Computing the period of an Ehrhart quasi-polynomial

Kevin Woods*

Department of Mathematics
University of California, Berkeley, USA
kwoods@math.berkeley.edu

Submitted: Jun 9, 2005; Accepted: Jun 24, 2005; Published: Jul 29, 2005
Mathematics Subject Classifications: 05A15, 68W30, 52C07

Abstract

If $P \subset \mathbb{R}^d$ is a rational polytope, then $i_P(t) := \#(tP \cap \mathbb{Z}^d)$ is a quasi-polynomial in t, called the Ehrhart quasi-polynomial of P. A period of $i_P(t)$ is $\mathcal{D}(P)$, the smallest $\mathcal{D} \in \mathbb{Z}_+$ such that $\mathcal{D} \cdot P$ has integral vertices. Often, $\mathcal{D}(P)$ is the minimum period of $i_P(t)$, but, in several interesting examples, the minimum period is smaller. We prove that, for fixed d, there is a polynomial time algorithm which, given a rational polytope $P \subset \mathbb{R}^d$ and an integer n, decides whether n is a period of $i_P(t)$. In particular, there is a polynomial time algorithm to decide whether $i_P(t)$ is a polynomial. We conjecture that, for fixed d, there is a polynomial time algorithm to compute the minimum period of $i_P(t)$. The tools we use are rational generating functions.

1 Introduction

Given a rational polytope $P \subset \mathbb{R}^d$ (that is, a bounded subset of $\mathbb{R}^d$ which is defined by a finite collection of integer linear inequalities), define the function

$$i_P(t) = \#(tP \cap \mathbb{Z}^d),$$

where tP is P dilated by a factor of t. Also define $\mathcal{D} = \mathcal{D}(P)$ to be the smallest $\mathcal{D} \in \mathbb{Z}_+$ such that $\mathcal{D} \cdot P$ has integral vertices. Ehrhart proved [Ehr62] that $i_P(t)$ is a quasi-polynomial function with a period of $\mathcal{D}$. In other words, there exist polynomial functions $f_0(t), f_1(t), \ldots, f_{\mathcal{D}-1}(t)$, called the *constituents* of $i_P(t)$, such that

$$i_P(t) = f_j(t) \text{ for } t \equiv j \pmod{\mathcal{D}}.$$

Example 1. $P = [0, \frac{1}{2}] \times [0, \frac{1}{2}] \subset \mathbb{R}^2$.

*Partially supported by a Clay Liftoff Fellowship and NSF Grant DMS 0402148.

Then

$$i_P(t) = \begin{cases} \left(\frac{t+2}{2}\right)^2, & \text{for } t \text{ even} \\ \left(\frac{t+1}{2}\right)^2, & \text{for } t \text{ odd} \end{cases} .$$

$\square$

We know that $\mathcal{D}$ is *a* period of the quasi-polynomial $i_P(t)$. What is the *minimum* period? Certainly, it must divide $\mathcal{D}$. In most cases, in fact, it is exactly $\mathcal{D}$. In certain interesting examples, however, the minimum period is smaller.

Example 2. Given partitions λ and μ, define the Gelfand-Tsetlin polytope $P = P_{\lambda\mu} \subset \mathbb{R}^N$, as in [DLM04] (following the classic [GC50]), where N is defined in terms of the lengths of λ and μ.

Then $\#(P \cap \mathbb{Z}^N)$ is the dimension of the weight μ subspace of the irreducible representation of $\mathrm{GL}_n\mathbb{C}$ with highest weight λ. Though

$$i_P(t) = \#(P_{t\lambda,t\mu} \cap \mathbb{Z}^N)$$

is a polynomial, that is, it has period one (see [KR86]), $\mathcal{D}(P)$ may be made arbitrarily large by suitable choice of λ and μ (see [DLM04]).

$\square$

Example 3. More generally, given partitions λ, μ, and ν such that $|\lambda| + |\mu| = |\nu|$, define the *hive polytope* $P = P^{\nu}_{\lambda\mu} \subset \mathbb{R}^N$ as in [Buc00] (an exposition of ideas from [KT99]), where N is defined in terms of the length of λ, μ, and ν.

