Limits and Applications of Group Algebras for Parameterized Problems

Ioannis Koutis
Computer Science Department
Carnegie Mellon University
ioannis.koutis@cs.cmu.edu

Ryan Williams
School of Mathematics
Institute of Advanced Study
ryanw@math.ias.edu

May 1, 2009

Abstract

The algebraic framework introduced in [Koutis, Proc. of the 35^{th} ICALP 2008] reduces several combinatorial problems in parameterized complexity to the problem of detecting multilinear degree-k monomials in polynomials presented as circuits. The best known (randomized) algorithm for this problem requires only $O^*(2^k)$ time and oracle access to an arithmetic circuit, i.e. the ability to evaluate the circuit on elements from a suitable group algebra. This algorithm has been used to obtain the best known algorithms for several parameterized problems. In this paper we use communication complexity to show that the $O^*(2^k)$ algorithm is essentially *optimal* within this *evaluation oracle* framework. On the positive side, we give new applications of the method: finding a copy of a given tree on k nodes, a spanning tree with at least k leaves, a minimum set of nodes that dominate at least t nodes, and an m-dimensional k-matching. In each case we achieve a faster algorithm than what was known. We also apply the algebraic method to problems in exact counting. Among other results, we show that a combination of dynamic programming and a variation of the algebraic method can break the trivial upper bounds for exact parameterized counting in fairly general settings.

1 Introduction

The algebraic framework introduced in [14] reduces several parameterized problems including the k-path problem, the m-set k-packing problem, and several graph packing problems to the following ground problem:

The k-monomial detection problem. Given a (commutative) arithmetic circuit C over a set of variables X, decide whether the polynomial $P(X)$ represented by C contains a multilinear monomial of degree k.

In other words, construing $P(X)$ as a sum of products, we wish to know if the sum contains a degree-k monomial with no squares. Here, an arithmetic circuit is a directed acyclic graph with a single sink, sources labelled by variables from a set X, and multiplication and addition gate labels at all other nodes. The decision algorithm given in [14] solves the *odd* k-monomial detection problem where the additional assumption that $P(X)$ contains an odd number of multilinear monomials is made. The technique used in [14] to reduce the parameterized packing problems to the odd k-monomial detection problem was extended and completed in [19] where the general k-term problem was essentially reduced to the odd problem by working over a larger group algebra.

The algebraic method of [14, 19] is based on choosing a commutative algebra $\mathcal{A}$ and randomized assignments $X \rightarrow \mathcal{A}$ such that (i) squares (and by commutativity, non-multilinear monomials) evaluate to the zero element of $\mathcal{A}$, and (ii) some multilinear monomial does not evaluate to zero, with good probability. For instance, the algorithm presented in [19] uses certain assignments from the group algebra $GF(2^{3+\log_2 k})[\mathbb{Z}_2^k]$ (for the definition of a group algebra, see the Appendix). Elements in $GF(2^\ell)[\mathbb{Z}_2^k]$ can be described in space $O(\ell 2^k)$ and the complexity of addition and multiplication in the algebra is $O^*(\ell 2^k)$.[1] This gives an $O^*(2^k t)$

[1] Throughout the paper the $O^*()$ notation hides factors polynomial in the instance size n and the parameter k.

algorithm for k-monomial detection, where t is the number of gates in C. The algorithm requires only *oracle access* to an "extended version" C' of C, and it can be implemented as one evaluation of C' over $GF(2^\ell)[\mathbb{Z}_2^k]$, or as $O^*(\ell 2^k)$ evaluations of C' over small polynomials in $\mathbb{Z}[x]$.

The technique yields the fastest known parameterized algorithms for the k-path problem, the m-set k-packing problem (packing k sets, each of size m), and more. In Section 2 we also show how to obtain faster algorithms for finding a copy of a given tree on k nodes, a spanning tree with k leaves, a minimum set of nodes that dominate at least t nodes in a graph, and an m-dimensional k-matching, all by simple reductions to k-monomial detection. Some of these algorithms match the best known upper bounds for the original NP-hard problems, i.e. the Hamiltonian path problem, and the k-set (n/k)-packing problem, in the sense that further improvement on the parameterized algorithms will imply further progress on the original problems. Thus, an intriguing and natural question is whether k-monomial detection can be solved faster, by evaluating circuits over a more exotic $\mathcal{A}$, with a significant reduction in the lengths of descriptions for elements in $\mathcal{A}$ (much less than 2^k), and the related complexity of arithmetic in $\mathcal{A}$. If so, this would have tremendous implications in exact algorithms.

