P_{i,2} \otimes R_{i,2})(\sum_{j=1}^{n} e_{i,j}^2 \otimes e_{i,j}^2)$ for linear maps $P_{i,2} : L_i^2 \to U$ and $R_{i,2} : L_i^2 \to W$, and $T_{i,3} = (P_{i,3} \otimes Q_{i,3})(\sum_{j=1}^{n} e_{i,j}^3 \otimes e_{i,j}^3)$ for linear maps $P_{i,3} : L_i^3 \to U$ and $Q_{i,3} : L_i^3 \to V$.

Let $Q_1 : L^1 \to V$ and $R_1 : L^1 \to W$ be linear maps which are equal to $Q_{i,1}$ and $R_{i,1}$, respectively, when restricted to the i-th slice L_i^1. Similarly we have maps $P_2 : L^2 \to U$ and $R_2 : L^2 \to W$ whose restrictions to i-th slices are $P_{i,2}$ and $R_{i,2}$, respectively, and $P_3 : L^3 \to U$ and $Q_3 : L^3 \to V$ have their restrictions as $P_{i,3}$ and $Q_{i,3}$.

Finally, we also have linear maps $P_1 : \tilde{U} \to U$ sending e_i^1 to $u_{i,1}$, $Q_2 : \tilde{V} \to V$ sending e_i^2 to $u_{i,2}$ and $R_3 : \tilde{W} \to W$ sending e_i^3 to $u_{i,3}$.

Thus $T = ((P_1 \otimes Q_1 \otimes R_1) \oplus (P_2 \otimes Q_2 \otimes R_2) \oplus (P_3 \otimes Q_3 \otimes R_3))S_{n,r_1,r_2,r_3}$ for some (r_2, r_2, r_3). The closure of $\mathrm{GL}_{s_1}, \mathrm{GL}_{s_2}$ and GL_{s_3} contains all linear endomorphisms of U', V' and W', respectively, and thus contains $(P_1 \otimes Q_1 \otimes R_1) \oplus (P_2 \otimes Q_2 \otimes R_2) \oplus (P_3 \otimes Q_3 \otimes R_3)$. Therefore, T lies in the closure $\overline{(\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3})S_{n,r_1,r_2,r_3}}$ for some (r_1, r_2, r_3) with $r_1 + r_2 + r_3 = r$. □

We now describe the stabilizers of the unit slice rank tensors.

Lemma 7.4. *The stabilizer of $\sum_{i=1}^{k} e_i \otimes e_i \in \mathbb{F}^k \otimes \mathbb{F}^k$ in $\mathrm{GL}_k \times \mathrm{GL}_k$ consists of elements of the form $(A, A^{-\mathsf{T}})$.*

Proof. For the left action of $\mathrm{GL}_k \times \mathrm{GL}_k$ on $\mathbb{F}^k \otimes \mathbb{F}^k$ consider the corresponding left-right action: $AXB := (A, B^{\mathsf{T}})X$ for $A, B \in \mathrm{GL}_k$ and $X \in \mathbb{F}^k \otimes \mathbb{F}^k$. If we interpret $F^k \otimes F^k$ as the space of $k \times k$ matrices, then $\sum_{i=1}^{k} e_i \otimes e_i$ is the identity matrix I and we observe that AXB is the usual product of matrices. Clearly $AIB = I$ iff $A = B^{-1}$. □

Theorem 7.3. *For $n \geq 2$, the stabilizer of S_{n,r_1,r_2,r_3} in $\mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}$ is isomorphic to $\bigoplus_{i=1}^{3}((\mathrm{GL}_n \times \mathrm{GL}_1)^{r_i} \rtimes \mathfrak{S}_{r_i})$. The element $(Z_{i1}, z_{i1}, \ldots, Z_{ir_i}, z_{ir_i}) \in (\mathrm{GL}_n \times \mathrm{GL}_1)^{r_i}$ for $i = 1, 2, 3$ is embedded into $\mathrm{GL}_{r_1} \times \mathrm{GL}_{nr_1} \times \mathrm{GL}_{nr_1}$, $\mathrm{GL}_{nr_2} \times \mathrm{GL}_{r_2} \times \mathrm{GL}_{nr_2}$ and $\mathrm{GL}_{nr_3} \times \mathrm{GL}_{nr_3} \times \mathrm{GL}_{r_3}$ respectively, via*

$$(\mathrm{diag}(z_{11}, \ldots, z_{1r_1}), \mathrm{diag}(Z_{11}, \ldots, Z_{1r_1}), \mathrm{diag}((z_{11}Z_{1r_1})^{-\mathsf{T}}, \ldots, (z_{1r_1}Z_{1r_1})^{-\mathsf{T}})),$$
$$(\mathrm{diag}(Z_{21}, \ldots, Z_{2r_2}), \mathrm{diag}(z_{21}, \ldots, z_{2r_2}), \mathrm{diag}((z_{21}Z_{2r_2})^{-\mathsf{T}}, \ldots, (z_{2r_2}Z_{2r_2})^{-\mathsf{T}})), \text{ and}$$
$$(\mathrm{diag}(Z_{31}, \ldots, Z_{3r_3}), \mathrm{diag}((z_{31}Z_{3r_3})^{-\mathsf{T}}, \ldots, (z_{3r_3}Z_{3r_3})^{-\mathsf{T}}), \mathrm{diag}(z_{31}, \ldots, z_{3r_3})),$$

respectively. The $\mathfrak{S}_{r_i}$ factor permutes the r_i coordinates of $\tilde{U}, \tilde{V}$ and $\tilde{W}$, and the r_i summands of $L^i \times L^i$ simultaneously.

Proof. Let $S := S_{n,r_1,r_2,r_3} = S_{n,r_1} \oplus S_{n,r_2} \oplus S_{n,r_3}$, and $(A, B, C) \in \mathrm{stab}\, S$, that is, $(A \otimes B \otimes C)S = S$. It will be useful to visualize A, B and C as

$$A = \left(\begin{array}{c|c|c} A_{11} & A_{12} & A_{13} \\ \hline A_{21} & A_{22} & A_{23} \\ \hline A_{31} & A_{32} & A_{33} \end{array}\right), \quad B = \left(\begin{array}{c|c|c} B_{11} & B_{12} & B_{13} \\ \hline B_{21} & B_{22} & B_{23} \\ \hline B_{31} & B_{32} & B_{33}, \end{array}\right), \quad C = \left(\begin{array}{c|c|c} C_{11} & C_{12} & C_{13} \\ \hline C_{21} & C_{22} & C_{23} \\ \hline C_{31} & C_{32} & C_{33} \end{array}\right).$$

Above, A_{11} is an $r_1 \times r_1$ matrix, A_{22} is an $nr_2 \times nr_2$ matrix, and A_{33} is an $nr_3 \times nr_3$ matrix, respectively. B_{11} is an $nr_1 \times nr_1$ matrix, B_{22} is an $r_2 \times r_2$ matrix, and B_{33} is

an $nr_3 \times nr_3$ matrix, respectively, and C_{11} is an $nr_1 \times nr_1$ matrix , C_{22} is an $nr_2 \times nr_2$ matrix, and C_{33} is an $r_3 \times r_3$ matrix, respectively.

Let $S = \sum_{i=1}^{s_1} e_i \otimes_1 S_i$, where S_i is the i-th slice of S. Then,

$$(A \otimes B \otimes C)S_{n,r_1,r_2,r_3} = \sum_{i=1}^{s_1}(Ae_i) \otimes (B \otimes C)S_i = \sum_{i=1}^{s_1} e_i \otimes (B \otimes C)(\sum_{j=1}^{s_1} a_{ij}S_j).$$

First of all we divide the set of slices into groups. These include:

- r_1 groups of size 1 each, $\{1\}, \ldots, \{n\}$,
- r_2 groups of size n each, $\{r_1+1, \ldots, r_1+n\}, \ldots, \{r_1+(r_2-1)n+1, \ldots, r_1+r_2n\}$,
- r_3 groups of size n each, $\{r_1+r_2n+1, \ldots, r_1+r_2n+n\}, \ldots, \{r_1+r_2n+(r_3-1)n+1, \ldots, r_1+r_2n+r_3n\}$.

We first consider the first r_1 groups of slices, i.e., slices S_i for $i \in \{1, \ldots, r_1\}$ to deduce about the first r_1 rows of A.

Recall that for $i \in \{1, \ldots, r_1\}$, $\mathrm{rk}(S_i) = n$ (by definition). Thus we have that $\mathrm{rk}(\sum_{j=1}^{s_1} a_{ij}S_j)$ and consequently the rank of the i-th slice of $(A \otimes B \otimes C)S$ will be at least qn, where q is the number of nonzero entries among $a_{i1}, \ldots, a_{ir_1}$. Therefore, there will be at most one $j \in \{1, \ldots, r_1\}$ such that a_{ij} is nonzero. Now consider the case when for some $i' \in \{1, \ldots, r_1\}$, $a_{i'j} \neq 0$ for some $j \in \{r_1+1, \ldots, r_1+nr_2, \ldots, r_1+nr_2+nr_3\}$. First of all, it implies that $a_{i'j} = 0$, for all $j \in \{1, \ldots, r_1\}$, otherwise $\mathrm{rk}(\sum_{j=1}^{s_1} a_{i'j}S_j) \geq n+1$. Now since A induces a bijection among the slices, every slice is involved in the linear combination of at least one of the slices. And since two slices of rank n cannot be involved in the linear combination of first r_1 slices, the above forces that at least one of the rank n slices is involved in the linear combination of a slice S_i for $i \in \{r_1+1, \ldots, r_1+nr_2, \ldots, r_1+nr_2+nr_3\}$, i.e., $a_{ij} \neq 0$ for some $i \in \{r_1+1, \ldots, r_1+nr_2, \ldots, r_1+nr_2+nr_3\}$, $j \in \{1, \ldots, r_1\}$. But this implies that $\mathrm{rk}(\sum_{j=1}^{s_1} a_{ij}S_j) \geq n$ and not 1, which cannot be the case if $A \otimes B \otimes C \in \mathrm{stab}\, S$. Thus $a_{ij} = 0$ for all $j \in \{r_1+1, \ldots, r_1+nr_2, \ldots, r_1+nr_2+nr_3\}$. Thus A_{11} is a product of a diagonal matrix and a permutation matrix, and $A_{12} = A_{13} = 0$. Symmetrical argument implies that $B_{21} = B_{23} = C_{31} = C_{32} = 0$, and both B_{22} and C_{33} are products of a diagonal matrix and a permutation matrix.

