

Solving Lattice Point Problems Using Rational Generating Functions

Kevin Woods
Oberlin College

An Easy Start

Question: How many even numbers are there between 100 and 250?

An Easy Start

Question: How many even numbers are there between 100 and 250?

List them all:

100, 102, 104, 106, 108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128,
130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158,
160, 162, 164, 166, 168, 170, 172, 174, 176, 178, 180, 182, 184, 186, 188,
190, 200, 202, 204, 206, 208, 210, 212, 214, 216, 218,
220, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240, 242, 244, 246, 248,
250

and count: **76**.

An Easy Start

This is the wrong way to answer the question.
Why?

An Easy Start

This is the wrong way to answer the question.
Why?

Because it doesn't take advantage of the **structure** of the set.

Theme of talk: **Generating functions** are a nice tool to take advantage of the special structure of certain sets.

An Easy Start

Given a set $S \subseteq \mathbb{N}$, define the generating function

$$f(S; x) = \sum_{a \in S} x^a.$$

In example,

$$\begin{aligned} f(S; x) &= x^{100} + x^{102} + x^{104} + \dots + x^{248} + x^{250} \\ &= \frac{x^{100} - x^{252}}{1 - x^2}. \end{aligned}$$

Then $|S| = f(S; 1)$.

Use l'Hospital's rule:

$$f(S; 1) = \frac{100 - 252}{-2} = 76.$$

An Easy Start

Given a set $S \subseteq \mathbb{N}$, define the generating function

$$f(S; x) = \sum_{a \in S} x^a.$$

In example,

$$\begin{aligned} f(S; x) &= x^{100} + x^{102} + x^{104} + \cdots + x^{248} + x^{250} \\ &= \frac{x^{100} - x^{252}}{1 - x^2}. \end{aligned}$$

Then $|S| = f(S; 1)$.

Use l'Hospital's rule:

$$f(S; 1) = \frac{100 - 252}{-2} = 76.$$

An Easy Start

Given a set $S \subseteq \mathbb{N}$, define the generating function

$$f(S; x) = \sum_{a \in S} x^a.$$

In example,

$$\begin{aligned} f(S; x) &= x^{100} + x^{102} + x^{104} + \dots + x^{248} + x^{250} \\ &= \frac{x^{100} - x^{252}}{1 - x^2}. \end{aligned}$$

Then $|S| = f(S; 1)$.

Use l'Hospital's rule:

$$f(S; 1) = \frac{100 - 252}{-2} = 76.$$

The Frobenius Problem

Let a_1, a_2, \dots, a_d be nonnegative integers such that $\gcd(a_1, a_2, \dots, a_d) = 1$. Let

$$S = \{\lambda_1 a_1 + \dots + \lambda_d a_d : \lambda_i \in \mathbb{N}\}.$$

Question: What is the largest integer not in S ?

Question: How many positive integers are not in S ?

The Frobenius Problem

Example: $a_1 = 3$, $a_2 = 7$.

$$S = \{0, 3, 6, 7, 9, 10, 12, 13, 14, \dots\}.$$

Question: What is the largest integer not in S ?

Answer: 11.

Question: How many positive integers are not in S ?

Answer: 6.

Generating Functions to the Rescue

Listing out the set is the “wrong” way to answer these questions, because there’s some **structure** we’re not using.

Let’s use **generating functions**.

$$\begin{aligned}f(S; x) &= 1 + x^3 + x^6 + x^7 + x^9 + x^{10} + \dots \\ &= \frac{1 - x^{21}}{(1 - x^3)(1 - x^7)}.\end{aligned}$$

In general

$$f(S; x) = \frac{1 - x^{a_1 a_2}}{(1 - x^{a_1})(1 - x^{a_2})}.$$

Generating Functions to the Rescue

Let $T = \mathbb{N} \setminus S$ (which is $\{1, 2, 4, 5, 8, 11\}$ in the example).

$$\begin{aligned} f(T; x) &= \frac{1}{1-x} - f(S; x) \\ &= \frac{(1-x^{a_1})(1-x^{a_2}) - (1-x)(1-x^{a_1 a_2})}{(1-x)(1-x^{a_1})(1-x^{a_2})}. \end{aligned}$$

