A Tutorial on Rational Generating Functions

Christopher Thomas Ryan
Sauder School of Business
University of British Columbia
2053 Main Mall, Vancouver, BC, Canada, V6T 1Z2
chris.ryan@sauder.ubc.ca

Albert Xin Jiang
Department of Computer Science
University of British Columbia
2366 Main Mall, Vancouver, BC, Canada, V6T 1Z4
jiang@cs.ubc.ca

Kevin Leyton-Brown
Department of Computer Science
University of British Columbia
2366 Main Mall, Vancouver, BC, Canada, V6T 1Z4
kevinlb@cs.ubc.ca

March 30, 2010

1 Rational generating functions

Many of the positive results in this paper are derived from results from the literature on rational generating functions and in particular from the method of Barvinok and Woods [2]. Generating functions have been applied in an analogous fashion in a variety of other contexts, including social choice theory [7], discrete optimization [6], combinatorics [4], and compiler optimization [8]. We here provide a brief and selective overview of this theory. We have two aims: to introduce some machinery that is used in proving our main theorem, and to invite other researchers to use these methods in future work.

Generating functions offer the key benefit of compactly representing exponential-cardinality sets of integer points while efficiently supporting the computational operations of counting and enumerating points in the set. We demonstrate the essence of the approach in the following simple example. Consider the set of integers on the line between 0 and n. We can represent these points by associating an exponent of a complex variable ξ with every integer $x \in [0, n]$. The choice of variable ξ plays an important role in later analysis, where residue calculus can be used to extract important information. Using this encoding we can represent the integers in the interval $[0, n]$ as the exponents in the polynomial expression

$$\sum_{x=0}^{n} \xi^x, \tag{1}$$

called a *generating function representation*. So far we have not gained much: there are exponentially many terms in (1) (in terms of the binary encoding size of n), just as there are in an explicit listing of the numbers $0, 1, \dots, n$. However, we can also write the expression as

$$\sum_{x=0}^{n} \xi^x = \frac{1}{1-\xi} - \frac{\xi^{n+1}}{1-\xi}. \tag{2}$$

Thus, we have written the *long* sum (1) as a *short* sum of two rational functions, called a *rational generating function representation*. The encoding length of this new representation is now *polynomial* in the encoding length of n. Not only does this representation appeal because of its compact size, but it also offers computational benefits. Observe that we can compute the cardinality of our set by setting $\xi = 1$ and then evaluating the sum. Working with Equation (1), this requires an exponential number of arithmetic operations. However, we can obtain the same result by working with (2) (letting $\xi \to 1$ and using L'Hôpital's rule), thereby performing exponentially-fewer arithmetic operations.

We are often interested in sets arising from unions, intersections and differences of simpler sets of integers. A basic illustration is as follows: suppose we have the set $\{0, \dots, n\}$ represented by the rational generating function $\frac{1-\xi^{n+1}}{1-\xi}$, as well as the set $\{n+1, \dots, 2n\}$ represented by the rational generating function $\frac{\xi^{n+1}-\xi^{2n+1}}{1-\xi}$. A rational generating function representing the *union* of these two *disjoint* sets can by found by *summing* the representations of $\{0, \dots, n\}$ and $\{n+1, \dots, 2n\}$:

$$\frac{1-\xi^{n+1}}{1-\xi} + \frac{\xi^{n+1}-\xi^{2n+1}}{1-\xi} = \frac{1-\xi^{2n+1}}{1-\xi}.$$

It is straightforward to verify that $\frac{1-\xi^{2n+1}}{1-\xi}$ is a rational generating function encoding of the union $\{0, \dots, 2n\}$. A more general result obtains when combining sets that are not disjoint (Lemma 2 below). Observe that the complication in this case is that we cannot simply sum the two generating function representations, as this will yield nonunitary weights on the terms in the new generating function that correspond to points lying in the intersection of the two sets.