Now consider the first group from the second case i.e. $i \in \{r_1+1, \ldots, r_1+n\}$: Here, first of all recall that $\mathrm{rk}(S_i) = 1$ for all i. Thus $\mathrm{rk}(\sum_{j=1}^{s_1} a_{ij}S_j)$ has to be 1. This immediately implies that $a_{ij} = 0$ for all $j \in \{1, \ldots, r_1\}$, otherwise the resulting rank will be at least n. We further argue that $a_{ij} = 0$ for all $j \in \{r_1+r_2n+1, \ldots, r_2n+n, \ldots, r_1+r_2n+r_3n\}$. Assume the contrary. Then $\sum_{j=1}^{s_1} a_{ij}S_j$ will have something in the bottom right $r_3 \times nr_3$ block. In order for (A, B, C) to be in $\mathrm{stab}\, S$, $(B \otimes C)(\sum_{j=1}^{s_1} a_{ij}S_j)$ should bring it back to its original place, i.e., in the central $nr_2 \times r_2$ block. However C_{33} being a product of a diagonal matrix and a permutation matrix, C will only permute the the last r_3 rows of $\sum_{j=1}^{s_1} a_{ij}S_j$ within themselves and hence $B \otimes C$ will not bring $\sum_{j=1}^{s_1} a_{ij}S_j$ to the central block as needed. Thus from the above discussion, we have $A_{21} = A_{23} = A_{31} = A_{32} = B_{12} = B_{13} = B_{31} = B_{32} = C_{12} = C_{13} = C_{31} = C_{32} = 0$. Finally, A_{22} will be a product of a block diagonal matrix and a block permutation matrix. For this, notice that for a fixed $i \in \{r_1+1, \ldots, r_1+r_2n\}$, a_{ij} cannot be non-zero for j's belonging to more than one group,

otherwise the rank of the resulting slice exceeds 1.

Thus A will be a product of a diagonal matrix with a permutation matrix in the top left block. In the central block, it will be a product of a block diagonal matrix with a block permutation matrix. Similarly, for the bottom right block, too, it will be a product of a block diagonal matrix with a block permutation matrix. Thus the picture becomes

$$A = \left(\begin{array}{c|c|c} A_{11} & 0 & 0 \\ \hline 0 & A_{22} & 0 \\ \hline 0 & 0 & A_{33} \end{array}\right), \quad B = \left(\begin{array}{c|c|c} B_{11} & 0 & 0 \\ \hline 0 & B_{22} & 0 \\ \hline 0 & 0 & B_{33} \end{array}\right), \quad C = \left(\begin{array}{c|c|c} C_{11} & 0 & 0 \\ \hline 0 & C_{22} & 0 \\ \hline 0 & 0 & C_{33} \end{array}\right),$$

where A_{11}, B_{22} and C_{33} are products of a diagonal matrix and a permutation matrix, and $A_{22}, A_{33}, B_{11}, B_{33}, C_{11}$ and C_{22} are all products of a block diagonal matrix and a block permutation matrix. Thus we can decompose $(A, B, C) \in \operatorname{stab} S$ as $((A_{11}, B_{11}, C_{11}) \oplus (A_{22}, B_{22}, C_{22}) \oplus (A_{33}, B_{33}, C_{33}))$ where (A_{11}, B_{11}, C_{11}) acts on $\tilde{U} \otimes L^1 \otimes L^1$, (A_{22}, B_{22}, C_{22}) acts on $L^2 \otimes \tilde{V} \otimes L^2$ and (A_{33}, B_{33}, C_{33}) acts on $L^3 \otimes L^3 \otimes \tilde{W}$, respectively. Hence for (A, B, C) to be in $\operatorname{stab} S$, with $S = S_{n,r_1} \oplus S_{n,r_2} \oplus S_{n,r_3}$, (A_{11}, B_{11}, C_{11}) must preserve S_{n,r_1}, (A_{22}, B_{22}, C_{22}) must preserve S_{n,r_2}, and (A_{33}, B_{33}, C_{33}) must preserve S_{n,r_3}, i.e., $\operatorname{stab} S = \operatorname{stab} S_{n,r_1} \oplus \operatorname{stab} S_{n,r_2} \oplus \operatorname{stab} S_{n,r_3}$, where $\operatorname{stab} S_{n,r_1} \subseteq \mathrm{GL}_{r_1} \times \mathrm{GL}_{nr_1} \times \mathrm{GL}_{nr_1}$, $\operatorname{stab} S_{n,r_2} \subseteq \mathrm{GL}_{nr_2} \times \mathrm{GL}_{r_2} \times \mathrm{GL}_{nr_2}$ and $\operatorname{stab} S_{n,r_3} \subseteq \mathrm{GL}_{nr_3} \times \mathrm{GL}_{nr_3} \times \mathrm{GL}_{r_3}$.

We consider $\operatorname{stab} S_{n,r_1}$ now. Let P_{σ_1} be an element of $\mathrm{GL}_{r_1} \times \mathrm{GL}_{nr_1} \times \mathrm{GL}_{nr_1}$ which permutes the r_1 coordinates of $\tilde{U} \subseteq U'$ and the r_1 summands of $L^1 \times L^1 \subseteq V' \times W'$ according to the permutation σ_1. It is easy to see that $P_{\sigma_1} \in \operatorname{stab} S_{n,r_1}$. Hence, $(A_{11}, B_{11}, C_{11}) P_{\sigma_1}^{-1} = (\tilde{A}_{11}, \tilde{B}_{11}, \tilde{C}_{11}) \in \operatorname{stab} S_{n,r_1}$. Using this and the previous discussion, we have that $\tilde{A}_{11}$ is a diagonal matrix $\operatorname{diag}(\tilde{a}_{11}^1, \ldots, \tilde{a}_{11}^{r_1})$. Let $\tilde{A}'_{11}$ be the linear map which scales elements of L_i^1 by $\tilde{a}_{11}^i$ for each $i \in [r_1]$. Clearly $(\tilde{A}_{11}^{-1}, \mathrm{id}, \tilde{A}'_{11})$ also preserves S_{n,r_1}. Therefore, $(\tilde{A}_{11}, \tilde{B}_{11}, \tilde{C}_{11}) \cdot (\tilde{A}_{11}^{-1}, \mathrm{id}, \tilde{A}'_{11}) = (\mathrm{id}, \tilde{B}_{11}, \hat{C}_{11})$ is in $\operatorname{stab} S_{n,r_1}$

Now, since the first component of $(\mathrm{id}, \tilde{B}_{11}, \hat{C}_{11})$ is the identity, it preserves S_{n,r_1} if and only if $\tilde{B}_{11} \otimes \hat{C}_{11}$ preserves each slice of S_{n,r_1}. If it preserves each slice, it also preserves its sum $\sum_{i=1}^{r_1} \sum_{j=1}^{n} e_{ij}^1 \otimes e_{ij}^1$, the full rank diagonal matrix of size $nr_1 \times nr_1$. Therefore, by Lemma 7.4, $\hat{C}_{11} = \tilde{B}_{11}^{-\mathsf{T}}$. Thus $(\mathrm{id}, \tilde{B}_{11}, \tilde{B}_{11}^{-\mathsf{T}}) \in \operatorname{stab} S_{n,r_1}$.

Thus, we decomposed an element $A_{11} \otimes B_{11} \otimes C_{11} \in \operatorname{stab} S_1$ into a product of three special elements $(\mathrm{id}, \operatorname{diag}(B_{11}^1, \ldots, B_{11}^{r_1}), \operatorname{diag}(B_{11}^1, \ldots, B_{11}^{r_1})^{-\mathsf{T}})$, $(\operatorname{diag}(\tilde{a}_{11}^1, \ldots, \tilde{a}_{11}^{r_1}), \mathrm{id}, \operatorname{diag}(\tilde{a}_{11}^1 \,\mathrm{id}, \ldots, \tilde{a}_{11}^{r_1} \,\mathrm{id})^{-1})$, and P_{σ_1} for some permutation $\sigma_1 \in \mathfrak{S}_{r_1}$. These three types of elements correspond to three subgroups of $\operatorname{stab} S_{n,r_1}$. The subgroups intersect only in the identity; elements of the first two types commute, and the conjugation with P_{σ_1} permutes $\tilde{a}_{11}^i$ and B_{11}^i according to σ_1, so the product of the first subgroups is direct, and the last product is semidirect. Symmetrical arguments for $\operatorname{stab} S_{n,r_2}$ and $\operatorname{stab} S_{n,r_3}$ finishes the proof. □

Now we show that the unit slice rank tensor S_{n,r_1,r_2,r_3} is almost characterized by its stabilizer. More precisely, it is a direct sum of three tensors S_{n,r_1}, S_{n,r_2} and S_{n,r_3} that are each characterized by their respective stabilizers.

Theorem 7.4. *Suppose T is a tensor in $U' \otimes V' \otimes W' = (\tilde{U} \oplus L^2 \oplus L^3) \otimes (L^1 \oplus \tilde{V} \oplus L^3) \otimes$*

$(L^1 \oplus L^2 \oplus \tilde{W})$. *If* $\operatorname{stab} T = \operatorname{stab} S_{n,r_1,r_2,r_3}$, *then*

$$T = ((\operatorname{diag}(\alpha,\dots,\alpha),\mathrm{id},\mathrm{id}),(\mathrm{id},\operatorname{diag}(\beta,\dots,\beta),\mathrm{id}),(\mathrm{id},\mathrm{id},\operatorname{diag}(\gamma,\dots,\gamma))S_{n,r_1,r_2,r_3},$$

for some $\alpha,\beta,\gamma \neq 0$, *i.e.,* $T = \alpha \cdot S_{n,r_1} \oplus \beta \cdot S_{n,r_2} \oplus \gamma \cdot S_{n,r_3}$.

Proof. Suppose T is stabilised by $\operatorname{stab} S_{n,r_1,r_2,r_3}$. We first establish that T also has the block structure like S_{n,r_1,r_2,r_3}, i.e., even though the ambient space of T is $(\tilde{U} \oplus L^2 \oplus L^3) \otimes (L^1 \oplus \tilde{V} \oplus L^3) \otimes (L^1 \oplus L^2 \oplus \tilde{W})$, it sits completely inside one of the smaller subspaces $(\tilde{U} \otimes L^1 \otimes L^1) \oplus (L^2 \otimes \tilde{V} \otimes L^2) \oplus (L^3 \otimes L^3 \otimes \tilde{W})$ which also contains S_{n,r_1,r_2,r_3}.

