Short Rational Functions for Toric Algebra and Applications ⋆

J. A. De Loera [a], D. Haws [a], R. Hemmecke [a], P. Huggins [a], B. Sturmfels [b], and R. Yoshida [a]

[a] *University of California at Davis, One Shields Ave. Davis, CA 95616, USA*

[b] *University of California at Berkeley, Berkeley, CA 94720, USA*

Abstract

We encode the binomials belonging to the toric ideal I_A associated with an integral $d \times n$ matrix A using a short sum of rational functions as introduced by Barvinok (1994); Barvinok and Woods (2003). Under the assumption that d and n are fixed, this representation allows us to compute a universal Gröbner basis and the reduced Gröbner basis of the ideal I_A, with respect to any term order, in time polynomial in the size of the input. We also derive a polynomial time algorithm for normal form computations which replaces in this new encoding the usual reductions typical of the division algorithm. We describe other applications, such as the computation of Hilbert series of normal semigroup rings, and we indicate applications to enumerative combinatorics, integer programming, and statistics.

Key words: Gröbner basis, toric ideals, Hilbert series, short rational function, Barvinok's algorithm, Ehrhart polynomial, lattice points, magic cubes and squares.

1 Introduction

In this note we present polynomial-time algorithms for computing with toric ideals and semigroup rings in fixed dimension. For background on these algebraic objects and their interplay with polyhedral geometry see (Stanley, 1996; Sturmfels, 1995; Villarreal, 2001). Our results are a direct application of recent results by Barvinok and Woods (2003) on short encodings of rational generating functions (such as Hilbert series).

⋆ Research supported by NSF Grants DMS-0309694, DMS-0073815, DMS-0070774, DMS-0200729, and by NSF VIGRE Grant DMS-0135345.

Let $A = (a_{ij})$ be an integral $d \times n$-matrix and $b \in \mathbb{Z}^d$ such that the convex polyhedron $P = \{ u \in \mathbb{R}^n \ : \ A \cdot u = b \text{ and } u \geq 0 \}$ is bounded. In 1994 Barvinok gave an algorithm for counting the lattice points in P in polynomial time when $n - d$ is a constant. The input for Barvinok's algorithm is the binary encoding of the integers a_{ij} and b_i, and the output is a formula for the multivariate generating function $f(P) = \sum_{a \in P \cap \mathbb{Z}^n} x^a$ where x^a is an abbreviation of $x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}$. This long polynomial with exponentially many monomials is encoded as a much shorter sum of rational functions of the form

$$f(P) \quad = \quad \sum_{i \in I} \pm \frac{x^{u_i}}{(1 - x^{c_{1,i}})(1 - x^{c_{2,i}}) \ldots (1 - x^{c_{n-d,i}})}. \tag{1}$$

Barvinok and Woods (2003) developed a set of powerful manipulation rules for using these short rational functions in Boolean constructions on various sets of lattice points. In this note we apply their techniques to problems in combinatorial commutative algebra. Our first theorem concerns the computation of the *toric ideal* I_A of the matrix A. This ideal is generated by all binomials $x^u - x^v$ such that $Au = Av$. In what follows, we encode any set of binomials $x^u - x^v$ in n variables as the formal sum of the corresponding monomials $x^u y^v$ in $2n$ variables $x_1, \ldots, x_n, y_1, \ldots, y_n$.

Theorem 1 *Let $A \in \mathbb{Z}^{d \times n}$ and a term order $\prec_W$, specified by a matrix W, be given. Assuming that n and d are fixed, then there are algorithms, that run in polynomial time in the size of the input data, to perform the following four tasks:*

(1) Compute a short rational function G which represents the reduced Gröbner basis of the toric ideal I_A with respect to the term order $\prec_W$.
(2) Decide whether the input monomial x^a is in normal form with respect to G.
(3) Perform one step of the division algorithm modulo G.
(4) Compute the normal form of the input monomial x^a modulo the Gröbner basis G.

Our research group at UC Davis has developed a computer program, called `LattE`, which efficiently counts the lattice points in any rational polytope by computing its Barvinok representation as a rational function (1). The Gröbner basis and normal form algorithms of Theorem 1 will be implemented in a future version of `LattE`. It is important to note that **the Gröbner basis** G, which will be output by future versions of `LattE`, is **a rational function**. It is not the long list of binomials produced by all other computer algebra systems today.

Example 2 Fix the integers $n = 4$ and $d = 2$. Let us imagine we input the following data into the future `LattE`: the matrix $A = \begin{bmatrix} m & m-1 & 1 & 0 \\ 0 & 1 & m-1 & m \end{bmatrix}$, where $m \geq 3$ is an integer, and the lexicographic term order is used. The task is to compute the kernel I_A of

$$k[x_1, x_2, x_3, x_4] \ \to \ k[s, t], \quad x_1 \mapsto s^m, \ x_2 \mapsto s^{m-1}t, \ x_3 \mapsto st^{m-1}, \ x_4 \mapsto t^m.$$

Then the output produced by future LattE would consist of the rational function

$$G(x,y) \;=\; x_1\, x_4\, y_2\, y_3 \;+\; x_2\, x_4^{m-2} y_3^{m-1} \;+\; \frac{x_1\, x_3\, {y_2}^2 \left((x_1\, y_2)^{m-2} - (x_3\, y_4)^{m-2} \right)}{x_1\, y_2 - x_3\, y_4}.$$

This rational function is a polynomial whose number of terms is m and hence grows exponentially in the size of the input. Yet, the running time for computing $G(x,y)$ is bounded by a polynomial in $\log(m)$. It is an interesting exercise to perform the tasks (1), (2), and (3) in Theorem 1 for $G(x,y)$ and the monomial $x_1^m x_2^m x_3^m x_4^m$. Note that this example shows that even when n, d are fixed constants the size of a Gröbner basis can be exponentially large in the size of the input.