The answer is, unfortunately, negative. In Section 3 we use communication complexity to show that for any commutative algebra $\mathcal{A}$ used to evaluate the circuit C, the lengths of elements in $\mathcal{A}$ must be at least $\Omega(2^k/k)$ in order to perform k-monomial detection. For all of the applications listed above, commutativity of multiplication is required.[2] Thus the $O^*(2^k)$ algorithm for k-monomial detection is optimal in some sense, and further progress on the relevant parameterized problems are only via algorithms that exploit specific properties of circuits for k-path and set packing, or perform different kinds of operations altogether. Much research in complexity theory focuses on understanding the limits of restricted algorithmic frameworks. Such cases include the study of deterministic oracle-based algorithms for the computation of convex volumes [7], of randomized oracle-based volume computation algorithms [16], of local minimum search algorithms for black-box functions [1], or in the area of parameterized kernelization [5]. Apart from the obvious potential of saving unneeded efforts on upper bounds, this type of research often reveals surprising connections.

Although the algebraic method is superior to older approaches such as color coding [3] and the divide-and-color approaches [11, 6] for *decision* problems, its potential has not been explored in the context of the related *counting* problem:

Exact multilinear k-monomial counting. Given a (commutative) arithmetic circuit C describing an n-variate polynomial P over $\mathbb{R}$, compute the sum of the coefficients of the degree-k multilinear monomials in P.

A simple enumeration-based algorithm for the problem runs in $O^*(2^k\binom{n}{k})$ time. Given the #W[1] hardness of exact counting of k-paths [8] (a special case), an $O(f(k)poly(n))$ algorithm for the counting problem is unlikely to exist. However, certain recent results indicate that improvements may be possible. Björklund et al. gave an $O^*\left(\binom{n}{k/2}\right)$ algorithm for computing the number of k-paths [4]. Using rectangular permanents, Vassilevska and Williams gave algorithms for counting small subgraphs with large independent sets [18]. In Section 4 we show that a *combination* of enumeration-based algorithm and the algebraic method is able to break the naive $\binom{n}{k}$ upper bound for this problem when the polynomial P is a product of two polynomials P_1, P_2 each of which have total degree less than k. One application of this is an $O^*(n^{mk/2})$ algorithm for counting k-packings of m-sets over a universe of size n.

2 Faster Parameterized Algorithms

To further demonstrate the power of k-monomial detection, we show how it can be used to obtain faster randomized algorithms for the following problems:

k-Tree. Given a tree T on k nodes and a graph G on n nodes, decide if there is a (not necessarily induced) copy of T in G.

[2] We state our result for commutative algebras, for the sake of clarity and coherence with our algorithmic results that all use commutativity. However, our lower bound holds in the non-commutative setting as well, under the appropriate definition.

k-Leaf Spanning Tree. Given an undirected graph G on n nodes, decide if G contains a spanning tree with at least k leaves.

t-Dominating Set. Given a graph $G = (V, E)$, find a minimum set of nodes S that dominate at least t nodes in the graph. That is, $|S \cup N(S)| \geq t$ where $N(S) = \{v \mid (u, v) \in E, u \in S\}$.

m-Dimensional k-Matching. Given mutually disjoint sets U_i, for $i = 1, \ldots, m$, and a collection $\mathcal{C}$ of m-tuples from $U_1 \times \ldots \times U_m$, decide whether $\mathcal{C}$ contains a sub-collection of k mutually disjoint m-tuples.

Each of these can be solved by formulating them as k-monomial detection instances in some way. To the best of our knowledge, the only other algorithm we know for k-Tree follows from the color-coding method [3], running in $O^*((2e)^k)$ time. For k-Leaf Spanning Tree, the best known algorithm is deterministic and runs in $O^*(4^k)$ time [10]. The best known (randomized) algorithm for t-Dominating Set runs in time $O^*((4+\varepsilon)^t)$ [12]. The best known (randomized) algorithm for the k m-Dimensional Matching Problem runs in time $O^*(2^{mk})$ [14]. In addition, in all these problems, given an algorithm for the decision version of the problem, standard reductions can be used to recover an algorithm for the search version, with the same exponential dependence on the parameter.

Before we proceed to showing the reductions we consider the following generalization of the results in [14, 19].