For this, we take the element $(A,B,C) \in \operatorname{stab} S_{n,r_1,r_2,r_3} \subseteq \mathrm{GL}_{s_1} \times \mathrm{GL}_{s_2} \times \mathrm{GL}_{s_3}$ where we have $A = (\operatorname{diag}(\alpha_1,\dots,\alpha_1),\operatorname{diag}(\alpha_2 \cdot \mathrm{id},\dots,\alpha_2 \cdot \mathrm{id}),\operatorname{diag}(\alpha_3 \cdot \mathrm{id},\dots,\alpha_3 \cdot \mathrm{id}))$, whereas $B = (\operatorname{diag}(\beta_1 \cdot \mathrm{id},\dots,\beta_1 \cdot \mathrm{id}),\operatorname{diag}(\beta_2,\dots,\beta_2),\operatorname{diag}(\beta_3 \cdot \mathrm{id},\dots,\beta_3 \cdot \mathrm{id}))$ and $C = (\operatorname{diag}(\gamma_1 \cdot \mathrm{id},\dots,\gamma_1 \cdot \mathrm{id}),\operatorname{diag}(\gamma_2 \cdot \mathrm{id},\dots,\gamma_2 \cdot \mathrm{id}),\operatorname{diag}(\gamma_3,\dots,\gamma_3))$ such that $\alpha_1\beta_1\gamma_1 = \alpha_2\beta_2\gamma_2 = \alpha_3\beta_3\gamma_3 = 1$. Now, $U' \otimes V' \otimes W'$ is a direct sum of 27 subspaces, and T can be decomposed into corresponding 27 blocks. On the action of (A,B,C) on T, only three of the blocks remain fixed, i.e., the ones corresponding to the subspaces $(\tilde{U} \otimes L^1 \otimes L^1)$, $(L^2 \otimes \tilde{V} \otimes L^2)$ and $(L^3 \otimes L^3 \otimes \tilde{W})$ because the entries in these blocks get scaled by $\alpha_1\beta_1\gamma_1, \alpha_2\beta_2\gamma_2$ and $\alpha_3\beta_3\gamma_3$ respectively, which are all equal to unity. The blocks corresponding to the other subspaces will be scaled by non-unity and hence will not remain fixed. Hence for (A,B,C) to be in the stabilizer of T, only the blocks corresponding to these three subspaces will be non-zero, which is also the case for S_{n,r_1,r_2,r_3}.

Recall from Definition 7.2 that S_{n,r_1,r_2,r_3} can be decomposed as $S_{n,r_1} \oplus S_{n,r_2} \oplus S_{n,r_3}$. Thus, we decompose T into blocks as $T = T_{n,r_1} \oplus T_{n,r_2} \oplus T_{n,r_3}$. We focus on $T_{n,r_1} =: T'$, where $T' \in (\tilde{U} \otimes L^1 \otimes L^1)$.

Let $T'_1,\dots,T'_{r_1}$ be the slices of T'. Decompose them into blocks $T'_i = (T'_{ijk})$ according to the decomposition of L^1 into L^1_i.

Let $A_i(\lambda) : \tilde{U} \to \tilde{U}$ be the map which scales the i-th coordinate by λ, leaving other in place, and $B_i(\lambda) : L^1 \to L^1$ be the map which scales L^1_i by λ and acts like identity on the other summands. Applying to T' the transformation $(A_i(\lambda^{-2}), B_i(\lambda), B_i(\lambda)) \in \operatorname{stab} S_{n,r_1}$, we see that all blocks of T'_i except T'_{iii} are zero, as they are multiplied by a coefficient λ^{-2} or λ^{-1} in this transformation.

Applying $(\mathrm{id},\operatorname{diag}(Z_{11},\dots,Z_{1r_1}),\operatorname{diag}(Z_{11},\dots,Z_{1r_1})^{-\mathsf{T}})$ with arbitrary Z_{1i} to T', we obtain that each T'_{iii} has the form $a_{1i}\sum_{j=1}^{\dim L^1_i} e^1_{ij} \otimes e^1_{ij}$. Applying permutations on the blocks of T'_{iii}, we see that $a_{11} = \cdots = a_{1r_1} =: \alpha$.

Therefore

$$T' = \alpha \sum_{i=1}^{r_1}\sum_{j=1}^{n} e_i \otimes_1 e^1_{ij} \otimes e^1_{ij} = \alpha \cdot S_{n,r_1}.$$

If $\alpha \neq 0$, then $T' = T_{n,r_1} = (\operatorname{diag}(\alpha,\dots,\alpha),\mathrm{id},\mathrm{id})S_{n,r_1}$. Applying the symmetrical arguments on T_{n,r_2} and T_{n,r_3}, we get that

$$T = (\operatorname{diag}(\alpha,\dots,\alpha),\mathrm{id},\mathrm{id})S_{n,r_1} \oplus (\mathrm{id},\operatorname{diag}(\beta,\dots,\beta),\mathrm{id})S_{n,r_2} \oplus (\mathrm{id},\mathrm{id},\operatorname{diag}(\gamma,\dots,\gamma))S_{n,r_3},$$

for some $\alpha,\beta,\gamma \neq 0$, or simply $T = \alpha \cdot S_{n,r_1} \oplus \beta \cdot S_{n,r_2} \oplus \gamma \cdot S_{n,r_3}$. □

7.3 Complexity of the slice rank problem

In this section, we show that the problem of testing if a given 3-tensor has slice rank at most r is NP-hard.

Theorem 7.5 (Theorem 7.1 restated)**.** *Given a 3-tensor T and a positive integer r, determining if the slice rank of T is at most is r, is* NP*-hard.*

We prove this by showing that a variant of hypergraph vertex cover testing is NP-hard. Tao and Sawin [175] showed the equivalence of the slice rank problem to this variant of hypergraph vertex cover testing. For stating this equivalence precisely, we now set up some notations.

We fix a field $\mathbb{F}$. Given a 3-uniform, 3-partite hypergraph H with 3 partitions U, V and W with $|U| = n_1$, $|V| = n_2$, and $|W| = n_3$, $n_i \in \mathbb{N}$, $i \in [3]$, and edge set $E \subseteq U \times V \times W$, we can define a 3-tensor $T_H(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3})$ corresponding to H in the following way, where $\mathbf{x_i}$ is a tuple of $[n_i]$ variables:

$$T_H(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}) = \sum_{(u_{i_1}, v_{i_2}, w_{i_3}) \in E} x_{1,i_1} \cdot x_{1,i_2} \cdot x_{3,i_3}$$

We label the nodes in U, V and W from the set of integers. For two hyperedges $e_1 := (u_{a_1}, v_{b_1}, w_{c_1})$ and $e_2 := (u_{a_2}, v_{b_2}, w_{c_2})$, we say that $e_1 \leq e_2$ iff $(a_1 \leq a_2) \wedge (b_1 \leq b_2) \wedge (c_1 \leq c_2)$. If neither $e_1 \leq e_2$ nor $e_2 \leq e_1$ holds, we say that e_1 and e_2 are incomparable. In E, if every pair of hyperedges is incomparable to each other, we say that E is an antichain.

Tao and Sawin (see [175, Proposition 4]) showed the following.

Lemma 7.5. *If the hyperedge set E is an antichain, then the slice rank of T_H is the same as the size of the minimum vertex cover of the hypergraph H.*

Thus, in order to show that computing the slice rank of 3-tensors is NP-hard, we show that the hypergraph minimum vertex cover problem for a 3-partite, 3-uniform graph, where the edge set is an antichain, is NP-hard.

Our reduction is inspired by [88] where they show the NP-hardness of the hypergraph vertex cover problem for 3-uniform 3-partite graphs. Their reduction involved reducing 3-SAT to this problem. Here we need to show the hardness under the extra condition that the hyperedge set of the graph is an antichain. This makes the reduction far more involved, and we also change the hard problem that we reduce to our problem.
The NP-hard problem that we use for our reduction is a bounded occurrence mixed SAT problem (bom-SAT), where we have 3-clauses and 2-clauses, such that every variable appears exactly thrice, once in a 3-clause, while the other two occurrences are in 2-clauses (note that the number of variables, $n = 3t$, for some t, where t is the number of 3-clauses).

Remark 7.1. *It is easy to see that the above mentioned bom-SAT is* NP*-hard. For this, start with any 3-SAT instance. Now assume that a variable Z appears m times. Introduce m copies $Z_1, ..., Z_m$ of X. Replace every occurrence of Z by one Z_i. We do this for all the variables. Now every variables appears only once. However, we have to ensure consistency, that is, $Z_1, ...Z_m$ should have the same value. So we add the 2-clauses: $(Z_1 \vee \neg Z_2) \wedge (Z_2 \vee \neg Z_3) \wedge \cdots \wedge (Z_m \vee \neg Z_1)$. These 2-clauses can only be satisfied if we*

set all the Z_i's to 0 *or all the Z_i's to* 1. *The resulting formula is a bom-SAT instance as described above.*

In the reduction, given a bom-SAT formula ϕ in n variables $X_1, \dots X_n$ with t 3-clauses and m 2-clauses, the construction of a 3-uniform 3-partite hypergraph G^ϕ with 3 vertex partitions U, V and W proceeds as follows. First of all we sort all the clauses such that all the 3-clauses precede all the 2-clauses. Next we rename all the variables such that the variables in the r-th 3-clause ($r \in t$) are $Y_{3(r-1)+1}, Y_{3(r-1)+2}$ and $Y_{3(r-1)+3}$ corresponding to the first, second and the third position of the clause respectively. We also say that $Y_{3(r-1)+1}, Y_{3(r-1)+2}$ and $Y_{3(r-1)+3}$ belong to the same triple of variables.
Now, we have a gadget G_k^ϕ corresponding to each variable Y_k, $k \in [n]$. G_k^ϕ consists of nodes $(i,j)^k$ and $\overline{(i,j)}^k, i, j \in \{1,2,3\}$. Here $(i,j)^k$ refers to the node corresponding to the i-th occurrence of the variable Y_k, and it occurs at the j-th position in the clause in which it appears. $\overline{(i,j)}^k$ refers to the negation of Y_k in its i-th occurrence at the j-th position in the clause. We will drop the superscript k, when it is clear from the context. Clearly, there are 18 such *literal nodes* in a gadget G_k^ϕ, which are ordered along a circle (see the outer circle in Figure 7.1). Since Y_k appears exactly thrice in ϕ, exactly 3 out of these 18 nodes will correspond to some occurrence of Y_k in ϕ. G_k^ϕ also consists of 18 other nodes, which we call *free nodes* (as they do not correspond to any literal), that are useful in the construction (see the inner circle in Figure 7.1). We have hyperedges connecting two literal nodes and a free node. There are total 18 hyperedges in G_k^ϕ each consisting of three vertices that form a triangle in Figure 7.1. Note that every literal node appears in exactly 2 hyperedges, while a free node appears in exactly one of them. We partition the set of nodes in 3 parts, as illustrated in the figure. Among the literal nodes, the nodes corresponding to the first-occurrences ($j = 1$) go to the set U, the ones corresponding to the second-occurrences ($j = 2$) go to the set V, while the ones corresponding to third occurrences ($j = 3$) go to the set W. We distribute the free nodes equally among the three sets, while maintaining the property of being 3-partite (see Figure 7.1).

