Deterministic Constrained Multilinear Detection

Cornelius Brand
Algorithms and Complexity Group, TU Wien, Austria

Viktoriia Korchemna
Algorithms and Complexity Group, TU Wien, Austria

Michael Skotnica
Department of Applied Mathematics, Charles University, Prague, Czech Republic

Abstract

We extend the algebraic techniques of Brand and Pratt (ICALP'21) for deterministic detection of k-multilinear monomials in a given polynomial with non-negative coefficients to the more general situation of detecting *colored* k-multilinear monomials that satisfy additional constraints on the multiplicities of the colors appearing in them. Our techniques can be viewed as a characteristic-zero generalization of the algebraic tools developed by Guillemot and Sikora (MFCS'10) and Björklund, Kaski and Kowalik (STACS'13)

As applications, we recover the state-of-the-art deterministic algorithms for the Graph Motif problem due to Pinter, Schachnai and Zehavi (MFCS'14), and give new deterministic algorithms for generalizations of certain questions on colored directed spanning trees or bipartite planar matchings running in deterministic time $O^*(4^k)$, studied originally by Gutin, Reidl, Wahlström and Zehavi (J. Comp. Sys. Sci. 95, '18). Finally, we give improved randomized algorithms for intersecting three and four matroids of rank k in characteristic zero, improving the record bounds of Brand and Pratt (ICALP'21) from $O^*(64^k)$ and $O^*(256^k)$, respectively, to $O^*(4^k)$.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Fixed-parameter algorithms, Algebraic algorithms, Motif discovery, Matroid intersection

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.25

Funding *Cornelius Brand*: Austrian Science Fund (FWF, project Y1329).
Viktoriia Korchemna: Austrian Science Fund (FWF, project Y1329).
Michael Skotnica: "Grant Schemes at CU" (reg. no. CZ.02.2.69/0.0/0.0/19_073/0016935), GAČR grant 22-19073S.

1 Introduction

The area of fixed-parameter algorithms, sprung from the seminal work of Downey and Fellows (see e.g. their monograph [11]), has produced an enormous amount of tools and techniques to facilitate the design of algorithms that can solve NP-hard problems in running times of the form $f(k) \cdot \text{poly}(n)$, where n is the input size and k is some parameter that can be interpreted to quantify the difficulty of the instance at hand.

Two prominent and highly successful techniques contributing to this toolbox are, on the one hand, of *algebraic* nature, focusing on formulations of combinatorial problems in the language of polynomials, and then employing mathematical means to solve these reformulations. On the other hand, one of the earliest approaches known to produce fixed-parameter algorithms is the combinatorial method of *representative families*, with their first algorithmic applications dating back at least to work of Monien [25].

Both the algebraic as well as the representative-family based approach have received a considerable amount of attention in recent years, often focusing on the same application problems. This has resulted in a flurry of competing results on variants of e.g., the notorious longest path problem, subgraph isomorphism, set packing, network design as well as matroid problems, to name a few.

An important variation on the basic combinatorial problems studied in these lines of research concerns *colored* variants of the problems. One of the most well-studied problems of this kind is the *graph motif* problem, which was originally motivated from the analysis of biological networks, and has since been the subject of many parameterized algorithmic studies. As with the uncolored variants of these problems, the colored counterparts attracted attention from researchers from both the perspective of representative families, as well as the algebraic point of view, with both techniques contributing methods that have remained the state-of-the-art within their respective regime. A possibly sweeping generalization of the results that together form this body of work might perhaps conclude that algebraic methods seem more adapted to produce fast randomized algorithms, whereas the representative families tend to yield record bounds for deterministic algorithms.

On the side of algebraic algorithms, in the uncolored regime, a common technique for all of the combinatorial problems above is to formulate them as so-called *multilinear monomial detection* problems. In these problems, one is given an arithmetic circuit computing a polynomial, and the task is to decide whether this polynomial contains a product of k variables that are all pairwise distinct, and k is the parameter. Similarly, in the colored variants, these detection problems are generalized to so-called *constrained* multilinear monomial detection, where additional coloring constraints are imposed on the products of variables to be detected. This is also the technique that is considered in the present article.

1.1 Related Work

For general background on fixed-parameter algorithms, we refer the reader to the snapshot of the state-of-the-art of the field as captured by e.g. the textbook of Cygan et al. [9], in particular Chapters 10 and 12, where diverse applications of the tools mentioned above are developed.

Algebraic Algorithms

One of the seminal works for algebraic methods in parameterized and exact algorithms is the work by Björklund et al. [2] on fast subset convolutions and its application for the parameterized Steiner tree problem. More specifically, for the problems considered in this article, the algorithms by Koutis and Williams on multilinear detection were highly influential [20, 22, 29]. These methods were first transported to the setting of constrained multilinear detection by Guillemot and Sikora [17] and subsequently improved by Koutis [21] as well as Björklund, Kowalik and Kaski [4, 5]. It is important to note that all their methods are inherently randomized, because they resort to a use of the DeMillo-Lipton-Schwartz-Zippel Lemma, which makes it seem hard to derandomize them in a black-box manner.

Graph Problems

These randomized algebraic methods allow to design the state-of-the-art algorithms for e.g. the maximum graph motif problem, running time $2^k \cdot \mathrm{poly}(n)$ [4, 5]. On the side of deterministic algorithms, which is the focus of this article, the relevant techniques are more combinatorial in nature. Indeed, the method of representative families was first considered

explicitly in the context of fixed-parameter algorithms by Marx [24] and has since been extended into a most intricate machinery, in particular in the works of Fomin et al. [15, 16]. These methods were then refined by Pinter, Shachnai and Zehavi and applied to design the state-of-the-art deterministic algorithms for the graph motif problem [26], running in time $2^{\omega k} \cdot \text{poly}(n)$.