Then $\#(P \cap \mathbb{Z}^N)$ is the Littlewood-Richardson coefficient $c^{\nu}_{\lambda\mu}$, defined to be the multiplicity of V_ν (the highest weight representation of $\mathrm{GL}_n(\mathbb{C})$ corresponding to ν) in $V_\lambda \otimes V_\mu$. Though

$$i_P(t) = \#(P^{t\nu}_{t\lambda,t\mu} \cap \mathbb{Z}^N)$$

is a polynomial (see [DW02]), $\mathcal{D}(P)$ need not be one. $\square$

Example 4. Given any $\mathcal{D} \subset \mathbb{Z}_+$ and any s dividing $\mathcal{D}$, let P be the pentagon with vertices $(0,0), (0, -\frac{1}{s}), (\mathcal{D}, -\frac{1}{s}), (\mathcal{D}, 0)$, and $(1, \frac{\mathcal{D}-1}{\mathcal{D}})$.

In [MW05], it is shown that this pentagon has $\mathcal{D}(P) = \mathcal{D}$, but has minimum period s. $\square$

These examples raise several questions: When is the minimum period of $i_p(t)$ less than $\mathcal{D}(P)$? When is $i_P(t)$ a polynomial? How can we tell what the minimum period of $i_P(t)$ is? These questions are wide-open, though [MW05] gives a geometric characterization of the polygons $P \subset \mathbb{R}^2$ such that $i_P(t)$ is a polynomial. Here, we attack these questions from a computational perspective. Can we find algorithms to answer these questions "quickly?"

Let us be more precise. We define the *input size* of an algorithm to be the number of bits needed to encode the input into binary. In particular, the input size of an integer a is approximately $1 + \log_2|a|$ (the number of digits needed to write a in binary). An algorithm is called *polynomial time* if the number of steps it takes is bounded by a certain polynomial in the input size. Proving that an algorithm is polynomial time is generally regarded as proving that it is "quick," at least theoretically. See [Pap94] for general background on algorithms and computation complexity.

Our algorithms will take as input a polytope P. The input size of a polytope defined by n linear inequalities $\langle c_i, x\rangle \le b_i$, where $c_i \in \mathbb{Z}^d$, $b_i \in \mathbb{Z}$, is approximately

$$nd + \sum_{i,j} \log_2|c_{ij}| + \sum_i \log_2|b_i|.$$

We can now state the main theorem, which we will prove in Section 4.

Theorem 5. *Fix d. There is a polynomial time algorithm which, given a rational polytope $P \subset \mathbb{R}^d$ and an integer $n > 0$, decides whether n is a period of the quasi-polynomial $i_P(t)$.*

In particular, there is a polynomial time algorithm which decides whether $i_P(t)$ is a polynomial (that is, whether $n = 1$ is a period).

It is important that we fix d in this theorem, because problems of this sort become intractable if d is allowed to vary. For example, the problem of deciding whether P even contains an integer point is NP-hard if d is not fixed.

Naïvely applying Theorem 5 yields an algorithm to find the minimum period of $i_P(t)$ which, unfortunately, is not polynomial time. We would have to factor $\mathcal{D}(P)$, which would give us a set of possible n, one of which must be the minimal period. We will prove the following corollary in Section 4. By a *polynomial-time reduction*, of Problem A to Problem B, we mean that, if there was some oracle which could solve Problem B instantaneously (more precisely, in the amount of time it takes to output the answer to Problem B), then we could use that oracle to get a polynomial time algorithm for Problem A. In other words, Problem A is "as easy as" Problem B.

Corollary 6. *Fix d. There is a polynomial-time reduction of the problem of finding the minimum period of $i_P(t)$, where P is a d-dimensional polytope, to the problem of factoring a natural number $\mathcal{D}$.*

Unfortunately, the problem of factoring is probably hard. It is not known to be polynomial time (read, not too hard) or NP-hard (read, very hard) and is probably somewhere in between. Nevertheless, we make the following conjecture.