The largest integer not in S is the degree of the polynomial $f(T; x)$, which is

$$(1 + a_1 a_2) - (1 + a_1 + a_2) = a_1 a_2 - a_1 - a_2.$$

The number of positive integers not in S is $f(T; 1)$, which is (taking the limit as $x \rightarrow 1$)

$$\frac{a_1 a_2 - a_1 - a_2 + 1}{2}.$$

Generating Functions to the Rescue

Let $T = \mathbb{N} \setminus S$ (which is $\{1, 2, 4, 5, 8, 11\}$ in the example).

$$\begin{aligned} f(T; x) &= \frac{1}{1-x} - f(S; x) \\ &= \frac{(1-x^{a_1})(1-x^{a_2}) - (1-x)(1-x^{a_1 a_2})}{(1-x)(1-x^{a_1})(1-x^{a_2})}. \end{aligned}$$

The largest integer not in S is the degree of the polynomial $f(T; x)$, which is

$$(1 + a_1 a_2) - (1 + a_1 + a_2) = a_1 a_2 - a_1 - a_2.$$

The number of positive integers not in S is $f(T; 1)$, which is (taking the limit as $x \rightarrow 1$)

$$\frac{a_1 a_2 - a_1 - a_2 + 1}{2}.$$

Generating Functions to the Rescue

Let $T = \mathbb{N} \setminus S$ (which is $\{1, 2, 4, 5, 8, 11\}$ in the example).

$$\begin{aligned} f(T; x) &= \frac{1}{1-x} - f(S; x) \\ &= \frac{(1-x^{a_1})(1-x^{a_2}) - (1-x)(1-x^{a_1 a_2})}{(1-x)(1-x^{a_1})(1-x^{a_2})}. \end{aligned}$$

The largest integer not in S is the degree of the polynomial $f(T; x)$, which is

$$(1 + a_1 a_2) - (1 + a_1 + a_2) = a_1 a_2 - a_1 - a_2.$$

The number of positive integers not in S is $f(T; 1)$, which is (taking the limit as $x \rightarrow 1$)

$$\frac{a_1 a_2 - a_1 - a_2 + 1}{2}.$$

Generating Functions to the Rescue

Let $T = \mathbb{N} \setminus S$ (which is $\{1, 2, 4, 5, 8, 11\}$ in the example).

$$\begin{aligned} f(T; x) &= \frac{1}{1-x} - f(S; x) \\ &= \frac{(1-x^{a_1})(1-x^{a_2}) - (1-x)(1-x^{a_1 a_2})}{(1-x)(1-x^{a_1})(1-x^{a_2})}. \end{aligned}$$

The largest integer not in S is the degree of the polynomial $f(T; x)$, which is

$$(1 + a_1 a_2) - (1 + a_1 + a_2) = a_1 a_2 - a_1 - a_2.$$

The number of positive integers not in S is $f(T; 1)$, which is (taking the limit as $x \rightarrow 1$)

$$\frac{a_1 a_2 - a_1 - a_2 + 1}{2}.$$

Generating Functions to the Rescue

Let $T = \mathbb{N} \setminus S$ (which is $\{1, 2, 4, 5, 8, 11\}$ in the example).

$$\begin{aligned} f(T; x) &= \frac{1}{1-x} - f(S; x) \\ &= \frac{(1-x^{a_1})(1-x^{a_2}) - (1-x)(1-x^{a_1 a_2})}{(1-x)(1-x^{a_1})(1-x^{a_2})}. \end{aligned}$$

The largest integer not in S is the degree of the polynomial $f(T; x)$, which is

$$(1 + a_1 a_2) - (1 + a_1 + a_2) = a_1 a_2 - a_1 - a_2.$$

The number of positive integers not in S is $f(T; 1)$, which is (taking the limit as $x \rightarrow 1$)

$$\frac{a_1 a_2 - a_1 - a_2 + 1}{2}.$$

What else?

Questions:

- ▶ What types of sets can be encoded as rational generating functions?
- ▶ What types of sets can be encoded as **short** rational generating functions, **quickly**?