Additionally, we have clause hyperedges, which for a 3-clause, connect the nodes corresponding to the three literals present in it. For every 2-clause, we first introduce another free node to the graph, added to set W (as there are no literals at the third position in a 2-clause). Now, there is an hyperedge for every 2-clause as well, connecting the two nodes corresponding to its literals and a free node. We refer to the hyperedges in a variable gadget either as *variable hyperedges* or *local hyperedges*. We refer to the hyperedges corresponding to the clauses as *clause hyperedges* or *global hyperedges*. We illustrate the set up with an example. See Figure 7.3.

The following two lemmas together finish the reduction and hence prove Theorem 7.5.

Lemma 7.6. *The size of the minimum vertex cover of the hypergraph G^ϕ is at most* $9n$ *if and only the* bom-SAT *instance ϕ is satisfiable.*

The proof of this lemma follows very closely the proof of hardness of hypergraph minimum vertex cover problem (see [88, Lemma 5.3]), which was itself inspired by the proof of NP-hardness of 3-dimensional matching given in Garey and Johnson [78]. We give a sketch here.

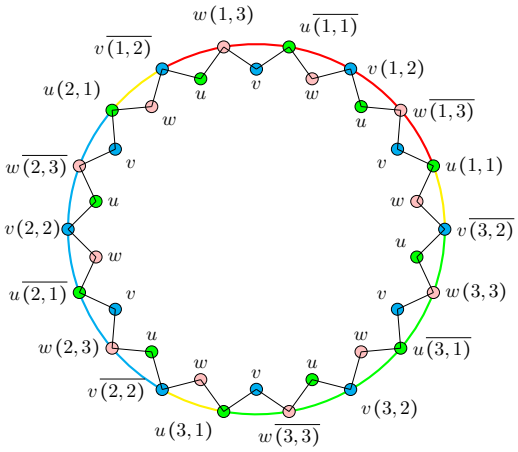

Figure 7.1: A variable gadget G_k^ϕ corresponding to the variable Y_k in ϕ. Nodes sharing the red, cyan and green arcs correspond to the first, second, and third occurrence of Y_k in a clause respectively. Exactly 3 out of 18 literal nodes are used in clause hyperedges. Nodes with an overline indicate that the negation of Y_k appeared in the corresponding clause. Nodes in the inner circle correspond to the free nodes.

Proof. Let ϕ be satisfiable with ν being a satisfying assignment on the variables $Y_1, \ldots, Y_n$. Now, we construct the vertex cover set S for G^ϕ of size $9n$ as follows. If $\nu(Y_k) = 0$, we add all the 9 overlined nodes from G_k^ϕ to S, otherwise we add the other 9 nodes to S. Note that S covers all the local hyperedges. Since ν is a satisfying assignment, all the clause hyperedges are also covered by S as well.

Conversely, assume there is a minimum vertex cover S of G^ϕ of size at most $9n$. Now, since all the free nodes appear in only one hyperedge each, we can assume that S does not contain any free node, since we can always replace them by a literal node of the same hyperedge. Now, for $i \in \{1, \ldots, n\}$ if S_i is the subset of S such that S_i only contains the vertices corresponding to the variable gadget G_i^ϕ, it can be easily seen that $|S_i| \geq 9$ for all the variable hyperedges to be covered. This implies that $|S_i| = 9$ since we assumed that $|S| = |\cup_{i=1}^n S_i| \leq 9n$. Thus S_i forms a vertex cover corresponding to the local gadget G_i^ϕ and hence covers the hyperedges in G_i^ϕ. However, there are only two vertex covers of G_i^ϕ of size 9, namely the one set containing all the overlined nodes, i.e., they correspond to $\neg Y_k$, and the other set where none of the nodes are overlined, i.e., they correspond to Y_k. In the first case, we assign the value 0 to Y_k, and we assign 1 in the second case. Thus we construct the assignment ν for $Y_1, \ldots, Y_n$. Now, since S is a vertex cover and hence span all the hyperedges including the clause hyperedges, ν satisfies all the clauses of ϕ. □

The following lemma ensures that the edge set E of the above constructed graph G^ϕ is indeed an antichain under some labelling.

Lemma 7.7. *For every formula ϕ, there exists a way of labelling of the nodes in hypergraph G^ϕ such that the hyperedge set of G^ϕ is an antichain.*

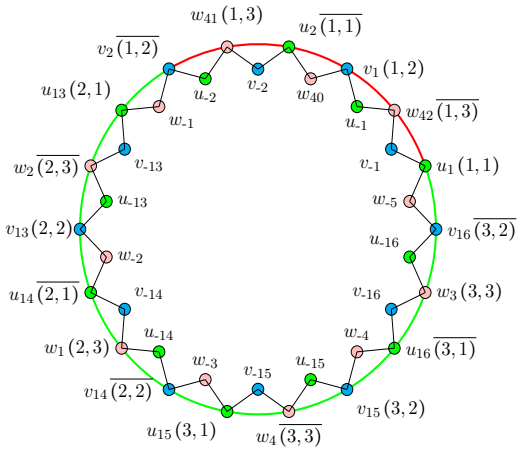

Figure 7.2: The labelling of variable gadgets G_1^ϕ for $n = 6$. The hyperedges with a red arc correspond to the first occurrence of variables. Notice the difference in labelling of W nodes. Literal nodes are all labelled positive. Free nodes are all labelled negative except the W node connecting the two first occurrence literal nodes.

Proof. We first give the labelling used. We have literal nodes and free nodes. The literal nodes either correspond to the first occurrence, the second occurrence or the third occurrence of a variable. In every gadget, we have 6 nodes corresponding to each occurrence, 2 from each partition U, V and W. The free nodes although do not correspond to any occurrences, we say that they correspond to first occurrence if the two literal nodes that they connect both correspond to the first occurrence. In every gadget, there are 5 such nodes, 2 each belonging to U and V, while one belonging to W. If a free node does not correspond to the first occurrence, we say that it corresponds to the second or third occurrence (we do not make distinction within them as it is not needed).

We first give the labelling corresponding to the nodes corresponding to the second and the third occurrences of variables:

- The position 1 literal nodes $(i,1)^k$ and $\overline{(i,1)}^k$ in G_k^ϕ are labelled $u_{2n+2(i-2)+4(k-1)+1}$ and $u_{2n+2(i-2)+4(k-1)+2}$, respectively, $\forall k$, for $i = 2, 3$.
- Similarly, the position 2 literal nodes $(i,2)^k$ and $\overline{(i,2)}^k$ are labelled $v_{2n+2(i-2)+4(k-1)+1}$ and $v_{2n+2(i-2)+2(k-1)+2}$, respectively, $\forall k$, for $i = 2, 3$.
- Likewise, the position 3 literal nodes $(i,3)^k$ and $\overline{(i,3)}^k$ are labelled $w_{2n+2(i-2)+4(k-1)+1}$ and $w_{2n+2(i-2)+4(k-1)+2}$ respectively, $\forall k$, for $i = 2, 3$.
- The 4 free U nodes in G_k^ϕ corresponding to the second or third occurrence are labelled $u_{-2n-4(k-1)-\ell}$, $\ell \in [4]$ (see Figure 7.2 to see which ones exactly).
- Similarly, the 4 such free V nodes in G_k^ϕ are labelled $v_{-2n-4(k-1)-\ell}$, $\ell \in [4]$.
- Finally, the 5 such free W nodes in G_k^ϕ are labelled $w_{-5(k-1)-\ell}$, $\ell \in [5]$.
- All the 2-clauses also correspond to the second and third occurrence of variables.

Each such 2-clause will have a corresponding hyperedge. Here we have a freedom to choose the position for the free node. We invariably choose it to be at the third position. Thus the first two nodes of the hyperedges will take the relevant literals as per the clause, while the W nodes will be free ones. For the $s-$th 2-clause (under an arbitrary order), $s \in [m]$ label the W nodes as w_{-5n-s}.

- We take all the hyperedges that include all the above labelled free W nodes. This will include all the 2-clause hyperedges along with 5 hyperedges per variable gadget. Now the tuple of U and V coordinates (u_a, v_b) of these hyperedges will have a partial order among themselves. We shuffle their W coordinates so that the order of the W coordinates becomes the reverse of the order of the tuple (u_a, v_b). We can do this without disturbing other hyperedges because these W nodes are all free and are used in only one hyperedge each.

Now it remains to label the literal nodes corresponding to the first occurrences and the free nodes pertaining to them. They are labelled differently so as to ensure that the antichain property indeed holds when the hyperedges connecting these would be compared with the 3-clause hyperedges. One key difference is that the labels of W nodes for G_k^ϕ in this case also depend on whether $k \equiv 1, 2$ or $0 \mod 3$.

- The position 1 literal nodes $(1,1)^k$ and $\overline{(1,1)}^k$ in G_k^ϕ are labelled $u_{2(k-1)+1}$ and $u_{2(k-1)+2}$, respectively, $\forall k$.

- The position 2 literal nodes $(1,2)^k$ and $\overline{(1,2)}^k$ are labelled $v_{2(k-1)+1}$ and $v_{2(k-1)+2}$, respectively, $\forall k$.