Lemma 2.1. *Let $P(X, z)$ be a polynomial represented by a commutative arithmetic circuit C. The existence of a term of the form $z^t Q(X)$ in $P(X, z)$, where $Q(X)$ is a multilinear monomial of degree at most k, can be decided in time $O^*(2^k t^2 |C|)$ and space $O(t|C|)$.*

The idea of the proof is to assign values $\mathcal{X}$ from the appropriate algebra $\mathcal{A}$ to the variables X, in exactly the same fashion as in [19], evaluate the polynomial $P(\mathcal{X}, z)$ symbolically, then read off the answer from the coefficient $Q(\mathcal{X}) \in \mathcal{A}$ of z^t. The details will appear in the full version of the paper.

Theorem 2.2. *The k-Tree problem can be solved in $O^*(2^k)$ time.*

Proof. Suppose T is a k-node tree we wish to find in G. Let the nodes of T be $\{1, \ldots, k\}$, and let the nodes of G be $\{1, \ldots, n\}$. We define an arithmetic circuit $C_{T,i,j}(x_1, \ldots, x_n)$ inductively, for all $i \in [k]$ and $j \in [n]$.

- If $|V(T)| = 1$, simply define $C_{T,i,j} := x_j$.
- If $|V(T)| > 1$, let $T_{i,1}, \ldots, T_{i,\ell}$ be the connected subtrees of T remaining after node i is removed from T. For all $t = 1, \ldots, \ell$, let $n_{i,t} \in [k]$ be the (unique) node in $T_{i,t}$ that is a neighbor of i in T. Define

$$C_{T,i,j} := \prod_{t=1}^{\ell} \left(\sum_{j' : (j,j') \in E(G)} x_j \cdot C_{T_{i,t}, n_{i,t}, j'} \right).$$

Observe that the size of $C_{T,i,j}(x_1, \ldots, x_n)$ is at most $O(|V(T)| \cdot |E(G)|)$. The polynomial $C_{T,i,j}$ enumerates those homomorphisms that map nodes of T into nodes of G, such that node i in T is mapped to node j in G. Each monomial represents the range of some homomorphism. More precisely, the monomial $x_{j_1} \cdots x_{j_k}$ is present in the polynomial if and only if there is a homomorphism where the nodes of T are mapped to the vertices $\{j_1, \ldots, j_k\} \subseteq V$ of G. These mappings are all homomorphisms, since $i' \in V(T)$ can only be mapped to $j' \in V(G)$ if $(i, i') \in E(T)$, $(j, j') \in E(G)$, and $i \in V(T)$ is already mapped to $j \in V(G)$.

Now consider the sum-product expansion of $Q = \sum_{j \in V(G), i \in V(T)} C_{T,i,j}$. Q is a sum over all connected subgraphs S of G on k vertices, where each monomial corresponds to the range of some homomorphism from T into S. Note Q also includes homomorphisms that map distinct nodes of T to a common node in G. However, those homomorphisms correspond precisely to monomials with *squares* in them. That is, the multilinear monomials of Q correspond to homomorphisms that map all nodes in T to distinct nodes in G, i.e. an isomorphic copy of T in G. Therefore, by calling k-monomial detection on the arithmetic circuit defined by Q, we can detect whether G contains a copy of T in $O^*(2^k)$ time. □ □

Theorem 2.3. *The k-Leaf Spanning Tree problem can be solved in $O^*(2^k)$ time.*

Proof. To find a spanning tree with at least k leaves, first observe that it suffices to find a connected subgraph S with at least k nodes of degree one. To get a spanning tree for G, compute a spanning tree T over G with the nodes and edges of S removed. Connect T and S by an edge, and take the spanning tree to be the union of these two subgraphs (removing any extraneous edges that may form cycles). This was observed in [10].

It suffices to show how to detect if G has a connected subgraph with at least k degree one nodes. Let the vertices of G be $\{1, \ldots, n\}$. We define an arithmetic circuit $C_{k,t,i}(x_1, \ldots, x_n, z)$ inductively.

- Define $C_{1,1,i} := x_i \cdot z$. If $t < k$ or $k \leq 0$, define $C_{k,t,i} := 1$.
- For $k > 1$, define

$$C_{k,t,i} := \sum_{t'=1}^{t} \sum_{k'=1}^{k} \sum_{j:(i,j)\in E} \left(C_{k',t',j} \cdot C_{k-k',t-t',i}\right).$$

By induction, one can prove that the monomials of $C_{k,t,i}$ correspond to the connected subgraphs on t nodes which include node i of G and have up to k degree ≤ 1 nodes. Each such subgraph has a monomial corresponding to its leaves. A multilinear monomial of degree k in the coefficient of z^k implies that there is a subgraph rooted at i that has k nodes of degree one. Testing whether the circuit $Q = \sum_{t=k+1}^{n} \sum_{i=1}^{n} C_{k,t,i}$ has a multilinear monomial in the coefficient of z^k determines if G has a connected subgraph with at least k degree-one nodes. By Lemma 2.1 this can be done. Finally, note that the size of Q as an arithmetic circuit is $\text{poly}(n, k)$. □ □

