

Schur Polynomials Do Not Have Small Formulas If the Determinant Doesn't

Prasad Chaugule

Department of Computer Science & Engineering, IIT Bombay, India
prasad@cse.iitb.ac.in

Mrinal Kumar

Department of Computer Science & Engineering, IIT Bombay, India
mrinal@cse.iitb.ac.in

Nutan Limaye

Department of Computer Science & Engineering, IIT Bombay, India
nutan@cse.iitb.ac.in

Chandra Kanta Mohapatra

Department of Computer Science & Engineering, IIT Bombay, India
ckm@cse.iitb.ac.in

Adrian She

Department of Computer Science, University of Toronto, Canada
ashe@cs.toronto.edu

Srikanth Srinivasan

Department of Mathematics, IIT Bombay, India
srikanth@math.iitb.ac.in

Abstract

Schur Polynomials are families of symmetric polynomials that have been classically studied in Combinatorics and Algebra alike. They play a central role in the study of Symmetric functions, in Representation theory [39], in Schubert calculus [26] as well as in Enumerative combinatorics [14, 38, 39]. In recent years, they have also shown up in various incarnations in Computer Science, e.g., Quantum computation [17, 31] and Geometric complexity theory [21].

However, unlike some other families of symmetric polynomials like the Elementary Symmetric polynomials, the Power Symmetric polynomials and the Complete Homogeneous Symmetric polynomials, the computational complexity of syntactically computing Schur polynomials has not been studied much. In particular, it is not known whether Schur polynomials can be computed efficiently by algebraic formulas. In this work, we address this question, and show that unless *every* polynomial with a small algebraic branching program (ABP) has a small algebraic formula, there are Schur polynomials that cannot be computed by algebraic formula of polynomial size. In other words, unless the algebraic complexity class VBP is equal to the complexity class VF, there exist Schur polynomials which do not have polynomial size algebraic formulas.

As a consequence of our proof, we also show that computing the determinant of certain *generalized* Vandermonde matrices is essentially as hard as computing the general symbolic determinant. To the best of our knowledge, these are one of the first hardness results of this kind for families of polynomials which are not *multilinear*. A key ingredient of our proof is the study of composition of *well behaved* algebraically independent polynomials with a homogeneous polynomial, and might be of independent interest.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Schur polynomial, Jacobian, Algebraic independence, Generalized Vandermonde determinant, Taylor expansion, Formula complexity, Lower bound

Digital Object Identifier 10.4230/LIPIcs.CCC.2020.14



© Prasad Chaugule, Mrinal Kumar, Nutan Limaye,
Chandra Kanta Mohapatra, Adrian She, and Srikanth Srinivasan;
licensed under Creative Commons License CC-BY

35th Computational Complexity Conference (CCC 2020).

Editor: Shubhangi Saraf; Article No. 14; pp. 14:1–14:27



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Related Version <https://eccc.weizmann.ac.il/author/1282/>,
<https://arxiv.org/abs/1911.12520>.

Funding *Mrinal Kumar*: A part of the work of the author was done during a postdoctoral stay at the University of Toronto.

Nutan Limaye: Supported by MATRICS grant MTR/2017/000909 awarded by SERB, Government of India.

Srikanth Srinivasan: Supported by MATRICS grant MTR/2017/000958 awarded by SERB, Government of India.

Acknowledgements Mrinal thanks Ramprasad Saptharishi, Amir Shpilka and Noam Solomon and Srikanth thanks Murali K. Srinivasan and Krishnan Sivasubramanian for many insightful discussions. The authors thank Amir for allowing us to include Question 42 and related discussion in this draft.

1 Introduction

In this paper, we explore a theme at the intersection of *Algebraic Complexity Theory*, which studies the computational complexity of computing multivariate polynomials using algebraic operations, and *Algebraic Combinatorics*, which studies, among other things, algebraic identities among polynomials associated to various combinatorial objects.

Specifically, the questions we study are related to the computational complexity of *Symmetric Polynomials*, which are polynomials in $\mathbb{C}[x_1, \dots, x_n]$ that are invariant under permutations of the underlying variable set x_1, \dots, x_n .¹ Examples of such polynomials include

- The *Elementary Symmetric polynomials* e_0, e_1, \dots, e_n where $e_d = \sum_{|S|=d} \prod_{j \in S} x_j$ is the sum of all multilinear monomials of degree exactly d ,
- The *Complete Homogeneous Symmetric polynomials* h_0, h_1, \dots where h_d is the sum of all monomials (multilinear or otherwise) of degree exactly d , and
- The *Power Symmetric polynomials* p_0, p_1, \dots where $p_d = \sum_{i=1}^n x_i^d$.

It is a standard fact that the above three families generate *all* symmetric polynomials in a well-defined sense. More precisely, the Fundamental Theorem of Symmetric Polynomials states that every symmetric polynomial f can be written *uniquely* as a polynomial in $\{e_1, \dots, e_n\}$, and similarly in $\{h_1, \dots, h_n\}$ and $\{p_1, \dots, p_n\}$, each of which is thus an algebraically independent set of polynomials. In particular, for $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_\ell)$ a non-increasing sequence of positive integers, if we define $e_\lambda = \prod_{i \in [\ell]} e_{\lambda_i}$, then the $\{e_\lambda\}_\lambda$ are linearly independent, and moreover the set $E_d := \{e_\lambda \mid \sum_i \lambda_i = d\}$ forms a basis for the vector space Λ_d of homogeneous symmetric polynomials of degree d ; the same is true also of h_λ and p_λ (defined analogously), yielding bases H_d and P_d respectively for Λ_d .

Symmetric Polynomials in Mathematics

The study of Symmetric Polynomials is a classical topic in Mathematics, with close connections to combinatorics and representation theory, among other fields (see, e.g., [28, 33]). In representation theory, it is known that the entries of the change-of-basis matrices between different bases for the space Λ_d yield important numerical invariants of various representations

¹ In Combinatorics literature, these are more commonly known as *Symmetric Functions*. One can also consider symmetric functions over fields other than the complex numbers, but throughout this paper, we will stick to \mathbb{C} .

of the symmetric group S_d . In algebraic and enumerative combinatorics, the study of symmetric polynomials leads to formulas and generating functions for interesting combinatorial quantities such as *Plane Partitions* (see, e.g. [39, Chapter 7]). These studies have in turn given rise to many interesting algebraic identities and generating functions for various families of symmetric polynomials. Some of these, as we note below, have already had consequences for computational complexity.

Algebraic Complexity of Symmetric Polynomials

Symmetric polynomials have also been intensively investigated by researchers in Algebraic complexity [30, 36, 35, 19, 13], with several interesting consequences. The famous “Ben-Or trick” in algebraic complexity (also known simply as “interpolation”) was discovered by Ben-Or [36] in the context of using a standard generating function for the Elementary Symmetric Polynomials to obtain small depth-3 formulas for e_1, \dots, e_n .² The same idea also yields small constant-depth formulas for the complete homogeneous symmetric polynomials. Symmetric polynomials have also been used to prove lower bounds for several interesting models of computation including homogeneous and inhomogeneous $\Sigma\Pi\Sigma$ formulas [30, 36, 35], homogeneous multilinear formulas [19] and homogenous $\Sigma\Pi\Sigma\Pi$ formulas [13]. Further, via reductions, the elementary and power symmetric polynomials have been used to define restricted models of algebraic computation known as the symmetric circuit model [35] and the $\Sigma \wedge \Sigma$ model (or Waring rank), which in turn have been significantly investigated (see, e.g. [34, 25, 32]).

Schur polynomials

In this paper, we study the complexity of an important family of symmetric polynomials called the *Schur Polynomials*, which we now define.

► **Definition 1.** Let $\lambda = (\lambda_1, \dots, \lambda_\ell)$ be a non-increasing sequence of positive integers with $\sum_i \lambda_i = d$. We define the Schur polynomial $s_\lambda(x_1, \dots, x_n)$ of degree d as follows.

$$s_\lambda = \frac{\det \left((x_i^{\lambda_j + n - j})_{i,j \in [n]} \right)}{\det \left((x_i^{n-j})_{i,j \in [n]} \right)}$$

(Here, if $\lambda = (\lambda_1, \dots, \lambda_\ell)$, then we define $\lambda_j = 0$ for $j > \ell$.)

The Schur polynomials are known to generalize the elementary symmetric polynomials as well as homogeneous symmetric polynomials. It is also known that the Schur polynomials of degree d form a basis for Λ_d , which is the vector space of all homogeneous symmetric polynomials of degree d .

The Schur polynomials occupy a central place in the study of symmetric polynomials. Their importance in representation theory can be seen for instance by the fact that, they describe the characters of representations of the general linear and symmetric groups. In particular, consider the general linear group $\text{GL}(V)$ over a complex vector space V of dimension n . If ρ is an irreducible representation of $\text{GL}(V)$ that is polynomial, meaning that the eigenvalues of $\rho(A)$ can be expressed as a polynomial in the eigenvalues of A , then the character $\text{Tr}(\rho(A))$ is a Schur polynomial $s_\lambda(x_1, \dots, x_n)$ evaluated at the eigenvalues

² The generating function referred to is $\prod_{i=1}^n (t - x_i) = \sum_{j=0}^n (-1)^{n-j} e_{n-j} t^j$.

14:4 Formula Complexity of Schur Polynomials

x_1, \dots, x_n of A , where λ is a partition with at most n non-zero parts. Furthermore, the entries of the change-of-basis matrix (for the vector space Λ_d) from the power symmetric polynomials to the Schur polynomials are exactly the values of the irreducible characters of the symmetric group S_d . Specifically, *the Murnaghan-Nakayama rule* states that when expanded into the basis of power symmetric polynomials we have

$$s_\lambda = \sum_{\mu=(\mu_1, \dots, \mu_l) \vdash n} \chi^\lambda(\mu) \prod_{i=1}^l p_{\mu_i}$$

where $\chi^\lambda(\mu)$ is an irreducible character of the symmetric group S_n evaluated at a permutation of cycle type μ . See [39, Chapter 7] for further details about these uses in representation theory.

Beyond representation theory, Schur polynomials are used in algebraic geometry in the Schubert calculus [26], which is used to calculate the number of ways in which Schubert subvarieties in the Grassmannian (the set of all k dimensional linear subspaces in an n -dimensional space) may intersect. Schur polynomials are also used in enumerative combinatorics, as they provide generating functions for counting various combinatorial objects, including plane partitions, tableaux [39, Chapter 7], reduced decompositions of permutations [38], and graph colourings [14].

Being one of the most well-studied objects in the theory of symmetric functions, Schur polynomials appear in many different avatars in the literature. The following classical definition is also known to capture Schur polynomials. The definition uses combinatorial structures called Ferrers diagrams. A Ferrers diagram (or a Young diagram or simply a diagram) of shape λ , is a left-aligned two-dimensional array of boxes with the i th row containing λ_i many boxes. (See, e.g. Stanley [39], for more about Ferrers diagrams.)

► **Definition 2.** Consider a Ferrers diagram of shape λ . For any non-decreasing sequence $\mu = (\mu_1, \dots, \mu_m)$ with $\sum_j \mu_j = d$, we define the Kostka number $K_{\lambda\mu}$ to be the number of ways of filling the boxes of the Ferrers diagram with numbers from $1, \dots, m$ such that each row is non-decreasing, each column is strictly increasing, and the number of i 's equals μ_i for each $i \in [m]$.

The Schur polynomial $s_\lambda(x_1, \dots, x_n) \in \Lambda_d$ is defined so that the coefficient of $x_1^{\mu_1} \cdots x_m^{\mu_m}$ is the Kostka number $K_{\lambda\mu}$ (the coefficients of other monomials are defined by symmetry).

In particular, $s_\lambda = 0$ if $n < \ell$. So we assume that $n \geq \ell$ throughout.

From this definition it is easy to see that Schur polynomials generalize both elementary symmetric polynomials (when $\ell = d$ and $\lambda_1 = \lambda_2 = \cdots = \lambda_d = 1$ in the definition above) and homogeneous symmetric polynomials (when $\ell = 1$ and $\lambda_1 = d$ in the definition above).

The Kostka numbers used in the definition above have been investigated extensively both from combinatorial and computational perspectives. (See for instance [39, 29].)

Algebraic Complexity of Schur Polynomials

In this work we focus on the algebraic complexity of Schur polynomials. As stated in Definition 1, which is also known as the *bialternant formula* of Jacobi, the Schur polynomial s_λ can be expressed as the ratio of two determinants. In particular, this implies that the Schur polynomials have algebraic circuits of size $\text{poly}(n, d)$. In fact, it also implies that these polynomials belong to the smaller algebraic complexity class VBP,³ for which the Determinant is the complete polynomial.