Matroid Problems

Another area in which representative-families based methods have proven fruitful is the realm of *matroid problems*. In this article, we consider the problem of *matroid intersection:* Given q (representations of) matroids of rank k, decide whether they share a single common basis. This is a classic problem in combinatorial optimization, and the polynomial-time solvable special case of intersecting $q = 2$ matroids is famously treated by Edmonds [13]. More generally, intersecting $q > 2$ matroids becomes NP-hard, and fixed-parameter algorithms for this problem were given first (without using this term) by Barvinok [1]. In a later development, Marx [24] revived the interest in this problem by giving the first single-exponential (in q and k) algorithm using representative families, which was superseded by the work of Fomin et al. [15]. The current state-of-the-art is 4^{qk} [8]. A recent manuscript of Eiben, Koana and Wahlström [14] shows that this can be improved to $4^{(q-2)k}$ using different algebraic methods.

Our Contribution

The contribution of this paper is three-fold. First, we show how to extend the (deterministic) algebraic machinery of Brand and Pratt to the colored, that is, constrained-multilinear setting. This is noteworthy insofar as until now, the only known algebraic tools for this task were inherently randomized, and the only deterministic algorithms for the respective combinatorial application problems were of decidedly combinatorial nature.

Secondly, we show how to use these adapted methods to, on the one hand, reproduce the deterministic state-of-the-art for the graph motif problem without any additional problem-specific adaptation as a generic application of the algebraic methodology laid out here. In addition, we provide examples of natural colorful extensions of several combinatorial problems, involving e.g. spanning trees and planar perfect matchings, that are not known to admit deterministic algorithms by using only the known combinatorial techniques.

Finally, we improve the state-of-the-art for matroid intersection in the case $q \leq 4$ by giving specialized polynomial formulations of these cases. It is worth noting that the speedup over the generic state-of-the-art technique is by a factor of 16 (or 4 with respect to [14]) in the exponential base.

Organization

We continue with a formal introduction of notation and all problems considered in the article. We then prove our main theorem about constrained multilinear detection and give our applications for graph problems. We then conclude with the improved algorithms for matroid problems.

2 Preliminaries

Generally, we denote the set of integers $\{1, \ldots, t\}$ by $[t]$, and use $[t]_0$ as a shorthand for $[t] \cup \{0\}$. We denote by $\mathbb{N}$ the set of natural numbers excluding zero, and by $\mathbb{Q}$ the set of rational numbers.

Matrices and Matroids

Let A be a matrix with row set X and column set Y, and let X_0 and Y_0 be subsets of X and Y correspondingly. We denote by $A[X_0, Y_0]$ the submatrix of A restricted to the rows in X_0 and the columns in Y_0 (while in $A[Y_0]$ the restriction is applied to columns alone). In particular, $A[X, Y] = A[Y] = A$. For two matrices A and B, we denote their direct sum by $A \oplus B$. For an arbitrary sequence $r_1, \ldots, r_N \in \mathbb{Q}$, we write $\mathrm{Vand}_k(r_1, \ldots, r_N)$ for the $k \times N$ *Vandermonde* matrix defined through

$$\mathrm{Vand}_k(r_1, \ldots, r_N)_{l,j} = r_j^l,\ l \in [k-1]_0,\ j \in [N].$$

By convention, we let $0^0 = 1$ in this definition, and we say $\mathrm{Vand}_k(r_1, \ldots, r_N)$ is the Vandermonde matrix of the sequence $r_1, \ldots, r_N$.

A *finite matroid* $\mathcal{M}$ is a pair $(E, \mathcal{I})$, where E is a finite set (called the *ground set*) and $\mathcal{I}$ is a family of subsets of E (called the *independent sets*) satisfying so-called independence axioms. In this article we only work with finite matroids that can be represented by matrices as follows. Any matrix M with entries in $\mathbb{Q}$ gives rise to a matroid $\mathcal{M}$ with the ground set being its set of columns. The independent sets of the matroid are those subsets of columns that are linearly independent as vectors. In particular, size of any *base* (i.e., maximal with respect to inclusion independent set) of $\mathcal{M}$ is equal to the rank of M. We say that such a matroid $\mathcal{M}$ is *represented* by M.

Polynomials

Let R be a commutative ring, and let $x_1, \ldots, x_n$ be formal indeterminates. Then $R[x_1, \ldots, x_n]$ is the ring of polynomials in $x_1, \ldots, x_n$ with coefficients in R, and we call the latter the *coefficient ring* of the polynomial ring.

Every polynomial f can be represented uniquely as a weighted, finite sum of monomials, that is, products of variables. We may therefore write

$$f = \sum_{a_1, \ldots, a_n \in \mathbb{N}} c_{a_1, \ldots, a_n} x_1^{a_1} \cdots x_n^{a_n},$$

where only finitely many of the coefficients $c_{a_1, \ldots, a_n} \in R$ are non-zero. When all monomials appearing in $f_{\mathcal{M}}$ are of degree k, we call f itself *homogeneous of degree k*. Furthermore, if for some choice of $a_1, \ldots, a_n \leq 1$ there is a coefficient $c_{a_1, \ldots, a_n} \neq 0$, we say that f has a *multilinear* monomial $x_1^{a_1} \cdots x_n^{a_n}$.

Our algorithms are based on the algebraic techniques found in [8]. In this approach, as with similar algebraic methods [20], the combinatorial objects (matroid bases, subgraphs, subsets, and so on) over a universe of size n are modeled using (rational) multivariate polynomials over the indeterminates $x_1, \ldots, x_n$. The crux of the approach in [8] is then to write down a polynomial whose coefficients encode some information about the combinatorial problem at hand, and evaluate algebraically some linear functional over this polynomial that reveals some sought-after combinatorial answers.