The proof of Theorem 1 will be given in Section 2. Special attention will be paid to the Projection Theorem (Barvinok and Woods, 2003, Theorem 1.7) since the projection of short rational functions is the most difficult step to implement. Its practical efficiency has yet to be investigated. Our proof of Theorem 1 does use the Projection Theorem, but our Proposition 10 in Section 2 shows that a *non-reduced* Gröbner basis can be computed in polynomial time without using the Projection Theorem.

In Section 3 we present what we call the *homogenized Barvinok algorithm*. This algorithm was first outlined in (De Loera et al., 2003) and it was recently implemented in LattE. Like the original version in (Barvinok, 1994), it runs in polynomial time when the dimension is fixed. But it performs much better in practice (1) when computing the Ehrhart series of polytopes with few facets but many vertices; (2) when computing the Hilbert series of normal semigroup rings. We show its effectiveness by solving the classical counting problems for 5×5 *magic squares* (all row, column and diagonal sums are equal) and $3 \times 3 \times 3 \times 3$ *magic cubes* (all line sums in the 4 possible coordinate directions and the sums along main diagonal entries are equal). Our computational results are presented in Theorem 15.

A *normal semigroup* S is the intersection of the lattice $\mathbb{Z}^n$ with a rational convex polyhedral cone in $\mathbb{R}^n$. The *Hilbert series* of S is the rational generating function $\sum_{a \in S} x^a$. Barvinok and Woods (2003) showed that this Hilbert series can be computed as a short rational generating function in polynomial time for fixed dimension. We show that this computation can be done without the Projection Theorem when the semigroup is known to be normal.

Theorem 3 *Under the hypothesis that the ambient dimension n is fixed,*

1) the Ehrhart series of a rational convex polytope given by linear inequalities can be computed in polynomial time. The Projection Theorem is not used in the algorithm.

2) The same applies to computing the Hilbert series of a normal semigroup S.

In the final section of the paper we sketch applications of our techniques to Integer Programming and Statistics. These results will be explored in detail elsewhere.

2 Computing Toric Ideals

We assume that the reader is familiar with toric ideals and Gröbner bases as presented in (Cox et al., 1992; Sturmfels, 1995). From now on and without loss of generality we will assume that $ker(A) \cap \mathbb{R}^n_{\geq 0} = \{0\}$. This condition is not restrictive because toric ideal problems can be reduced to this particular case via homogenization of the problem. Our assumption implies that for all b, the convex polyhedron $P = \{ u \in \mathbb{R}^n \ : \ A \cdot u = b \text{ and } u \geq 0 \}$ is a polytope (i.e. a bounded polytope) or the empty set. We begin by recalling some useful results of Barvinok and Woods (2003):

Lemma 4 (Theorem 3.6 in (Barvinok and Woods, 2003)) *Let S_1, S_2 be finite subsets of $\mathbb{Z}^n$, for n fixed. Let $f(S_1, x)$ and $f(S_2, x)$ be their generating functions, given as short rational functions with at most k binomials in each denominator. Then there exists a polynomial time algorithm, which, given $f(S_i, x)$, computes*

$$f(S_1 \cap S_2, x) \quad = \quad \sum_{i \in I} \gamma_i \cdot \frac{x^{u_i}}{(1 - x^{v_{i1}}) \ldots (1 - x^{v_{is}})}$$

with $s \leq 2k$, where the γ_i are rational numbers, u_i, v_{ij} are nonzero integer vectors, and I is a polynomial-size index set.

The following lemma was proved by Barvinok and Woods using Lemma 4:

Lemma 5 (Corollary 3.7 in (Barvinok and Woods, 2003)) *Let $S_1, S_2, \ldots, S_m$ be finite subsets of $\mathbb{Z}^n$, for n fixed. Let $f(S_i, x)$ for $i = 1 \ldots m$ be their generating functions, given as short rational functions with at most k binomials in each denominator. Then there exists a polynomial time algorithm, in the input size, which computes*

$$f(S_1 \cup S_2 \cup \ldots S_m, x) \quad = \quad \sum_{i \in I} \gamma_i \cdot \frac{x^{u_i}}{(1 - x^{v_{i1}}) \ldots (1 - x^{v_{is}})}$$

with $s \leq 2k$, where the γ_i are rational numbers, u_i, v_{ij} are nonzero integer vectors, and I is a polynomial-size index set. Similarly one can compute in polynomial time $f(S_1 \backslash S_2, x)$ as a short rational function.

We will use the *Intersection Lemma* and the *Boolean Operation Lemma* to extract special monomials present in the expansion of a generating function. The essential step in the intersection algorithm is the use of the *Hadamard product* (Barvinok and Woods, 2003, Definition 3.2) and a special monomial substitution. The Hadamard product is a bilinear operation on rational functions (we denote it by $*$). The computation is carried out for pairs of summands as in (1). Note that the Hadamard product $m_1 * m_2$ of two monomials m_1, m_2 is zero unless $m_1 = m_2$. We present an example of computing intersections.

Example 6 Let $S_i = \{x \in \mathbb{R} : i-2 \leq x \leq i\} \cap \mathbb{Z}$ for $i = 1, 2$. We rewrite their rational generating functions as in the proof of Theorem 3.6 in (Barvinok and Woods, 2003): $f(S_1, z) = \frac{z^{-1}}{(1-z)} + \frac{z}{(1-z^{-1})} = \frac{-z^{-2}}{(1-z^{-1})} + \frac{z}{(1-z^{-1})} = g_{11} + g_{12}$, and $f(S_2, z) = \frac{1}{(1-z)} + \frac{z^2}{(1-z^{-1})} = \frac{-z^{-1}}{(1-z^{-1})} + \frac{z^2}{(1-z^{-1})} = g_{21} + g_{22}$.