³ The class of polynomial families which can be efficiently computed by algebraic branching programs.

However, this upper bound is quite a bit weaker than what is known for other well-studied symmetric polynomials mentioned above, all of which have *constant-depth* formulas of polynomial size. We consider the question of whether the Schur polynomials have constant-depth formulas of polynomial size or even general (arbitrary depth) formulas of polynomial size.

Our main result is that under reasonable complexity assumptions, the answer to the above question is negative for many different λ . (Note that since the elementary and complete homogeneous symmetric polynomials are particular examples of Schur polynomials, there are *some* Schur polynomials that have formulas of polynomial size.)

► **Theorem 3 (Main Theorem).** *Assume that $\lambda = (\lambda_1, \dots, \lambda_\ell)$ is such that $\lambda_i \geq \lambda_{i+1} + (\ell - 1)$ for all $i \in [\ell - 1]$, also $\lambda_\ell \geq \ell$ and let $d = \sum_i \lambda_i$. Then, for $n \geq \lambda_1 + \ell$, if $s_\lambda(x_1, \dots, x_n)$ has an algebraic formula of size s and depth Δ , then the $\ell \times \ell$ determinant (\det_ℓ) has an algebraic formula of size $\text{poly}(s)$ and depth $\Delta + O(1)$.*

For suitable choices of ℓ, d, n above, we can ensure that these parameters are all polynomially related. The theorem then implies that the Schur polynomials do not have algebraic formulas of polynomial size unless the entire complexity class VBP collapses to the complexity class VF which consists of polynomials with small formulas. Moreover, the Schur polynomials do not have *constant-depth* formulas of subexponential size unless the determinant does, a result that would greatly improve the state-of-the-art in this direction [16].

The above theorem and its proof have several interesting aspects that we now elaborate on.

Newton iteration and formula complexity

Theorem 3 is motivated in part by a recent result of Bläser and Jindal [1] who prove the following interesting result about symmetric polynomials. As mentioned earlier, it is known that any symmetric polynomial $f_{sym} \in \mathbb{C}[x_1, \dots, x_n]$ can be written *uniquely* as a polynomial in (say) the elementary symmetric polynomials e_1, \dots, e_n . I.e., there exists a unique $f_E \in \mathbb{C}[x_1, \dots, x_n]$ such that

$$f_{sym}(x_1, \dots, x_n) = f_E(e_1, \dots, e_n).$$

Motivated by a question of Lipton and Regan [27], Bläser and Jindal studied the computational complexity of f_{sym} vis-a-vis that of f_E . It is clear that if f_E has algebraic circuits of polynomial size (resp. formulas) then so does f_{sym} , since the elementary symmetric polynomials have algebraic formulas of polynomial size. Interestingly, Bläser and Jindal showed a converse to this statement: they showed that if f_{sym} has small algebraic circuits, then so does f_E .⁴

At first sight, this looks highly relevant to our theorem, since by the classical *Jacobi-Trudi* identity (see, e.g. Theorem 24 in this paper or [39, Theorem 7.16.1]), when f_{sym} is a Schur polynomial of the type assumed in Theorem 3, then f_E is in fact the determinant (on a subset of its variables). We could hope to use the theorem of Bläser and Jindal to prove that if the Schur polynomial has a small formula, then so does the determinant. However, this doesn't quite work, since the proof of [1] only yields small *circuits* for the polynomial f_E , even if we assume that the polynomial f_{sym} has small *formulas*.

⁴ Bläser and Jindal work throughout with the elementary symmetric polynomials. However, using algebraic identities that link various symmetric polynomials with each other, we observe in this paper that their result also holds for the complete homogeneous and power symmetric polynomials.

14:6 Formula Complexity of Schur Polynomials

We briefly outline the reason for this, noting that this hurdle occurs quite often in trying to adapt results in algebraic circuit complexity to algebraic formulas. As mentioned above, the polynomials e_1, \dots, e_n are algebraically independent. A standard proof of this (see, e.g., [35]) goes via showing that the map

$$\bar{e} : \mathbb{C}^n \rightarrow \mathbb{C}^n, \text{ defined by } \bar{a} = (a_1, \dots, a_n) \mapsto (e_1(\bar{a}), \dots, e_n(\bar{a}))$$

is surjective. Hence, for each $\bar{b} \in \mathbb{C}^n$, there exists an $\bar{a} \in \mathbb{C}^n$ such that $\bar{e}(\bar{a}) = \bar{b}$. The reason this is relevant to the result of [1] is that if we have an efficient algorithm for “inverting” \bar{e} in this way and we additionally have an efficient algorithm for computing f_{sym} , then we immediately obtain an efficient algorithm for computing f_E on any given input $\bar{b} \in \mathbb{C}^n$ by first inverting the map \bar{e} to obtain an \bar{a} as above, and then applying the algorithm for f_{sym} to obtain $f(\bar{a}) = f_E(\bar{b})$. The main technical result in Bläser and Jindal’s work is to show how to invert the map \bar{e} as above using an algebraic circuit. The inversion is done by carefully applying a standard algebraic version of *Newton iteration*, which can be performed by an efficient algebraic circuit. Having done this, we plug the output of this circuit into the circuit for f_{sym} to obtain the circuit for f_E .

The reason the above proof does not work in the setting of algebraic formulas is the use of Newton iteration, which is not known to be doable with small formulas (or even within the seemingly larger class VBP). Indeed, this is the main bottleneck in translating several results in algebraic complexity on polynomial factorization [23, 8, 6] and hardness-randomness tradeoffs [22, 9, 5] that are known in the context of algebraic circuits to the setting of algebraic formulas.

In the proof of the main theorem, we show how to get around the use of Newton iteration in this setting and use it to prove a (slightly weaker) version of the result of Bläser and Jindal for algebraic formulas. We hope that the ideas we use here can be adapted and extended to circumvent the use of Newton iteration in some of the other settings mentioned above as well. Our main technical lemma is the following.

► **Lemma 4 (Main Technical Lemma (informal)).** *Let $g_1, \dots, g_n \in \mathbb{C}[x_1, \dots, x_n]$ be “well-behaved” algebraically independent polynomials. Then, for any homogeneous polynomial \tilde{f} , if $f = \tilde{f}(g_1, \dots, g_n)$ has a formula of size s and depth Δ , the polynomial \tilde{f} has a formula of size $\text{poly}(s)$ and depth $\Delta + O(1)$.*

For a formal definition of what we mean by “well-behaved” and for a formal statement of this lemma, we refer the reader to Definition 26 and Lemma 27 respectively. This lemma, and some of the ideas in its (very simple) proof might be of independent interest and may have other applications, e.g. in Subsection 4.4 we discuss an application of this lemma to some special cases of a question of Amir Shpilka on proving lower bounds on the partial derivative complexity of a product of algebraically independent polynomials.

Generalized Vandermonde determinants

The Vandermonde matrix $(x_i^{n-j})_{i,j \in [n]}$ and its determinant are ubiquitous in computation because of their relation to polynomial interpolation. More precisely, the problem of finding a degree- $(n-1)$ univariate polynomial that takes prescribed values at a specified set of n distinct points involves solving a linear system of equations where the underlying matrix is precisely the Vandermonde matrix. It is, therefore, an important fact that the Vandermonde determinant is computationally much easier than the general determinant: in fact, it has the following standard depth-2 formula

$$\det_n \left((x_i^{n-j})_{i,j \in [n]} \right) = \prod_{i,j \in [n]: i < j} (x_i - x_j).$$

However, it is unclear whether such small expressions continue to exist if we allow the exponents of the variables to vary more generally. For integers $\mu_1 > \mu_2 > \dots > \mu_n \geq 0$, consider the *generalized Vandermonde matrix* $(x_i^{\mu_j})_{i,j \in [n]}$. Similar to the Vandermonde matrix, the determinant of this matrix is related to the problem of *sparse polynomial interpolation*, where we are trying to interpolate a polynomial only involving the monomials of degree μ_1, \dots, μ_n through the given points.

Can we expect that computing any generalized Vandermonde determinant is much easier than computing the determinant itself? It follows from Theorem 3 and the bialternant formula from Definition 1 above that the answer to this question is negative: for certain (polynomially large) exponents, the generalized Vandermonde determinant is not much easier than the determinant.

Discussion on Schur Polynomials and Generating functions

In algebraic and enumerative combinatorics, we often study a family of related combinatorial objects by considering a *generating function* that combines them, in the hope that the generating function yields a nice closed-form expression which can further be used to estimate or otherwise understand these objects better. (See e.g. [41, 10] for much more about this.) For instance, we know that the generating functions for the elementary and complete homogeneous symmetric polynomials

$$E(t) = \sum_{i=0}^n t^i e_i(\mathbf{x}) \quad \text{and} \quad H(t) = \sum_{i=0}^n t^i h_i(\mathbf{x})$$

have small expressions given by

$$E(t) = \prod_{i \in [n]} (1 + tx_i) \quad \text{and} \quad H(t) = \frac{1}{\prod_{i \in [n]} (1 - tx_i)}.$$

Furthermore, as such expressions are algebraic formulas using additions, multiplications and divisions, we can use these formulas along with division elimination and interpolation to construct small algebraic formulas for the e_i s and h_j s themselves.

Recall that both the elementary and complete homogeneous symmetric polynomials are special cases of Schur polynomials. It therefore is natural to ask if generating functions can be obtained for other simple sequences of Schur polynomials. Our results imply that the generating function for certain sequences of Schur polynomials do *not* have small closed-form expressions with small formulas unless the determinant has small formulas. This seems like an interesting statement in algebraic combinatorics, conditioned upon a well-known conjecture in Computational Complexity theory.

For concreteness, here is one such “hard” generating function made up of Schur polynomials. For any $\ell \geq 0$, let $\lambda_\ell = (\ell^2, \ell^2 - \ell, \ell^2 - 2\ell, \dots, \ell)$. Define

$$S(t) = \sum_{\ell \geq 0} t^\ell s_{\lambda_\ell}.$$

Note that this is a finite sum for any fixed n as $s_{\lambda_\ell} = 0$ if $\ell > n$. In algebraic combinatorics, it is common to consider symmetric polynomials in an *infinite* number of variables in which case the above is truly an infinite sum. A simple expression in the infinite case typically leads to a simple expression in the finite case by simply setting all variables other than x_1, \dots, x_n to 0 in the expression.

Proving hardness of non-multilinear polynomial families

The most natural and widely studied notion of completeness in the algebraic setting is the notion of *projections*. A polynomial $P \in \mathbb{C}[x_1, \dots, x_n]$ is said to be a projection of a polynomial $Q \in \mathbb{C}[y_1, \dots, y_m]$ if there is a setting σ of y_1, \dots, y_m to either field constants or to variables from the set $\{x_1, \dots, x_n\}$, such that the polynomial $Q(\sigma(y_1), \sigma(y_2), \dots, \sigma(y_m))$ equals P . While this notion of reductions is very natural and intuitive and in particular, it is clear that *easiness* of Q (with respect to having a small algebraic circuit or formula, for instance) immediately implies the *easiness* of P , there is an inherent difficulty in using this notion of reductions for proving the hardness of families of non-multilinear polynomials. To see this, observe that if Q is non-multilinear in each of its variables, and P is a multilinear polynomial which depends on at least one variable, then P cannot be expressed as a projection of Q . In particular, this notion of reductions cannot be used to prove the hardness of non-multilinear Schur polynomials or the hardness of generalized Vandemonde determinant, assuming the hardness of determinant for algebraic formulas.⁵ We avoid this issue by showing that there is a *c-reduction* from the Determinant to the non-multilinear polynomials we study.⁶ More precisely, we show that a small formula for any of our hard non-multilinear polynomials can be used to construct a small formula for the Determinant polynomial with only a small blow-up in size and depth.

Other related work

The algebraic complexity of Schur polynomials has been studied in various restricted models of computation. Koev [24], Chan et al. [3] and Fomin et al. [11] consider the complexity of computing Schur polynomials in *the subtraction-free* algebraic circuit model. An algebraic circuit is *subtraction-free* if it uses only addition, multiplication and division operators.⁷ They showed that $s_\lambda(x_1, \dots, x_n)$ has subtraction-free circuits of size polynomial in n and λ_1 . In Fomin et al. [11], the authors also proved polynomial bounds on the size of the subtraction-free circuits computing other interesting variants of Schur polynomials such as *double Schur polynomials* and *skew Schur polynomials*. All the algorithms presented in [24, 3, 11] for computing Schur polynomials used division in non-trivial ways.