This functional is defined via the following inner product: Let f and g be any two such polynomials that are homogeneous of degree k, and say that f is as above and

$$g = \sum_{a_1, \ldots, a_n \in \mathbb{N}} d_{a_1, \ldots, a_n} x_1^{a_1} \cdots x_n^{a_n}.$$

Then we define their *apolar inner product* $\langle f, g \rangle$ in an almost entirely straightforward way, by

$$\langle f, g \rangle = \sum_{a_1,\ldots,a_n \in \mathbb{N}} a_1! \cdot a_2! \cdots a_n! \cdot c_{a_1,\ldots,a_n} \cdot d_{a_1,\ldots,a_n}. \tag{1}$$

Including the product of factorials serves the purpose of normalization, which allows to connect $\langle f, g \rangle$ with the partial derivatives of f and g.

As an inner product, this mapping can be intuitively interpreted as a measure of similarity between f and g. In particular, when fixing g (or, equivalently, f) to, for example, the k-th elementary symmetric polynomial

$$e_k(x_1, \ldots, x_n) = \sum_{\substack{S \subseteq [n] \\ |S|=k}} \prod_{s \in S} x_s,$$

which has all, and only, multilinear monomials with coefficient one, it is easy to see that $f \mapsto \langle f, e_k \rangle$ is a linear functional that yields the sum of coefficients of multilinear monomials of degree k of f. If f maps to a non-zero value under this functional, we may conclude that f has a multilinear monomial. As shown in [8], the complexity of evaluating this functional depends on certain algebraic properties of f and e_k (or g, in general).

Arithmetic Circuits

Even before these algebraic properties, the complexity of this evaluation depends on the *encoding* of f and g. Indeed, if f and g are given by their list of coefficients, then all questions treated in this article become trivial. However, this would imply inputs that are of exponential size in n, making this *sparse* encoding a poor choice for polynomials enumerating e.g. combinatorial objects, as an algebraic analog to brute-force search. Instead, polynomials are encoded using *arithmetic circuits*, which are directed acyclic graphs with a single sink, labeled as follows: Every vertex of in-degree zero (*inputs*) is labeled with either an indeterminate or a constant from the coefficient ring. Every vertex with non-zero in-degree is labeled either $+$ or $\times$. The labeled nodes of the arithmetic circuit are referred to as *gates*. An arithmetic circuit *computes* a polynomial in the obvious inductive manner. Finally, we call an arithmetic circuit *skew* if every $\times$-gate has at most one edge coming from a non-input gate.

Constrained Monomials

In the context of this article, mutlilinear monomials that are not only multilinear, but satisfy additional constraints are relevant. More precisely, suppose $C = \{1, \ldots, q\}$ is a set of q colors with *multiplicities* $\mu_1, \ldots, \mu_q \in \mathbb{N}$, and $\chi : [n] \to C$ is a *coloring* of $[n]$. A multilinear monomial $x_1^{a_1} \cdots x_n^{a_n}$ is called *well-colored* if $\sum_{i:\chi(i)=c} a_i \leq \mu_c$ for all colors $c \in C$, that is, every color appears at most μ_c times in the monomial.

Problem Statements

Let us now formally introduce the problems studied in this article. The most prominent one, which will be used to reduce the combinatorial application problems to, is the following algebraic problem.

Constrained k-Multilinear Detection	
Input:	A number k, an arithmetic circuit computing a polynomial f, a coloring of the variables of f, together with multiplicity constraints on each color.
Question:	Does f have a well-colored multilinear monomial of degree k?

In our algorithms, we will encounter the restriction that the computed polynomial f (but not necessarily the circuit itself) have non-negative coefficients in order to make them deterministic. This is a natural restriction when dealing with combinatorial problems in many cases.

Graph Problems

In analogy to the definition of well-colored monomials, the notion of a vertex or edge coloring includes those mappings that are not necessarily *proper* colorings, that is, two neighboring vertices in a graph, or two edges sharing a vertex, may very well receive the same color. To be precise, given a coloring $\chi : V(G) \to \{1, \ldots, q\}$, or analogously $\chi : E(G) \to \{1, \ldots, q\}$, and multiplicites $\mu_1, \ldots, \mu_q$, we call a set S of vertices (or edges, respectively) *well-colored* if $\chi^{-1}(i) \cap S \leq \mu_i$ for all $i = 1, \ldots, q$. With this in mind, the following problems can be defined:

Maximum Graph Motif	
Input:	A vertex-colored undirected graph G together with multiplicity constraints on each color.
Question:	Does G have a well-colored set of k vertices that induce a connected subgraph of G?

In algebraic terms, Maximum Graph Motif, while amenable to our techniques, doesn't showcase their full strength, for reasons explained in the article. In contrast, the following problems share the important property that they can be expressed succinctly by computations of certain determinants, which allows us to give the first fixed-parameter algorithms for them that are probably hard to come by using other approaches. These are made in analogy to the non-well-colored graph problems studied by Gutin et al. [18].

Well-Colored Spanning Tree	
Input:	A number k and an edge-colored directed graph G together with multiplicity constraints on each color.
Question:	Does G have a well-colored subset of k edges that can be extended to a directed spanning tree of G?

Well-Colored Planar Perfect Matching	
Input:	A number k and an edge-colored planar graph G together with multiplicity constraints on each color.
Question:	Does G have a well-colored subset of k edges that can be extended to a perfect matching?

Internally Well-Colored Spanning Tree	
Input:	A number k and a vertex-colored planar graph G together with multiplicity constraints on each color.
Question:	Does G have a spanning tree such that its internal vertices contain a well-colored subset of at least k vertices?