Theorem 2.4. *The t-Dominating Set problem can be solved in $O^*(2^t)$ time.*

Proof. Let the vertex set be $V = \{1, \ldots, n\}$ and $X = \{x_1, \ldots, x_n\}$, where x_i corresponds to the vertex i. Consider the polynomial

$$P(X, z) = \left(\sum_{i\in V} \left((1 + z \cdot x_i) \cdot \prod_{j\,:\,(i,j)\in E} (1 + z \cdot x_j)\right)\right)^k,$$

where z is an extra indeterminate. P is a sum of monomials in which each monomial of the form $z^t x_{i_1} \cdots x_{i_t}$ for *distinct* i_j appears if and only if the t nodes $\{i_1, \ldots, i_t\}$ are dominated by a set of at most k nodes. In addition, every other term of the form $z^t Q(X)$ contains a square since $Q(X)$ has total degree t. Then, the proof follows from Lemma 2.1 and trials with increasing values of k. □ □

Theorem 2.5. *The m-Dimensional k-Matching problem can be solved in time $O^*(2^{(m-1)k})$.*

Proof. [Sketch] Encode each element u in $U = \bigcup_{i=2}^{m} U_i$ by a variable $x_u \in X$. Encode each m-tuple $t = (u_1, \ldots, u_m) \in \mathcal{C} \subseteq U_1 \times \cdots \times U_m$ by the monomial $M_t = \prod_{i=2}^{m} x_{u_i}$. Assume $U_1 = \{u_{1,1}, \ldots, u_{1,n}\}$, and let $T_j \subseteq \mathcal{C}$ denote the subset of m-tuples whose first coordinate is $u_{1,j}$. Consider the polynomial

$$P(X, z) = \prod_{j=1}^{n} \left(1 + \sum_{t\in T_j} (z \cdot M_t)\right),$$

where z is an extra indeterminate. The coefficient of z^k is a polynomial $Q(X)$ which contains a multilinear $((m-1)k)$-term if and only if $\mathcal{C}$ contains a k-matching. The proof follows from Lemma 2.1. □ □

3 Lower Bound for Multilinear Detection

We now investigate whether the approach to finding k-paths in [14, 19] can be improved upon further. One burning question from this work is whether is it possible to determine if an arbitrary arithmetic circuit C has a multilinear k-monomial in much less time than $O(2^k|C|)$, by evaluating C over a more exotic algebraic structure. We shall prove that this is *not* the case. For any commutative $\mathcal{A}$, we show that if it can be used to detect multilinear monomials in an arithmetic circuit (even randomly), then $|\mathcal{A}| \geq 2^{\Omega(2^k/\sqrt{k})}$. This implies a lower bound of $\Omega(\frac{2^k}{\sqrt{k}}|C|)$. That is, the group algebras used in [14, 19] (which have $|\mathcal{A}| \leq 2^{O(2^k \log k)}$) are essentially optimal for their purpose: commutativity of $\mathcal{A}$ is required for all the applications of k-monomial detection that we have shown.

At a high level, our proof uses hypothetical fast multilinear monomial detection in order to design communication protocols that are too efficient to exist. Let us informally recall some notions from communication complexity. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Suppose two parties wish to compute $f(x, y)$, but one party is given x (the x-party), the other is given y (the y-party). The parties must communicate bits about their inputs in order to compute f. A *simultaneous public-coin protocol for f on n-bits* is specified by a pair of functions (g_1, g_2), and works as follows on all n-bit strings. Initially, the x-party and y-party see a common string z chosen at random from a distribution independent of x and y (to maximize the parties' capabilities, we assume the distribution is uniform). The x-party computes $g_1(x, z)$, the y-party computes $g_2(y, z)$, and both send their answers to a third party. We require that the third party holding the two messages can compute f (with high probability), on all x and y of length n. The *randomized public-coin simultaneous communication complexity of* (g_1, g_2) is the maximum length $L(n)$ of a message sent in the protocol (g_1, g_2) over all n-bit strings. The corresponding communication complexity of f is the minimum $L(n)$ achieved by any n-bit protocol (g_1, g_2) for f.