Demmel et al. [7] and Fomin et al. [12] studied the monotone complexity of Schur polynomials. In the monotone setting, only addition and multiplication operators are used. (Both division and subtraction operators are not allowed.) They proved exponential upper bounds on the monotone complexity of Schur polynomials and conjectured an exponential lower bound. The exact complexity of Schur polynomials is not resolved in the monotone setting. However, Grigoriev et al. [15] proved an exponential monotone circuit lower bound for a related family of symmetric polynomials, called the *monomial symmetric polynomials*.

⁵ There is a more general notion of projections which involves substituting y_1, \dots, y_m with affine linear functions in x_1, \dots, x_n . While such projections can be used to reduce multilinear polynomials to non-multilinear ones, the reduction must rely on clever cancellations in order to achieve this. Designing such reductions is in general challenging. We do not know of such a reduction in our setting.

⁶ A *c-reduction* in algebraic complexity is similar in spirit to Turing reductions in standard Computational Complexity. See for example [2, 20].

⁷ For example, consider the polynomial $x^2 - xy + y^2$. It is computed by the following subtraction-free circuit: $(x^3 + y^3)/(x + y)$.

Organization of the paper

The rest of the paper is organized as follows. We start with a brief discussion of some of the preliminaries in Section 2 and a brief introduction to Symmetric polynomials and Schur polynomials in Section 3. We formally state and prove Lemma 4 in Subsection 4.1, followed by its application to the proof of Theorem 3 in Subsection 4.2. We discuss further applications of some of these ideas to extending the result of Bläser and Jindal’s [1] in Subsection 4.3 and to the question of proving lower bounds on the partial derivative complexity of a product of algebraically independent polynomials in Subsection 4.4. We conclude with some open questions in Section 5.

2 Notations and Preliminaries

Throughout this paper, we assume that we are working over the field \mathbb{C} . It is not very hard to see that the results we present can be made to work for fields of characteristic zero or fields of sufficiently large characteristic. Boldface letters are used for tuples of variables e.g. \mathbf{x} for (x_1, x_2, \dots, x_n) . For $\mathbf{b} = (b_1, b_2, \dots, b_n) \in \mathbb{N}^n$ and $\mathbf{x} = (x_1, x_2, \dots, x_n)$, we use $\mathbf{x}^{\mathbf{b}}$ to denote $\prod_{i=1}^n x_i^{b_i}$. We use $|\mathbf{b}|_1$ to denote $\sum_{i \in [n]} b_i$.

2.1 Models of computation

We start by defining some of the standard models of algebraic computation that we work with in the rest of the paper.

► **Definition 5** (Algebraic circuit). *An algebraic circuit C is a directed acyclic graph with a unique sink node, called the root node. The source nodes are called leaves and are labelled with field constants or variables. All the other nodes are labelled with $+$ or \times . Each $+$ node computes the addition of the polynomials computed by its children and similarly, each \times node computes the multiplication of the polynomials computed by its children. The circuit is said to compute the polynomial computed by the root node.*

► **Definition 6** (Algebraic formula). *If the underlying graph of an algebraic circuit is a directed tree (which is a special type of a directed acyclic graph) then the circuit is called a formula.*

► **Definition 7** (Algebraic branching program). *An algebraic branching program (ABP) is a layered directed acyclic graph with a unique source vertex denoted s and a unique sink vertex denoted t . Each edge is labelled by a linear polynomial. The weight of a path p is the product of the labels of the edges in p . The polynomial that the ABP computes is the sum of all the weights of paths from s to t .*

2.2 Interpolation and Division elimination

We now state two fairly standard facts about algebraic formulas. The first of these relates the formula complexity of a polynomial to the formula complexity of each of its homogeneous components.

► **Lemma 8.** *Let $P(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ be a polynomial which can be computed by a formula of size at most s and depth Δ . Then, for every d , the homogeneous component of P of degree d can be computed by a formula of size at most $O(s^2)$ and depth $\Delta + O(1)$.*

14:10 Formula Complexity of Schur Polynomials

The proof of this lemma is via a standard interpolation argument, where we consider the polynomial $Q(t) = P(x_1t, x_2t, \dots, x_nt) \in \mathbb{C}(\mathbf{x})[t]$ as a univariate in t . The point to note is that the homogeneous components of P are coefficients of various powers of t in this new polynomial, and hence can be computed as a linear combination of sufficiently many evaluations of $Q(t)$ for distinct values of t in the base field (which we assume to be large enough). For every $\alpha \in \mathbb{C}$, the formula size of $Q(\alpha)$ is upper bounded by the formula size of P . Similarly the depth of the formula of $Q(\alpha)$ is bounded by the depth of P . The number of such distinct evaluations needed is upper bounded by one more than the degree of Q , which is one more than the degree of P itself. The final observation needed for proving the size upper bound is that a polynomial which can be computed by a formula of size s has degree upper bounded by s . Thus, we need to take an appropriately weighted linear combination of $s + 1$ distinct substitutions of t in Q , each of which has a formula of size at most s ; thereby giving us an upper bound of $O(s^2)$. Taking linear combinations of such substitutions can be done in depth $O(1)$, which gives the overall depth bound of $\Delta + O(1)$.

The next statement we need is about the formula complexity of a polynomial which can be written as quotient of two polynomials with small formulas.

► **Lemma 9.** *Let P and R be polynomials in $\mathbb{C}[\mathbf{x}]$ of formula (ABP/circuit) size at most s and depth at most Δ such that R divides P . Then, the polynomial $Q = \frac{P}{R}$ can be computed by a formula (an ABP/circuit resp.) of size at most $\text{poly}(s)$ and depth at most $\Delta + O(1)$.*

The proof of this lemma goes via the standard division elimination argument of Strassen and that of Lemma 8. We refer the reader to the excellent survey of Shpilka and Yehudayoff [37] for formal details on division elimination.

2.3 Algebraic independence and the Jacobian

The notion of algebraic independence that we now define plays a crucial role in the proofs in the paper. We start with a formal definition.

► **Definition 10.** *Polynomials $q_1, q_2, \dots, q_k \in \mathbb{C}[\mathbf{x}]$ are said to be algebraically independent over \mathbb{C} if there is no non-zero polynomial $g(y_1, y_2, \dots, y_k) \in \mathbb{C}[\mathbf{y}]$ such that $g(q_1, q_2, \dots, q_k)$ is identically zero.*

This definition generalizes the notion of *linear* independence, which is the special case when there is no non-zero polynomial g in k variables and degree 1 such that $g(q_1, q_2, \dots, q_k)$ is identically zero. As we shall see next, over fields of characteristic zero (or sufficiently large characteristic), the notion of algebraic independence is characterized by the rank of the Jacobian matrix defined below.

► **Definition 11.** *The Jacobian matrix of a tuple (q_1, q_2, \dots, q_k) of n -variate polynomials in $\mathbb{C}[\mathbf{x}]$, denoted by $\mathcal{J}(q_1, q_2, \dots, q_k)$ is a $k \times n$ matrix with entries from $\mathbb{C}[\mathbf{x}]$ whose $(i, j)^{\text{th}}$ entry equals $\frac{\partial q_i}{\partial x_j}$.*

Thus, the row corresponding to q_i in $\mathcal{J}(q_1, q_2, \dots, q_k)$ contains all of the n first order partial derivatives of q_i . In other words, the i^{th} row of the Jacobian gives us the gradient of q_i . The connection between algebraic independence and the Jacobian stems from the following (almost folklore) theorem.

► **Theorem 12 (Jacobian and Algebraic Independence).** *Let (q_1, q_2, \dots, q_k) be a k tuple of n -variate polynomials in $\mathbb{C}[\mathbf{x}]$ of degree at most d . Then, q_1, q_2, \dots, q_k are algebraically independent over \mathbb{C} if and only if the rank of the Jacobian matrix $\mathcal{J}(q_1, q_2, \dots, q_k)$ over the field $\mathbb{C}(\mathbf{x})$ is equal to k .*

A proof of this theorem can be found in the survey of Chen, Kayal and Wigderson [4, Chapter 3].

2.4 Taylor's expansion

For our proof, we need the following well-known form of Taylor's expansion.

► **Theorem 13.** *Let $P \in \mathbb{C}[\mathbf{x}]$ be an n -variate polynomial of degree at most d , and let $\mathbf{a} \in \mathbb{C}^n$ be a point. Then, for an n -tuple of variables \mathbf{z}*

$$P(\mathbf{a} + \mathbf{z}) = \sum_{i=0}^d \left(\sum_{\mathbf{u} \in \mathbb{N}^n, |\mathbf{u}|_1=i} \frac{\mathbf{z}^{\mathbf{u}}}{\mathbf{u}!} \cdot \frac{\partial P}{\partial \mathbf{x}^{\mathbf{u}}}(\mathbf{a}) \right)$$

where, for $\mathbf{u} = (u_1, u_2, \dots, u_n)$, $\mathbf{u}! = u_1! \cdot u_2! \cdots u_n!$.

Note that for $i = 0$, the summand $\left(\sum_{\mathbf{u} \in \mathbb{N}^n, |\mathbf{u}|_1=i} \frac{\mathbf{z}^{\mathbf{u}}}{\mathbf{u}!} \cdot \frac{\partial P}{\partial \mathbf{x}^{\mathbf{u}}}(\mathbf{a}) \right)$ is just equal to $P(\mathbf{a})$, and for every positive integer i at most d , this summand is a homogeneous polynomial of degree equal to i in \mathbf{z} . Of particular utility to us is the following easy corollary of Theorem 13.

► **Corollary 14.** *Let $P \in \mathbb{C}[\mathbf{x}]$ be an n -variate polynomial of degree $d \geq 1$, and let $\mathbf{a} \in \mathbb{C}^n$ be a point. Then, for an n -tuple of variables \mathbf{z}*

$$P(\mathbf{a} + \mathbf{z}) = P(\mathbf{a}) + \sum_{j=1}^n z_j \cdot \frac{\partial P}{\partial x_j}(\mathbf{a}) \pmod{\langle \mathbf{z} \rangle^2}.$$

2.5 Two useful lemmas

We use the following (well known) lemma in our arguments. While the lemma is essentially folklore, we sketch a proof for completeness.

► **Lemma 15.** *Let $f(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ and $P(\mathbf{x}, y) \in \mathbb{C}[\mathbf{x}, y]$ be polynomials such that $P(\mathbf{x}, f(\mathbf{x}))$ is identically zero. Then, there exists a polynomial $Q(\mathbf{x}, y) \in \mathbb{C}[\mathbf{x}, y]$ such that $P(\mathbf{x}, y) = (y - f(\mathbf{x})) \cdot Q(\mathbf{x}, y)$.*

Proof. Let d be the degree of P in y and let $C_0(\mathbf{x}), C_1(\mathbf{x}), \dots, C_d(\mathbf{x})$ be polynomials in $\mathbb{C}[\mathbf{x}]$ such that

$$P(\mathbf{x}, y) = \sum_{i=0}^d C_i(\mathbf{x}) \cdot y^i.$$

Therefore, $P(\mathbf{x}, f(\mathbf{x}))$ can be written as

$$P(\mathbf{x}, f(\mathbf{x})) = \sum_{i=0}^d C_i(\mathbf{x}) \cdot f(\mathbf{x})^i.$$

Subtracting the two expressions above, we get

$$P(\mathbf{x}, y) - P(\mathbf{x}, f(\mathbf{x})) = \sum_{i=0}^d C_i(\mathbf{x}) \cdot y^i - \sum_{i=0}^d C_i(\mathbf{x}) \cdot f(\mathbf{x})^i.$$

Now, on the right hand side, the term for $i = 0$ cancels out and on further simplification, we get

$$P(\mathbf{x}, y) - P(\mathbf{x}, f(\mathbf{x})) = \sum_{i=1}^d C_i(\mathbf{x}) \cdot (y^i - f(\mathbf{x})^i).$$

14:12 Formula Complexity of Schur Polynomials

Note that for every natural number $i \geq 1$, $y^i - f(\mathbf{x})^i$ is divisible by $(y - f(\mathbf{x}))$. Therefore, every summand on the right hand side has $(y - f(\mathbf{x}))$ as a factor, and thus there is a polynomial $Q(\mathbf{x}, y)$ such that

$$P(\mathbf{x}, y) - P(\mathbf{x}, f(\mathbf{x})) = (y - f(\mathbf{x})) \cdot Q(\mathbf{x}, y).$$

Moreover, since $P(\mathbf{x}, f(\mathbf{x}))$ is identically zero, we have that $P(\mathbf{x}, y) = (y - f(\mathbf{x})) \cdot Q(\mathbf{x}, y)$, thereby completing the proof of the lemma. \blacktriangleleft

In particular, if $|S| \geq d + 1$, then there exists some $\mathbf{a} \in S^n$ satisfying $P(\mathbf{a}) \neq 0$. This gives us a brute force deterministic algorithm, running in time $(d + 1)^n$, to test if an arithmetic circuit computing a polynomial of degree at most d in n variables is identically zero.