3 Constrained Multilinear Detection

For any multiplicities $\mu = (\mu_1, \ldots, \mu_q)$ and natural numbers n, k, consider the polynomial ring $\mathbb{Q}[y_{1,1}, \ldots, y_{1,n}, \ldots, y_{q,1}, \ldots, y_{q,n}]$ in nq variables. Following the nomenclature from randomized algebraic methods by Björklund, Kaski and Kowalik [4, 5], we refer to the variables $y_{i,1}, \ldots, y_{i,n}$ as the *shades* of the color i. We call a multilinear monomial of degree k in y-variables *well-colored* if for every color i, at most μ_i shades of the color i appear in the monomial. Furthermore, we associate with every subset M of $[nq]$ of size k a multilinear monomial y_M of degree k in the obvious manner.

▶ **Lemma 1.** *There is an algorithm that, given μ, constructs in time $\mathrm{poly}(n, k, q)$ a skew arithmetic circuit computing a polynomial $\chi_\mu = \sum_{M \subseteq [nq],\, |M|=k} c_M y_M$ such that $c_M \geq 0$ for all M, and strict inequality holds if and only if y_M is well-colored.*

Proof. For every $i \in [q]$, let

$$\mu_{\leq i} = \sum_{j=1}^{i} \mu_j$$

be the i-th partial sum of μ. We set $\mu_{\leq 0} = 0$. Then, we define a matrix $S \in \mathbb{Q}^{k \times nq}$ as follows: $S = (S_1 | S_2 | \ldots | S_q)$ is the concatenation of q blocks $S_1, \ldots, S_q \in \mathbb{Q}^{k \times n}$, one for each color. Each block S_i is in turn defined as $S_i = U_i \cdot V_i$, where $U_i \in \mathbb{Q}^{k \times \mu_i}$ is the a Vandermonde matrix of dimension $k \times \mu_i$, and $V_i \in \mathbb{Q}^{\mu_i \times n}$ is a Vandermonde matrix of dimension $\mu_i \times n$, namely

$$U_i = \mathrm{Vand}_k(\mu_{\leq i-1} + 1, \mu_{\leq i-1} + 2, \ldots, \mu_{\leq i}),$$
$$V_i = \mathrm{Vand}_{\mu_i}(1, \ldots, n).$$

For a subset $M \subseteq [nq]$ of size k, let $\sigma_M = \det(S[M])$. We claim that $\sigma_M \geq 0$ holds, which can be seen as follows. Let $m = \mu_{\leq q}$, and consider the auxiliary matrices

$$U = (U_1 | U_2 | \ldots | U_q) = \mathrm{Vand}_k(1, \ldots, m) \in \mathbb{Q}^{k \times m},$$

$$V = V_1 \oplus \ldots \oplus V_q \in \mathbb{Q}^{m \times nq}.$$

Then per definition, we have $S = UV$, and moreover $S[M] = U \cdot V[M]$. Therefore, by the Cauchy-Binet formula,

$$\sigma_M = \sum_{\substack{L \subseteq [m] \\ |L| = k}} \det(U[L]) \cdot \det(V[L, M]). \tag{2}$$

Note now that $U[L]$ is a Vandermonde matrix of an increasing sequence, hence $\det(U[L])$ is strictly positive, as witnessed by the well-known formula for the Vandermonde determinant:

$$\det(\mathrm{Vand}_k(r_1, \ldots, r_N)) = \prod_{1 \leq i < j \leq N} (r_j - r_i).$$

On the other hand, we observe that $V[L, M]$ is either a direct sum of submatrices of Vandermonde matrices of positive increasing sequences, or has determinant zero. In the former case, $\det(V[L, M])$ is the product of the determinants of these submatrices. It is also well-known that Vandermonde matrices of positive, increasing sequences are *totally positive*, that is, *all* their minors, not just maximal minors, are positive. Hence, the determinant of

each submatrix is positive, and so is their product, i.e., $\det(V[L, M]) > 0$. Consequently, since we already argued that $\det(U[L]) > 0$ holds, we have shown that $\det(V[L, M]) \cdot \det(U[L]) \geq 0$ holds for all L in (2), and thus in particular $\sigma_M \geq 0$.

Furthermore, if M contains more than μ_i indices that belong to the i-th block of S, then $\sigma_M = 0$: The i-th block of S is defined as $S_i = U_i V_i$ with $U_i \in \mathbb{Q}^{k \times \mu_i}$ and $V_i \in \mathbb{Q}^{\mu_i \times n}$. Hence, S_i is of rank at most μ_i, and any set of more than μ_i columns from S_i will necessarily be linearly dependent. Conversely, if M contains $\rho_i \leq \mu_i$ indices belonging to the i-th block of S for each i, pick arbitrary subset L of rows of V containing precisely ρ_i rows from each V_i. Then $V[L, M]$ is a direct sum of square submatrices of Vandermonde matrices of increasing sequences, which makes the corresponding term $\det(U[L]) \det(V[L, M])$ strictly positive. Since we have just argued that all summands in (2) are non-negative, this proves that in this case, $\sigma_M > 0$.

Overall, we have shown that $\sigma_M \neq 0$ if and only if M contains no more than μ_i indices from the same block of S, which is equivalent to the corresponding monomial y_M being well-colored. Another application of Cauchy-Binet then provides us with the saught circuit: Letting Y be the matrix with diagonal entries $y_{1,1}, y_{1,2}, \ldots, y_{n,q}$, we observe that

$$\det(S \cdot Y \cdot S^T) = \sum_{M \subseteq [nq],\, |M|=k} \sigma_M^2 \cdot y_M.$$

By the preceding argument, this polynomial has the desired properties demanded in the statement, and the witnessing skew circuit can be written down in polynomial time, using the known constructions for skew determinant circuits [23]. ◀

▶ Remark 2. The proof of the preceding Lemma can also be seen as constructing an explicit representation of the k-truncation of the partition matroid corresponding to μ.