The set disjointness function $DISJ_n(x, y)$ on $x, y \in \{0,1\}^n$ is defined to be $\vee_{i=1}^n (x_i \wedge y_i)$. We utilize the following fact:

Theorem 3.1. *[[15],p.79] The randomized public-coin communication complexity of $DISJ_n$ is $\Omega(n)$.*

Theorem 3.2. *Let $\mathcal{A}$ be a commutative semiring. Suppose there is a distribution $\mathcal{D}$ on elements from $\mathcal{A}$, an element $e \in \mathcal{A}$, and constants $d_1, d_2 \in [0,1]$, $d_1 < d_2$ such that for every circuit $C(x_1, \ldots, x_n)$ representing a degree-k polynomial:*

- *C has a multilinear monomial $\implies \Pr_{(e_1,\ldots,e_n)\in\mathcal{D}^n}[C(e_1, \ldots, e_n) = e] \leq d_1$*
- *C does not have a multilinear monomial $\implies \Pr_{(e_1,\ldots,e_n)\in\mathcal{D}^n}[C(e_1, \ldots, e_n) = e] \geq d_2$.*

Then $|\mathcal{A}| \geq 2^{\Omega(2^k/\sqrt{k})}$. Furthermore, for every k there is a circuit C with $n = k$ for which this lower bound holds.

In other words, the group algebra utilized in [19] is optimal within poly(k) factors: any other algebra could only yield a slightly better asymptotic upper bound. As a consequence it is not possible to solve Hamilton Path much faster than $O(2^n)$ by solving k-monomial detection over a more interesting algebra.

Proof. Using multilinear detection, we design a protocol for $DISJ_N$. Let $N > 0$ and $k = n = \log N + \frac{1}{2}\log\log N + c$ where $c > 0$ is sufficiently large. Without loss of generality, assume k is even. Let $\mathcal{S} = \{S_1, \ldots, S_\ell\}$ be an intersecting set system over $[k]$ such that $|S_i| = k/2$, for all i, and $\ell \geq N$. That is, we have $S_i \cap S_j \neq \varnothing$ for all i, j. For example, we may take $\mathcal{S} = \{S \subseteq [k] \mid |S| = k/2 \ \&\ 1 \in S\}$. Observe that for this collection, when c is large enough we have

$$|\mathcal{S}| = \binom{k-1}{k/2} \geq \Omega\left(\frac{2^k}{\sqrt{k}}\right) = \Omega\left(\frac{2^{\log N + \frac{1}{2}\log\log N + c}}{\sqrt{\log N + \frac{1}{2}\log\log N + c}}\right) \geq N$$

by Stirling's inequality. Hence $\ell \geq N$.

We now give a protocol for set disjointness, assuming the existence of a good $\mathcal{A}$. Let $a, b \in \{0,1\}^N$. For all $i = 1, \ldots, \ell$, define the monomials

$$P_i = \prod_{j \in S_i} x_j \quad \text{and} \quad Q_i = \prod_{j \notin S_i} x_j.$$

Now define an arithmetic circuit

$$C(x_1, \ldots, x_k) = \left(\sum_{i=1}^{n} a_i P_i \right) \cdot \left(\sum_{i=1}^{n} b_i Q_i \right).$$

Note that C represents a homogeneous polynomial of degree k. We claim that C has a square-free monomial if and only if $DISJ_N(a, b) = 1$. This follows from the fact that the monomial $P_i \cdot Q_j$ has a square if and only if $i \neq j$.[3]

Let $C_a = \sum_{i=1}^{n} a_i P_i$ and $C_b = \sum_{i=1}^{n} b_i Q_i$. To get a communication protocol for set disjointness, the x-party uses public randomness to obtain $e_i \in \mathcal{A}$ for each x_i, computes $v = C_a(e_1, \ldots, e_k) \in \mathcal{A}$, and sends an $O(\log |\mathcal{A}|)$-bit string corresponding to v. Similarly, the y-party obtains all $e_i \in \mathcal{A}$, evaluates $w = C_b(e_1, \ldots, e_k)$, and sends w. The third party outputs *disjoint* if and only if $e = v \cdot w$ (where $e \in \mathcal{A}$ has the properties of the theorem's hypothesis).

Under the hypotheses of the theorem (and repeating the protocol $O(1)$ times to obtain a good estimate of $\Pr_{(e_1,\ldots,e_n) \in \mathcal{D}^n}[C(e_1, \ldots, e_n) = e]$), the above is a correct public-coin protocol for $DISJ_n$. It follows from Theorem 3.1 that $\log |\mathcal{A}| \geq cN$, for some constant $c > 0$. Finally, note that $N \geq \Omega\left(\frac{2^k}{\sqrt{k}}\right)$. □ □

4 Exact Parameterized Multilinear Counting

We will base our algorithm on the following lemma.