If $S \subseteq \mathbb{N}^n$, then let

$$f(S; \mathbf{x}) = \sum_{s=(s_1, \dots, s_n) \in S} x_1^{s_1} x_2^{s_2} \cdots x_n^{s_n} = \sum_{s \in S} \mathbf{x}^s.$$

What else?

Question: What types of sets can be encoded as rational generating functions?

Answer (W): Anything like

$$S = \{x \in \mathbb{N} \mid \forall y_1 \in \mathbb{N}, \exists y_2 \in \mathbb{N} : \\ (3y_1 + 5y_2 - x \geq 0) \text{ and} \\ (5y_1 + 2y_2 + 3x < 5 \text{ or } 3y_1 - x = 7)\},$$

using quantifiers (\exists and \forall), boolean operations (**and**, **or**, **not**), and linear (in)equalities (\leq , $=$, $>$).

These are sentences in the **Presburger arithmetic**.

What else?

Examples:

$$S = \{x \in \mathbb{N} \mid \exists y \in \mathbb{N} : 2y = x \text{ and } 100 \leq x \leq 250\}.$$

$$S = \{x \in \mathbb{N} \mid \exists \lambda_1 \in \mathbb{N}, \dots, \exists \lambda_d \in \mathbb{N} : \\ x = a_1 \lambda_1 + \dots + a_d \lambda_d\}.$$

Quick now!

Question: When can we find $f(S; \mathbf{x})$ **quickly**?

We want an algorithm that inputs a Presburger sentence and outputs $f(S; \mathbf{x})$.

The **input size** is the number of bits needed to encode the input for the algorithm.

The input size of a number a is

$$1 + \log_2(a).$$

For a Presburger sentence, the input size is approximately

$$\sum_i 1 + \log_2(a_i),$$

where the a_i are the coefficients appearing in the linear inequalities.

Quick now!

An algorithm is **polynomial time** if there is a polynomial p such that the algorithm runs in at most $p(\text{input size})$ steps.

polynomial time = **quick**

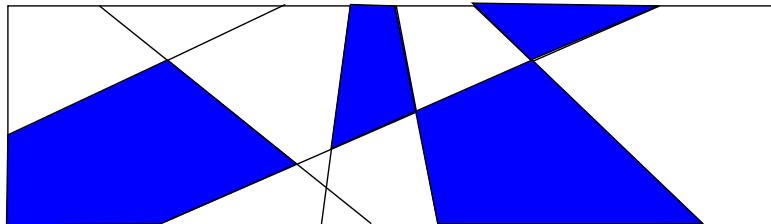
The Barvinok Algorithm

Theorem (Barvinok)

Fix n . There is a polynomial time algorithm which, given a set $S \subseteq \mathbb{N}^n$ defined **without** quantifiers, computes $f(S; \mathbf{x})$ in the form

$$\sum_{i \in I} \pm \frac{\mathbf{x}^{p_i}}{(1 - \mathbf{x}^{q_{i1}}) \cdots (1 - \mathbf{x}^{q_{in}})},$$

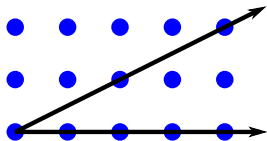
where $p_i, q_{ij} \in \mathbb{N}^n$.



The hard part is, given a **polyhedron** $P \subseteq \mathbb{R}^n$, compute $f(P \cap \mathbb{Z}^n; \mathbf{x})$.

The Barvinok Algorithm

Step 1: Unimodular Cones



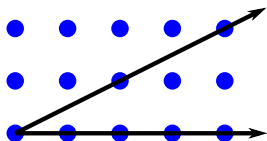
Let $u_1, \dots, u_k \in \mathbb{Z}^k$ be a **basis** for \mathbb{Z}^k , and let $\alpha \in \mathbb{Q}^k$.

Let

$$K = \{\alpha + \lambda_1 u_1 + \dots + \lambda_k u_k \in \mathbb{R}^k : \lambda_i \geq 0\}.$$

The Barvinok Algorithm

Step 1: Unimodular Cones



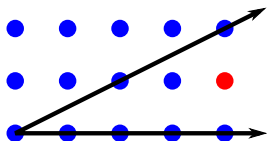
Example: $u_1 = (2, 1)$, $u_2 = (1, 0)$.