▶ **Theorem 3.** *There is a deterministic algorithm that, given an n-variate homogeneous polynomial f of degree k with non-negative coefficients, represented as an arithmetic circuit of size s, as well as multiplicities μ, decides in time $2^{\omega k} \cdot \mathrm{poly}(n, k, s)$ whether or not f contains a multilinear monomial that is well-colored with respect to μ. Here $\omega < 2.373$ denotes the exponent of matrix multiplication. This running time can be reduced to $4^k \cdot \mathrm{poly}(n, k, s)$ if the circuit computing f is skew.*

Proof. We invoke [8, Theorem 25] with f and χ_μ to compute $\langle f, \chi_\mu \rangle$. Since both f (by assumption) and χ_μ (by Lemma 1) have non-negative coefficients, this inner product is zero if and only if no well-colored multilinear monomial exists in f, and positive otherwise. The claim on the improved running time follows from [8, Theorem 7] and the fact that χ_μ is a determinant polynomial. ◀

4 Graph Problems

This theorem allows us to recover the best known bounds for deterministic detection of maximum graph motifs using an entirely different approach.

▶ **Corollary 4** ([26])**.** *There is a deterministic algorithm for* Maximum Graph Motif *running in time $2^{\omega k}\mathrm{poly}(n, k)$.*

Proof. All that is needed is a polynomial representation of the set of all k-vertex connected subgraphs of the input graph. This is possible due to a construction first employed by Guillemot and Sikora [17] on so-called *branching walks*. Consider the following sequence of polynomials: $P_{i,0} = 1$ for all i, and

$$P_{i,s} = x_i \sum_{j \in N_G(i)} \sum_{t_1+t_2=s-1} P_{i,t_1} \cdot P_{j,t_2}.$$

The multilinear monomials in $\beta_k = \sum_{i \in V} P_{i,k}$ can be shown to correspond bijectively to k-vertex connected subgraphs, and clearly all coefficients are non-negative. Hence, given a coloring $\chi : [n] \to C$, it suffices to evaluate β_k with $x_j = y_{c,j}$ for all j such that $\chi(j) = c$, for all colors $c \in C$. Invoking Theorem 3 then answers whether or not there is a graph motif as sought. ◀

▶ Remark 5. If one were able to design a skew circuit computing β_k, the running time in the preceding theorem would drop immediately to 4^k. From the perspective of algebraic complexity, it is an interesting problem whether such skew circuits for β_k exist, or whether one can rule out their existence under common complexity-theoretic assumptions. The latter could be accomplished e.g. by a completeness proof of β_k for the algebraic complexity class VP. However, despite heavy research efforts in the past (see e.g. [12] and references therein), very few natural VP-complete polynomials are known, and the family β_k is not among them.

One conspicuous property of the polynomial β_k is that its computation is *monotone,* that is, no cancellations can arise during its computation. However, the method desribed here is able to deal also with such cases where cancellations due to negations occur (but the resulting final coefficients are still non-negative). Indeed, this distinction is subtle but crucial: for instance, the determinant can only be computed without using cancellations by circuits of exponential size [19], whereas it is well-known to admit general arithmetic circuits with cancellations of polynomial size. A large number of polynomials that enumerate combinatorial objects can be expressed as determinants, which makes this situation particularly relevant. Moreover, determinants are the prototypical example for polynomials computable by skew circuits, which allows to use the faster running time mentioned in Theorem 3. For instance, this allows us to solve e.g. the following problems in deterministic time $O^*(4^k)$:

▶ **Theorem 6.** *There are deterministic algorithms running in time* $4^k \cdot \mathrm{poly}(n, k)$ *for each of the following problems:*

- *Well-Colored Spanning Tree,*
- *Well-Colored Planar Perfect Matching, and*
- *Internally Well-Colored Spanning Tree.*

Proof. The algorithms for these three problems all follow the same basic principle and build upon the algorithms for the variants where *well-colored* monomials are replaced by monomials with at least k distinct colors, which is the special case of having multiplicities $\mu_i = 1$ for all i. The core idea is to make use of determinantal generating functions for the sought objects in each case. These generating functions are provided by the directed Matrix-Tree theorem and the Pfaffian of planar graphs. Details on these formulations can be found in [6] and [3]. Once these generating functions are available, all that remains to check is that by a standard trick of substituting $x_i \mapsto (1 + x_i)$ for every variable, these become generating functions for all subsets of solutions (that is, all subsets of edges of spanning trees, all subsets of perfects matching, etc.), and the claim follows by applying Theorem 3. ◀