We need to compute four Hadamard products between rational functions g_{ij},whose denominators are products of binomials and whose numerators are monomials. Lemma 3.4 in Barvinok and Woods (2003) says that, these Hadamard products are essentially the same as computing the rational function, as in Equation (1), of the auxiliary polyhedron $\{(\epsilon_1, \epsilon_2) | p_1 + a_1\epsilon_1 = p_2 + a_2\epsilon_2,\ \epsilon_i \geq 0\}$. Here p_1, p_2 are the exponents of numerators of g_{ij} 's involved and a_1, a_2 are the exponents of the binomial denominators. For example, the Hadamard product $g_{11} * g_{22}$ corresponds to the polyhedron $\{(\epsilon_1, \epsilon_2) | \epsilon_2 = 4 + \epsilon_1,\ \epsilon_i \geq 0\}$. The contribution of this half line is $-\frac{z^{-2}}{(1-z^{-1})}$. We find

$$\begin{aligned} f(S_1, z) * f(S_2, z) &= g_{11} * g_{21} + g_{12} * g_{22} + g_{12} * g_{21} + g_{11} * g_{22} \\ &= \frac{z^{-2}}{(1-z^{-1})} + \frac{z}{(1-z^{-1})} - \frac{z^{-1}}{(1-z^{-1})} - \frac{z^{-2}}{(1-z^{-1})} \\ &= \frac{z - z^{-1}}{1 - z^{-1}} = 1 + z = f(S_1 \cap S_2, z). \end{aligned}$$

Another key subroutine introduced by Barvinok and Woods is the following *Projection Theorem.* In Lemmas 4, 5, and 7, the dimension n is assumed to be fixed.

Lemma 7 (Theorem 1.7 in (Barvinok and Woods, 2003)) *Assume the dimension n is a fixed constant. Consider a rational polytope $P \subset \mathbb{R}^n$ and a linear map $T : \mathbb{Z}^n \to \mathbb{Z}^k$. There is a polynomial time algorithm which computes a short representation of the generating function $f(T(P \cap \mathbb{Z}^n), x)$.*

We represent a term order $\prec$ on monomials in $x_1, \ldots, x_n$ by an integral $n \times n$-matrix W as in (Mora and Robbiano, 1998). Two monomials satisfy $x^\alpha \prec x^\beta$ if and only if $W\alpha$ is lexicographically smaller than $W\beta$. In other words, if $w_1, \ldots, w_n$ denote the rows of W, there is some $j \in \{1, \ldots, n\}$ such that $w_i\alpha = w_i\beta$ for $i < j$, and $w_j\alpha < w_j\beta$. For example, $W = I_n$ describes the lexicographic term ordering. In what follows, we will denote by $\prec_W$ the term order defined by W.

Lemma 8 *Let $S \subset \mathbb{Z}^n_+$ be a finite set of lattice points in the positive orthant. Suppose the polynomial $f(S, x) = \sum_{\beta \in S} x^\beta$ is represented as a short rational function and let $\prec_W$ be a term order. We can extract the (unique) leading monomial of $f(S, x)$ with respect to $\prec_W$ in polynomial time.*

Proof: The term order $\prec_W$ is represented by an integer matrix W. For each of the rows w_j of W we perform a monomial substitution $x_i := x_i' t^{w_{ji}}$. Note that t is a "dummy variable" that we will use to keep track of elimination. Such a monomial substitution can be computed in polynomial time by (Barvinok and Woods, 2003, Theorem 2.6). The effect is that the polynomial $f(S, x)$ gets replaced by a polynomial in the t and the $x's$. After each substitution we determine the degree in t. This is done as follows: We want to do calculations in univariate polynomials since this is faster so we consider the polynomial $g(t) = f(S, 1, t)$, where all variables except t are set to the constant one. Clearly the degree of $g(t)$ in t is the same as the degree of $f(S, x', t)$. We create the *interval polynomial* $i_{[p,q]}(t) = \sum_{i=p}^{q} t^i$ which obviously has a short rational function representation. Compute the Hadamard product of $i_{[p,q]}(t)$ with $g(t)$. This yields those monomials whose degree in the variable t lies between p and q. We will keep shrinking the interval $[p, q]$ until we find the degree. We need a bound for the degree in t of $g(t)$ to start a binary search. An upper bound U can be found via linear programming or via the estimate in Theorem 3.1 of (Lasserre, 2003) which is an easy manipulation of the numerator and denominator of the fractions in $g(t)$. It is clear that $\log(U)$ is polynomially bounded. In no more than $\log(U)$ steps one can determine the degree in t of $f(S, x, t)$ by using a standard binary search algorithm.

Let α be a polynomial-size upper bound on the highest total degree of a monomial appearing in the generating function $f(S, x)$. We can again apply linear programming or the estimate of (Lasserre, 2003) to compute such an α (just as we computed U before). Once the highest degree r in t is known, we compute the Hadamard product of $f(S, x, t)$ and $t^r h(x)$, where $h(x)$ is the rational generating function encoding the lattice points contained inside the box $[0, \alpha]^n$. This will capture only the desired monomials. Then compute the limit as t approaches 1. This can be done in polynomial time using residue techniques. The limit represents the subseries $H(S, x) = \sum_{\beta \cdot w_j = r} x^\beta$. Repeat the monomial and highest degree search for the row w_{j+1},w_{j+2}, etc. Since $\prec_W$ is a term order, after doing this n times we will have only one single monomial left, the desired leading monomial. □

One has to be careful when using earlier Lemmas (especially the projection theorem) that the sets in question are finite. We need the following well-known bound:

Lemma 9 (Lemma 4.6 and Theorem 4.7 in (Sturmfels, 1995)) *Let M be equal to $(d + 1)(n - d)D(A)$, where A is a $d \times n$ integral matrix and $D(A)$ is the biggest $d \times d$ subdeterminant of A in absolute value. Any entry of an exponent vector of any reduced Gröbner basis for the toric ideal I_A is less than M.*

Proposition 10 *Let $A \in \mathbb{Z}^{d \times n}$, $W \in \mathbb{Z}^{n \times n}$ specifying a term order $\prec_W$. Assume that d and n are fixed.*

1) There is a polynomial time algorithm to compute a short rational function G which represents a universal Gröbner basis of I_A.