3 Symmetric polynomials

A polynomial is said to be symmetric if it is invariant under a permutation of variables. We now define some of the families of symmetric polynomials that are discussed in this paper and briefly discuss some of their properties. For a more detailed introduction on symmetric polynomials, we refer the reader to the book [28]. We start with the definitions.

► **Definition 16** (Elementary symmetric polynomials). *The elementary symmetric polynomial of degree k on n variables denoted by $e_k(\mathbf{x})$ is defined as follows:*

$$e_k(\mathbf{x}) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i.$$

The following fact states a property of the elementary symmetric polynomials which will be useful for our proofs in the later sections.

► **Fact 17.** *For all $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{C}$, if c_1, c_2, \dots, c_n are field elements such that*

$$\prod_{i=1}^n (z - \alpha_i) = z^n - c_1 \cdot z^{n-1} + c_2 \cdot z^{n-2} - \dots + (-1)^n c_n,$$

then, for every $j \in [n]$, $c_j = e_j(\alpha_1, \alpha_2, \dots, \alpha_n)$.

► **Definition 18** (Homogeneous (Complete) symmetric polynomials). *The homogeneous symmetric polynomial of degree k on n variables denoted by $h_k(\mathbf{x})$ is defined as follows:*

$$h_k(\mathbf{x}) = \sum_{\mathbf{b} \in \mathbb{N}^n: |\mathbf{b}|_1 = k} \mathbf{x}^{\mathbf{b}}.$$

► **Definition 19** (Homogeneous (Complete) symmetric polynomials). *The homogeneous symmetric polynomial of degree k on n variables denoted by $h_k(\mathbf{x})$ is defined as follows:*

$$h_k(\mathbf{x}) = \sum_{\mathbf{b} \in \mathbb{N}^n: |\mathbf{b}|_1 = k} \mathbf{x}^{\mathbf{b}}.$$

► **Definition 20** (Power symmetric polynomials). *The power symmetric polynomial of degree k on n variables denoted by $p_k(\mathbf{x})$ is defined as follows: $p_k = \sum_{i=1}^n x_i^k$.*

These sets of polynomials are algebraically independent. The following fact states this formally.

► **Fact 21.** *Let \mathbf{x} be an n tuple of variables. Then, elementary symmetric polynomials $e_1(\mathbf{x}), \dots, e_n(\mathbf{x})$ are algebraically independent over \mathbb{C} . Similarly, homogeneous symmetric polynomials $h_1(\mathbf{x}), \dots, h_n(\mathbf{x})$ and power symmetric polynomials $p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_n(\mathbf{x})$ are also algebraically independent over \mathbb{C} .*

We now formally state the fundamental theorem of symmetric polynomials, which essentially says that over field of characteristic zero, every symmetric polynomial can be written as a unique polynomial combinations of the elementary symmetric polynomials (similarly, for power symmetric polynomials or homogeneous symmetric polynomials).

► **Theorem 22** (The fundamental theorem of symmetric polynomials). *For a symmetric polynomial $f_{sym} \in \mathbb{C}[\mathbf{x}]$ there exists a unique polynomial $f \in \mathbb{C}[\mathbf{x}]$ s.t. $f_{sym} = f_E(e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_n(\mathbf{x}))$ where $e_i(\mathbf{x})$ is the elementary symmetric polynomial of degree i .*

Similarly, there exists a unique polynomial $f_H \in \mathbb{C}[\mathbf{x}]$ such that $f_{sym} = f_H(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_n(\mathbf{x}))$ and a unique polynomial $f_P \in \mathbb{C}[\mathbf{x}]$ such that $f_{sym} = f_P(p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_n(\mathbf{x}))$, where $h_i(\mathbf{x})$ is the homogeneous symmetric polynomial of degree i and $p_i(\mathbf{x})$ is the power symmetric polynomial of degree i .

3.1 Schur polynomials

A partition of a natural number d is any sequence $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_\ell)$ of non-negative integers in a non-increasing order $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_\ell \geq 0$ such that $\sum_{i=1}^{\ell} \lambda_i = d$.⁸ The number of non-zero parts of λ is called the length of λ and is denoted by $l(\lambda)$. The weight of λ , denoted by $|\lambda|$ is defined to be the sum of each individual component, i.e. $|\lambda| = \lambda_1 + \lambda_2 + \dots + \lambda_{l(\lambda)}$. If $|\lambda| = d$, then we say that λ is a partition of the number d or alternatively a partition of degree d .

Let λ be a partition of the number d . A Ferrers diagram (or simply a diagram) of shape λ is a left-aligned two-dimensional array of boxes with the i th row containing λ_i many boxes. The conjugate of λ , denoted by λ' , is the diagram obtained by switching the rows and columns of the diagram of λ .

► **Definition 23** (Schur polynomials). *Let λ be a partition of degree d and let $l(\lambda) \leq n$. Then the Schur polynomial $s_\lambda(\mathbf{x})$ is defined as*

$$s_\lambda(\mathbf{x}) = \frac{a_{\lambda+\delta}(\mathbf{x})}{a_\delta(\mathbf{x})}$$

where,

$$\delta = (n-1, n-2, \dots, 2, 1, 0)$$

$$a_{\lambda+\delta}(\mathbf{x}) = \det(x_i^{\lambda_j+n-j})_{1 \leq i, j \leq n}$$

$$a_\delta(\mathbf{x}) = \det(x_i^{n-j})_{1 \leq i, j \leq n} = \prod_{1 \leq i < j \leq n} (x_i - x_j)$$

⁸ Usually the λ_i s are assumed to be positive integers as defined earlier. For the sake of notational convenience we allow trailing zeroes in the definition of λ .

14:14 Formula Complexity of Schur Polynomials

We first observe that $s_\lambda(\mathbf{x})$ is a symmetric polynomial. To see this, note that if $x_i = x_j$ for any $i \neq j$ then $a_{\lambda+\delta}(\mathbf{x})$ is 0. Thus, by Lemma 15 and the fact that $x_i - x_j$ and $x_{i'} - x_{j'}$ do not share a common factor unless $\{i, j\} = \{i', j'\}$, $\prod_{i < j} (x_i - x_j)$ is factor of $a_{\lambda+\delta}(\mathbf{x})$ i.e., $a_\delta(\mathbf{x})$ is a factor of $a_{\lambda+\delta}(\mathbf{x})$. Therefore, $s_\lambda(\mathbf{x})$ is a polynomial. Moreover, for any permutation of variables, the sign changes in the numerator and the denominator are the same, and thus their ratio does not see a change in sign. This implies that $s_\lambda(\mathbf{x})$ is a symmetric polynomial.

We now state the classical Jacobi-Trudi identities which relates Schur polynomials to the elementary symmetric and homogeneous symmetric polynomials.

► **Theorem 24** (Jacobi-Trudi identities).

(1) $s_\lambda(\mathbf{x}) = \det(h_{\lambda_i - i + j}(\mathbf{x}))_{1 \leq i, j \leq \ell}$, where $\lambda = (\lambda_1, \dots, \lambda_\ell)$.

(2) $s_\lambda(\mathbf{x}) = \det(e_{\lambda'_i - i + j}(\mathbf{x}))_{1 \leq i, j \leq m}$, where λ' is the conjugate of λ and $m = l(\lambda')$.

In particular,

$$s_\lambda = \begin{vmatrix} h_{\lambda_1} & h_{\lambda_1+1} & \dots & h_{\lambda_1+\ell-1} \\ h_{\lambda_2-1} & h_{\lambda_2} & \dots & h_{\lambda_2+\ell-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{\lambda_\ell-\ell+1} & h_{\lambda_\ell-\ell+2} & \dots & h_{\lambda_\ell} \end{vmatrix}_{\ell \times \ell}$$

$$= \begin{vmatrix} e_{\lambda'_1} & e_{\lambda'_1+1} & \dots & e_{\lambda'_1+m-1} \\ e_{\lambda'_2-1} & e_{\lambda'_2} & \dots & e_{\lambda'_2+m-2} \\ \vdots & \vdots & \ddots & \vdots \\ e_{\lambda'_m-m+1} & e_{\lambda'_m-m+2} & \dots & e_{\lambda'_m} \end{vmatrix}_{m \times m}$$

Note that the identity depends only on the properties of λ (i.e. ℓ or m) and does not depend on the number of variables n .

► **Theorem 25.** For any λ , $s_\lambda(\mathbf{x})$ can be computed using a small ABP, hence by a small algebraic circuit.

Proof. For polynomials $P, Q \in \mathbb{C}[x_1, x_2 \dots x_n]$ such that both P and Q have small ABPs, then by Lemma 9; $R = \frac{P}{Q}$ also has a small ABP. Homogenization or interpolation can be used to extract the required polynomial without much blow up. Here both $a_{\lambda+\delta}(\mathbf{x})$, $a_\delta(\mathbf{x})$ have small ABPs (as they are small determinants), thus $s_\lambda(\mathbf{x})$ has an ABP of polynomial size which also implies that it has an algebraic circuit of polynomial size. ◀

It is well-known that $a_\delta(\mathbf{x})$, also known as the Principal Vandermonde Determinant, has a small algebraic formula. However much less is known about the complexity of $a_{\lambda+\delta}(\mathbf{x})$. These polynomials are also known as Generalized Vandermonde determinants and are well studied (see for instance [18]). To the best of our knowledge, before this work it was not known whether for all λ , $a_{\lambda+\delta}(\mathbf{x})$ s have small formulas. Suppose they did, then by Lemma 9, we get that $s_\lambda(\mathbf{x})$ also have small formulas for all λ . In this paper we show that there exists some λ for which $s_\lambda(\mathbf{x})$ does not have a small formula unless the Determinant has a small formula. This in particular implies that there exist λ such that $a_{\lambda+\delta}(\mathbf{x})$ cannot be computed using small formulas (unless the Determinant can be computed by a small formula).

4 Proofs of main results

4.1 Proof of Lemma 4

We start with the following definition, which is crucial for our proofs.

► **Definition 26** (Property S). *A set of n -variate polynomials $\{q_1, q_2, \dots, q_k\} \subseteq \mathbb{C}[\mathbf{x}]$ is said to satisfy Property S, if there exists an $\mathbf{a} \in \mathbb{C}^n$ such that*

- For all $i \in [k]$, $q_i(\mathbf{a}) = 0$, and,
- The rank of the Jacobian matrix of q_1, q_2, \dots, q_k when evaluated at \mathbf{a} is equal to its symbolic rank, i.e. $\text{rank}_{\mathbb{C}}(\mathcal{J}(q_1, q_2, \dots, q_k)(\mathbf{a})) = \text{rank}_{\mathbb{C}(\mathbf{x})}(\mathcal{J}(q_1, q_2, \dots, q_k))$.

Property S gives us a concrete way to capture an appropriate notion of *niceness* of a set of algebraically independent polynomials. The following lemma which uses this notion is a key technical ingredient of our proofs.