5 Intersecting Four Matroids

The general method for intersecting q matroids shown in [8] exploits a well-known connection to matroid parity, and solves the latter problem instead. Indeed, given q matroids each of rank k represented by matrices with entries in $\mathbb{Q}$, the algorithm in [8] runs in randomized

time $O^*(4^{kq})$. In particular, for the cases $q = 3, 4$, this specializes to algorithms running in time $O^*(64^k)$ and $O^*(256^k)$, respectively, and [14] improve this to $O^*(4^k)$ and $O^*(16^k)$ (and $O^*(4^{k(q-2)})$ in general). We will now show how to obtain a running time of $O^*(4^k)$ in both cases, and examine the conditions under which the algorithms can be made deterministic. First, we define an extension of the apolar inner product to tensor products of polynomial rings, that is, polynomial rings that have themselves as a coefficient rings another ring of polynomials. For instance, consider $f \in \mathbb{Q}[x_1, \ldots, x_n] \otimes \mathbb{Q}[y_1, \ldots, y_n] \cong \mathbb{Q}[x_1, \ldots, x_n][y_1, \ldots, y_n]$. In general, f has the form

$$f = \sum_{a_1,\ldots,a_n \in \mathbb{N}} \sum_{b_1,\ldots,b_n \in \mathbb{N}} c^{b_1,\ldots,b_n}_{a_1,\ldots,a_n} \cdot x_1^{a_1} \cdots x_n^{a_n} \cdot y_1^{b_1} \cdots y_n^{b_n},$$

with only finitely many of the $c^{b_1,\ldots,b_n}_{a_1,\ldots,a_n}$ non-zero. Moreover, for fixed $a_1, \ldots, a_n$, we can collect all the corresponding terms into a single polynomial $\hat{c}_{a_1,\ldots,a_n} \in \mathbb{Q}[y_1, \ldots, y_n]$ via

$$\hat{c}_{a_1,\ldots,a_n} = \sum_{b_1,\ldots,b_n \in \mathbb{N}} c^{b_1,\ldots,b_n}_{a_1,\ldots,a_n} \cdot y_1^{b_1} \cdots y_n^{b_n},$$

and then recover the familiar

$$f = \sum_{a_1,\ldots,a_n \in \mathbb{N}} \hat{c}_{a_1,\ldots,a_n} x_1^{a_1} \cdots x_n^{a_n}.$$

While it is true that

$$\mathbb{Q}[x_1, \ldots, x_n][y_1, \ldots, y_n] \cong \mathbb{Q}[x_1, \ldots, x_n, y_1, \ldots, y_n],$$

it is important that we distinguish these two ways of looking at f. In particular, there is nothing new to say about the inner product on the latter polynomial ring, which is well-defined already through Eq. (1). However, for our purposes, we extend the definition Eq. (1) to $f, g \in \mathbb{Q}[x_1, \ldots, x_n][y_1, \ldots, y_n]$, where f is as above and $g = \sum_{a_1,\ldots,a_n \in \mathbb{N}} \hat{d}_{a_1,\ldots,a_n} x_1^{a_1} \cdots x_n^{a_n}$, with $\hat{d}_{a_1\ldots,a_n} \in \mathbb{Q}[y_1, \ldots, y_n]$ defined analogously to $\hat{c}_{a_1,\ldots,a_n}$. Then, we set

$$\alpha(f, g) := \sum_{a_1,\ldots,a_n \in \mathbb{N}} a_1! \cdot a_2! \cdots a_n! \cdot \hat{c}_{a_1,\ldots,a_n} \cdot \hat{d}_{a_1,\ldots,a_n} \in \mathbb{Q}[y_1, \ldots, y_n].$$

Note that such a mapping cannot possibly be an inner product anymore (after all, its codomain is not the field $\mathbb{Q}$, but $\mathbb{Q}[y_1, \ldots, y_n]$), and thus the need for a separate treatment arises. In particular, the algorithms for computing the apolar inner product from [8] do not extend, at least not within the same running time bound, to the case where $\langle \cdot, \cdot \rangle$ is replaced by α. However, we can use their results to obtain the following:

▶ **Lemma 7.** *Let $f = \det(M_1)$ and $g = \det(M_2)$, where M_i for $i = 1, 2$ are matrices of dimension $k \times k$ having bilinear polynomials in $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ as entries. Then, there is an algorithm that, given $\bar{y}_1, \ldots, \bar{y}_n \in \mathbb{Q}$, evaluates the polynomial $\alpha(f, g) \in \mathbb{Q}[y_1, \ldots, y_n]$ at $\bar{y}_1, \ldots, \bar{y}_n$ in $O^*(4^k)$ arithmetic operations over $\mathbb{Q}$. Moreover, if the evaluation points $\bar{y}_1, \ldots, \bar{y}_n$ can be encoded using $O^*(1)$ bits, then $\alpha(f, g)$ can be evaluated in time $O^*(4^k)$.*

Proof. By substituting $\bar{y}_1, \ldots, \bar{y}_n$ into M_1 and M_2 and using the fact that evaluation of polynomials at a fixed point is a homomorphism, we find that $\langle f(\bar{y}_1, \ldots, \bar{y}_n), g(\bar{y}_1, \ldots, \bar{y}_n) \rangle = \alpha(f, g)(\bar{y}_1, \ldots, \bar{y}_n)$ for all $\bar{y}_1, \ldots, \bar{y}_n$. Since the determinant is well-known to have skew circuits [23], we are in position to apply [8, Theorem 7] to evaluate $\langle f(\bar{y}_1, \ldots, \bar{y}_n), g(\bar{y}_1, \ldots, \bar{y}_n) \rangle$ in the required time bound. ◀

▶ **Theorem 8.** *Let $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ be four matroids over a common ground set E of size n, each represented by a matrix $R_i \in \mathbb{Q}^{k\times n}$. Then, we can decide in randomized time $O^*(4^k)$ whether these four matroids share a common basis.*