2) Suppose we are given the term order $\prec_W$ and a short rational function encoding a finite set of binomials $x^u - x^v$ now expressed as the sum of monomials $\sum x^u y^v$. Assume M is an integer positive bound on the degree of any variable for any of the monomials. One can compute in polynomial time a short rational function encoding only those binomials $x^u - x^v$ that satisfy $x^v \prec_W x^u$.

3) Suppose we are given a sum of short rational functions $f(x)$ which is identical, in its monomial expansion, to a single monomial x^a. Then in polynomial time we can recover the (unique) exponent vector a.

Proof: 1) Set $M = (d+1)(n-d)D(A)$ where $D(A)$ is again the largest absolute value of any $d \times d$-subdeterminant of A. Using Barvinok's algorithm in (Barvinok, 1994), we compute the following generating function in $2n$ variables:

$$G(x,y) \quad = \quad \sum \{\, x^u y^v \;:\; Au = Av \text{ and } 0 \leq u_i, v_i \leq M \,\}.$$

This is the sum over all lattice points in a rational polytope. Lemma 9 above implies that the toric ideal I_A is generated by the finite set of binomials $x^u - x^v$ corresponding to the terms $x^u y^v$ in $G(x,y)$. Moreover, these binomials are a universal Gröbner basis of I_A.

2) Denote by w_i the i-th row of the matrix W which specifies the term order. Suppose we are given a short rational generating function $G_0(x,y) = \sum x^u y^v$ representing a set of binomials $x^u - x^v$ in I_A, for instance $G_0 = G$ in part (1). In the following steps, we will alter the series so that a term $x^u y^v$ gets removed whenever u is not bigger than v in the term order $\prec_W$. Starting with $H_0 = G_0$, we perform Hadamard products with short rational functions $f(S; x, y)$ for $S \subset \mathbb{Z}^{2n}$.

Set $H_i = H_{i-1} * f(\{(u,v) : w_i u = w_i v,\ 0 \leq u_j, v_j \leq M,\ j = 1 \ldots n\})$, and $G_i = H_{i-1} * f(\{(u,v) : w_i u \geq w_i v + 1,\ 0 \leq u_j, v_j \leq M\ j = 1 \ldots n\})$. All monomials $x^u y^v \in G_j$ have the property that $w_i u = w_i v$ for $i < j$, $w_j u > w_j v$, and thus $v \prec_W u$. On the other hand, if $v \prec_W u$ then there is some j such that $w_i u = w_i v$ for $i < j$, $w_j u > w_j v$, and we can conclude that $x^u y^v \in G_j$. Note that $H = G_1 \cup G_2 \cup \ldots \cup G_n$ is actually a disjoint union of sets. The rational function that gives the union, can be computed in polynomial time by Lemma 5. In practice, the rational generating functions representing the G_i's can be simply added together. The short rational function H encodes exactly those binomials in G_0 that are correctly ordered with respect to $\prec_W$. We have proved our claim since all of the above constructions can be done in polynomial time.

3) Given $f(x)$ we can compute in polynomial time the partial derivative $\partial f(x)/\partial x_i$. This puts the exponent of x_i as a coefficient of the unique monomial. Computing the derivative can be done in polynomial time by the quotient and product derivative rules. Each time we differentiate a short rational function of the form

$$\frac{x^{b_i}}{(1 - x^{c_{1,i}})(1 - x^{c_{2,i}}) \ldots (1 - x^{c_{n,i}})}$$

we add polynomially many (binomial type) factors to the numerator. The factors in the numerators should be expanded into monomials to have again summands in short rational canonical form $\frac{x^{b_i}}{(1-x^{c_{1,i}})(1-x^{c_{2,i}})\ldots(1-x^{c_{n,i}})}$. Note that at most 2^n monomials appear each time (n is a constant). Finally, if we take the limit when all variables x_i go to one we will get the desired exponent. $\square$

Example 11 Using `LattE` we compute the set of all binomials of degree less than or equal 10000 in the toric ideal I_A of the matrix $A = \begin{bmatrix} 1\ 1\ 1\ 1 \\ \\ 0\ 1\ 2\ 3 \end{bmatrix}$. This matrix represents the *Twisted Cubic Curve* in algebraic geometry. We find that there are exactly 195281738790588958143425 such binomials. Each binomial is encoded as a monomial $x_1^{u_1}x_2^{u_2}x_3^{u_3}x_4^{u_4}y_1^{v_1}y_2^{v_2}y_3^{v_3}y_4^{v_4}$. The computation takes about 40 seconds. The output is a sum of 538 simple rational functions of the form a monomial divided by a product such as $\left(1-\frac{x_3y_4}{x_1y_2}\right)\left(1-\frac{x_1x_4y_2}{x_3}\right)(1-x_1y_1)(1-x_1x_3y_2{}^2)(1-x_3y_3)(1-x_2y_2)$. $\square$

Proof of Theorem 1

The proof of Theorem 1 will require us to project and intersect sets of lattice points represented by rational functions. We cannot, in principle, do those operations for *infinite* sets of lattice points. Fortunately, in our setting it is possible to restrict our attention to finite sets. Besides Lemma 9 for the size of exponents of Gröbner bases, we need a bound for the exponents of normal form monomials:

Lemma 12 *Let x^u be the normal form of x^a with respect to the reduced Gröbner basis G of a toric ideal I_A for the term order $\prec_W$ (associated to the matrix W). Every coordinate of u is bounded above by $L = (d+1)nD(A)\tilde{a}$, where $D(A)$ is the biggest subdeterminant of A in absolute value, $\tilde{a}$ denotes the largest coordinate of the exponent vector a.*