Conjecture 7. *Fix d. There is a polynomial time algorithm, which, given a d-dimensional polytope P, computes the minimum period of $i_P(t)$.*

The tools we will use are *rational generating functions*. Given a set $S \subset \mathbb{Z}^d$, define the generating function

$$f(S;\mathbf{x}) = \sum_{a=(a_1,\ldots,a_d)\in S} x_1^{a_1} x_2^{a_2} \cdots x_d^{a_d} = \sum_{a\in S} \mathbf{x}^a.$$

Sets that are very large can sometimes be written compactly as rational generating functions in the form

$$f(S;\mathbf{x}) = \sum_{i\in I} \alpha_i \frac{\mathbf{x}^{p_i}}{(1-\mathbf{x}^{b_{i1}})(1-\mathbf{x}^{b_{i2}})\cdots(1-\mathbf{x}^{b_{ik_i}})}, \tag{1}$$

where $\mathbf{x} \in \mathbb{C}^d$, $\alpha_i \in \mathbb{Q}$, $p_i \in \mathbb{Z}^d$, and $b_{ij} \in \mathbb{Z}^d \setminus \{0\}$.

Example 8. $S = \{0, 1, 2, \ldots, n\}$, for some n.

Then

$$\begin{aligned} f(S;x) &= 1 + x + x^2 + \cdots + x^n \\ &= \frac{1-x^{n+1}}{1-x}. \end{aligned}$$

□

In Section 2, we present several tools to compute and to manipulate rational generating functions, most of which were proved in either [BP99] or [BW03].

Given a rational polytope $P \subset \mathbb{R}^d$, define the generating function

$$F_P(t,z) = f_0(t) + f_1(t)z + \cdots + f_{\mathcal{D}-1}(t)z^{\mathcal{D}-1},$$

where the $f_i(t)$ are the constituents of $i_P(t)$. In Section 3, we will prove the following proposition, which will be useful in the proof of Theorem 5.

Proposition 9. *Fix d. There is a polynomial time algorithm which, given a rational polytope P, computes $F_P(t,z)$ as a rational generating function of the form (1).*

Finally, in Section 4, we prove Theorem 5 and Corollary 6.

2 Rational generating function tools

In this section, we present several tools to compute and manipulate rational generating functions. Except for Lemma 16, which is proved here, they were proved in either [BP99] or [BW03].

First we present a tool for creating rational generating functions.

Theorem 10. *(Theorem 4.4 of [BP99]) Fix d. Then there exists a polynomial time algorithm which, for any given rational polyhedron $P \subset \mathbb{R}^d$, computes $f(P \cap \mathbb{Z}^d; \mathbf{x})$ in the form*

$$f(P \cap \mathbb{Z}^d; \mathbf{x}) = \sum_{i \in I} \epsilon_i \frac{\mathbf{x}^{p_i}}{(1 - \mathbf{x}^{a_{i1}})(1 - \mathbf{x}^{a_{i2}}) \cdots (1 - \mathbf{x}^{a_{id}})},$$

where $\epsilon_i \in \{-1, +1\}$, $p_i, a_{ij} \in \mathbb{Z}^d$, and $a_{ij} \neq 0$ for all i, j. In fact, for each i, $a_{i1}, a_{i2}, \ldots, a_{id}$ is a basis of $\mathbb{Z}^d$.

Example 11. P is the interval $[0, n]$.

Then $P \cap \mathbb{Z} = \{0, 1, 2, \ldots, n\}$, and we have already computed $f(P \cap \mathbb{Z}) = \frac{1 - x^{n+1}}{1 - x}$. □

Once we have computed some rational generating functions, we also have several tools to manipulate them.

Let $f(\mathbf{x})$, with $\mathbf{x} \in \mathbb{C}^d$, be a rational function in the form (1), and let $l_1, l_2, \ldots, l_d \in \mathbb{Z}^n$ be integer vectors. These vectors define the *monomial map* $\phi : \mathbb{C}^n \to \mathbb{C}^d$ given by

$$\mathbf{z} = (z_1, z_2, \ldots, z_n) \mapsto (\mathbf{z}^{l_1}, \mathbf{z}^{l_2}, \ldots, \mathbf{z}^{l_d}).$$

If the image of ϕ does not lie entirely in the poles of $f(\mathbf{x})$, we can define the function $g : \mathbb{C}^n \to \mathbb{C}$ by

$$g(\mathbf{z}) = f\big(\phi(\mathbf{z})\big),$$

which is regular at almost every point in $\mathbb{C}^n$. Then $g(\mathbf{z})$ is $f(\mathbf{x})$ specialized at $x_i = \mathbf{z}^{l_i}$. In particular, if $l_i = 0$ for all i, then $g(\mathbf{z})$ is $f(1, 1, \ldots, 1)$.