► **Lemma 27.** *Let $\{q_1, q_2, \dots, q_k\} \in \mathbb{C}[x_1, x_2, \dots, x_n]$ be a set of algebraically independent polynomials which satisfy Property S. Let $g \in \mathbb{C}[z_1, z_2, \dots, z_k]$ be a homogeneous k -variate polynomial of degree equal to d such that the composed polynomial $g(q_1, q_2, \dots, q_k) \in \mathbb{C}[\mathbf{x}]$ has an algebraic formula of size s and depth Δ . Then, $g(z_1, z_2, \dots, z_k)$ can be computed by an algebraic formula of size $O(s^2n)$ and depth $\Delta + O(1)$.*

Proof. Let Φ be the formula of size s which computes the polynomial $g(q_1(\mathbf{x}), q_2(\mathbf{x}), \dots, q_k(\mathbf{x}))$. Since q_1, q_2, \dots, q_k satisfy Property S, there is an $\mathbf{a} \in \mathbb{C}^n$ such that $q_1(\mathbf{a}) = q_2(\mathbf{a}) = \dots = q_k(\mathbf{a}) = 0$ and $\text{rank}_{\mathbb{C}}(\mathcal{J}(q_1, q_2, \dots, q_k)(\mathbf{a})) = \text{rank}_{\mathbb{C}(\mathbf{x})}(\mathcal{J}(q_1, q_2, \dots, q_k))$. Moreover, since they are algebraically independent, the rank of $(\mathcal{J}(q_1, q_2, \dots, q_k)(\mathbf{a}))$ is equal to k . Thus,

$$\text{rank}_{\mathbb{C}}(\mathcal{J}(q_1, \dots, q_k)(\mathbf{a})) = k.$$

Applying Corollary 14 to each $q_i(\mathbf{x})$ around this point $\mathbf{a} \in \mathbb{C}^n$, we get

$$q_i(\mathbf{a} + \mathbf{x}) = \sum_{j=1}^n x_j \cdot \frac{\partial q_i}{\partial x_j}(\mathbf{a}) \pmod{\langle \mathbf{x} \rangle^2}.$$

Observe that i^{th} row of the matrix $(\mathcal{J}(q_1, q_2, \dots, q_k)(\mathbf{a}))$ is the vector $\left(\frac{\partial q_i}{\partial x_1}(\mathbf{a}), \frac{\partial q_i}{\partial x_2}(\mathbf{a}), \dots, \frac{\partial q_i}{\partial x_n}(\mathbf{a})\right)$ and by the choice of \mathbf{a} , these vectors are linearly independent. Thus, the homogeneous linear forms $u_1(\mathbf{x}), u_2(\mathbf{x}), \dots, u_k(\mathbf{x})$ are linearly independent, where $u_i(\mathbf{x})$ is defined as

$$u_i(\mathbf{x}) = \sum_{j=1}^n x_j \cdot \frac{\partial q_i}{\partial x_j}(\mathbf{a}).$$

The rest of the proof follows immediately from the following two claims. We state the claims and use them to complete the proof of this lemma, and then move on to prove the claims.

▷ **Claim 28.** Let d be the degree of $g(\mathbf{x})$. Then, the homogeneous component of degree d of the polynomial $g(q_1, q_2, \dots, q_k)$ is equal to $g(u_1, u_2, \dots, u_k)$.

▷ **Claim 29.** If $g(u_1, u_2, \dots, u_k)$ has a formula of size s' and depth Δ , then $g(\mathbf{z})$ has a formula of size at most $s'n$ and depth $\Delta + O(1)$.

To complete the proof of the lemma, observe that given the formula Φ of size at most s and depth at most Δ which computes $g(q_1, q_2, \dots, q_k)$, we know from Lemma 8 that the homogeneous component of degree d of $g(q_1, q_2, \dots, q_k)$ can be computed by a formula Φ_1

14:16 Formula Complexity of Schur Polynomials

of size at most $O(s^2)$ and depth at most $\Delta + O(1)$. Moreover, from the homogeneity of g and Claim 28, we also know that Φ_1 computes the polynomial $g(u_1, u_2, \dots, u_k)$, where u_1, u_2, \dots, u_k are linearly independent linear forms. Thus, from Claim 29, this implies that $g(z_1, z_2, \dots, z_k)$ has a formula of size at most $O(s^2n)$ and depth $\Delta + O(1)$. This completes the proof of the lemma, modulo the two claims which we prove next. \blacktriangleleft

Proof of Claim 28. Let $f_1, f_2, \dots, f_k \in \mathbb{C}[\mathbf{x}]$ be polynomials which are zero modulo $\langle \mathbf{x} \rangle^2$ (i.e. every monomial in f_1, f_2, \dots, f_k has degree at least 2) such that for every i , $q_i(\mathbf{a} + \mathbf{x}) = u_i(\mathbf{x}) + f_i(\mathbf{x})$. Since g is a homogeneous polynomial of degree d , it can be expressed as

$$g(\mathbf{z}) = \sum_{\mathbf{b} \in \mathbb{N}^k, |\mathbf{b}|_1 = d} \alpha_{\mathbf{b}} \mathbf{z}^{\mathbf{b}},$$

for field constants $\alpha_{\mathbf{b}}$. Let $\mathbf{b} \in \mathbb{N}^k$ be any vector such that $|\mathbf{b}|_1 = d$. Observe that the homogeneous component of degree d of the polynomial

$$\prod_{j=1}^k q_j(\mathbf{a} + \mathbf{x})^{b_j} = \prod_{j=1}^k (u_j(\mathbf{x}) + f_j(\mathbf{x}))^{b_j}$$

is equal to $\prod_{j=1}^k u_j^{b_j}$. Thus, by linearity, the homogeneous component of degree d of

$$g(q_1(\mathbf{a} + \mathbf{x}), q_2(\mathbf{a} + \mathbf{x}), \dots, q_k(\mathbf{a} + \mathbf{x})) = \sum_{\mathbf{b} \in \mathbb{N}^k, |\mathbf{b}|_1 = d} \alpha_{\mathbf{b}} \cdot \prod_{j=1}^k (q_j(\mathbf{a} + \mathbf{x}))^{b_j}$$

equals

$$\sum_{\mathbf{b} \in \mathbb{N}^k, |\mathbf{b}|_1 = d} \alpha_{\mathbf{b}} \cdot \prod_{j=1}^k (u_j)^{b_j},$$

which, in turn is equal to the polynomial $g(u_1, u_2, \dots, u_k)$. \triangleleft

Proof of Claim 29. The idea for the proof of this claim is to show that each variable x_j can be replaced by a homogeneous linear form $\ell_j(\mathbf{z})$ in the variables \mathbf{z} such that for every $i \in [k]$, the linear form u_i satisfies $u_i(\ell_1(\mathbf{z}), \ell_2(\mathbf{z}), \dots, \ell_n(\mathbf{z})) = z_i$. Thus, under this linear transformation, the composed polynomial $g(u_1(\mathbf{x}), u_2(\mathbf{x}), \dots, u_k(\mathbf{x})) \in \mathbb{C}[\mathbf{x}]$ gets mapped to the polynomial $g(z_1, z_2, \dots, z_k)$. Once we can show an existence of these linear forms $\ell_1, \ell_2, \dots, \ell_n$, the bounds on the formula size and depth follow immediately since all we need to do to obtain a formula for $g(\mathbf{z})$ is to replace every occurrence of a variable in \mathbf{x} , e.g. x_j by the linear form $\ell_j(\mathbf{z})$. Since every such linear form has a formula of size at most k and depth $O(1)$, this process blows up the formula size by a multiplicative factor of at most $O(k)$ and the depth by an additive factor of $O(1)$.

Intuitively, to obtain these linear forms, we just *solve* the system of linear equations $U \cdot \mathbf{x}^T = \mathbf{z}^T$, where U is the $k \times n$ matrix whose i^{th} row is u_i . Since the rank of U is equal to k , let U' be an invertible $k \times k$ submatrix of U , and let V be the inverse of U' and let v_1, v_2, \dots, v_k be the rows of V . Moreover, for brevity, let us assume that U' consists of the first k columns of U . Observe that $(V \cdot U)$ is a $k \times n$ matrix such that its leftmost $k \times k$ sub-matrix is the identity matrix. We are now ready to define the linear forms $\{\ell_j : j \in [n]\}$. For $j \in [k]$, let $\ell_j(\mathbf{z})$ be equal to $v_j(\mathbf{z})$ and for $j > k$, $\ell_j(\mathbf{z})$ is defined to be zero. It is straightforward to check that this definition satisfies the desired property and we skip the details. \triangleleft

4.2 Formula complexity of Schur polynomials

We are now ready to prove Theorem 3. Our first stop, which we reach in the next two lemmas, is to show that which shows that the elementary symmetric polynomials of degree at most $n - 1$ on n variables are *well behaved*. We start by establishing a sufficient condition for their Jacobian matrix to have full rank at a point.

► **Lemma 30.** *Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ be an n -tuple of variables. Let $J(\mathbf{x})$ be the Jacobian of $e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_{n-1}(\mathbf{x})$, and let $\mathbf{b} = (b_1, b_2, \dots, b_n) \in \mathbb{C}^n$ be such that for all $i \neq j$, $b_i \neq b_j$. Then,*

$$\text{rank}_{\mathbb{C}(\mathbf{x})}(J(\mathbf{x})) = \text{rank}_{\mathbb{C}}(J(\mathbf{b})) = n - 1.$$

Proof. Using Fact 21, we can observe that the n -variate polynomials $e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_{n-1}(\mathbf{x})$ are algebraically independent. Therefore, from Theorem 12 we know that $\text{rank}_{\mathbb{C}(\mathbf{x})}(J(\mathbf{x}))$ is equal to $n - 1$.

Let $J'(\mathbf{x})$ be any $(n - 1) \times (n - 1)$ submatrix of $J(\mathbf{x})$ of rank equal to $n - 1$ and by symmetry we can assume that the columns in $J'(\mathbf{x})$ come from the first $n - 1$ columns of $J(\mathbf{x})$. Thus, for $i, j \in [n - 1]$, the (i, j) entry of $J'(\mathbf{x})$ is equal to $\frac{\partial e_i}{\partial x_j}$. We now show that the rank of $J'(\mathbf{b})$ over \mathbb{C} is equal to $n - 1$ and this would complete the proof. To this end, we observe that the determinant of $J'(\mathbf{x})$ is a non-zero scalar multiple of $\prod_{i, i' \in [n-1], i \neq i'} (x_i - x_{i'})$. Since the coordinates of \mathbf{b} are all distinct, this determinant remains non-zero on \mathbf{b} .

From the definition of $J'(\mathbf{x})$, we know that the entries in its i^{th} row are homogeneous polynomials of degree equal to $i - 1$. Thus, $\det(J'(\mathbf{x}))$ is a homogeneous polynomial in \mathbf{x} of degree equal to

$$0 + 1 + 2 + \dots + n - 2 = \frac{(n - 2)(n - 1)}{2}.$$

Recall that the (i, j) entry of $J'(\mathbf{x})$, is equal to $\frac{\partial e_i}{\partial x_j}$, which is equal to $\sum_{S \subseteq [n] \setminus \{i\}} \prod_{k \in S} x_k$. Thus, if we replace every occurrence of the variable x_i by the variable $x_{i'}$ for $i \neq i' \in [n - 1]$, then columns i, i' in $J'(\mathbf{x})$ become identical, and hence $\det(J'(\mathbf{x}))$ is identically zero. Thus, by Lemma 15, $(x_i - x_{i'})$ is a factor of $\det(J'(\mathbf{x}))$. Also, for any two distinct sets $\{i_1, i'_1\}$ and $\{i_2, i'_2\}$ where $i_1 \neq i'_1$ and $i_2 \neq i'_2$, the polynomials $(x_{i_1} - x_{i'_1})$ and $(x_{i_2} - x_{i'_2})$ do not share a non-trivial divisor. Thus, the determinant of $J'(\mathbf{x})$ must be divisible by the polynomial $\prod_{i, i' \in [n-1], i \neq i'} (x_i - x_{i'})$. Moreover, we observed that the degree of determinant of $J'(\mathbf{x})$ is equal to $\frac{(n-2)(n-1)}{2}$, which is also equal to the degree of $\prod_{i, i' \in [n-1], i \neq i'} (x_i - x_{i'})$. Thus, they must be non-zero scalar multiples of each other. This observation, together with the fact that the coordinates of \mathbf{b} are all distinct, shows that $\det(J'(\mathbf{x}))$ is non-zero at \mathbf{b} and hence, $\text{rank}(J(\mathbf{b}))$ equals $n - 1$ over \mathbb{C} . ◀

We now use Lemma 27 to show that $e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_{n-1}(\mathbf{x})$ satisfy Property S, where n is the number of variables.