Proof: We note that u is a point in the (bounded) convex polytope defined by the following inequalities in v: $Av = Aa$, and $v \geq 0$ (it is forced to be bounded for all a because we assumed $ker(A) \cap \mathbb{R}^n_{\geq 0} = \{0\}$). Thus each coordinate of u is bounded above by the corresponding coordinate of some vertex of this polytope. Let v be such a vertex. The non-zero entries of v are given by $B^{-1}Aa$ where B is a maximal non-singular square submatrix of A. Clearly, each entry of $B^{-1}A$ is bounded above by $D(A)$, and hence each entry of v is bounded above by L. We conclude that L is an upper bound for the coordinates of u. $\square$

Proof of Theorem 1: Proposition 10 gives a Gröbner basis for the toric ideal I_A in polynomial time. We now show how to get the reduced Gröbner basis from it in three easy polynomial time steps. The input is the the $d \times n$ integral matrix A and the $n \times n$ term order matrix W. The algorithm for claim (1) of Theorem 1 has three steps:

Step 1. Let M be equal to $(d+1)(n-d)D(A)$, as in Lemma 9, for given input matrix A. As in Proposition 10, compute the generating function which encodes binomials of highest degree M on variables that generate I_A:

$$f(x,y) \quad = \quad \sum\{\, x^u y^v \ : \ Au = Av \text{ and } 0 \leq u_j, v_j \leq M \text{ for } j = 1 \dots n\},$$

Next we wish to remove from $f(x,y)$ all incorrectly ordered binomials (i.e. those monomials $x^u y^v$ with $u \prec_W v$ instead of the other way around). We do this using part 2 of Proposition 10. We obtain from it a collection $G_0, G_1, \ldots, G_n$ of rational functions encoding disjoint sets of lattice points. We call $\bar{f}(x,y)$ the generating function representing the union of $G_0, \ldots, G_n$. This can be computed in polynomial time by adding the rational functions of the G_i together (since they are disjoint). The reader should notice that this updated $\bar{f}(x,y)$ contains only those monomials of the old $f(x,y)$ that are now correctly ordered.

Let $g_i(x)$ be the projection of G_i onto the first group of x-variables and denote by $g(x)$ the rational function that represents the union of the $g_i(x)$. The rational function $g(x)$ can be computed in polynomial time by the projection theorem of Barvinok-Woods, i.e. Lemma 7. It is important to note that $g(x)$ is the result of projecting $\bar{f}(x,y)$ into the first group of variables. This is true because a linear projection of the union of disjoint lattice point sets (i.e. those represented by G_i) equals the union of the projections of the individual sets. In conclusion, $g(x)$ is the sum over all non-standard monomials having degree at most M in any variable.

Step 2. Write $r(x,M) = \prod_{i=1}^{n} \left(\frac{1}{1-x_i} + \frac{x_i^M}{1-x_i^{-1}}\right)$ for the generating function of all x-monomials having degree at most M in any variable. Note that this is a large, but finite, set of monomials. We compute the following Hadamard product of n rational functions in x and Boolean complements (we denote them by $\backslash$):

$$\Big(r(x,M)\backslash x_1 \cdot g(x)\Big) * \Big(r(x,M)\backslash x_2 \cdot g(x)\Big) * \cdots * \Big(r(x,M)\backslash x_n \cdot g(x)\Big).$$

This is the generating function over those monomials all of whose proper factors are standard modulo the toric ideal I_A and whose degree in any variable is at most M.

Step 3. Let $h(x,y)$ denote the ordinary product of the resulting rational function from Step 2 with

$$r(y,M)\backslash g(y) \quad = \quad \sum\{\, y^v \ : \ v \text{ standard monomial modulo } I_A \text{ of highest degree } M\}.$$

Thus $h(x,y)$ is the sum of all monomials $x^u y^v$ such that x^v is standard and x^u is a monomial all of whose proper factors are standard monomials modulo the toric ideal I_A and, finally, the highest degree in any variable is at most M.

Compute the Hadamard product $G(x,y) := \bar{f}(x,y) * h(x,y)$. This is a short rational representation of a polynomial, namely, it is the sum over all monomials $x^u y^v$ such that the binomial $x^u - x^v$ is in the reduced Gröbner basis of I_A with respect to W and $x^v \prec_W x^u$. This completes the proof of the first claim of Theorem 1.

We next give the algorithm that solves claims 2 and 3 of Theorem 1. This will be done in four steps (1,2,3,4). We are given an input monomial x^a for which we aim to determine whether it is already in normal form.

Step 4 Perform Steps 1,2,3. Let $G(x,y)$ be the reduced Gröbner basis of I_A with respect to the term order W encoded by the rational function obtained at the end of Step 3. Let $r(x,\tilde{a})$ be, as before, the rational function of all monomials having degree less than $\tilde{a}$ on any variable. Thus $G'(x,y) = r(x,\tilde{a}) \cdot G(x,y)$ consists of all monomials of the form $x^s(x^u y^v)$ where $x^u - x^v$ is a binomial of the Gröbner basis and where $0 \leq s \leq \tilde{a}$. Thus $x^s x^u$ is a monomial divisible by some leading term of the Gröbner basis.

Given a monomial x^a consider $b(x,y)$, the rational function representing the lattice points of $\{(u,v) : u = a, 0 \leq v_j \leq L \text{ for } j = 1 \ldots n\}$. The Hadamard product $\bar{G}(x,y) = G'(x,y) * b(x,y)$ is computable in polynomial time and corresponds to those binomials in $G(x,y)$ that can reduce x^a. If $\bar{G}(x,y)$ is empty then x^a is in normal form already, otherwise we use Lemma 8 and part 3 of Proposition 10 to find an element $x^u y^v \in \bar{G}(x,y)$ and reduce x^a to x^{a-u+v}. We may assume that the coefficient of the encoded monomial is one, because we can compute the coefficient in polynomial time using residue techniques, and divide our rational function through by it.