Working with rational generating functions is of course unnecessary in the simple setting we have discussed so far, but it becomes useful when extended to deal with more general sets of integer points in higher dimensions. A key milestone was the work of Barvinok [1], who introduced a polynomial-time algorithm for representing as a rational generating function the integer points inside of a *rational polytope* $P \subseteq \mathbb{R}^m$ given by an inequality system $\{\mathbf{x} \in \mathbb{R}^m : M\mathbf{x} \leq \mathbf{b}\}$, provided the dimension m is fixed. Note that representing the integer set $P \cap \mathbb{Z}^m$ by the inequality description of P gives little hint as to its cardinality. However, as in our simple example, we shall see that a generating function representation allows us to count the number of integer points in P exactly in polynomial time.

Now we briefly describe Barvinok's algorithm and its useful extensions and applications. Consider the *generating function* of the lattice point set $P \cap \mathbb{Z}^m$, which is defined as

$$g(P \cap \mathbb{Z}^m; \boldsymbol{\xi}) = \sum_{\mathbf{x} \in P \cap \mathbb{Z}^m} \boldsymbol{\xi}^{\mathbf{x}} \\ = \sum_{\mathbf{x} \in P \cap \mathbb{Z}^m} \xi_1^{x_1} \cdots \xi_m^{x_m}. \quad (3)$$

Note that each lattice point $\mathbf{x}$ in P is mapped to the exponent of a monomial $\boldsymbol{\xi}^{\mathbf{x}}$ in $g(P \cap \mathbb{Z}^m; \boldsymbol{\xi})$.

Lemma 1 (Barvinok's Theorem [1]) *Let P be a polytope in $\mathbb{R}^m$ and $S = P \cap \mathbb{Z}^m$ with generating function $g(S, \boldsymbol{\xi})$ given by (3) which encodes the lattice points inside P. Then, there exists an algorithm which computes an equivalent representation of the form*

$$g(S; \boldsymbol{\xi}) = \sum_{i \in I} \gamma_i \frac{\boldsymbol{\xi}^{\mathbf{c}_i}}{\prod_{k=1}^{m} (1 - \boldsymbol{\xi}^{\mathbf{d}_{ik}})}, \quad (4)$$

where I is a polynomial-size index set and all data are integer. A formula of this type is called a short rational generating function. *The algorithm runs in* polynomial time *when the dimension m is fixed.*

Note that the number of binomial terms in the denominator of each rational term is m and thus fixed when the dimension is fixed. When a lattice point set S is expressed in the form of (4) we refer to $g(S; \boldsymbol{\xi})$ as its *Barvinok encoding*. In the algorithms that follow, when a set S is given as input or output using its Barvinok encoding $g(S, \boldsymbol{\xi})$, the encoding size is the binary encoding of the integer vectors $\gamma, \mathbf{c}_i, \mathbf{d}_{i1}, \ldots, \mathbf{d}_{im}$ for $i \in I$.

It is important to note that Lemma 1 only describes how to encode sets of integer points inside of polytopes. The key result that makes this theory useful in our setting is that some more general lattice point sets arising from simple operations on polytopal lattice point sets also admit short rational generating function encodings. Barvinok and Woods [2] developed powerful algorithms that apply to these more general settings. For our purposes the most important algorithm concerns constant-length Boolean combinations of polyhedra. A *Boolean combination* of the sets $S_1, \ldots, S_k$ is any combination of unions, intersections and set differences of those sets. For instance, $(S_1 \cap S_2) \setminus S_3$ is a Boolean combination of the sets S_1, S_2 and S_3.