Example 12. S is a finite set.

Then $f(S; 1, 1, \ldots, 1) = |S|$. □

We have the following theorem, which states that, given $f(\mathbf{x})$ as a short rational generating function, we can find $g(\mathbf{z})$ quickly.

Theorem 13. *(Theorem 2.6 of [BW03]) Let us fix k, an upper bound on the k_i in (1). Then there exists a polynomial time algorithm, which, given $f(\mathbf{x})$ in the form (1) and a monomial map $\phi : \mathbb{C}^n \to \mathbb{C}^d$ such that the image of ϕ does not lie entirely in the poles of $f(\mathbf{x})$, computes $g(\mathbf{z}) = f\big(\phi(\mathbf{z})\big)$ in the form*

$$g(\mathbf{z}) = \sum_{i \in I'} \beta_i \frac{\mathbf{z}^{q_i}}{(1-\mathbf{z}^{b_{i1}})(1-\mathbf{z}^{b_{i2}})\cdots(1-\mathbf{z}^{b_{is}})},$$

where $s \le k$, $\beta_i \in \mathbb{Q}$, $q_i, b_{ij} \in \mathbb{Z}^n$, and $b_{ij} \neq 0$ for all i, j.

Now let $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ be Laurent power series given by

$$g_1(\mathbf{x}) = \sum_{m \in \mathbb{Z}^d} \alpha_m \mathbf{x}^m \text{ and } g_2(\mathbf{x}) = \sum_{m \in \mathbb{Z}^d} \beta_m \mathbf{x}^m.$$

Then the Hadamard product $g = g_1 \star g_2$ is defined to be the power series

$$g(\mathbf{x}) = \sum_{m \in \mathbb{Z}^d} \alpha_m \beta_m \mathbf{x}^m.$$

Example 14. S_1, S_2 are subsets of Z^d,

$$g_1(\mathbf{x}) = \sum_{m \in S_1} \mathbf{x}^m, \text{ and } g_2(\mathbf{x}) = \sum_{m \in S_2} \mathbf{x}^m.$$

Then

$$(g_1 \star g_2)(\mathbf{x}) = \sum_{m \in S_1 \cap S_2} \mathbf{x}^m.$$

□

More generally, we may take the Hadamard product with respect to a proper subset of the variables, by defining

$$g_1(\mathbf{y}, \mathbf{z}) \star_z g_2(\mathbf{y}, \mathbf{z}) = \sum_{m \in \mathbb{Z}^d} \alpha_m(\mathbf{y}) \beta_m(\mathbf{y}) \mathbf{z}^m,$$

where α_m and β_m are functions of $\mathbf{y}$ such that

$$g_1(\mathbf{y}, \mathbf{z}) = \sum_{m \in \mathbb{Z}^d} \alpha_m(\mathbf{y}) \mathbf{z}^m \text{ and } g_2(\mathbf{y}, \mathbf{z}) = \sum_{m \in \mathbb{Z}^d} \beta_m(\mathbf{y}) \mathbf{z}^m.$$

We have the following theorem (which is a slightly more general version of Lemma 3.4 of [BW03], but the proof is the same).