If $\alpha = (0, 0)$,

$$\begin{aligned} f(K \cap \mathbb{Z}^2; x, y) &= (1 + x^2y + x^4y^2 + \dots)(1 + x + x^2 + \dots) \\ &= \frac{1}{(1 - x^2y)(1 - x)}. \end{aligned}$$

The Barvinok Algorithm

Step 1: Unimodular Cones



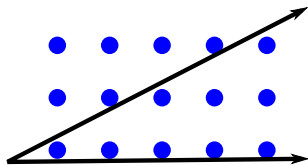
Example: $u_1 = (2, 1)$, $u_2 = (1, 0)$.

If $\alpha = (0, 0)$,

$$\begin{aligned} f(K \cap \mathbb{Z}^2; x, y) &= (1 + x^2y + x^4y^2 + \dots)(1 + x + x^2 + \dots) \\ &= \frac{1}{(1 - x^2y)(1 - x)}. \end{aligned}$$

The Barvinok Algorithm

Step 1: Unimodular Cones



Example: $u_1 = (2, 1)$, $u_2 = (1, 0)$.

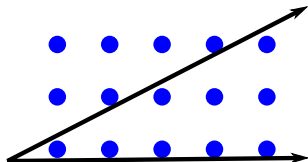
If $\alpha = \left(-\frac{3}{4}, -\frac{1}{4}\right)$,

$$f(K \cap \mathbb{Z}^2; x, y) = \frac{1}{(1-x)(1-x^2y)}.$$

(the same)

The Barvinok Algorithm

Step 1: Unimodular Cones



In general,

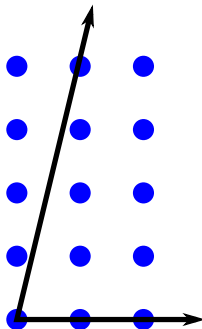
$$f(K \cap \mathbb{Z}^k; \mathbf{x}) = \frac{\mathbf{x}^p}{(1 - \mathbf{x}^{u_1}) \cdots (1 - \mathbf{x}^{u_k})},$$

where p is a function of α and the u_i .

The Barvinok Algorithm

Step 2: Unimodular Decomposition

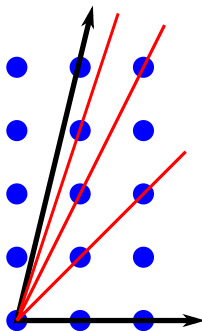
What about non-unimodular cones? Break into unimodular ones.



The Barvinok Algorithm

Step 2: Unimodular Decomposition

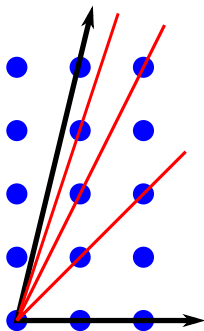
What about non-unimodular cones? Break into unimodular ones.



The Barvinok Algorithm

Step 2: Unimodular Decomposition

What about non-unimodular cones? Break into unimodular ones.

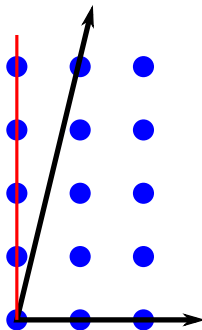


Too many cones!

The Barvinok Algorithm

Step 2: Unimodular Decomposition

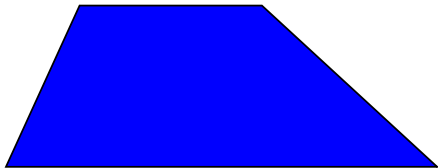
What about non-unimodular cones? Break into unimodular ones.