Lemma 2 (Boolean Operations Lemma [Cor. 3.7 in 2]) *Given fixed integers k and ℓ there exists a constant s and a polynomial-time algorithm for the following problem. Given as* input*, in binary encoding,*

(I$_1$) *the dimension m and*

(I$_2$) *Barvinok encodings of k finite sets $S_p \subseteq \mathbb{Z}^m$, $g(S_p; \boldsymbol{\xi})$ such that for each rational term the number of binomials in the denominator is at most ℓ,*

output*, in binary encoding,*

(O$_1$) *rational numbers γ_i, integer vectors $\mathbf{c}_i$, $\mathbf{d}_{ij}$ for $i \in I$, $j = 1, \ldots, s_i$, where $s_i \leq s$, such that*

$$g(S; \boldsymbol{\xi}) = \sum_{i \in I} \gamma_i \frac{\boldsymbol{\xi}^{\mathbf{c}_i}}{(1 - \boldsymbol{\xi}^{\mathbf{d}_{i1}}) \ldots (1 - \boldsymbol{\xi}^{\mathbf{d}_{is_i}})}$$

is a rational generating function of the finite set S that is the Boolean combination *of the sets $S_1, \ldots, S_k$, and where each rational term in the expression has at most s terms in its denominator and where I is a polynomial-sized index set.*

We remark that if k were allowed to vary, the number of binomials in the denominators would become exponential (essentially doubling with each Boolean operation); thus, we must require that k be fixed in order to achieve a polynomial run time. We also note that if the input sets $S_p \subseteq \mathbb{Z}^m$ are integer points inside of polyhedra whose Barvinok encodings $g(S; \boldsymbol{\xi})$ arise from applying Lemma 1 then the condition that the number of binomials ℓ in the denominators are fixed follows under the assumption that the dimension m is fixed.

Disjoint unions are a special case of combining sets. Thus, we could address them as well using the Boolean Operations Lemma. However, in this case we can prove a stronger result, analogous to our method in the simple example above. This allows the number of sets k in the union to be polynomial in the input size instead of fixed.

Lemma 3 (Disjoint Unions) *If two lattice point sets S and T are disjoint then the generating function for $S \cup T$ is the sum of generating functions for S and T. More generally, for disjoint lattice point sets $S_1, \ldots, S_k$:*

$$g\left(\biguplus_{i=1}^{k} S_i, \boldsymbol{\xi}\right) = \sum_{i=1}^{k} g(S_i, \boldsymbol{\xi}),$$

where $\biguplus$ denotes disjoint union.

Once a rational generating function of a set S has been computed, various pieces of information can be extracted from it. First, we consider computing the cardinality of S.

Lemma 4 (Counting Lemma) *Let the dimension m be a fixed constant. Given a lattice point set $S \in \mathbb{Z}^m$ input as its Barvinok encoding $g(S, \boldsymbol{\xi})$, there exists a polynomial-time algorithm for computing $|S|$.*

The idea behind the proof of this result is analogous to the basic example at the beginning of this section. Given a Barvinok encoding of a lattice point set S as in (4), each of the basic rational functions has poles (the point $\boldsymbol{\xi} = \mathbf{1}$ in particular is a pole of all the basic rational functions), but after summing up only removable singularities remain. Obtaining the exact number of lattice points of lattice point set S is easy in (3), since clearly $|S| = g(S; \mathbf{1})$. Since (4) is a formula for the same function (except for removable singularities), we also have $|S| = \lim_{\boldsymbol{\xi} \to \mathbf{1}} g(S; \boldsymbol{\xi})$, which can be evaluated in polynomial time by performing a residue calculation with each basic rational function in the sum (4).

We can also *explicitly enumerate* all elements of S. We note that the cardinality of S can be exponential in the encoding length. Nevertheless there exists a *polynomial-space, polynomial-delay enumeration algorithm*. The following result is derived from Theorem 7 of [5].