Theorem 15. *Fix k, d_1, and d_2. Let $\mathbf{y} \in \mathbb{C}^{d_1}, \mathbf{z} \in \mathbb{C}^{d_2}$, and $\mathbf{x} = (\mathbf{y}, \mathbf{z})$. Then there exists a polynomial time algorithm which, given $l \in \mathbb{Z}^{d_1+d_2}$ and functions*

$$g_1(\mathbf{x}) = \sum_{i \in I_1} \alpha_i \frac{\mathbf{x}^{p_i}}{(1-\mathbf{x}^{a_{i1}}) \cdots (1-\mathbf{x}^{a_{ik}})} \text{ and}$$
$$g_2(\mathbf{x}) = \sum_{i \in I_2} \beta_i \frac{\mathbf{x}^{q_i}}{(1-\mathbf{x}^{b_{i1}}) \cdots (1-\mathbf{x}^{b_{ik}})}$$

such that $\langle l, a_i \rangle, \langle l, b_i \rangle \neq 0$, computes $g = g_1 \star_z g_2$ (where the Laurent power series are convergent on a neighborhood of $(e^{l_1}, e^{l_2}, \ldots, e^{l_d})$).

Note that l in the input of the algorithm is important. For example, if $f(x) = \frac{1}{1-x}$, then f has two possible Laurent power series expansions

$$f(x) = 1 + x + x^2 + \cdots \text{ and } f(x) = -x^{-1} - x^{-2} - x^{-3} - \cdots$$

convergent on $|x| < 1$ and $|x| > 1$, respectively. In this paper, however, the power series we examine will actually be Laurent polynomials (which are convergent on all of $\mathbb{C}^d$), so we will not have to worry about l.

We present one final generating function tool.

Lemma 16. *Fix d and k. There is a polynomial time algorithm which, given rational generating functions $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ in the form (1) which are known to be Laurent polynomials, decides whether $g_1 \equiv g_2$.*

Remark: The lemma is also true if $g_1(\mathbf{x})$ and $g_2(x)$ are Laurent power series with an infinite number of terms, but there are several complications which will be noted in the proof.

Proof: Let $h(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$. We want to decide whether $h \equiv 0$. Suppose that

$$h(\mathbf{x}) = \sum_{a \in \mathbb{Z}^d} c_a \mathbf{x}^a,$$

and let

$$\tilde{h}(\mathbf{x}) = h(\mathbf{x}) \star h(\mathbf{x}) = \sum_{a \in \mathbb{Z}^d} c_a^2 \mathbf{x}^a.$$

We can compute $\tilde{h}$ in polynomial time, using Theorem 15. Then $h \equiv 0$ if and only if $\tilde{h} \equiv 0$. Since we know that h is a polynomial, we must simply check whether $\tilde{h}(1) = \sum_{a \in \mathbb{Z}^d} c_a^2$ is zero, which we can do in polynomial time using Theorem 13. If we did not know that h is polynomial, we would have to be a little more careful, and here is a sketch of what to do. We can find bounds M such that if $c_a = 0$ for all a with $\|a\|_\infty \leq M$, then h is identically zero, using, for example, ideas from Section 5.1 of [Woo04]. Then if we take the Hadamard product

$$\bar{h} = \tilde{h} \star \left(\frac{x_1^{-M} - x_1^{M+1}}{1 - x_1} \frac{x_2^{-M} - x_2^{M+1}}{1 - x_2} \cdots \frac{x_d^{-M} - x_d^{M+1}}{1 - x_d} \right),$$

we now have something which is known to be a Laurent polynomial, and h is identically zero if and only if $\bar{h}(1) = 0$. □

3 Computing the generating function

Proof of Proposition 9: Computing, say, $f_0(t)$ alone would be easy, by interpolation. Indeed, first define

$$g_0(s) = f_0(s\mathcal{D}).$$

We may find $g_0(0), g_0(1), \ldots, g_0(d)$ in polynomial time, using Theorems 10 and 13, and then interpolate, as follows. Let V be the $(d+1) \times (d+1)$ Vandermonde matrix whose i, j entry is $(i-1)^{j-1}$ as $1 \leq i, j \leq d+1$. Then, if $g_0(s) = a_0 + a_1 s + a_2 s^2 + \cdots + a_d s^d$, we have the following equation:

$$V \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_d \end{bmatrix} = \begin{bmatrix} g_0(0) \\ \vdots \\ g_0(d) \end{bmatrix}.$$

Multiplying by the inverse of V, we get the coefficients of $g_0(s)$, and can then easily recover the coefficients of $f_0(t)$.