Signed Decomposition (Barvinok)

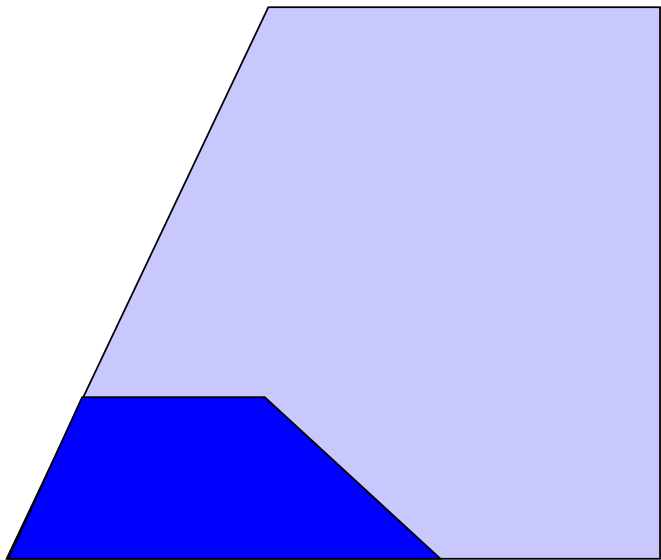
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



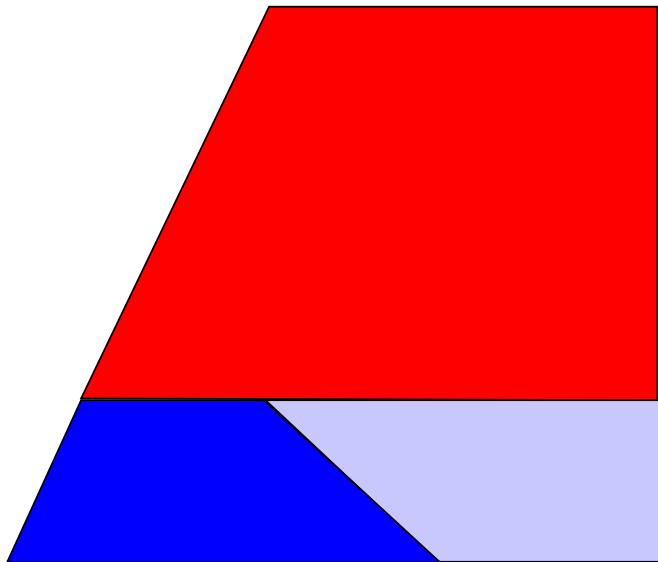
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



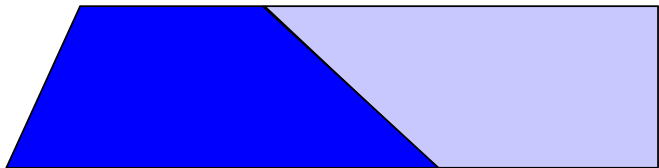
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



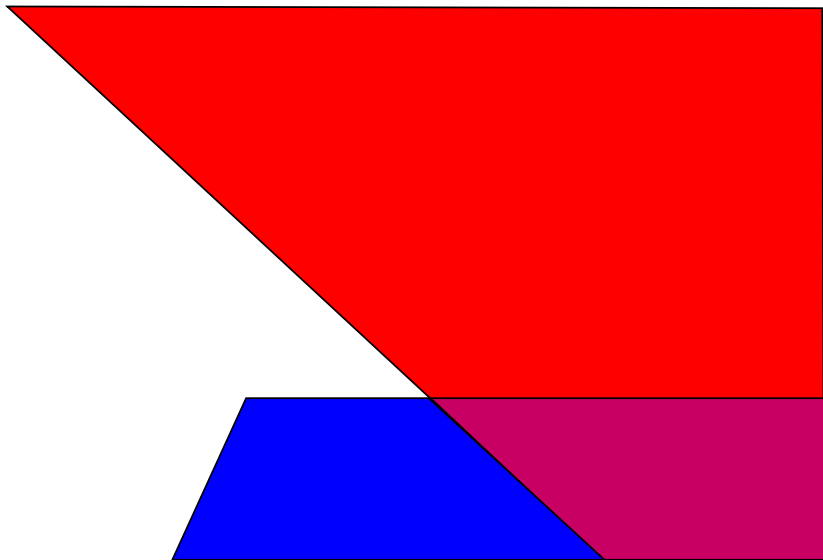
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