Lemma 4.1. *Let $P(x_1, \ldots, x_n)$ be a commutative polynomial with coefficients from $\mathbb{R}$, given by an arithmetic circuit C. For all $\{i_1, \ldots, i_k\} \subseteq [n]$, the coefficient of the monomial $x_{i_1} \cdots x_{i_k}$ in P can be computed in time $O^*(2^k)$. It follows that the restriction of P to its multilinear monomials of degree k can be computed in time $O^*(2^k \binom{n}{k})$.*

Proof. (Sketch) Without loss of generality, we can assume that all gates of C have two inputs. Consider the coefficient of $x_1 \cdots x_k$. Note that it is easy to find the sub-circuit C' that computes $P' = P(x_1, \ldots, x_k, 0, \ldots, 0)$. We will view C' as a circuit acting on polynomials. Let $\mathcal{R}(Q)$ denote the operation that restricts the polynomial Q to its multilinear terms. The key idea is to place an $\mathcal{R}$-gate in the output of every multiplication and addition gate of C'. Note that this does not change the coefficient of $x_1 \cdots x_k$ in the output of C', because all the terms that get deleted produce only non-multilinear terms in the rest of the computation.

Note now that the inputs of all multiplication and addition gates in C' contain only multilinear terms. There are at most 2^k such terms. It follows that the complexity of computing $\mathcal{R}(P_1 + P_2)$ is $O^*(2^k)$. It remains to show that the complexity of computing $\mathcal{R}(P_1 \cdot P_2)$ is also $O^*(2^k)$. First observe that the operation can be written as the sum of at most k^2 operations of the form $\mathcal{R}(Q_1 \cdot Q_2)$ where Q_1, Q_2 are k-variate *homogeneous* polynomials consisting of those multilinear terms in P_1, P_2 that have degrees *exactly* d and $t - d$ respectively, for $t \in [1, k]$ and $d \in [1, t]$. Let $Q_1'(z)$ and $Q_2'(z)$ be the polynomials resulting by the substitution $x_i \to z^{2^{i-1}}$ in Q_1, Q_2. The degrees of Q_1' and Q_2' are at most $2^k - 1$, hence $Q_1' \cdot Q_2'$ can be computed in $O^*(2^k)$ via the

[3]Note this is the portion of the argument where commutativity of multiplication is used. In the non-commutative setting, the condition is that the monomial $P_i \cdot Q_j$ contains two instances of the same variable in its product. This condition is precisely the same as that required for all the algorithmic applications.

Fast Fourier Transform. Now observe that $\mathcal{R}(Q_1 \cdot Q_2)$ can be recovered by the restriction of $Q_1' \cdot Q_2'$ to its terms of the form z^j where the binary form of j contains exactly t bits. □ □

This lemma gives an upper bound for the exact multilinear k-monomial counting problem. We show that this upper bound can be improved upon for polynomials of the form $P \cdot Q$, where the multilinear terms of P, Q both have total degree smaller than k. We need the following (properly adapted) theorem from the theory of error correcting codes [17][Chap. 5, Theorem 8], also used in the deterministic construction of k-wise probability spaces [2][Proposition 6.5].

Theorem 4.2. *There is a 0-1 matrix M with n columns and m rows with $2^m \leq 2(n+1)^{\lfloor k/2 \rfloor}$, such that every k columns of M are linearly independent over $\mathbb{Z}_2$. Moreover, the matrix can be constructed in $O(n^{k/2+1})$ time.*

We are now ready to prove the key theorem of this section. We use the group algebra $\mathbb{R}[\mathbb{Z}_2^k]$.

Theorem 4.3. *Let $P \cdot Q \in \mathbb{R}[X]$ be an n-variate polynomial over $\mathbb{R}$, such that P and Q contain only multilinear monomials of degree at most d. The sum of the coefficients of the multilinear k-terms in $P \cdot Q$, can be computed in time $O^*(2^d \binom{n}{d} + n^{\lfloor k/2 \rfloor})$.*

Proof. Applying the algorithm of Lemma 4.1 to the circuits for P and Q, we can assume that P, Q contain only multilinear terms. Furthermore, assume (for now) that all multilinear terms of $P \cdot Q$ have degree exactly k. Let m be the number of rows of the matrix M in Theorem 4.2. Note $m \approx \frac{k}{2} \log n$. The algorithm is as follows. Let $A : X \to \mathbb{R}[\mathbb{Z}_2^m]$ be the assignment $x_i \mapsto (\mathbf{v_0} + M_i)$, where $M_i \in \mathbb{Z}_2^m$ is the i^{th} column of M, and $\mathbf{v_0} \in \mathbb{Z}_2^m$ is the zero vector. Let $\bar{A}$ be the assignment $x_i \mapsto (\mathbf{v_0} - M_i)$. Compute $\Pi = P(A) \cdot Q(\bar{A})$ over $\mathbb{R}[\mathbb{Z}_2^m]$, and return the coefficient of $\mathbf{v_0}$ in Π.