We cannot, however, do this for each $f_i(t)$, sequentially, in polynomial time: there are $\mathcal{D}$ of them, and $\mathcal{D}$ may be exponential in the input size. Instead, we perform all $\mathcal{D}$ interpolations simultaneously, using generating functions.

For $0 \leq i \leq \mathcal{D} - 1$, let

$$g_i(s) = f\left(s\mathcal{D} + i\right).$$

For $0 \leq j \leq d$, let

$$h_j(z) = g_0(j) + g_1(j)z + g_2(j)z^2 + \cdots + g_{\mathcal{D}-1}(j)z^{\mathcal{D}-1}.$$

For $0 \leq i \leq \mathcal{D} - 1$ and $0 \leq k \leq d$, let a_{ik} be such that

$$g_i(s) = a_{i0} + a_{i1}s + a_{i2}s^2 + \cdots + a_{id}s^d,$$

and let

$$a_k(z) = a_{0k} + a_{1k}z + a_{2k}z^2 + \cdots + a_{\mathcal{D}-1,k}z^{\mathcal{D}-1}.$$

Then we have that

$$V \cdot \begin{bmatrix} a_0(z) \\ \vdots \\ a_d(z) \end{bmatrix} = \begin{bmatrix} h_0(z) \\ \vdots \\ h_d(z) \end{bmatrix}.$$

Therefore, if we can compute each $h_j(z)$ in polynomial time as short rational generating functions, then we could compute the $a_k(z)$ as short rational generating functions by multiplying by the inverse of V.

We compute

$$\begin{aligned} h_j(z) &= g_0(j) + g_1(j)z + g_2(j)z^2 + \cdots + g_{\mathcal{D}-1}(j)z^{\mathcal{D}-1} \\ &= f_0(j\mathcal{D}) + f_1(j\mathcal{D}+1)z + f_2(j\mathcal{D}+2) + \cdots + f_{\mathcal{D}-1}(j\mathcal{D}+\mathcal{D}-1)z^{\mathcal{D}-1} \\ &= i_P(j\mathcal{D}) + i_P(j\mathcal{D}+1)z + i_P(j\mathcal{D}+2)z^2 + \cdots + i_P(j\mathcal{D}+\mathcal{D}-1)z^{\mathcal{D}-1}, \end{aligned}$$

as follows. Given j, define the polyhedron

$$Q_j = \big\{(z, \mathbf{y}) \in \mathbb{R} \oplus \mathbb{R}^d : \ 0 \le z \le \mathcal{D}-1 \text{ and } \mathbf{y} \in (j\mathcal{D}+z)P\big\}.$$

Then

$$f(Q_j; z, \mathbf{y}) = \sum_{0 \le a \le \mathcal{D}-1} z^a \sum_{b \in (j\mathcal{D}+a)P} \mathbf{y}^b,$$

and

$$h_j(z) = f\big(Q_j; z, (1, 1, \ldots, 1)\big).$$

We may compute $f(Q_j; z, \mathbf{y})$ in polynomial time, using Theorem 10, and then perform the substitution $\mathbf{y} = (1, 1, \ldots, 1)$, using Theorem 13.

We have shown that we can construct the generating functions $a_k(z)$, for $1 \le k \le d$, in polynomial time. We must now use these generating functions to compute

$$F_P(t, z) = f_0(t) + f_1(t)z + \cdots + f_{\mathcal{D}-1}(t)z^{\mathcal{D}-1}.$$

Since, for $0 \le j \le \mathcal{D}-1$,

$$g_j(s) = a_{j0} + a_{j1}s + \cdots + a_{jd}s^d$$

and

$$f_j(t) = g_j\left(\frac{t-j}{\mathcal{D}}\right),$$

we have that

$$f_j(t) = a_{j0} + a_{j1}\frac{t-j}{\mathcal{D}} + \cdots + a_{jd}\left(\frac{t-j}{\mathcal{D}}\right)^d$$

and

$$F_P(t, z) = \left\{ \begin{array}{cccccccc} & a_{00} & + & a_{01}\frac{t}{\mathcal{D}} & + & \cdots & + & a_{0d}\left(\frac{t}{\mathcal{D}}\right)^d \\ + & a_{10}z & + & a_{11}\frac{t-1}{\mathcal{D}}z & + & \cdots & + & a_{1d}\left(\frac{t-1}{\mathcal{D}}\right)^d z \\ & \vdots & & \vdots & & & & \vdots \\ + & a_{\mathcal{D}-1,0}z^{\mathcal{D}-1} & + & a_{\mathcal{D}-1,1}\frac{t-\mathcal{D}+1}{\mathcal{D}}z^{\mathcal{D}-1} & + & \cdots & + & a_{\mathcal{D}-1,d}\left(\frac{t-\mathcal{D}+1}{\mathcal{D}}\right)^d z^{\mathcal{D}-1} \end{array} \right. .$$