Finally, we present the algorithm for claim 4 in Theorem 1 in four steps (1,2,5,6). A curious byproduct of representing Gröbner bases with short rational functions is that the reduction to normal form need not be done by dividing several times anymore.

Step 5. Redo all the calculations of the Steps 1,2,3 using $L = (d+1)nD(A)\tilde{a}$ from Lemma 12 instead of M. Note that the logarithm of L is still bounded by a polynomial in the size of the input data (A, W, a). Let $\bar{f}(x,y)$ and $g(x)$ from Step 1,2 (now recomputed with the new bound L) and compute the Hadamard product

$$H(x,y) \quad := \quad \bar{f}(x,y) * \Big(r(x,L) \cdot (r(y,L) \backslash g(y)) \Big).$$

This is the sum over all monomials $x^u y^v$ where x^v is the normal form of x^u and highest degree of x^u on any variable is L. Since we took a high enough degree, by Lemma 12, the monomial $x^a y^p$, with x^p the normal form of x^a, is sure to be present.

Step 6. We use $H(x,y)$ as one would use a traditional Gröbner basis of the ideal I_A. The normal form of a monomial x^a is computed by forming the Hadamard product $H(x,y) * (x^a \cdot r(y,L))$. Since this is strictly speaking a sum of rational functions equal to a single monomial, applying Part 3 of Proposition 10 completes the proof of Theorem 1. □

Remark: As an alternative to our algorithm for claim 2 of Theorem 1, analytic calculations may now be used to decide whether x^a is in reduced normal form or not. Take $G(x,y)$ as before and compute $F(x) = G(x,1)$. This can be done using monomial substitution (Barvinok and Woods, 2003, Section 2). Next compute the integral

$$\frac{1}{(2\pi i)^n} \int\limits_{|x_1|=\epsilon_1} \cdots \int\limits_{|x_d|=\epsilon_d} \frac{(x_1^{-a_1} \cdots x_n^{-a_n}) F(x)}{(1-x_1)\cdots(1-x_n)} \, dx \ .$$

Here $0 < \epsilon_1, \ldots, \epsilon_d < 1$ are different numbers such that we can expand all the $\frac{1}{1-x_k}$ into power series about 0. From basic facts in complex analysis and generating functions, the integral equals the number of ways that the monomial x^a can be written in terms of the leading monomials of the Gröbner bases G. It would be interesting to explore the practical aspects of this approach.

3 Computing Normal Semigroup Rings

We observed in (De Loera et al., 2003) that a major practical bottleneck of the original Barvinok algorithm in (Barvinok, 1994) is the fact that a polytope may have too many vertices. Since originally one visits each vertex to compute a rational function at each tangent cone, the result can be costly. For example, the well-known polytope of semi-magic cubes in the $4 \times 4 \times 4$ case has over two million vertices, but only 64 linear inequalities describe the polytope. In such cases we propose a homogenization of Barvinok's algorithm working with a single cone.

There is a second motivation for looking at the homogenization. Barvinok and Woods (Barvinok and Woods, 2003) proved that the Hilbert series of semigroup rings can be computed in polynomial time. We show that for *normal semigroup rings* this can be done simpler and more directly, without using the Projection Theorem.

Given a rational polytope P in $\mathbb{R}^{n-1}$, we set $i(P,m) = \#\{z \in \mathbb{Z}^{n-1} : z \in mP\}$. The *Ehrhart series* of P is the generating function $\sum_{m=0}^{\infty} i(P,m)t^m$.

Algorithm 13 (Homogenized Barvinok algorithm)

Input: *A full-dimensional, rational convex polytope P in $\mathbb{R}^{n-1}$ specified by linear inequalities and linear equations.*

Output: *The Ehrhart series of P.*

(1) Place the polytope P into the hyperplane defined by $x_n = 1$ in $\mathbb{R}^n$. Let K be the n-dimensional cone over P, that is, $K = \textit{cone}(\{(p,1) : p \in P\})$.
(2) Compute the polar cone K^*. The normal vectors of the facets of K are exactly the extreme rays of K^*. If the polytope P has far fewer facets then vertices, then the number of rays of the cone K^* is small.

(3) Apply Barvinok's decomposition of K^* into unimodular cones. Polarize back each of these cones. It is known, e.g. Corollary 2.8 in (Barvinok and Pommersheim, 1999), that by dualizing back we get a unimodular cone decomposition of K. All these cones have the same dimension as K. Retrieve a signed sum of multivariate rational functions which represents the series $\sum_{a \in K \cap \mathbb{Z}^n} x^a$.
(4) Replace the variables x_i by 1 for $i \leq n-1$ and output the resulting series in $t = x_n$. This can be done using the methods in (De Loera et al., 2003).

We recall that one of the key steps in Barvinok's algorithm is that any cone can be decomposed as the signed sum of (indicator functions of) unimodular cones.

Theorem 14 (see (Barvinok, 1994)) *Fix the dimension n. Then there exists a polynomial time algorithm which decomposes a rational polyhedral cone $K \subset \mathbb{R}^n$ into unimodular cones K_i with numbers $\epsilon_i \in \{-1, 1\}$ such that*

$$f(K) \;=\; \sum_{i \in I} \epsilon_i f(K_i), \qquad |I| < \infty.$$

Originally, Barvinok had pasted together such formulas, one for each vertex of a polytope, using a result of Brion. The point is that this can be avoided:

Proof of Theorem 3: We first prove part (1). The algorithm solving the problems is Algorithm 13. Steps 1 and 2 are polynomial when the dimension is fixed. Step 3 follows from Theorem 14. We require a special monomial substitution, possibly with some poles. This can be done in polynomial time by (Barvinok and Woods, 2003).