- The position 3 literal nodes $(1,3)^k$ and $\overline{(1,3)}^k$ get the labels $w_{7n-9(q-1)}$ and $w_{7n-9(q-1)-1}$, respectively, for $k = 3(q-1)+1$, whereas $w_{7n-9(q-1)-3}$ and $w_{7n-9(q-1)-4}$, respectively, for $k = 3(q-1)+2$, and $w_{7n-9(q-1)-5}$ and $w_{7n-9(q-1)-6}$, respectively, for $k = 3(q-1)+3$

- The 2 free U nodes corresponding to the first occurrence of the variable get the labels $u_{-2(k-1)-1}$ and $u_{-2(k-1)-2}$, respectively. Similarly such free V nodes get the labels $v_{-2(k-1)-1}$ and $v_{-2(k-1)-2}$ respectively, whereas the such free W nodes (1 per gadget) get the labels $w_{7n-9(q-1)-2}$ for $k = 3(q-1)+1$ and $w_{7n-9(q-1)-7}$ for $k = 3(q-1)+2$, and $w_{7n-9(q-1)-8}$ for $k = 3(q-1)+3$.

Figure 7.3 illustrates the labelling for $k = 1, 2, 3$ when $n = 6$.

We now show that with the above ordering, the set of hyperedges E of the hypergraph G^ϕ indeed is an antichain.

To simplify the argument, we divide the set of hyperedges in two parts $E = \mathcal{A} \cup \mathcal{B}$:

- Set $\mathcal{A}$: This set consists of local hyperedges in which both the literal nodes correspond to the first occurrence of variables. We also include the 3-clause hyperedges.

- Set $\mathcal{B}$: The set consisting of the remaining hyperedges, i.e., the ones in which at least one of the literal nodes correspond to the second or the third occurrences of variables. We also include the 2-clause hyperedges.

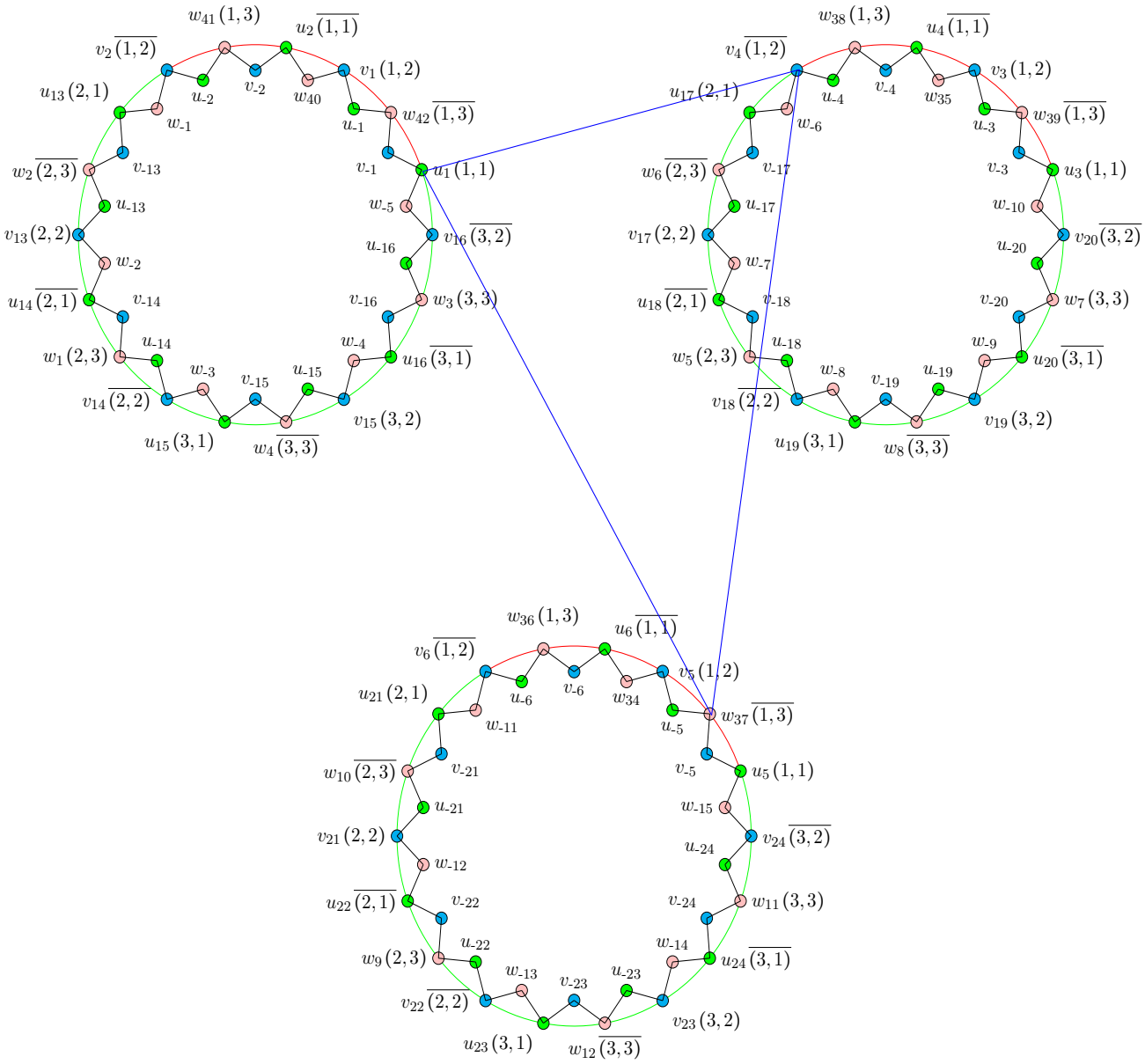

Figure 7.3: The variable gadgets $G_k^\phi, k = 1, 2, 3$ for $n = 6$. The hyperedges with a red arc correspond to the first occurrence of variables. Notice the difference in labelling of W nodes. The clause edge corresponds to the clause $Y_1 \vee \overline{Y_2} \vee \overline{Y_3}$.

We first argue that the subset $\mathcal{B}$ is an antichain.
We note that in $\mathcal{B}$, the literal nodes are all labelled positive $(2n + 2(i-2) + 4(k-1) + j)$, $i \in \{2,3\}$, $k \in [n]$, $j \in [4]$, while the free nodes are all labelled negative $(-2n - 4(k-1) - \ell)$, $k \in [n]$, $\ell \in [4]$, for U and V nodes, whereas $(-5(k-1) - \ell)$, $k \in [n]$, $\ell \in [5]$ for W nodes, and it is easy to verify that as the labels of the literal node increase, the labels along the free node decrease.
Now we take two arbitrary elements of the set $\mathcal{B}$. Recall that every hyperedge in $\mathcal{B}$ contains exactly one free node. Now the free node will either be in the same partition or in different ones.
If they are in different ones, we are done because we have a pair of coordinates such that, in one of them, one hyperedge is labelled positive while the other is labelled negative, while the opposite happens in the other coordinate. If the free nodes are in the same coordinate, we are done again because as the literal coordinate increases, the free coordinate decreases.
Note that, since we have already shuffled the nodes with free W nodes taking the 2-clause hyperedges into account, the 2-clause hyperedges are also taken care off.

Now, we argue that given an arbitrary hyperedge of the set $\mathcal{A}$, and an arbitrary hyperedge of the set $\mathcal{B}$, they are incomparable too.
For this, we notice that, the labels of the W nodes of all the hyperedges in $\mathcal{A}$ are higher than the labels of all the W nodes of the hyperedges in $\mathcal{B}$. For this, we simply note that range of the W labels of the second and the third occurrence (set $\mathcal{B}$) is $\{-5n, \ldots, 4n\} \setminus \{0\}$, whereas the W labels of the first occurrence ($\mathcal{A}$) has the range from $\{4n+1, \ldots, 7n\}$. Secondly, notice that the labels of the U and V literal nodes at the second and third occurrences, i.e., from the edges of set $\mathcal{B}$ (range $\{2n+1, \ldots, 6n\}$) are all higher than that of the first occurrence, i.e., from the edges of the set $\mathcal{A}$ (range $\{1, \ldots, 2n\}$).

We are done since for every pair of hyperedges (h_a, h_b), where $h_a \in \mathcal{A}$ and $h_b \in \mathcal{B}$, we have that the W coordinate of h_a will be higher than that of h_b, whereas the among the other two coordinates, whichever is positive (i.e., corresponds to a literal node) in h_b will be higher than the correpsonding coordinate in h_a.

Finally we are left to show that $\mathcal{A}$ is also an antichain.

We remind the reader that we have named the variables such that every 3-clause comprises of variables from only one triple of variables, i.e., every 3-clause involves $Y_{3(q-1)+1}, Y_{3(q-1)+2}, Y_{3(q-1)+3}$ at first, second and third position respectively, for some $q > 0$. Now first of all we notice that for a pair of hyperedges which come from a different triple of variables, we are done, because W coordinates of a higher triple are all lower than the W coordinates of a lower triple, since the labels are $(7n - 9(q-1) - \ell), \ell \in \{0, \ldots, 8\}$ for $q-$th triple of variables $Y_{3(q-1)+1}, Y_{3(q-1)+2}, Y_{3(q-1)+3}$, whereas the positive coordinate among U or V will be higher for the higher triple (lables are $4(k-1) + \ell, \ell \in [2]$). When they are in the same triple of variables, it helps to remark that there are three kinds of hyperedges in $\mathcal{A}$, i.e. $\mathcal{A} = \mathcal{A}_1 \uplus \mathcal{A}_2 \uplus \mathcal{A}_c$:

- $\mathcal{A}_1$: the ones where the free nodes belong to U or V. These hyperedges have exactly one negative coordinate, which will either be in the U coordinate or the V coordinate.

- $\mathcal{A}_2$: the ones where the free nodes belong to W. All the coordinates are positive.
- $\mathcal{A}_c$: the set of 3-clause hyperedges: All the coordinates are again positive, as all the nodes are literal nodes.

Now, we need to compare the hyperedges of $\mathcal{A}_1, \mathcal{A}_2$, and $\mathcal{A}_c$ with each other and within themselves when they all belong to the same triple of variables, say the q–th triple, $Y_{3(q-1)+1}, Y_{3(q-1)+2}, Y_{3(q-1)+3}$ for some $q \in [t]$. We remind the reader that the labelling of the W nodes that appear in $\mathcal{A}$ varies depending on whether the corresponding index $k = 3(q-1)+1$, $3(q-1)+1$, or $3(q-1)+3$.