► **Lemma 31.** *Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ be an n -tuple of variables. Then, the set of elementary symmetric polynomials of degree at most $n - 1$ on \mathbf{x} , i.e. the set $\{e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_{n-1}(\mathbf{x})\}$ of polynomials satisfies Property S.*

Proof. Let $\mathbf{a} = (1, \omega, \omega^2, \dots, \omega^{n-1})$, where ω is the primitive n^{th} root of unity. So, we have the following identity

$$z^n - 1 = \prod_{i=1}^n (z - \omega^{i-1}).$$

14:18 Formula Complexity of Schur Polynomials

However, from Fact 17, we also know that $z^n - 1$, which equals $\prod_{i=1}^n (z - a_i)$ can be expressed as

$$\prod_{i=1}^n (z - a_i) = z^n - c_1 \cdot z^{n-1} + c_2 \cdot z^{n-2} - \dots + (-1)^n c_n,$$

where $c_i = e_i(\mathbf{a})$. Comparing these two expressions for $z^n - 1$, we get that for all $1 \leq i < n$, $c_i = e_i(\mathbf{a}) = 0$. Thus, $\mathbf{a} = (1, \omega, \omega^2, \dots, \omega^{n-1})$ is a common zero of e_1, e_2, \dots, e_{n-1} , which satisfies the first item in Definition 26. Moreover, since ω is a primitive n^{th} root of one, we also know that for all pairs $i \neq j$, $a_i \neq a_j$. Therefore, by Lemma 30, \mathbf{a} satisfies the second condition in Definition 26. Thus, the n -variate polynomials $e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_{n-1}(\mathbf{x})$ satisfy Property S. \blacktriangleleft

We now observe that analogous statements are also true for homogeneous symmetric polynomials and power symmetric polynomials as well.

► **Lemma 32.** *Let $\mathbf{x} = (x_1, x_2, \dots, x_n)$ be an n -tuple variables. Then the set of complete symmetric polynomials of degree at most $n - 1$ on \mathbf{x} , i.e. the set $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_{n-1}(\mathbf{x})\}$ of polynomials satisfies Property S. Similarly, the set $\{p_1(\mathbf{x}), p_2(\mathbf{x}), \dots, p_{n-1}(\mathbf{x})\}$ of power symmetric polynomials of degree at most $n - 1$ also satisfies Property S.*

Proof sketch. The proof goes via generating functions of e_i, h_i, p_i and the relations among them. We denote the generating function of e_i, h_i, p_i by $E(t), H(t), P(t)$ respectively. The following relations are known between these polynomials. (See for instance [28].)

$$\begin{aligned} E(t) &= \prod_{i \geq 1} (1 + x_i t) = \sum_{k \geq 0} e_k t^k \\ H(t) &= \prod_{i \geq 1} \frac{1}{1 - x_i t} = \sum_{k \geq 0} h_k t^k \\ P(t) &= \prod_{i \geq 1} \frac{x_i}{1 - x_i t} = \sum_{k \geq 1} p_k t^{k-1} \end{aligned}$$

It is easy to see that $E(-t)H(t) = 1$. Therefore we get,

$$H(t) = \frac{1}{E(-t)} = \frac{1}{1 - e_1 t + e_2 t^2 - \dots + (-1)^n e_n t^n} \quad (1)$$

From Lemma 31, we know there exists a point $\mathbf{a} = (a_1, a_2, \dots, a_n)$, where $e_1(\mathbf{a}), e_2(\mathbf{a}), \dots, e_{n-1}(\mathbf{a})$ vanish and $e_n(\mathbf{a})$ is non-zero. We evaluate the above equation at the same \mathbf{a} . We denote the above evaluation by $H(t)|_{\mathbf{x}=\mathbf{a}}$.

$$H(t)|_{\mathbf{x}=\mathbf{a}} = 1 - (-1)^n e_n(\mathbf{a}) t^n + ((-1)^n e_n(\mathbf{a}) t^n)^2 \dots$$

Observing the equation for $H(t)|_{\mathbf{x}=\mathbf{a}}$, it follows that at point \mathbf{a} , $h_1(\mathbf{a}), h_2(\mathbf{a}), \dots, h_{n-1}(\mathbf{a})$ are zero and $h_n(\mathbf{a})$ is non-zero. An analogous relation between $P(t)$ and $E(t)$, given by

$$P(t) = \frac{E'(-t)}{E(-t)},$$

where $E'(-t)$ is the first derivative of $E(-t)$ with respect to t , can be used to show that the power symmetric polynomials $p_1(\mathbf{a}), \dots, p_{n-1}(\mathbf{a})$ are zero and $p_n(\mathbf{a})$ is non-zero.

Thus, we are halfway towards showing that the set of power symmetric polynomials of degree at most $n - 1$ and homogeneous symmetric polynomials of degree at most $n - 1$ also satisfy Property S. It remains to be argued that the rank of the Jacobian of these polynomials at \mathbf{a} is equal to $n - 1$. The proof is analogous to the argument for the elementary symmetric polynomials, as in the proof of Lemma 31. We skip the remaining details for brevity. \blacktriangleleft

From Definition 26 of Property S and Theorem 12, it follows that if a set A of polynomials satisfies Property S, then all the non-empty subsets of A also satisfy Property S. Thus, we have the following corollary of Lemma 31 and Lemma 32.

► **Corollary 33.** *Let $i_1 < \dots < i_k < n$ be positive integers. Consider the sets $E = \{e_{i_1}(\mathbf{x}), \dots, e_{i_k}(\mathbf{x})\}$ and $H = \{h_{i_1}(\mathbf{x}), \dots, h_{i_k}(\mathbf{x})\}$ of elementary and homogeneous symmetric polynomials respectively in $n > i_k$ variables. Then both E and H satisfy property S.*

We are now ready to prove the main theorem.

► **Theorem 34 (Main theorem).** *Let $\lambda = (\lambda_1, \dots, \lambda_\ell)$ be a partition of d such that $\lambda_i \geq \lambda_{i+1} + (\ell - 1)$ for all $i \in [\ell - 1]$, and $\lambda_\ell \geq \ell$. Then, for all n , such that $n \geq \lambda_1 + \ell$, if $s_\lambda(x_1, \dots, x_n)$ has an algebraic formula of size s and depth Δ , then the $\ell \times \ell$ determinant (\det_ℓ) has an algebraic formula of size $\text{poly}(s)$ and depth $\Delta + O(1)$.*

Proof. We use Jacobi-Trudi Identity from Theorem 24, which expresses $s_\lambda(\mathbf{x})$ in the form of homogeneous symmetric determinant.

$$s_\lambda = \begin{vmatrix} h_{\lambda_1} & h_{\lambda_1+1} & \dots & h_{\lambda_1+\ell-1} \\ h_{\lambda_2-1} & h_{\lambda_2} & \dots & h_{\lambda_2+\ell-2} \\ \vdots & \vdots & \ddots & \vdots \\ h_{\lambda_\ell-\ell+1} & h_{\lambda_\ell-\ell+2} & \dots & h_{\lambda_\ell} \end{vmatrix}$$

Let M_λ denote the matrix on the right hand side in the above equality. By our choice of λ , observe that the highest degree entry of M_λ is $h_{\lambda_1+\ell-1}$, which has degree at most $n - 1$, and the lowest degree entry of M_λ is $h_{\lambda_\ell-\ell+1}$ which for $\lambda_\ell \geq \ell$ has degree at least 1. Moreover, from the choice of λ , it also follows that all the entries of M_λ are distinct.

From Lemma 32, we know that there exists a point \mathbf{a} for which the n -variate polynomials $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_{n-1}(\mathbf{x})\}$ satisfy the Property S, where n can be taken to be strictly greater than ℓ^2 . Thus, now if we take the ℓ^2 -variate homogeneous polynomial g to be the symbolic determinant of an $\ell \times \ell$ matrix, then s_λ is obtained by a composition of g with a subset of polynomials h_1, h_2, \dots, h_{n-1} .

Thus, by Lemma 27, we get that if $g(h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_{\ell^2}(\mathbf{x}))$ (i.e s_λ) has an algebraic formula of size s and depth Δ , then $g(x_1, x_2, \dots, x_{\ell^2})$ also has a small formula of size $\text{poly}(s)$ and depth $\Delta + O(1)$. ◀

► **Remark 35 (Contrasting hard and easy λ s).** Let $\lambda = (\ell^2, \ell^2 - \ell, \dots, 2\ell, \ell, 0, 0, \dots, 0, 0, 0)$ and let $\tilde{\lambda} = (n\ell, (n-1)\ell, \dots, \ell^2, \ell^2 - \ell, \dots, 3\ell, 2\ell, \ell)$, where we will take $n \geq \ell^2 + \ell$ and $\ell > 0$. It is easy to see that $\tilde{\lambda}$ forms an arithmetic progression in which the difference between the successive terms is ℓ . Whereas λ is a truncated arithmetic progression, in which $n - \ell$ -many elements are zeroes and the non-zero elements form an arithmetic progression.

Here the structure of λ and $\tilde{\lambda}$ is quite similar, but the algebraic complexities of s_λ and $s_{\tilde{\lambda}}$ are different. In particular, Theorem 34 is applicable to λ . Therefore we can conclude that s_λ does not have a small algebraic formula unless the determinant has a small algebraic formula.

14:20 Formula Complexity of Schur Polynomials

On the other hand, there are small formulas for $s_{\tilde{\lambda}}$. To see this, observe that $\tilde{\lambda} + \delta = ((n-1)(\ell+1) + \ell, (n-2)(\ell+1) + \ell, \dots, (\ell+1) + \ell, \ell)$, which is also an arithmetic progression in which the difference between successive terms is $\ell + 1$. Therefore, we have

$$\begin{aligned}
 a_{\tilde{\lambda}+\delta} &= \begin{vmatrix} x_1^\ell & x_2^\ell & \dots & x_n^\ell \\ x_1^{(\ell+1)+\ell} & x_2^{(\ell+1)+\ell} & \dots & x_n^{(\ell+1)+\ell} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n-1)(\ell+1)+\ell} & x_2^{(n-1)(\ell+1)+\ell} & \dots & x_n^{(n-1)(\ell+1)+\ell} \end{vmatrix}_{n \times n} \\
 a_{\tilde{\lambda}+\delta} &= \left(\prod_{i=1}^n x_i^\ell \right) \begin{vmatrix} 1 & 1 & \dots & 1 \\ x_1^{\ell+1} & x_2^{\ell+1} & \dots & x_n^{\ell+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(n-1)(\ell+1)} & x_2^{(n-1)(\ell+1)} & \dots & x_n^{(n-1)(\ell+1)} \end{vmatrix}_{n \times n} \\
 s_{\tilde{\lambda}} &= \frac{\left(\prod_{i=1}^n x_i^\ell \right) \prod_{i < j} (x_j^{\ell+1} - x_i^{\ell+1})}{\prod_{i < j} (x_j - x_i)}
 \end{aligned}$$

In the above expression the numerator and denominator have small formulas and therefore using Lemma 9 we get that $s_{\tilde{\lambda}}$ has a small formula of size $\text{poly}(\ell, n)$.

4.2.1 Generalization to Skew Schur Polynomials

A straightforward generalization of the previous result is to prove that a class of skew Schur polynomials (defined in [28]) also hard for formulas. They can be defined via a Jacobi-Trudi like formula.

► **Theorem 36.** *Let μ and λ be partitions with $\mu_i \leq \lambda_i$ for every part i . Then, the skew Schur polynomial $s_{\lambda/\mu}$ satisfies*

- (1) $s_{\lambda/\mu} = \det(h_{\lambda_i - \mu_j - i + j})_{1 \leq i, j \leq k}$ where $k = l(\lambda)$.
- (2) $s_{\lambda/\mu} = \det(e_{\lambda'_i - \mu'_j - i + j})_{1 \leq i, j \leq k}$ where $k = l(\lambda')$.

From these definitions of skew Schur polynomials, we can see that they also have ABPs of polynomial size, like Schur polynomials. However, skew Schur polynomials are in general linear combinations of Schur polynomials, by the Littlewood-Richardson rule

$$s_{\lambda/\mu} = \sum_{\nu \vdash n-m} c_{\mu, \nu}^\lambda s_\nu.$$

The Littlewood-Richardson coefficients $c_{\mu, \nu}^\lambda$ count tableau whose Young diagram is of shape λ/μ and whose content satisfy certain technical conditions. They are also important in representation theory as they describe how Schur polynomials multiply, or equivalently how a tensor product of polynomial $\text{GL}(n)$ representations decomposes into irreducible representations. They are also known to be #P-hard to compute [29].

Hardness of skew Schur polynomials assuming hardness of determinant follows from the following lemma, Corollary 33 and Lemma 27.