Part (2): Recall the characterization of the integral closure of the semigroup S as the intersection of a pointed polyhedral cone with the lattice $\mathbb{Z}^n$. From this it is clear that Algorithm 13 computes the desired Hilbert series, with the only modification that the input cone K is given by the rays of the cone and not the facet inequalities. The rays are the generators of the monomial algebra. But, in fixed dimension, one can transfer from the extreme rays representation of the cone to the facet representation of the cone in polynomial time. □

Each pointed affine semigroup $S \subset \mathbb{Z}^n$ can be *graded*. This means that there is a linear map $deg : S \to \mathbb{N}$ with $deg(x) = 0$ if and only if $x = 0$. Given a pointed graded affine semigroup define S_r to be the set of all degree r elements, i.e. $S_r = \{x \in S : deg(x) = r\}$. The *Hilbert series* of S is the formal power series $H_S(t) = \sum_{k=0}^{\infty} \#(S_r) t^r$. Algebraically, this is just the Hilbert series of the semigroup ring $\mathbb{C}[S]$. It is a well-known property that H_S is represented by a rational function of the form

$$\frac{Q(t)}{(1-t^{d_1})(1-t^{d_2})\ldots(1-t^{d_n})}$$

where $Q(t)$ is a polynomial of degree less than $d_1 + \ldots + d_n$ (see Chapter 4 (Stanley, 1997)). Several other methods had been tried to compute the Hilbert series explicitly (see (Ahmed et al., 2003) for references). One of the most well-known challenges was that of counting the 5×5 magic squares of magic sum n. Similarly several authors had tried before to compute the Hilbert series of the $3 \times 3 \times 3 \times 3$ magic cubes. It is not difficult to see this is equivalent to determining an Ehrhart series. Using Algorithm 13 we finally present the solution, which had been inaccessible using Gröbner bases methods. For comparison, the reader familiar with Gröbner bases computations should be aware that the 5×5 magic squares problem required a computation of a Gröbner bases of a toric ideal of a matrix A with 25 rows and over 4828 columns. Our attempts to handle this problem with `CoCoA` and `Macaulay2` were unsuccessful. We now give the numerator and then the denominator of the rational functions computed with the software `LattE`:

Theorem 15

The generating function $\sum_{n \geq 0} f(n)t^n$ *for the number* $f(n)$ *of* 5×5 *magic squares of magic sum* n *is given by the rational function* $p(t)/q(t)$ *with numerator*

$p(t) = t^{76}+28\,t^{75}+639\,t^{74}+11050\,t^{73}+136266\,t^{72}+1255833\,t^{71}+9120009\,t^{70}+54389347\,t^{69}+ 274778754\,t^{68} + 1204206107\,t^{67} + 4663304831\,t^{66} + 16193751710\,t^{65} + 51030919095\,t^{64} + 147368813970\,t^{63} + 393197605792\,t^{62} + 975980866856\,t^{61} + 2266977091533\,t^{60} + 4952467350549\,t^{59} + 10220353765317\,t^{58} + 20000425620982\,t^{57} + 37238997469701\,t^{56} + 66164771134709\,t^{55} + 112476891429452\,t^{54} + 183365550921732\,t^{53} + 287269293973236\,t^{52} + 433289919534912\,t^{51}+630230390692834\,t^{50}+885291593024017\,t^{49}+1202550133880678\,t^{48}+ 1581424159799051\,t^{47} + 2015395674628040\,t^{46} + 2491275358809867\,t^{45} + 2989255690350053\,t^{44} + 3483898479782320\,t^{43} + 3946056312532923\,t^{42} + 4345559454316341\,t^{41} + 4654344257066635\,t^{40} + 4849590327731195\,t^{39} + 4916398325176454\,t^{38} + 4849590327731195\,t^{37} + 4654344257066635\,t^{36} + 4345559454316341\,t^{35} + 3946056312532923\,t^{34} + 3483898479782320\,t^{33} + 2989255690350053\,t^{32} + 2491275358809867\,t^{31} + 2015395674628040\,t^{30} + 1581424159799051\,t^{29} + 1202550133880678\,t^{28} + 885291593024017\,t^{27} + 630230390692834\,t^{26}+433289919534912\,t^{25}+287269293973236\,t^{24}+183365550921732\,t^{23}+ 112476891429452\,t^{22} + 66164771134709\,t^{21} + 37238997469701\,t^{20} + 20000425620982\,t^{19} + 10220353765317\,t^{18} + 4952467350549\,t^{17} + 2266977091533\,t^{16} + 975980866856\,t^{15} + 393197605792\,t^{14}+147368813970\,t^{13}+51030919095\,t^{12}+16193751710\,t^{11}+4663304831\,t^{10}+ 1204206107\,t^{9}+274778754\,t^{8}+54389347\,t^{7}+9120009\,t^{6}+1255833\,t^{5}+136266\,t^{4}+11050\,t^{3}+ 639\,t^{2} + 28\,t + 1$ *and denominator*

$$q(t) = \left(t^2-1\right)^{10}\left(t^2+t+1\right)^{7}\left(t^7-1\right)^{2}\left(t^6+t^3+1\right)\left(t^5+t^3+t^2+t+1\right)^{4}(1-t)^{3}\left(t^2+1\right)^{4}.$$

The generating function $\sum_{n \geq 0} f(n)t^n$ *for the number* $f(n)$ *of* $3 \times 3 \times 3 \times 3$ *magic cubes with magic sum* n *is given the rational function* $r(t)/s(t)$ *where*