Lemma 5 (Enumeration Lemma) *Let the dimension m and the maximum number ℓ of binomials in the denominators be fixed. Then there exists a polynomial-space, polynomial-delay enumeration algorithm for the following enumeration problem. Given as* input*, in binary encoding, a bound M and the Barvinok encoding $g(S, \boldsymbol{\xi})$ of a lattice point set $S \in [-M, M]^m \cap \mathbb{Z}^n$,* output*, in binary encoding, all points in S in lexicographic order.*

Note, the proof relies on a binary search procedure that uses cardinality counts to test for emptiness of each search region. Binary search can also give useful results for *optimizing*. By applying Lemma 2 and a binary search argument, we can optimize in polynomial time any linear objective over a set of lattice points in Barvinok encoding.

When the objective function is an arbitrary polynomial function (without any assumptions on convexity) that is nonnegative on a set with by a Barvinok encoding, then it is still possible to use a fully polynomial time approximation scheme (FPTAS).

Lemma 6 (FPTAS for maximizing non-negative polynomials [5]) *Let the dimension n and the maximum number ℓ of binomials in the denominator be fixed. There exists a polynomial-time algorithm for the following problem. Given as* input *an*

(I$_1$) *two vectors $\mathbf{x}_{\mathrm{L}}$, $\mathbf{x}_{\mathrm{U}} \in \mathbb{Z}^k$,*

(I$_2$) *a Barvinok encoding of a finite set $S \subseteq \mathbb{Z}^k$ of lattice points that is contained in the box $\{\, \mathbf{x} : \mathbf{x}_{\mathrm{L}} \leq \mathbf{x} \leq \mathbf{x}_{\mathrm{U}} \,\}$,*

(I$_3$) *a list of coefficients $f_i \in \mathbb{Q}$, encoded in binary encoding, and exponent vectors $\boldsymbol{\alpha}_i \in \mathbb{Z}_+$, encoded in unary encoding, representing a polynomial*

$$f = \sum_i f_i \mathbf{x}^{\boldsymbol{\alpha}_i} \in \mathbb{Q}[x_1, \ldots, x_n]$$

that is non-negative on S,

(I$_4$) *a positive rational number $1/\epsilon$ encoded in unary encoding,*

output*, in binary encoding,*

(O$_1$) *a point $\mathbf{x}_\epsilon \in S$ that satisfies*

$$f(\mathbf{x}_\epsilon) \geq (1 - \epsilon) f^* \quad \textit{where} \quad f^* = \max_{\mathbf{x} \in S} f(\mathbf{x}).$$

Here we have offered only some highlights from the theory of rational generating functions, focusing on results needed for the analysis in this paper. A more complete picture of this theory can be obtained from the excellent textbook by Beck and Robbins [3].

References

[1] A. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Mathematics of Operations Research*, 19(4):769–779, 1994.

[2] A. Barvinok and K. M. Woods. Short rational generating functions for lattice point problems. *Journal of the American Mathematical Society*, 16(957-979), 2003.

[3] M. Beck and S. Robbins. *Computing the Continuous Discretely: Integer-point Enumeration in Polyhedra*. Springer, 2007.

[4] J. A. De Loera. The many aspects of counting lattice points in polytopes. *Mathematische Semesterberichte*, 52(2):175–195, August 2005.

[5] J. A. De Loera, R. Hemmecke, and M. Köppe. Pareto optima of multicriteria integer linear programs. *INFORMS Journal on Computing*, 21(1):39–48, 2009.

[6] J. A. De Loera, R. Hemmecke, J. Tauzer, and R. Yoshida. Effective lattice point counting in rational convex polytopes. *Journal of Symbolic Computation*, 38(4):1273–1302, October 2004.

[7] D. Lepelley, A. Louichi, and H. Smaoui. On Ehrhart polynomials and probability calculations in voting theory. *Social Choice and Welfare*, 30(3):363–383, 2008.

[8] S. Verdoolaege, R. Seghir, K. Beyls, V. Loechner, and M. Bruynooghe. Analytical computation of Ehrhart polynomials: Enabling more compiler analyses and optimizations. In *CASES: Proceedings of the International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pages 248–258, 2004.