Proof. First, let us assume that $E = [n]$ without loss of generality. We then begin by introducing n fresh indeterminates $y_1, \ldots, y_n$. Let then Y be the diagonal matrix of dimension $n \times n$ having y_i in its i-th diagonal entry. Observe that $\hat{R}_i := R_i Y$ is the matrix R_i where the i-th column was scaled by a factor of y_i. Moreover, for any set $S \subset [n]$ of k column indices, the maximal minor of $\hat{R}_i$ corresponding to S, that is, the determinant of the matrix $\hat{R}_i[S]$ obtained from $\hat{R}_i$ by restricting to the columns in S, is a polynomial in $y_1, \ldots, y_n$. Indeed, since the determinant is a multilinear functional in its columns, we have

$$\det(\hat{R}_i[S]) = \prod_{s\in S} y_s \cdot \det(R_i[S]). \tag{3}$$

In analogy to Y, let X be the diagonal matrix having x_i in its i-th diagonal entry. Now, the Cauchy-Binet formula gives the following expression for the determinant of the product $\hat{R}_i \cdot X \cdot R_j^T$, with a similar reasoning as in Eq. (3):

$$\begin{aligned}\det(\hat{R}_i \cdot X \cdot R_j^T) = \\ \sum_{S\subseteq[n],|S|=k} \det(\hat{R}_i[S]) \cdot \det(R_j[S]) \cdot \prod_{s\in S} x_s = \\ \sum_{S\subseteq[n],|S|=k} \left(\prod_{s\in S} y_s \cdot \det(R_i[S]) \cdot \det(R_j[S])\right) \cdot \prod_{s\in S} x_s.\end{aligned}$$

In the last line, we grouped the products in order to highlight that we consider this expression foremost as a polynomial in the variables x_i, that is, an element of $\mathbb{Q}[y_1, \ldots, y_n][x_1, \ldots, x_n]$. Of course, the analogous expression holds for $\det(R_i \cdot X \cdot R_j^T)$ (note the missing hat on R_i), namely

$$\det(R_i \cdot X \cdot R_j^T) = \sum_{S\subseteq[n],|S|=k} \det(R_i[S]) \cdot \det(R_j[S]) \cdot \prod_{s\in S} x_s,$$

which is a polynomial from $\mathbb{Q}[x_1, \ldots, x_n] \subset \mathbb{Q}[y_1, \ldots, y_n][x_1, \ldots, x_n]$.

Since by assumption, $\det(R_i[S]) \neq 0$ if and only if S is a basis of $\mathcal{M}_i$, it follows from the definition of $\alpha(\cdot,\cdot)$ (and noting that all the x_i have exponent zero or one) that

$$\alpha(\det(\hat{R}_1 \cdot X \cdot R_2^T),\ \det(R_3 \cdot X \cdot R_4^T)) = \tag{4}$$

$$\sum_{S\subseteq[n],|S|=k} (\det(R_1[S]) \cdot \det(R_2[S])) \cdot \left(\prod_{s\in S} y_s \cdot \det(R_3[S]) \cdot \det(R_4[S])\right) = \tag{5}$$

$$\sum_{\substack{S \text{ basis of} \\ \mathcal{M}_1,\ldots,\mathcal{M}_4}} \underbrace{\det(R_1[S]R_2[S]R_3[S]R_4[S])}_{\neq 0} \cdot \prod_{s\in S} y_s. \tag{6}$$

As witnessed in the last line of the preceding calculation, $\alpha(\det(\hat{R}_1 \cdot X \cdot R_2^T),\ \det(R_3 \cdot X \cdot R_4^T))$ is the zero polynomial if and only if the four input matroids share no common basis. Therefore, all that remains is to test the polynomial $\alpha(\det(\hat{R}_1 \cdot X \cdot R_2^T),\ \det(R_3 \cdot X \cdot R_4^T))$ for zero, using the DeMillo–Lipton–Schwartz–Zippel Lemma [10, 30, 28] in combination with Lemma 7, we obtain the desired randomized algorithm: Choosing random evaluation points, we can ensure that $\alpha(\det(\hat{R}_1 \cdot X \cdot R_2^T),\ \det(R_3 \cdot X \cdot R_4^T))$ evaluates to a non-zero value, in case

it truly is non-zero, with constant probability. Of course, if the polynomial is identically zero, so is every evaluation, and the algorithm will always correctly recognize this. As usual, repeating this procedure a polynomial number of times allows us to decrease the one-sided error probability exponentially. ◀

▶ **Corollary 9.** *Given three matroids $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$ as in Theorem 8, we can decide in randomized time $O^*(4^k)$ whether they share a common basis.*

Proof. This follows from choosing $\mathcal{M}_1 = \mathcal{M}_2$ in Theorem 8. ◀

▶ Remark 1. During the preparation of the present article, a manuscript by Eiben, Koana and Wahlström [14] appeared, where they give a different algebraic approach to some of the matroid problems considered here. In particular, their Theorem 4.6 coincides with Corollary 9, and they show how to use this as a base case to obtain an algorithm for intersecting q rank-k matroids, running in time $4^{k(q-2)} \cdot \mathrm{poly}(n)$. It would be interesting to see if our Theorem 8 can be expedited in a similar manner to obtain a running time of $4^{k(q-3)} \cdot \mathrm{poly}(n)$. Their techniques are based on exterior algebra, which is related to the methods used in this article through a general algebraic connection [7].

5.1 Intersecting Positroids

Using the same strategy as for general matroids, we obtain deterministic algorithms for the following important class of matroids:

▶ Definition 10. Let $k \leq n$, and let $\mathcal{M}$ be a matroid over a ground set of size n of rank k. Suppose $\mathcal{M}$ is represented by a matrix M such that for each submatrix $M[S]$ with $S \subset [n]$ and $|S| = k$ it holds that the corresponding maximal minor satisfies $\det(M[S]) \geq 0$. In this case, $\mathcal{M}$ is called a *positroid*.