► **Lemma 37.** *Let $l \geq 2$ and $\mu_1 \geq 1$ be positive integers. Let λ, μ be partitions with l parts with $\lambda_i = (l - (i - 1))l + \mu_1$ and $\mu_i = \mu_1$ for all $i < l$ and $\mu_l = \mu_1 - 1$. Then all entries of the homogeneous Jacobi-Trudi determinant $s_{\lambda/\mu}$ are distinct.*

Proof. For simplicity, we call the integer k the label of the homogeneous symmetric polynomial h_k . Here, the (i, j) entry of the matrix is $h_{(l-(i-1))l-i+j}$ for $1 \leq j < l$ and $h_{(l-(i-1))l+1-i+j}$ for $j = l$. Hence, for a fixed row i , all entries are distinct and the labels increase from left to right. Furthermore,

$$(l-i)l+1-(i+1)+l=(l-i)(l-1)+1-i < l(l-(i-1))-i+1$$

so the label of the last entry of row $(i+1)$ is strictly less than the label of the first entry of row i . Hence, all entries of the Jacobi-Trudi determinant are distinct. ◀

4.3 Extensions of the results of Bläser and Jindal [1]

Shifted variants for formulas

A fairly direct consequence of our techniques is the following theorem which can be considered a partial generalization of the result of Bläser-Jindal [1] for algebraic formulas.

▶ **Theorem 38.** *There exist field constants a_1, a_2, \dots, a_n such that the following is true: if for an n -variate homogeneous polynomial g over \mathbb{C} , the composition $g(e_1 - a_1, \dots, e_n - a_n)$ has a formula of size at most s , then the polynomial $g(\mathbf{y})$ has a formula of size at most $O(s^2n)$.*

Proof. We will primarily use Lemma 27 to prove the above theorem. Note that the set $\{e_1 - a_1, \dots, e_n - a_n\}$ continue to be algebraically independent as the Jacobian matrix $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n) = \mathcal{J}(e_1, \dots, e_n)$. Recall that $(e_1 \dots e_n)$ are algebraically independent over n variables and thus has a full rank Jacobian matrix, which also implies the algebraic independence of $(e_1 - a_1, \dots, e_n - a_n)$ using Theorem 12. The next step would be to find a point \mathbf{c} where $e_i(\mathbf{c}) - a_i$ is zero for all i . We also need to make sure that at point \mathbf{c} the $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)$ matrix has full rank. It is easy to verify that the determinant of $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)$ is a polynomial of degree $\binom{n}{2}$. As per our assumption, the field we use is quite large, in fact much larger than n^2 . Thus, from the Polynomial Identity lemma, we know that over every large enough set $S \subseteq \mathbb{C}$, there exists a point $\mathbf{c} \in S^n$ at which the determinant of $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)$ is non-zero and hence, the matrix $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)(\mathbf{c})$ is full rank. By setting $a_i = e_i(\mathbf{c})$ for all i , $\{e_1 - a_1, \dots, e_n - a_n\}$ satisfy the Property S mentioned in Definition 26 for point \mathbf{c} . The proof of this theorem is now an immediate corollary of Lemma 27. ◀

The proof above gives a slightly stronger statement than what is stated in Theorem 38. Moreover, the statement of Theorem 38 holds for many such a_1, a_2, \dots, a_n . To see this, note that the only property that the proof above uses about a_1, a_2, \dots, a_n is that $\mathbf{a} = (a_1, a_2, \dots, a_n)$ is a point in the image of the polynomial map σ from \mathbb{C}^n to \mathbb{C}^n which is given by mapping $\mathbf{y} \in \mathbb{C}^n$ to $(e_1(\mathbf{y}), e_2(\mathbf{y}), \dots, e_n(\mathbf{y}))$ such that there is a point \mathbf{c} in the pre-image of \mathbf{a} where the Jacobian $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)(\mathbf{b})$ (which is equal to $\mathcal{J}(e_1, \dots, e_n)(\mathbf{b})$) is full rank. Now, observe that the map σ is invertible. To see this, note that σ can be thought of as mapping n roots b_1, b_2, \dots, b_n of the univariate polynomial $\prod_{i=1}^n (z - b_i)$ to its coefficients, and hence its inverse is the map which maps the n coefficients of a monic degree n polynomial to its roots.

Thus, if we take \mathbf{b} to be a random point from a large enough grid, then the Jacobian $\mathcal{J}(e_1 - a_1, \dots, e_n - a_n)$ has rank n with high probability. Moreover, whenever this event happens, Theorem 38 holds with \mathbf{a} being set to be the image of \mathbf{b} under σ .

Generalizing the results in [1] to other bases

We know that the fundamental theorem of symmetric polynomials also holds for other bases and not just for elementary symmetric basis. For any given n -variate symmetric polynomial f_{sym} , Bläser-Jindal efficiently finds f such that $f_{sym} = f(e_1, e_2 \dots e_n)$. The degree of f is also given apriori. We generalize the Bläser-Jindal work to other bases such as homogeneous symmetric base and power symmetric base efficiently. In order to prove that, we need to show there exists an efficient transformation which can represent any elementary symmetric polynomial in the form of homogeneous symmetric or power symmetric polynomial. The following lemma illustrates that.

► **Lemma 39.** *Any n -variate elementary symmetric polynomial of degree k can be written as an algebraic combination of homogeneous symmetric polynomials (or power symmetric polynomials) using a small formula.*

Proof. It is well known that these transformations are doable using ABP of polynomial size as the transformation uses small determinants [28]. We prove the same for formula. Recall that $E(t)H(-t) = 1$

$$E(t) = \frac{1}{H(-t)} = \frac{1}{1 - h_1 t + h_2 t^2 \dots + (-1)^n h_n t^n} = \frac{1}{1 - z} = \sum_{i \geq 0} z^i$$

where $z = h_1 t - h_2 t^2 \dots + (-1)^{n-1} h_n t^n$.

Consider the truncated polynomial $A(t) = E(t) \bmod \langle z \rangle^{k+1}$, where $\langle z \rangle$ is the ideal generated by z . Now we use interpolation to find the coefficient of t^k in the polynomial $A(t)$, which is precisely e_k . $A(t)$ can have degree at most nk , which implies a formula size of $O(nk)$ for $A(t)$ (assuming the field to be algebraically closed). But, $k \leq n$, hence the trivial formula complexity for expressing e_k in the form of h_i 's is $O(n^4)$ using Lemma 8.

From the definition of the generating functions $P(t)$ and $E(t)$, it is easy to verify that

$$E(t) = e^{\int P(-t) dt} = e^{\int (\sum_{m \geq 1} p_m t^{m-1}) dt} = e^{(\sum_{m \geq 1} \frac{p_m t^m}{m!})} = 1 + q + \frac{q^2}{2!} \dots,$$

where $q = \sum_{m \geq 1} \frac{p_m t^m}{m!}$.

Now we consider the truncated polynomial containing degree up to q^k and interpolate this polynomial to get the coefficient of t^k . The formula complexity for expressing e_k as power symmetric polynomials is $O(n^4)$. Also, the proof outline is very similar to the homogeneous symmetric case. ◀

► **Theorem 40** (Bläser-Jindal for other bases). *For any n -variate symmetric polynomial $f_{sym} \in \mathbb{C}[\mathbf{x}]$, we can efficiently compute the polynomials $f_E, f_H, f_P \in \mathbb{C}[\mathbf{x}]$ that are n -variate and unique s.t $f_{sym} = f_E(e_1, e_2 \dots e_n) = f_H(h_1, h_2 \dots h_n) = f_P(p_1, p_2 \dots p_n)$.*

Proof. Bläser-Jindal proves that f_E can be efficiently computed using an algebraic circuit. They prove that the circuit size for computing f_E is bounded by $O(d^2 S(f_{sym}) + \text{poly}(n, d))$ where d is the degree of f_E , $S(f_{sym})$ denotes the circuit size of f_{sym} and n is the number of working variables. We extend their work for computing f_H and f_P . To prove that, we shall use Bläser-Jindal method as a black box. We use Bläser-Jindal technique and get the circuit for f_E . We denote this circuit by \mathcal{C}_{f_E} . \mathcal{C}_{f_E} can be visualized as a circuit having elementary symmetric polynomials(e_i 's) as its input. But, from Lemma 39, we know e_i 's can be uniquely expressed in the form of h_i 's and p_i 's using a formula of polynomial size. We

denote the modified circuit as \mathcal{C}_{f_H} after transforming e_i 's to h_i 's in \mathcal{C}_{f_E} . The output of \mathcal{C}_{f_H} is f_H because h_i 's are algebraically independent and also satisfy fundamental theorem of symmetric polynomials. The proof for f_P is similar. Thus f_H and f_P can still be computed using a circuit of size $O(d^2 S(f_{sym}) + \text{poly}(n, d))$ as the transformation just adds some poly factor to the previously calculated size. \blacktriangleleft

Bläser-Jindal method works for the circuit and we do not know whether it works for ABPs or formula. If it works for ABP, then the basis transformation is trivial using a small determinant (i.e a small ABP). If it works for formula, then Theorem 40 would be useful to extend the notion for other bases while still staying in the formula regime.

4.4 Partial derivatives of a product of algebraically independent polynomials

We digress a little in this section to discuss another application of the ideas used in the proof of Lemma 27 to a question of Amir Shpilka on the partial derivative complexity of a product of algebraically independent polynomials. We start with the definition of the partial derivative complexity.

► **Definition 41.** *The partial derivative complexity of a polynomial $P \in \mathbb{C}[\mathbf{x}]$ is the dimension of the linear space of polynomials over \mathbb{C} spanned by all the partial derivatives of P .*

The following question was asked by Amir Shpilka and our techniques provide a partial answer. As far as we are aware, the general question remains open.

► **Question 42 (Shpilka).** *Let $g_1, g_2, \dots, g_k \in \mathbb{C}[\mathbf{x}]$ be algebraically independent polynomials. Then, prove (or disprove) that the partial derivative complexity of the the product $\prod_{i=1}^k g_i(\mathbf{x})$ is at least $\exp(\Omega(k))$.*

A canonical example of polynomials satisfying the hypothesis is when $g_i(\mathbf{x}) = x_i$. Thus, the product polynomial $\prod_{i=1}^k g_i(\mathbf{x})$ is equal to the monomial $x_1 \cdot x_2 \cdots x_k$, and indeed the partial derivative complexity of this monomial is at least 2^k , since for every $S \subseteq [k]$, the monomial $\prod_{i \in S} x_i$ is a partial derivative and these monomials are all linearly independent over \mathbb{C} . Thus, in general, we cannot hope for a better lower bound on the partial derivative complexity of such polynomials. Using our techniques, we observe the following two statements which answer special cases of this question.

► **Theorem 43.** *Let $g_1, g_2, \dots, g_k \in \mathbb{C}[\mathbf{x}]$ be algebraically independent polynomials which satisfy Property S. Then, the partial derivative complexity of $\prod_{i=1}^k g_i(\mathbf{x})$ is at least 2^k .*

► **Theorem 44.** *Let $g_1, g_2, \dots, g_k \in \mathbb{C}[\mathbf{x}]$ be algebraically independent polynomials. Then, there are field constants a_1, a_2, \dots, a_n such that the partial derivative complexity of $\prod_{i=1}^k (g_i(\mathbf{x}) + a_i)$ is at least 2^k .*

In fact, the lower bound holds for almost all choices of a_1, a_2, \dots, a_k .

The following observation essentially follows from the definition of Property S (Definition 26) and Theorem 13. The proof is also implicit in the proof of Lemma 27.

► **Observation 45.** *Let $q_1(\mathbf{x}), q_2(\mathbf{x}), \dots, q_k(\mathbf{x}) \in \mathbb{C}[\mathbf{x}]$ be algebraically independent polynomials which satisfy Property S. Then, there is an $\mathbf{a} \in \mathbb{C}^n$ such that the degree zero homogeneous component of the the polynomials $q_1(\mathbf{x} + \mathbf{a}), q_2(\mathbf{x} + \mathbf{a}), \dots, q_k(\mathbf{x} + \mathbf{a})$ are all zero, and their homogeneous components of degree one are all linearly independent.*

Theorem 43 and Theorem 44 are essentially immediate consequences of Observation 45 and some standard properties of partial derivatives, which we now discuss.

► **Lemma 46.** *Let $P \in \mathbb{C}[\mathbf{x}]$ be a polynomial. Then,*

1. *For every $\mathbf{a} \in \mathbb{C}^n$, the partial derivative complexity of P is equal to the partial derivative complexity of $P(\mathbf{x} + \mathbf{a})$. More generally, the partial derivative complexity is invariant under invertible linear transformation of variables.*
2. *Let i be the degree of the lowest degree homogeneous component of P which is non-zero and let P_i denote this homogeneous component. Then, the partial derivative complexity of P is at least as large as the partial derivative complexity of P_i .*
3. *The partial derivative complexity of a product of k linearly independent homogeneous linear forms is at least 2^k .*

We briefly sketch the main ideas in the proof.