There are six possible cases:

(1) $\mathcal{A}_1$: same proof that was given for the elements of $\mathcal{B}$, where also we had exactly one negative coordinate.

(2) $\mathcal{A}_2$: for the higher variable, the W coordinate is lower (labels are $7n-9(q-1)-2$ for $k = 3(q-1)+1$, $7n-9(q-1)-7$ for $k = 3(q-1)+2$ and $7n-9(q-1)-8$ for $k = 3(q-1)+3$), while the other two coordinates are higher, since both U and V labels are $2(k-1)+1, 2$.

(3) $\mathcal{A}_c$: two different clauses clearly belong to different triple of variables: already taken care of above.

(4) $\mathcal{A}_1 - \mathcal{A}_2$ ($h_{a_1} \in \mathcal{A}_1, h_{a_2} \in \mathcal{A}_2$): Here we have two cases: namely, either h_{a_1} belonging to a higher variable, or h_{a_1} belonging to the same or lower variable as compared to h_{a_2}. In the first case, one of the U or V coordinate of h_{a_1} (whichever is positive) will be higher, while the other coordinate being negative will be lower than that of h_{a_2} (whose all coordinates are positive). In the second case, we note that the W coordinate of h_{a_2} will be lower, since for the same variable, it has the lowest W coordinate (being $7n-9(q-1)-2$ versus $7n-9(q-1)$, $7n-9(q-1)-1$ for $k = 3(q-1)+1$, $7n-9(q-1)-7$ versus $7n-9(q-1)-3$, $7n-9(q-1)-4$ for $k = 3(q-1)+2$ and $7n-3(k-1)-8$ versus $7n-9(q-1)-5$, $7n-9(q-1)-6$ for $k = 3(q-1)+3$), and as we go up the variables, W coordinate decreases, while at least one of the other two coordinate will be higher, i.e., in the coordinate in which h_{a_1} is negative and h_{a_2} is positive.

(5) $\mathcal{A}_1 - \mathcal{A}_c$ ($h_{a_1} \in \mathcal{A}_1, h_{a_c} \in \mathcal{A}_c$): When h_{a_1} belongs to $G^{\phi}_{3(q-1)+1}$ or $G^{\phi}_{3(q-1)+2}$, its W coordinate will be higher than that of h_{a_c}, since for the clause hyperedge h_{a_c}, the W node is picked from $G^{\phi}_{3(q-1)+3}$. However, one of the other two coordinates in h_{a_1} is negative. So, it will be lower than that of h_{a_c}. So, we are done. When h_{a_1} belongs to $G^{\phi}_{3(q-1)+3}$, both h_{a_1} and h_{a_c} might share the W coordinate. However, in such h_{a_1}, the positive node among the U and V coordinate will be higher than that of h_{a_c}, since h_{a_1} comes from the highest variable among the triple, and both U and V coordinate increase with higher variables, being labelled $2(k-1)+1, 2$, whereas the negative coordinate will of course be lower than that of h_{a_c} which has no negative coordinate.

(6) $\mathcal{A}_2 - \mathcal{A}_c$ ($h_{a_2} \in \mathcal{A}_2, h_{a_c} \in \mathcal{A}_c$): Here when $h_{a_2} \in G^{\phi}_{3(q-1)+1}$, its V coordinate will be less since $Y_{3(q-1)+1}$ is the lowest variable, whereas the V coordinate of the

clause hyperedge h_{a_c} is picked from $G^{\phi}_{3(q-1)+2}$. However, the W coordinate will be higher for h_{a_2} as it is labelled $7n-9(q-1)-2$, whereas the clause gets the W coordinate corresponding to the $G^{\phi}_{3(q-1)+3}$ and hence the label $7n-9(q-1)-5$ or $7n-9(q-1)-6$. Whereas when $h_{a_2} \in G^{\phi}_{3(q-1)+2}$ or $G^{\phi}_{3(q-1)+3}$, the W coordinate will be lower for h_{a_2} (labelled $7n-9(q-1)-7$ or $7n-9(q-1)-8$ respectively) than h_{a_c} (labelled $7n-9(q-1)-5$ or $7n-9(q-1)-6$), whereas the U coordinate of h_{a_2} will be higher, since the clause hyperedge h_{a_c} gets the U coordinate corresponding to variable $Y_{3(q-1)+1}$ which is the lowest variable within the triple and hence has the lowest U coordinate (U labels being $2(k-1)+1, 2$).

□

List of Figures

List of Algorithms

Bibliography

[1] Shagnik Das. A brief note on estimates of binomial coefficients. http://page.mi.fu-berlin.de/shagnik/notes/binomials.pdf. Accessed on 2020-02-18.

[2] A. Abdesselam, C. Ikenmeyer, and G. Royle. 16,051 formulas for Ottaviani's invariant of cubic threefolds. *J. Algebra*, 447:649–663, 2016.

[3] M. Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of software technology and theoretical computer science*, volume 3821 of *Lecture Notes in Comput. Sci.*, pages 92–105. Springer, Berlin, 2005.

[4] M. Agrawal and S. Biswas. Primality and identity testing via Chinese remaindering. *J. ACM*, 50(4):429–443, 2003.

[5] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Ann. of Math. (2)*, 160(2):781–793, 2004.

[6] M. Agrawal, N. Kayal, and N. Saxena. Errata: PRIMES is in P. *Ann. of Math. (2)*, 189(1):317–318, 2019.

[7] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena. Jacobian hits circuits: Hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 599–614, 2012.

[8] M. Agrawal, N. Saxena, and S. S. Srivastava. Integer Factoring Using Small Algebraic Dependencies. In *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:14, 2016.

[9] Z. Allen-Zhu, A. Garg, Y. Li, R. M. de Oliveira, and A. Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In I. Diakonikolas, D. Kempe, and M. Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 172–181. ACM, 2018.

[10] E. Allender and F. Wang. On the power of algebraic branching programs of width two. *Comput. Complex.*, 25(1):217–253, Mar. 2016.

[11] J. Alman. Limits on the universal method for matrix multiplication. In *34th Computational Complexity Conference*, volume 137 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 12, 24. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019.

[12] N. Alon. Combinatorial Nullstellensatz. volume 8, pages 7–29. 1999. Recent trends in combinatorics (Mátraháza, 1995).

[13] J. Alper, T. Bogart, and M. Velasco. A lower bound for the determinantal complexity of a hypersurface. *Foundations of Computational Mathematics*, pages 1–8, 2015.

[14] M. Anderson, M. A. Forbes, R. Saptharishi, A. Shpilka, and B. L. Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, Dagstuhl, DEU, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[15] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[16] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.

[17] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[18] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. In *31st Annual Symposium on Foundations of Computer Science, Vol. I, II (St. Louis, MO, 1990)*, pages 16–25. IEEE Comput. Soc. Press, Los Alamitos, CA, 1990.

[19] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[20] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical Computer Science*, 22(3):317–330, 1983.

[21] M. Ben-Or and R. Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992.

[22] J. Berthomieu, J.-C. Faugère, and L. Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: the regular case. *J. Complexity*, 31(4):590–616, 2015.

[23] V. Bhargava, M. Bläser, G. Jindal, and A. Pandey. A deterministic PTAS for the algebraic rank of bounded degree polynomials. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 647–661. SIAM, Philadelphia, PA, 2019.

[24] D. Bini. Relations between exact and approximate bilinear algorithms. applications. *CALCOLO*, 17(1):87–97, Jan 1980.

[25] D. Bini, M. Capovani, F. Romani, and G. Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979.

[26] M. Bläser. Fast matrix multiplication. *Theory of Computing, Graduate Surveys*, 5:1–60, 2013.

[27] M. Bläser and C. Engels. Randomness efficient testing of sparse black box identities of unbounded degree over the reals. In *28th International Symposium on*

Theoretical Aspects of Computer Science, volume 9 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 555–566. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2011.

[28] M. Bläser, M. Hardt, R. J. Lipton, and N. K. Vishnoi. Deterministically testing sparse polynomial identities of unbounded degree. *Inform. Process. Lett.*, 109(3):187–192, 2009.

[29] M. Bläser, M. Hardt, and D. Steurer. Asymptotically optimal hitting sets against polynomials. In *Automata, languages and programming. Part I*, volume 5125 of *Lecture Notes in Comput. Sci.*, pages 345–356. Springer, Berlin, 2008.

[30] M. Bläser and C. Ikenmeyer. Introduction to geometric complexity theory. http://pcwww.liv.ac.uk/~iken/teaching_sb/summer17/introtogct/gct.pdf, version from July 25, 2018.

[31] M. Bläser, C. Ikenmeyer, G. Jindal, and V. Lysikov. Generalized matrix completion and algebraic natural proofs. In Diakonikolas et al. [63], pages 1193–1206.

[32] M. Bläser, C. Ikenmeyer, M. Mahajan, A. Pandey, and N. Saurabh. Algebraic Branching Programs, Border Complexity, and Tangent Spaces. In S. Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:24, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[33] M. Bläser, G. Jindal, and A. Pandey. A deterministic PTAS for the commutative rank of matrix spaces. *Theory of Computing*, 14(3):1–21, 2018.

[34] M. Bläser and A. Pandey. Polynomial Identity Testing for Low Degree Polynomials with Optimal Randomness. In J. Byrka and R. Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:13, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[35] J. Blasiak, T. Church, H. Cohn, J. A. Grochow, E. Naslund, W. F. Sawin, and C. Umans. On cap sets and the group-theoretic approach to matrix multiplication. *Discrete Anal.*, pages Paper No. 3, 27, 2017.

[36] M. Blum, A. K. Chandra, and M. N. Wegman. Equivalence of free Boolean graphs can be decided probabilistically in polynomial time. *Inform. Process. Lett.*, 10(2):80–82, 1980.

[37] M. Blum and S. Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, Jan. 1995.

[38] A. Bogdanov. Pseudorandom generators for low degree polynomials. In *STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 21–30. ACM, New York, 2005.

[39] J. Bondy and U. Murty. *Graph Theory*. Springer Publishing Company, Incorporated, 2008.

[40] C. Bouillaguet, P.-A. Fouque, and A. Véber. Graph-theoretic algorithms for the "isomorphism of polynomials" problem. In *Advances in cryptology—EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Comput. Sci.*, pages 211–227. Springer, Heidelberg, 2013.