It is worth pointing out that while these objects seem not to have made any significant algorithmic appearances so far, they are of great importance in geometry, where they correspond to the so-called *totally non-negative Graßmannian*. They have many desirable properties that general matroids are lacking, most notably a beautiful combinatorial correspondence with certain planar bicolored (or *plabic*) graphs. We refer the reader to Postnikov's groundbreaking work on the subject [27].[1]

Let us furthermore remark that the following results will only apply to the situation where the input matroids are *promised* to be positroids, since there doesn't seem to be a way to decide efficiently whether a given matroid representation does indeed only have non-negative minors. While the aforementioned correspondence with plabic graphs does in principle allow for a full-fledged decision problem (by taking as an input not the matrix of the positroid, but its corresponding plabic graph, and then computing a representation from this graph), this is beyond the scope of the present article.

▶ **Theorem 11.** *Let $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4$ be four positroids over a common ground set E of size n, each represented by a matrix $R_i \in \mathbb{Q}^{k \times n}$ with non-negative minors. Then, we can decide in deterministic time $O^*(4^k)$ whether these four matroids share a common basis.*

Proof. The proof relies on the fact that all determinants in Eq. (6) are not only non-zero, but in fact positive. Therefore, it is not necessary to include the variables y_i in the calculation, and a direct application of [8, Theorem 7] to $\langle \det(R_1 \cdot X \cdot R_2^T), \det(R_3 \cdot X \cdot R_4^T) \rangle$ is already enough. The resulting value is non-zero if and only if the four positroids share a common basis, and the running time bound follows directly from [8]. ◀

[1] This particular manuscript, despite being cited hundreds of times, didn't appear in a journal.

References

1 Alexander I Barvinok. New algorithms for linear k-matroid intersection and matroid k-parity problems. *Mathematical Programming*, 69:449–470, 1995.

2 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Fourier meets möbius: fast subset convolution. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 67–74. ACM, 2007. doi:10.1145/1250790.1250801.

3 Andreas Björklund, Petteri Kaski, and Ioannis Koutis. Directed hamiltonicity and out-branchings via generalized laplacians. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 91:1–91:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.91.

4 Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Probably optimal graph motifs. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013, February 27 – March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 20–31. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.STACS.2013.20.

5 Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Constrained multilinear detection and generalized graph motifs. *Algorithmica*, 74(2):947–967, 2016. doi:10.1007/s00453-015-9981-1.

6 Cornelius Brand. Patching colors with tensors. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 25:1–25:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.25.

7 Cornelius Brand. A note on algebraic techniques for subgraph detection. *Inf. Process. Lett.*, 176:106242, 2022. doi:10.1016/j.ipl.2021.106242.

8 Cornelius Brand and Kevin Pratt. Parameterized applications of symbolic differentiation of (totally) multilinear polynomials. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 38:1–38:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ICALP.2021.38.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

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

11 Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. doi:10.1007/978-1-4612-0515-9.

12 Arnaud Durand, Meena Mahajan, Guillaume Malod, Nicolas de Rugy-Altherre, and Nitin Saurabh. Homomorphism polynomials complete for VP. *Chic. J. Theor. Comput. Sci.*, 2016, 2016. URL: http://cjtcs.cs.uchicago.edu/articles/2016/3/contents.html.

13 Jack Edmonds. Matroid intersection. In *Annals of discrete Mathematics*, volume 4, pages 39–49. Elsevier, 1979.

14 Eduard Eiben, Tomohiro Koana, and Magnus Wahlström. Determinantal sieving, 2023. arXiv:2304.02091.

15 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

16 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Representative families of product families. *ACM Trans. Algorithms*, 13(3):36:1–36:29, 2017. doi:10.1145/3039243.

17 Sylvain Guillemot and Florian Sikora. Finding and counting vertex-colored subtrees. In Petr Hlinený and Antonín Kucera, editors, *Mathematical Foundations of Computer Science 2010, 35th International Symposium, MFCS 2010, Brno, Czech Republic, August 23-27, 2010. Proceedings*, volume 6281 of *Lecture Notes in Computer Science*, pages 405–416. Springer, 2010. doi:10.1007/978-3-642-15155-2_36.

18 Gregory Z. Gutin, Felix Reidl, Magnus Wahlström, and Meirav Zehavi. Designing deterministic polynomial-space algorithms by color-coding multivariate polynomials. *J. Comput. Syst. Sci.*, 95:69–85, 2018. doi:10.1016/j.jcss.2018.01.004.

19 Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982. doi:10.1145/322326.322341.

20 Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 575–586. Springer, 2008. doi:10.1007/978-3-540-70575-8_47.

21 Ioannis Koutis. Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett.*, 112(22):889–892, 2012. doi:10.1016/j.ipl.2012.08.008.

22 Ioannis Koutis and Ryan Williams. Limits and applications of group algebras for parameterized problems. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 653–664. Springer, 2009. doi:10.1007/978-3-642-02927-1_54.

23 Meena Mahajan and V. Vinay. A combinatorial algorithm for the determinant. In Michael E. Saks, editor, *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana, USA*, pages 730–738. ACM/SIAM, 1997. URL: http://dl.acm.org/citation.cfm?id=314161.314429.

24 Dániel Marx. A parameterized view on matroid optimization problems. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, volume 4051 of *Lecture Notes in Computer Science*, pages 655–666. Springer, 2006. doi:10.1007/11786986_57.

25 Burkhard Monien. How to find long paths efficiently. In *North-Holland Mathematics Studies*, volume 109, pages 239–254. Elsevier, 1985.

26 Ron Y. Pinter, Hadas Shachnai, and Meirav Zehavi. Deterministic parameterized algorithms for the graph motif problem. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 – 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 589–600. Springer, 2014. doi:10.1007/978-3-662-44465-8_50.

27 Alexander Postnikov. Total positivity, grassmannians, and networks, 2006. arXiv:math/0609764.

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

29 Ryan Williams. Finding paths of length k in $o^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009. doi:10.1016/j.ipl.2008.11.004.

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