Let us first consider the algorithm's correctness. Every term in $P \cdot Q$ is a multiple of the form $t_P(A) \cdot t_Q(\bar{A})$ where t_P, t_Q are multilinear monomials in P and Q respectively. If t_P and t_Q both contain a variable x_i, then $t_P(A) \cdot t_Q(\bar{A})$ is a multiple of $(\mathbf{v_0} + M_i)(\mathbf{v_0} - M_i)$ which is equal to 0, since $\mathbf{v_0}^2 = M_i^2 = \mathbf{v_0}$. If $t_P t_Q$ is multilinear, then because columns of M_i are independent and by Lemma 2.2 of [14], the coefficient of $\mathbf{v_0}$ in $t_P(A) \cdot t_Q(\bar{A})$ equals 1. This proves correctness.

Now we concentrate on the time complexity. We first note that we can compute each product $t(A)$ inductively, as follows. Assume we have evaluated $t'(A)$ where t' consists of the first d' factors of t. By Lemma 2.2 of [14] the product $t'(A)$ is a sum of $2^{d'}$ distinct vectors from $\mathbb{Z}_2^m$. Hence, we can store $t'(A)$ in a sparse form, where only the identities of $2^{d'}$ vectors are kept. The identity of each of the 2^m vectors of $\mathbb{Z}_2^m$ requires just m bits. Given this sparse storage, a sparse form product of the form $t'(A)(\mathbf{v_0} + M_i)$ can be evaluated fairly simply in time $O^*(2^{d'})$. Hence, $t(A)$ can be evaluated in sparse form in time $\sum_{i=1}^{d} 2^i = O(2^d)$. For the sum $P(A)$, we keep a single vector of size 2^m, which we update every time a term $t(A)$ is computed. The discussion is very similar for the assignment $\bar{A}$. Hence, $P(A)$ and $Q(A)$ can be computed in time $O(2^d \binom{n}{d})$, as there are at most $\binom{n}{d}$ monomials in P and Q.

Finally, we want $P(A) \cdot Q(\bar{A})$. Since $P(A)$ and $Q(\bar{A})$ are both elements of $\mathbb{R}[\mathbb{Z}_2^m]$, their multiplication can be performed via a Fast Fourier Transform method in time $O^*(2^m)$ [19]. Finally note that if $P \cdot Q$ contains multilinear terms of degree different than k, these can be treated by the introduction of a free indeterminate z, as in Lemma 2.1. □ □

To show an application of Theorem 4.3 we consider the m-set k-packing problem, over a universe of n elements and an instance consisting of t sets. It is clear that the number of k-packings can be counted exactly in time $O^*(\binom{t}{k})$. For large enough t, we can do better. The "algebraization" of this problem gives a simple polynomial I which is a product of sums [13], satisfying the requirements of Theorem 4.3. This gives the following corollary.

Corollary 4.4. *The number of m-set k-packings can be counted in $O^*(n^{\lceil mk/2 \rceil})$.*

It is also interesting that the above construction can be used to compute the parity of the coefficient without restrictions in the type of the circuit, as stated in the following theorem whose proof is given in the full version of the paper.

Theorem 4.5. *Let C be a circuit describing a degree-k polynomial P over $\mathbb{Z}$. The parity of the sum of the coefficients of the multilinear monomials in P can be computed in time $O^*(n^{k/2})$ and poly$(|C|)$ space.*

5 Final Remarks

Using the techniques of this paper the following theorem can also be shown.

Theorem 5.1. *Let A be a $k \times n$ matrix with entries from a commutative semiring S. The permanent of A can be computed in time and space $O^*(2^k)$, assuming a unit cost for the addition and multiplication operations over S. If $S = \mathbb{R}$, the space complexity can be reduced to poly(n).*

To the best of our knowledge the only previous related results were an $O^*(3^k)$ time, $O^*(2^k)$ space algorithm in [9], and more recently an $O^*(4^k)$ time poly(n) space algorithm for matrices over arbitrary commutative semirings [18]. The proof of this result along with other proofs omitted in this paper will appear in the full version of the paper. A number of interesting questions remain open:

- Theorem 4.2 can be used to get a weak derandomization of our earlier results, and solve more sophisticated problems. To what extent can we improve on the construction of M in this result?
- Is there an algorithm for multilinear k-monomial counting that improves over the naive upper bounds for *general* circuits?
- Can the algebraic method be applied to improve the time complexity for *approximate* counting? For this question, color coding is still the best known approach.