[41] K. Bringmann, C. Ikenmeyer, and J. Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5):32:1–32:29, 2018.

[42] N. H. Bshouty. Testers and their applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:11, 2012.

[43] P. Bürgisser. Erratum to the complexity of factors of multivariate polynomials. https://www.math.tu-berlin.de/fileadmin/i26_fg-buergisser/erratum_factors.pdf.

[44] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2000.

[45] P. Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004.

[46] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[47] P. Bürgisser, C. Franks, A. Garg, R. M. de Oliveira, M. Walter, and A. Wigderson. Efficient algorithms for tensor scaling, quantum marginals, and moment polytopes. In M. Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 883–897. IEEE Computer Society, 2018.

[48] P. Bürgisser, C. Franks, A. Garg, R. M. de Oliveira, M. Walter, and A. Wigderson. Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes. In D. Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 845–861. IEEE Computer Society, 2019.

[49] P. Bürgisser, A. Garg, R. M. de Oliveira, M. Walter, and A. Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. *CoRR*, abs/1711.08039, 2017.

[50] P. Bürgisser, A. Garg, R. M. de Oliveira, M. Walter, and A. Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. In A. R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 24:1–24:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[51] P. Bürgisser and C. Ikenmeyer. Explicit lower bounds via geometric complexity theory. *Proceedings 45th Annual ACM Symposium on Theory of Computing 2013*, pages 141–150, 2013.

[52] P. Bürgisser, J. Landsberg, L. Manivel, and J. Weyman. An overview of mathematical issues arising in the Geometric complexity theory approach to VP v.s. VNP. *SIAM J. Comput.*, 40(4):1179–1209, 2011.

[53] J. F. Buss, G. S. Frandsen, and J. Shallit. The computational complexity of some problems of linear algebra. *J. Comput. Syst. Sci.*, 58(3):572–596, 1999.

[54] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation with applications to perfect matching and related problems. *SIAM J. Comput.*, 24(5):1036–1050, 1995.

[55] P. Chatterjee, M. Kumar, A. She, and B. L. Volk. A Quadratic Lower Bound for Algebraic Branching Programs. In S. Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:21, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[56] Z.-Z. Chen and M.-Y. Kao. Reducing randomness via irrational numbers. In *STOC '97 (El Paso, TX)*, pages 200–209. ACM, New York, 1999.

[57] M. Christandl, P. Vrana, and J. Zuiddam. Universal points in the asymptotic spectrum of tensors. In Diakonikolas et al. [63], pages 289–296.

[58] G. Cohen and A. Ta-Shma. Pseudorandom generators for low degree polynomials from algebraic geometry codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:155, 2013.

[59] N. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem minrank. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 402–421, 2001.

[60] E. Croot, V. F. Lev, and P. P. Pach. Progression-free sets in $\mathbb{Z}_4^n$ are exponentially small. *Ann. of Math. (2)*, 185(1):331–337, 2017.

[61] R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

[62] H. Derksen and V. Makam. Polynomial degree bounds for matrix semi-invariants. *Adv. Math.*, 310:44–63, 2017.

[63] I. Diakonikolas, D. Kempe, and M. Henzinger, editors. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. ACM, 2018.

[64] J. Dörfler, C. Ikenmeyer, and G. Panova. On geometric complexity theory: multiplicity obstructions are stronger than occurrence obstructions. *SIAM J. Appl. Algebra Geom.*, 4(2):354–376, 2020.

[65] Z. Dvir, A. Gabizon, and A. Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18(1):1–58, 2009.

[66] J. R. Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards Sect. B*, 71:241–245, 1967.

[67] J. S. Ellenberg and D. Gijswijt. On large subsets of $\mathbb{F}_q^n$ with no three-term arithmetic progression. *Ann. of Math. (2)*, 185(1):339–343, 2017.

[68] S. A. Fenner, R. Gurjar, and T. Thierauf. Bipartite perfect matching is in quasi-NC. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 754–763, 2016.

[69] M. Forbes. Personal communication with Christian Ikenmeyer.

[70] M. Forbes. Some concrete questions on the Border Complexity of polynomials, 2016. Talk at the Workshop on Algebraic Complexity Theory (WACT) 2016 in Tel Aviv.

[71] M. A. Forbes, S. Ghosh, and N. Saxena. Towards blackbox identity testing of log-variate circuits. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 54, 16. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

[72] M. A. Forbes, R. Saptharishi, and A. Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 867–875, New York, NY, USA, 2014. ACM.

[73] M. A. Forbes and A. Shpilka. Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, randomization, and combinatorial optimization*, volume 8096 of *Lecture Notes in Comput. Sci.*, pages 527–542. Springer, Heidelberg, 2013.

[74] M. A. Forbes and A. Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 243–252, Washington, DC, USA, 2013. IEEE Computer Society.

[75] M. Fortin and C. Reutenauer. Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank. *Séminaire Lotharingien de Combinatoire*, 52:B52f, 2004.

[76] H. Fournier, G. Malod, M. Szusterman, and S. Tavenas. Nonnegative Rank Measures and Monotone Algebraic Branching Programs. In A. Chattopadhyay and P. Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[77] H. N. Gabow and M. F. M. Stallmann. An augmenting path algorithm for linear matroid parity. *Combinatorica*, 6(2):123–150, 1986.

[78] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[79] A. Garg and R. M. de Oliveira. Recent progress on scaling algorithms and applications. *Bull. EATCS*, 125, 2018.

[80] A. Garg, L. Gurvits, R. M. de Oliveira, and A. Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In I. Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 109–117. IEEE Computer Society, 2016.

[81] A. Garg, L. Gurvits, R. M. de Oliveira, and A. Wigderson. Algorithmic and optimization aspects of Brascamp-Lieb inequalities, via operator scaling. In H. Hatami, P. McKenzie, and V. King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 397–409. ACM, 2017.

[82] A. Garg, L. Gurvits, R. M. de Oliveira, and A. Wigderson. Operator scaling: Theory and applications. *Found. Comput. Math.*, 20(2):223–290, 2020.

[83] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson. Operator scaling; theory and applications. *CoRR*, abs/1511.03730, 2015.

[84] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *57th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117, Oct 2016.

[85] F. Gesmundo. Geometric aspects of iterated matrix multiplication. *Journal of Algebra*, 461:42 – 64, 2016.

[86] F. Gesmundo, C. Ikenmeyer, and G. Panova. Geometric complexity theory and matrix powering. *Differential Geometry and its Applications*, 55:106 –6 127, 2017. Geometry and complexity theory.

[87] S. Goldwasser and O. Grossman. Bipartite perfect matching in pseudo-deterministic NC. In *44th International Colloquium on Automata, Languages, and Programming*, volume 80 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 87, 13. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.

[88] G. Gottlob and P. Senellart. Schema mapping discovery from data instances. *J. ACM*, 57(2):6:1–6:37, 2010.

[89] T. Gowers. Gowers weblog: What are dense Sidon subsets of {1,2,...,n} like? https://gowers.wordpress.com/2012/07/13/what-are-dense-sidon-subsets-of-12-n-like/. Published on 2012-07-13.

[90] B. Grenet. An upper bound for the permanent versus determinant problem. Manuscript, 2011.

[91] B. Grenet, P. Koiran, and N. Portier. On the complexity of the multivariate resultant. *J. Complexity*, 29(2):142–157, 2013.

[92] D. Grigoriev and N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *J. Symb. Comput.*, 5(1/2):37–64, 1988.

[93] J. A. Grochow. Unifying known lower bounds via geometric complexity theory. *Comput. Complex.*, 24(2):393–475, 2015.

[94] J. A. Grochow, K. D. Mulmuley, and Y. Qiao. Boundaries of VP and VNP. In I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 34:1–34:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[95] Z. Guo, N. Saxena, and A. Sinhababu. Algebraic dependencies and PSPACE algorithms in approximative complexity. In *Proceedings of the 33rd Computational Complexity Conference, CCC '18*, volume 102 of *LIPIcs*, pages 10:1–10:21, 2018.

[96] R. Gurjar, A. Korwar, and N. Saxena. Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *Theory Comput.*, 13:Paper No. 2, 21, 2017.

[97] R. Gurjar, A. Korwar, N. Saxena, and T. Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complexity*, 26(4):835–880, 2017.

[98] R. Gurjar and T. Thierauf. Linear matroid intersection is in quasi-NC. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 821–830, 2017.

[99] V. Guruswami and C. Xing. Hitting sets for low-degree polynomials with optimal density. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 161–168. IEEE Computer Society, 2014.

[100] M. Hardt, R. Meka, P. Raghavendra, and B. Weitz. Computational limits for matrix completion. In M. Balcan, V. Feldman, and C. Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 703–725. JMLR.org, 2014.

[101] N. J. Harvey. Algebraic algorithms for matching and matroid problems. *SIAM Journal on Computing*, 39(2):679–702, 2009.

[102] N. J. A. Harvey, D. R. Karger, and S. Yekhanin. The complexity of matrix completion. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 1103–1111. ACM Press, 2006.

[103] J. D. Hauenstein, C. Ikenmeyer, and J. M. Landsberg. Equations for lower bounds on border rank. *Exp. Math.*, 22(4):372–383, 2013.

[104] U. Heide-Jørgensen. On the determinantal complexity of the 2-hook-immanant. *PhD Dissertation, Department of Mathematics, Aarhus University*, 2012.

[105] J. Heintz and M. Sieveking. Lower bounds for polynomials with algebraic coefficients. *Theoretical Computer Science*, 11(3):321–330, 1980.

[106] G. Higman. The units of group-rings. *Proceedings of the London Mathematical Society*, 2(1):231–248, 1940.

[107] C. J. Hillar and L. Lim. Most tensor problems are np-hard. *J. ACM*, 60(6):45:1–45:39, 2013.

[108] J. Hüttenhain. *Geometric Complexity Theory and Orbit Closures of Homogeneous Forms*. PhD thesis, Technische Universität Berlin, July 2017.

[109] J. Hüttenhain and P. Lairez. The boundary of the orbit of the 3-by-3 determinant polynomial. *Comptes Rendus Mathematique*, 354(9):931 – 935, 2016.

[110] C. Ikenmeyer and U. Kandasamy. Implementing geometric complexity theory: On the separation of orbit closures via symmetries. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 713–726, New York, NY, USA, 2020. Association for Computing Machinery.