For $0 \le k \le d$, define

$$b_k(t,z) = a_{0k}\left(\frac{t}{\mathcal{D}}\right)^k + a_{1k}\left(\frac{t-1}{\mathcal{D}}\right)^k z + \cdots + a_{\mathcal{D}-1,k}\left(\frac{t-\mathcal{D}+1}{\mathcal{D}}\right)^k z^{\mathcal{D}-1}.$$

Then

$$F_P(t,z) = b_0(t,z) + b_1(t,z) + \cdots + b_d(t,z).$$

For each k, we will compute $b_k(t,z)$ from

$$a_k(z) = a_{0k} + a_{1k}z + a_{2k}z^2 + \cdots + a_{\mathcal{D}-1,k}z^{\mathcal{D}-1}.$$

In fact

$$b_k(t,z) = a_k(z) \star_z \left[\left(\frac{t}{\mathcal{D}}\right)^k + \left(\frac{t-1}{\mathcal{D}}\right)^k z + \cdots + \left(\frac{t-\mathcal{D}+1}{\mathcal{D}}\right)^k z^{\mathcal{D}-1}\right],$$

and

$$\left(\frac{t}{\mathcal{D}}\right)^k + \left(\frac{t-1}{\mathcal{D}}\right)^k z + \cdots + \left(\frac{t-\mathcal{D}+1}{\mathcal{D}}\right)^k z^{\mathcal{D}-1}$$

can be computed as a short rational generating function in polynomial time, by expanding all of the terms and repeatedly using the fact that, for any k, $\sum_{i=1}^{\infty} i^k z^i$ is $\left(z\frac{d}{dz}\right)^k\left(\frac{1}{1-z}\right)$. Therefore we can compute the $b_k(t,z)$ and hence $F_P(t,z)$ in polynomial time. $\square$

4 Deciding whether n is a period

Proof of Theorem 5: Given n and P, we want to decide whether n is a period of the quasi-polynomial $i_P(t)$. Using Proposition 9, we may compute the generating function

$$F_P(t,z) = f_0(t) + f_1(t)z + \cdots + f_{\mathcal{D}-1}(t)z^{\mathcal{D}-1}$$

in polynomial time. Define the generating function

$$\begin{aligned} G_{n,P}(t,z) = \quad & f_n(t) + f_{n+1}(t)z + \cdots + f_{\mathcal{D}-1}(t)z^{\mathcal{D}-n-1} \\ & + f_0(t)z^{\mathcal{D}-n} + f_1(t)z^{\mathcal{D}-n+1} + \cdots + f_{n-1}(t)z^{\mathcal{D}-1}. \end{aligned}$$

Then n is a period of $i_P(t)$ if and only if $F_P(t,z) \equiv G_{n,p}(t,z)$. We must show how to compute $G_{n,P}$ in polynomial time. Note that

$$\begin{aligned} F_P(t,z) \star_z \left(\frac{z^n - z^{\mathcal{D}}}{1-z}\right) &= F_P(t,z) \star_z \left(z^n + z^{n+1} + \cdots + z^{\mathcal{D}-1}\right) \\ &= f_n(t)z^n + f_{n+1}(t)z^{n+1} + \cdots + f_{\mathcal{D}-1}(t)z^{\mathcal{D}-1} \end{aligned}$$

and

$$F_P(t,z) \star_z \left(\frac{1-z^n}{1-z}\right) = F_P(t,z) \star_z \left(1 + z + \cdots + z^{n-1}\right)$$
$$= f_0(t) + f_1(t)z + \cdots + f_{n-1}(t)z^{n-1}.$$

Then

$$G_{n,P}(t,z) = \left[F_P(t,z) \star_z \left(\frac{z^n - z^{\mathcal{D}}}{1-z}\right)\right] z^{-n} + \left[F_P(t,z) \star_z \left(\frac{1-z^n}{1-z}\right)\right] z^{\mathcal{D}-n}.$$

This can be computed in polynomial time, using Theorem 15.