6 Appendix. Group algebras of $\mathbb{Z}_2^k$

Let $\mathbb{Z}_2^k$ be the group consisting of k-dimensional 0-1 vectors with the group multiplication being entry-wise addition modulo 2. The group algebra $R[\mathbb{Z}_2^k]$, where R is a ring, is the set of all linear combinations of the form $\sum_{v\in\mathbb{Z}_2^k} a_v v$ where $a_v \in K$. The addition operator of $R[\mathbb{Z}_2^k]$ is defined by

$$\sum_{v\in\mathbb{Z}_2^k} a_v v + \sum_{v\in\mathbb{Z}_2^k} b_v v = \sum_{v\in\mathbb{Z}_2^k} (a_v + b_v)v.$$

Multiplication by a scalar $\alpha \in R$ is defined by

$$\alpha \sum_{v\in\mathbb{Z}_2^k} a_v v = \sum_{v\in\mathbb{Z}_2^k} (\alpha a_v)v.$$

The multiplication operator of $R[\mathbb{Z}_2^k]$ is defined by

$$\left(\sum_{v\in\mathbb{Z}_2^k} a_v v\right) \cdot \left(\sum_{u\in\mathbb{Z}_2^k} b_u u\right) = \sum_{v,u\in\mathbb{Z}_2^k} (a_v b_u)(uv).$$

It can be verified that $R[\mathbb{Z}_2^k]$ is commutative, provided that R is commutative.

References

[1] David Aldous. Minimization algorithms and random walk on the d-cube. *Annals of Probability*, 2:403–413, 1983.

[2] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.

[3] Noga Alon, Raphael Yuster, and Uri Zwick. Color coding. *Journal of the ACM*, 42(4):844–856, 1995.

[4] Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. The fast intersection transform with applications to counting paths. *CoRR*, abs/0809.2489, 2008.

[5] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels (extended abstract). In *ICALP '08: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part I*, pages 563–574, Berlin, Heidelberg, 2008. Springer-Verlag.

[6] Jianer Chen, Songjian Lu, Sing-Hoi Sze, and Fenghui Zhang. Improved algorithms for path, matching, and packing problems. In *Proc. 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, pages 298–307, 2007.

[7] György Elekes. A geometric inequality and the complexity of computing volume. *Discrete & Computational Geometry*, 1:289–292, 1986.

[8] Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. In *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*, page 538, Washington, DC, USA, 2002. IEEE Computer Society.

[9] Tsutomu Kawabata and Jun Tarui. On complexity of computing the permanent of a rectangular matrix. *IEICE Trans. Fundamentals*, E82-A.

[10] Joachim Kneis, Alexander Langer, and Peter Rossmanith. A new algorithm for finding trees with many leaves. In *Algorithms and Computation, 19th International Symposium, ISAAC 2008*, pages 270–281.

[11] Joachim Kneis, Daniel Mölle, Stefan Richter, and Peter Rossmanith. Divide-and-color. In *WG: Graph-Theoretic Concepts in Computer Science, 32nd International Workshop*, pages 58–67, 2006.

[12] Joachim Kneis, Daniel Mölle, and Peter Rossmanith. Partial vs. complete domination: t-dominating set. In *SOFSEM: 33rd Conference on Current Trends in Theory and Practice of Computer Science*, pages 367–376, 2007.

[13] Ioannis Koutis. A faster parameterized algorithm for set packing. *Information Processing Letters*, 94(1):4–7, 2005.

[14] Ioannis Koutis. Faster algebraic algorithms for path and packing problems. In *Automata, Languages and Programming, 35th International Colloquium, ICALP*, pages 575–586, 2008.

[15] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge Univ. Press, 1996.

[16] Luis Rademacher and Santosh Vempala. Dispersion of mass and the complexity of randomized geometric algorithms. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 729–738, Washington, DC, USA, 2006. IEEE Computer Society.

[17] Neil J. Sloane and Florence Jessie MacWilliams. *The Theory of Error-Correcting Codes*. North Holland, 1983.

[18] Virginia Vassilevska and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. In *STOC, to appear*, 2009.

[19] Ryan Williams. Finding paths of length k in $O^*(2^k)$. *Information Processing Letters*, 109(6):301–338, 2009.