$r(t) = t^{54} + 150\,t^{51} + 5837\,t^{48} + 63127\,t^{45} + 331124\,t^{42} + 1056374\,t^{39} + 2326380\,t^{36} + 3842273\,t^{33}+5055138\,t^{30}+5512456\,t^{27}+5055138\,t^{24}+3842273\,t^{21}+2326380\,t^{18}+1056374\,t^{15}+ 331124\,t^{12} + 63127\,t^{9} + 5837\,t^{6} + 150\,t^{3} + 1$ *and*

$$s(t) = \left(t^3+1\right)^{4}\left(t^{12}+t^9+t^6+t^3+1\right)\left(1-t^3\right)^{9}\left(t^6+t^3+1\right).$$

4 Applications

As explained in Chapter 5 of the book Sturmfels (1995), Gröbner bases can be useful in the context of integer programming, serving as universal test sets of families of integer programs, and in statistics, where they can be used as the Markov basis for sampling from conditional distributions (e.g. on contingency tables). The fact that we can compute Gröbner bases and normal forms in polynomial time (under certain hypotheses) implies the following well-known result (details will appear elsewhere).

Corollary 16 *Let $A \in \mathbb{Z}^{d\times n}$, $b \in \mathbb{Z}^d$, $W \in \mathbb{Z}^n$. Assume that d and n are fixed. There is a polynomial time algorithm to solve the integer programming problem $min_{x\in P\cap\mathbb{Z}^n} Wx$ where $P(b) = \{x|Ax = b, x \geq 0\}$.*

Sketch of proof: Make the cost vector W into a term order by breaking ties of the order $m_1 > m_2$ if $Wm_1 > Wm_2$. You can do this via lexicographic ordering. From Chapter 5 of Sturmfels (1995) the integral optimum of P can be obtained from the Gröbner basis obtained in Theorem 1 and then computing the normal form of a monomial x^u, $Au = b$ with respect to the Gröbner basis. Since both steps can be done efficiently the corollary follows.

Another application is to the uniform sampling of lattice points inside polyhedra of the form $P(b) = \{x \in \mathbb{R}^d|Ax = b, x \geq 0\}$. Let M be a finite set such that $M \subset \{x \in \mathbb{Z}^d|Ax = 0\}$. We define the graph G_b such that its nodes are all the lattice points inside of $P(b)$ and there is an undirected edge between a node u and a node v iff $u - v \in M$. In general this graph may not be connected for arbitrary choices of M. We say M is a *Markov basis* if G_b is a connected graph for all b.

Corollary 17 *Let $A \in \mathbb{Z}^{d\times n}$, where d and n are fixed. There is a polynomial time algorithm to compute a multivariate rational generating function for a Markov basis M associated to A. This is presented as a short sum of rational functions.*

We conclude with a numerical example from statistics. Ian Dinwoodie communicated to us the problem of counting all 7×7 contingency tables whose entries are nonnegative integers x_i, with diagonal entries multiplied by a constant as presented in Table 1. The row sums and column sums of the entries are given there too. Using `LattE` we obtained the exact answer *8813835312287964978894*.

References

Ahmed, M., De Loera, J.A., and Hemmecke, R. *Polyhedral cones of magic cubes and squares,* in "New Directions in Combinatorial Geometry", the Goodman-Pollack Festschrift (eds. B. Aronov et al.), Springer Verlag, 2003, 25–41.

Barvinok, A.I. *Polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed*, Math. of Operations Research 19, 1994, 769–779.

$2x_1$	x_2	x_3	x_4	x_5	x_6	x_7	205
x_2	$2x_8$	x_9	x_{10}	x_{11}	x_{12}	x_{13}	600
x_3	x_9	$2x_{14}$	x_{15}	x_{16}	x_{17}	x_{18}	61
x_4	x_{10}	x_{15}	$2x_{19}$	x_{20}	x_{21}	x_{22}	17
x_5	x_{11}	x_{16}	x_{20}	$2x_{23}$	x_{24}	x_{25}	11
x_6	x_{12}	x_{17}	x_{21}	x_{24}	$2x_{26}$	x_{27}	152
x_7	x_{13}	x_{18}	x_{22}	x_{25}	x_{27}	$2x_{28}$	36
205	600	61	17	11	152	36	1082

Table 1
The conditions for retinoblastoma RB1-VNTR genotype data from the Ceph database.

Barvinok, A.I. and Pommersheim, J. *An algorithmic theory of lattice points in polyhedra*, in: *New Perspectives in Algebraic Combinatorics* (Berkeley, CA, 1996-1997), 91–147, Math. Sci. Res. Inst. Publ. 38, Cambridge Univ. Press, 1999.

Barvinok, A.I. and Woods, K. *Short rational generating functions for lattice point problems*, Journal of the American Mathematical Society, 16, 2003, 957–979.

Cox, D., Little, J., and O'Shea D. *Ideals, Varieties and Algorithms*, Undergraduate Texts in Mathematics, Springer, New York, 1992.

De Loera, J.A, Hemmecke, R., Tauzer, J., and Yoshida, R. *Effective lattice point counting in rational convex polytopes*, to appear in the Journal of Symbolic Computation.

Lasserre, J.B. *Integer programming, Barvinok's counting algorithm and Gomory relaxations*, Oper. Res. Letters 32, 2003, 133-137.

Mora, T. and Robbiano, L. *The Gröbner fan of an ideal,* J. Symbolic Comput. 6, 1988, no. 2-3, 183–208.

Stanley, R.P. *Combinatorics and Commutative Algebra*, Second edition. Progress in Mathematics, 41. Birkhäuser, Boston, 1996.

Stanley, R.P. *Enumerative Combinatorics*, Volume I, Cambridge, 1997.

Sturmfels, B. *Gröbner Bases and Convex Polytopes*, University Lecture Series, vol. 8, AMS, Providence RI, 1995.

Villarreal, R. H. *Monomial Algebras*, Monographs and Textbooks in Pure and Applied Mathematics, 238. Marcel Dekker, Inc., New York, 2001.