We can decide whether $F_P(t,z) \equiv G_{n,p}(t,z)$ using Lemma 16, in polynomial time, and the proof follows. □

Proof of Corollary 6: Compute $\mathcal{D} = \mathcal{D}(P)$ by taking the least common multiple of the denominators of all of the coordinates of the vertices of P. Assume that we can find the prime factorization of $\mathcal{D}$ using an oracle. Initialize the following loop with $n_0 := \mathcal{D}$.

1. After the jth iteration of the loop, n_j is known to be a period of $i_P(t)$.
2. For each prime factor p of n_j, decide whether $\frac{n_j}{p}$ is a period of $i_P(t)$.
   - If none are periods, then n_j is the minimum period of $i_P(t)$, and we are done.
   - if $\frac{n_j}{p}$ is a period of $i_P(t)$ for some p, then repeat the process with $n_{j+1} = \frac{n_j}{p}$.

This loop must terminate, because eventually we would have $n_j = 1$. □

Acknowledgements

Many thanks to Matthias Beck for helpful conversations. These results were originally presented at the Mathematisches Forschungsinstitut Oberwolfach mini-workshop "Ehrhart-Quasipolynomials: Algebra, Combinatorics, and Geometry."

References

[BP99] Alexander Barvinok and James Pommersheim. An algorithmic theory of lattice points in polyhedra. In *New Perspectives in Algebraic Combinatorics (Berkeley, CA, 1996–97)*, volume 38 of *Math. Sci. Res. Inst. Publ.*, pages 91–147. Cambridge Univ. Press, Cambridge, 1999.

[Buc00] Anders Skovsted Buch. The saturation conjecture (after A. Knutson and T. Tao). *Enseign. Math. (2)*, 46(1-2):43–60, 2000. With an appendix by William Fulton.

[BW03] Alexander Barvinok and Kevin Woods. Short rational generating functions for lattice point problems. *J. Amer. Math. Soc.*, 16(4):957–979 (electronic), 2003.

[DLM04] Jesús De Loera and Tyrrell McAllister. Vertices of Gelfand-Tsetlin polytopes. *Discrete Comput. Geom.*, 32(4):459–470, 2004.

[DW02] Harm Derksen and Jerzy Weyman. On the Littlewood-Richardson polynomials. *J. Algebra*, 255(2):247–257, 2002.

[Ehr62] Eugène Ehrhart. Sur les polyèdres rationnels homothétiques à n dimensions. *C. R. Acad. Sci. Paris*, 254:616–618, 1962.

[GC50] Izrail Gelfand and M. L. Cetlin. Finite-dimensional representations of the group of unimodular matrices. *Doklady Akad. Nauk SSSR (N.S.)*, 71:825–828, 1950.

[KR86] Anatoli Kirillov and Nikolai Reshetikhin. The Bethe ansatz and the combinatorics of Young tableaux. *Zap. Nauchn. Sem. Leningrad. Otdel. Mat. Inst. Steklov. (LOMI)*, 155(Differentsialnaya Geometriya, Gruppy Li i Mekh. VIII):65–115, 194, 1986.

[KT99] Allen Knutson and Terence Tao. The honeycomb model of $\mathrm{GL}_n(\mathbf{C})$ tensor products. I. Proof of the saturation conjecture. *J. Amer. Math. Soc.*, 12(4):1055–1090, 1999.

[MW05] Tyrrell McAllister and Kevin Woods. The minimum period of the Ehrhart quasi-polynomial of a rational polytope. *J. Combin. Theory Ser. A*, 109(2):345–352, 2005.

[Pap94] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.

[Woo04] Kevin Woods. *Rational Generating Functions and Lattice Point Sets*. PhD thesis, University of Michigan, 2004.