[111] C. Ikenmeyer and J. Landsberg. On the complexity of the permanent in various computational models. *Journal of Pure and Applied Algebra*, 221(12):2911 – 2927, 2017.

[112] R. Impagliazzo and A. Wigderson. P = BPP if E requires exponential circuits: derandomizing the XOR lemma. In *STOC '97 (El Paso, TX)*, pages 220–229. ACM, New York, 1999.

[113] G. Ivanyos and Y. Qiao. Algorithms based on $*$-algebras, and their applications to isomorphism of polynomials with one secret, group isomorphism, and polynomial identity testing. *SIAM J. Comput.*, 48(3):926–963, 2019.

[114] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Computational Complexity*, 26:1–33, 2017.

[115] C. Jacobi. De determinantibus functionalibus. *J. Reine Angew. Math.*, 22(4):319–359, 1841.

[116] G. Jindal. *On approximate polynomial identity testing and real root finding*. PhD thesis, 2019.

[117] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[118] K. Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM J. Comp.*, 14(3):678–687, 1985. (Conference version in ICALP 1982).

[119] N. Kayal, V. Nair, C. Saha, and S. Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Trans. Comput. Theory*, 11(1), Nov. 2018.

[120] A. D. King. Moduli of representations of finite-dimensional algebras. *Quart. J. Math. Oxford (2)*, 45:515–530, 1994.

[121] A. R. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 216–223, New York, NY, USA, 2001. ACM.

[122] P. Koiran. Hilbert's nullstellensatz is in the polynomial hierarchy. *J. Complex.*, 12(4):273–286, 1996.

[123] M. Kumar. On top fan-in vs formal degree for depth-3 arithmetic circuits. https://eccc.weizmann.ac.il/report/2018/068/revision/1/download, 2018.

[124] M. Kumar. A quadratic lower bound for homogeneous algebraic branching programs. *Computational Complexity*, 28(3):409–435, 2019.

[125] M. Kumar, R. Saptharishi, and A. Tengse. Near-optimal bootstrapping of hitting sets for algebraic circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646. SIAM, Philadelphia, PA, 2019.

[126] M. Kumar and S. Saraf. Arithmetic circuits with locally low algebraic rank. In *31st Conference on Computational Complexity, CCC 2016*, volume 50 of *LIPIcs*, pages 34:1–34:27, 2016.

[127] M. Kumar and B. L. Volk. Lower bounds for matrix factorization. *CoRR*, abs/1904.01182, 2019.

[128] J. M. Landsberg. *Tensors: Geometry and Applications*. AMS, 2012.

[129] J. M. Landsberg. Geometric complexity theory: an introduction for geometers. *ANNALI DELL'UNIVERSITA' DI FERRARA*, 61(1):65–117, May 2015.

[130] J. M. Landsberg. *Geometry and Complexity Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017.

[131] J. M. Landsberg, L. Manivel, and N. Ressayre. Hypersurfaces with degenerate duals and the geometric complexity theory program. *arXiv preprint arXiv:1004.4802*, 2010.

[132] D. Lewin and S. Vadhan. Checking polynomial identities over any field: towards a derandomization? In *STOC '98 (Dallas, TX)*, pages 438–447. ACM, New York, 1999.

[133] L. Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of computation theory (Proc. Conf. Algebraic, Arith. and Categorical Methods in Comput. Theory, Berlin/Wendisch-Rietz, 1979)*, volume 2 of *Math. Res.*, pages 565–574. Akademie-Verlag, Berlin, 1979.

[134] L. M. Lovász and L. Sauermann. A lower bound for the k-multicolored sum-free problem in $\mathbb{Z}_m^n$. *Proc. Lond. Math. Soc. (3)*, 119(1):55–103, 2019.

[135] S. Lovett. The analytic rank of tensors and its applications. *Discrete Anal.*, pages Paper No. 7, 10, 2019.

[136] C. Lu. Hitting set generators for sparse polynomials over any finite fields. In *2012 IEEE 27th Conference on Computational Complexity*, pages 280–286, June 2012.

[137] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *31st Annual Symposium on Foundations of Computer Science, Vol. I, II (St. Louis, MO, 1990)*, pages 2–10. IEEE Comput. Soc. Press, Los Alamitos, CA, 1990.

[138] M. Mahajan and V. Vinay. A combinatorial algorithm for the determinant. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '97, pages 730–738, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.

[139] M. Mahajan and V. Vinay. Determinant: combinatorics, algorithms, and complexity. *Chicago J. Theoret. Comput. Sci.*, pages Article 5, 26 pp. (electronic), 1997.

[140] V. Makam and A. Wigderson. Symbolic determinant identity testing (SDIT) is not a null cone problem; and the symmetries of algebraic varieties. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 881–888. IEEE, 2020.

[141] G. Malod and N. Portier. Characterizing valiant's algebraic complexity classes. *Journal of Complexity*, 24(1):16 – 38, 2008.

[142] K. Mulmuley and M. Sohoni. Geometric Complexity Theory. I. An approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526 (electronic), 2001.

[143] K. Mulmuley and M. Sohoni. Geometric Complexity Theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

[144] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[145] D. Mumford. *Algebraic geometry. I: Complex projective varieties.* Classics in mathematics. Springer-Verlag, Berlin, 1995. Reprint of the 1976 edition in Grundlehren der mathematischen Wissenschaften, vol. 221.

[146] E. Naslund and W. Sawin. Upper bounds for sunflower-free sets. *Forum Math. Sigma*, 5:e15, 10, 2017.

[147] N. Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 410–418. ACM, 1991.

[148] K. O'Bryant. A complete annotated bibliography of work related to Sidon sequences. *The Electronic Journal of Combinatorics*, DS11:39 p., 2004.

[149] J. B. Orlin. A fast, simpler algorithm for the matroid parity problem. In *Proc. 13th Internat. Conf. on Integer Programming and Combinatorial Optimization (IPCO'08)*, volume 5035 of *Lecture Notes in Comp. Sci.*, pages 240–258. Springer, 2008.

[150] J. G. Oxley. *Matroid theory*, volume 3. Oxford University Press, 2006.

[151] A. Pandey, N. Saxena, and A. Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low-algebraic-rank circuits. *Computational Complexity*, 27:617–670, 2018.

[152] J. Patarin. Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In U. Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 33–48, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[153] R. Peeters. Orthogonal representations over finite fields and the chromatic number of graphs. *Combinatorica*, 16(3):417–431, 1996.

[154] R. Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):Art. 40, 15, 2013.

[155] R. Raz and A. Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[156] I. Z. Ruzsa. An infinite Sidon sequence. *J. Number Theory*, 68(1):63–71, 1998.

[157] W. Sawin. Bounds for matchings in nonabelian groups. *Electron. J. Combin.*, 25(4):Paper 4.23, 21, 2018.

[158] N. Saxena. Diagonal circuit identity testing and lower bounds. In *International Colloquium on Automata, Languages, and Programming*, pages 60–71. Springer, 2008.

[159] N. Saxena. Progress on polynomial identity testing. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, (99):49–79, 2009.

[160] N. Saxena. Progress on polynomial identity testing-II. In *Perspectives in computational complexity*, volume 26 of *Progr. Comput. Sci. Appl. Logic*, pages 131–146. Birkhäuser/Springer, Cham, 2014.

[161] M. Schaefer. Realizability of graphs and linkages. In *Thirty essays on geometric graph theory*, pages 461–482. Springer, New York, 2013.

[162] M. Schaefer and D. Stefankovic. The complexity of tensor rank. *Theory Comput. Syst.*, 62(5):1161–1174, 2018.

[163] C.-P. Schnorr. Improved lower bounds on the number of multiplications/divisions which are necessary to evaluate polynomials. In *International Symposium on Mathematical Foundations of Computer Science*, pages 135–147. Springer, 1977.

[164] J. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

[165] I. R. Shafarevich. *Basic Algebraic Geometry 1.* Springer, Berlin, 1994.

[166] A. Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[167] F. B. Shepherd and A. Vetta. The demand-matching problem. *Math. Oper. Res.*, 32(3):563–578, 2007.

[168] Y. Shitov. How hard is the tensor rank? *CoRR*, abs/1611.01559, 2016.

[169] A. Shpilka and I. Volkovich. Improved polynomial identity testing for read-once formulas. In *Approximation, randomization, and combinatorial optimization*, volume 5687 of *Lecture Notes in Comput. Sci.*, pages 700–713. Springer, Berlin, 2009.

[170] A. Shpilka and A. Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[171] A. Silverberg. Applications to cryptography of twisting commutative algebraic groups. *Discrete Appl. Math.*, 156(16):3122–3138, 2008.

[172] S. Srinivasan. Strongly exponential separation between monotone VP and monotone VNP. arXiv:1903.01630, 2019.

[173] V. Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM Journal on Computing*, 3(2):128–149, 1974.

[174] T. Tao. A symmetric formulation of the Croot-Lev-Pach-Ellenberg-Gijswijt capset bound. https://terrytao.wordpress.com/2016/05/18/a-symmetric-formulation-of-the-croot-lev-pach-ellenberg-gijswijt-capset-bound/. Published on 2016-05-18.

[175] T. Tao and W. Sawin. Notes on the "slice rank" of tensors. https://terrytao.wordpress.com/2016/08/24/notes-on-the-slice-rank-of-tensors/. Published on 2016-08-24.

[176] I. C. Teodorescu. The module isomorphism problem for finite rings and related results. *ACM Commun. Comput. Algebra*, 49(1):14, June 2015.

[177] S. Toda. Classes of Arithmetic Circuits Capturing the Complexity of the Determinant. *IEICE TRANS. INF. & SYST.*, E75-D(1):116–124, 1992.

[178] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 1(2):107–111, 1947.

[179] L. G. Valiant. Completeness classes in algebra. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. ACM.

[180] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Information Theory*, 43(6):1757–1766, 1997.

[181] A. Wigderson. Operator scaling: theory, applications and connections. *Computational Complexity Conference*, (Invited talk), 2017.

[182] I. B. Yaacov. The Vandermonde determinant identity in higher dimension. *arXiv preprint arXiv:1405.0993*, 2014.

[183] A. Yehudayoff. Separating monotone VP and VNP. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 425–429, New York, NY, USA, 2019. Association for Computing Machinery.

[184] R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. Symbolic and Algebraic Comput. (EUROSAM'79)*, volume 72 of *Lecture Notes in Comp. Sci.*, pages 216–226. Springer, 1979.