Proof Sketch. For the first item, we prove the statement for first order partial derivatives. The argument easily extends to higher order derivatives as well. By the chain rule, $\frac{\partial P(\mathbf{x} + \mathbf{a})}{\partial x_i}$ equals $\frac{\partial P(\mathbf{x} + \mathbf{a})}{\partial(x_i + a_i)} \cdot \frac{\partial(x_i + a_i)}{\partial x_i}$. Observe that this is equal to the polynomial obtained by taking the partial derivative $\frac{\partial P(\mathbf{x})}{\partial x_i}$ of the original polynomial and then shifting the variables by \mathbf{a} , i.e. replacing x_j by $x_j + a_j$ for every j . Thus, the linear space spanned by the first order partial derivatives of $P(\mathbf{x} + \mathbf{a})$ is equal to the linear space obtained by taking the space of first order partial derivatives of $P(\mathbf{x})$ and shifting the variables by \mathbf{a} . It is not hard to see that this preserves the dimension of the space. The proof of the *moreover* part needs a bit more care, but follows similarly.

For the second item, observe that for any set of polynomials $\{Q_1, Q_2, \dots, Q_t\}$, the dimension of the linear span of $\{Q_1, Q_2, \dots, Q_t\}$ is at least as large as the dimension of the linear span of the lowest degree non-zero homogeneous components of Q_1, Q_2, \dots, Q_t . Also, observe that for any monomial $\mathbf{x}^{\mathbf{b}}$, if the partial derivative $\frac{\partial P_i}{\partial \mathbf{x}^{\mathbf{b}}}$ is non-zero, then the lowest degree non-zero homogeneous component of $\frac{\partial P}{\partial \mathbf{x}^{\mathbf{b}}}$ equals $\frac{\partial P_i}{\partial \mathbf{x}^{\mathbf{b}}}$. Now, let S be the set of monomials such that the space of partial derivatives of P_i with respect to monomials in S is a basis for the linear space of all partial derivatives of P_i . From the two earlier observations in this paragraph, it follows that the derivatives of P with respect to monomials in the set S are all linearly independent, thereby implying the desired lower bound.

The third item is an immediate consequence of the observation that the partial derivative complexity of the monomial $\prod_{i=1}^k x_i$ is equal to 2^k and the “moreover” part of the first item of this lemma, which says that partial derivative complexity is invariant under invertible linear transformations. ◀

We now sketch the main ideas in the proof of Theorem 43.

Proof of Theorem 43. The goal is to prove a lower bound on the partial derivative complexity of the polynomial $\prod_{i=1}^k q_i(\mathbf{x})$. Let $\mathbf{a} \in \mathbb{C}^n$ be the point guaranteed by Observation 45. Thus, $q_1(\mathbf{x} + \mathbf{a}), q_2(\mathbf{x} + \mathbf{a}), \dots, q_k(\mathbf{x} + \mathbf{a})$ are all zero, and their homogeneous components of degree one are all linearly independent. From the first item of Lemma 46, we know that it suffices to prove a lower bound on the partial derivative complexity of $\prod_{i=1}^k q_i(\mathbf{x} + \mathbf{a})$.

The claim now is that the lowest degree homogeneous component of $\prod_{i=1}^k q_i(\mathbf{x} + \mathbf{a})$ of which is non-zero is the homogeneous component of degree equal to k , and this is equal to the product of the homogeneous components of degree one of $q_1(\mathbf{x} + \mathbf{a}), q_2(\mathbf{x} + \mathbf{a}), \dots, q_k(\mathbf{x} + \mathbf{a})$. This immediately follows from Claim 28 in the proof of Lemma 27. But once we have this claim, the theorem follows from the third item of Lemma 46. We skip rest of the details. ◀

Theorem 44 follows from observing that we can pick a_1, a_2, \dots, a_n so that $q_1 + a_1, q_2 + a_2, \dots, q_k + a_k$ satisfy Property S. This follows from a similar observation in the proof of Theorem 38. Once we have this observation, we are back in the setting of Theorem 43.

Before we conclude this section, we note that in order to generalize Theorem 43 and Theorem 44 to completely answer Question 42 in affirmative, it would suffice to prove the following conjecture.

► **Conjecture 47.** *For all constants $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{C}$ and linearly independent homogeneous linear forms $\ell_1, \ell_2, \dots, \ell_k$ the following is true : if q_1, q_2, \dots, q_k are polynomials such that their minimum degree non-zero monomial has degree at least two, then the partial derivative complexity of the polynomial $\prod_{i=1}^k (\alpha_i + \ell_i + q_i)$ is at least 2^k .*

If the conjecture is false, a counterexample to the conjecture may be instructive towards understanding how the partial derivative complexity behaves over taking a product of polynomials.

5 Open problems

We conclude with some open problems.

- (i) Perhaps the most natural question would be to characterize the formula complexity of all Schur polynomials. As discussed in Remark 35, there exist partitions λ for which the corresponding s_λ have small (polynomial sized) algebraic formulas. On the other hand, as shown in this work, there exist families of λ s which do not have polynomial sized formulas unless the determinant does. Due to classical results such as [40], we know that the latter class of λ s in fact have formulas of size $n^{O(\log n)}$. It would be of great interest to extend these two results and get a complete characterization of the formula complexity of s_λ , as a function of the partition λ .
- (ii) Bläser and Jindal gave a computationally efficient version of the fundamental theorem for symmetric polynomials. In particular, they showed that if f_{sym} is a symmetric polynomial computed by a polynomial-sized algebraic circuit then the unique polynomial f_E such that $f_{sym} = f_E(e_1, \dots, e_n)$, also has a polynomial-sized algebraic circuit. A natural question one can ask is: if f_{sym} has a polynomial-sized algebraic formula (or ABP) then does f_E also have a polynomial-sized algebraic formula (ABP resp.)? In Theorem 38 we take a step towards proving this statement. We show that there exists $\mathbf{a} \in \mathbb{C}^n$ such that if f_{sym} can be expressed as $f_E(e_1 - a_1, \dots, e_n - a_n)$ for a homogeneous f_E then f_E has a small algebraic formula if f_{sym} does. To get the exact Bläser-Jindal-like statement in the formula setting, we would have to improve our result in two ways. We would have to prove it for general f_E rather than for homogeneous f_E and we would have to prove it for $\mathbf{a} = 0^n$. We believe that both of these are interesting directions to pursue.
- (iii) Another interesting extension of the results here would be to show that there are families of Generalized Vandermonde matrices such that circuit complexity of computing their permanent is essentially as large as the circuit complexity of the Permanent. This would be a VNP analogue of Theorem 3.
- (iv) Yet another interesting direction would be to extend Theorem 43 to answer Question 42 completely.

References

- 1 Markus Bläser and Gorav Jindal. On the Complexity of Symmetric Polynomials. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ITCS.2019.47.

- 2 Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Algorithms and Computation in Mathematics. Springer, 2000. URL: <https://www.springer.com/gp/book/9783540667520>.
- 3 Cy Chan, Vesselin Drensky, Alan Edelman, Raymond Kan, and Plamen Koev. On computing schur functions and series thereof. *Journal of Algebraic Combinatorics*, 50(2):127–141, September 2019. doi:10.1007/s10801-018-0846-y.
- 4 Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity. *Foundations and Trends in Theoretical Computer Science*, 2011.
- 5 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Some closure results for polynomial factorization and applications. *CoRR*, abs/1803.05933, 2018. arXiv:1803.05933.
- 6 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure of VP under taking factors: a short and simple proof. *CoRR*, abs/1903.02366, 2019. arXiv:1903.02366.
- 7 James Demmel and Plamen Koev. Accurate and efficient evaluation of schur and jack functions. *Mathematics of Computation*, 75(253):223–239, 2006. URL: <http://www.jstor.org/stable/4100151>.
- 8 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *STOC*, pages 1152–1165, 2018. doi:10.1145/3188745.3188760.
- 9 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2010. Preliminary version in STOC’08. doi:10.1137/080735850.
- 10 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2014.
- 11 Sergey Fomin, Dima Grigoriev, and Gleb Koshevoy. Subtraction-free complexity, cluster transformations, and spanning trees. *Found. Comput. Math.*, 16(1):1–31, February 2016. doi:10.1007/s10208-014-9231-y.
- 12 Sergey Fomin, Dima Grigoriev, Dorian Nogneng, and Éric Schost. On semiring complexity of schur polynomials. *Comput. Complex.*, 27(4):595–616, December 2018. doi:10.1007/s00037-018-0169-3.
- 13 Hervé Fournier, Nutan Limaye, Meena Mahajan, and Srikanth Srinivasan. The shifted partial derivative complexity of elementary symmetric polynomials. *Theory of Computing*, 13(1):1–34, 2017. doi:10.4086/toc.2017.v013a009.
- 14 Vesselin Gasharov. Incomparability graphs of $(3+1)$ -free posets are s -positive. *Discrete Mathematics*, 157(1-3):193–197, 1996.
- 15 Dima Grigoriev and Gleb Koshevoy. Complexity of tropical schur polynomials. *Journal of Symbolic Computation*, 74:46–54, 2016. doi:10.1016/j.jsc.2015.05.005.
- 16 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Satharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. doi:10.1137/140957123.
- 17 Sean Hallgren, Alexander Russell, and Amnon Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 627–635. ACM, 2000.
- 18 E. R. Heineman. Generalized vandermonde determinants. *Transactions of the American Mathematical Society*, 31(3):464–476, 1929. URL: <http://www.jstor.org/stable/1989528>.
- 19 Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011. doi:10.1007/s00037-011-0007-3.
- 20 Christian Ikenmeyer and Stefan Mengel. On the relative power of reduction notions in arithmetic circuit complexity. *Information Processing Letters*, 130:7–10, 2018.
- 21 Christian Ikenmeyer and Greta Panova. Rectangular kronecker coefficients and plethysms in geometric complexity theory. *Advances in Mathematics*, 319:40–66, 2017.
- 22 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*. doi:10.1007/s00037-004-0182-6.

- 23 Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.
- 24 Plamen Koev. Accurate computations with totally nonnegative matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(3):731–751, 2007. doi:10.1137/04061903X.
- 25 J. M. Landsberg and Zach Teitler. On the ranks and border ranks of symmetric tensors. *Foundations of Computational Mathematics*, 10(3):339–366, June 2010. doi:10.1007/s10208-009-9055-3.
- 26 Veerle Ledoux and Simon JA Malham. *Introductory schubert calculus*, 2010.
- 27 Dick Lipton and Ken Regan. Arithmetic complexity and symmetry, 2009. URL: <https://rjlipton.wordpress.com/2009/07/10/arithmetic-complexity-and-symmetry/>.
- 28 Ian G. Macdonald. *Symmetric functions and Hall polynomials*. Oxford University Press, 1979.
- 29 Hariharan Narayanan. On the complexity of computing kostka numbers and littlewood-richardson coefficients. *Journal of Algebraic Combinatorics*, 24(3):347–354, 2006.
- 30 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. Available on citeseer:10.1.1.90.2644. doi:10.1007/BF01294256.
- 31 Ryan O’Donnell and John Wright. Quantum spectrum testing, 2015. arXiv:1501.05028.
- 32 Luke Oeding. Border ranks of monomials, 2016. arXiv:1608.02530.
- 33 Bruce Sagan. *The Symmetric Group: Representations, Combinatorial Algorithms, and Symmetric Functions*. Springer, 2001.
- 34 Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 60–71, 2008. doi:10.1007/978-3-540-70575-8_6.
- 35 Amir Shpilka. Affine projections of symmetric polynomials. In *In Proc. 16th Annual IEEE Conference on Computational Complexity*, pages 160–171, 2001.
- 36 Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001. Preliminary version in the *14th Annual IEEE Conference on Computational Complexity (CCC 1999)*. doi:10.1007/PL00001609.
- 37 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. doi:10.1561/04000000039.
- 38 Richard P Stanley. On the number of reduced decompositions of elements of coxeter groups. *European Journal of Combinatorics*, 5(4):359–372, 1984.
- 39 Richard P Stanley. Enumerative combinatorics. vol. 2, volume 62 of. *Cambridge Studies in Advanced Mathematics*, 1999.
- 40 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in the *6th International Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*. doi:10.1137/0212043.
- 41 Herbert S. Wilf. *Generatingfunctionology*. A. K. Peters, Ltd., third edition, 2006.