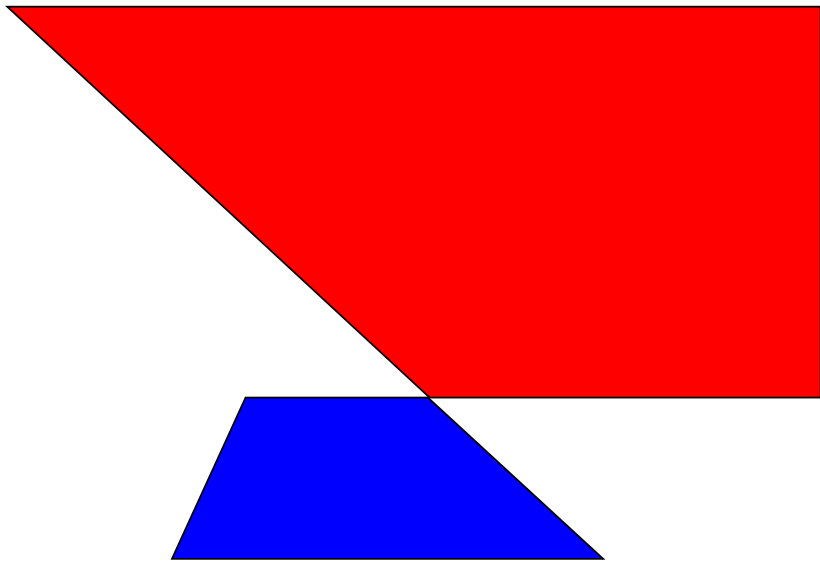
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



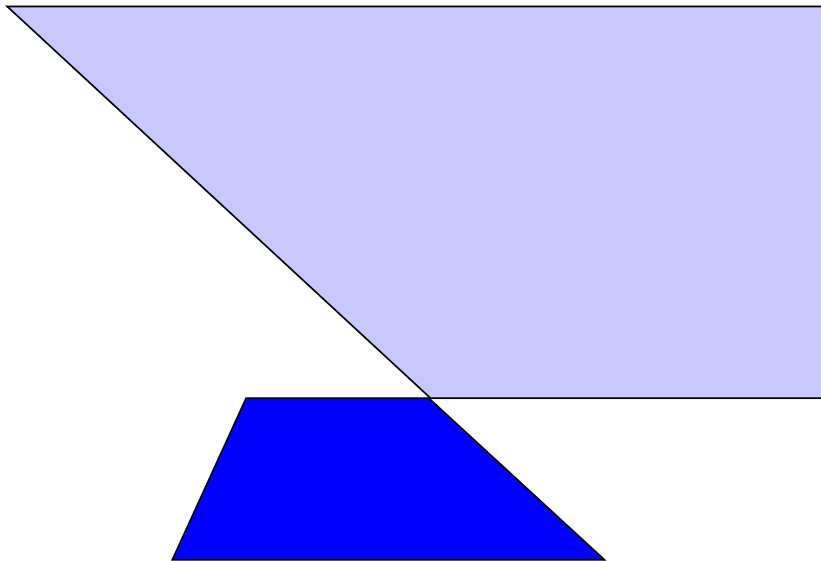
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



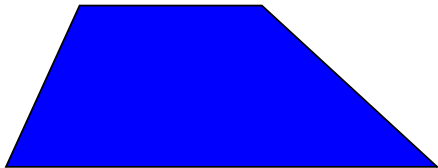
The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



The Barvinok Algorithm

Step 3: All Polyhedra (Brion)



Quantifiers?

Theorem (W)

Fix d , n , and m . There is a polynomial time algorithm which, given a set $S \subseteq \mathbb{N}^n$ defined by

$$S = \{x \in \mathbb{N}^n \mid \exists y_1 \in \mathbb{N}, \dots, \exists y_d \in \mathbb{N} : \\ F(x, y)\},$$

where $F(x, y)$ is a boolean combination of at most m linear inequalities, computes $f(S; \mathbf{x})$ in the form

$$\sum_{i \in I} \alpha_i \frac{\mathbf{x}^{p_i}}{(1 - \mathbf{x}^{q_{i1}}) \cdots (1 - \mathbf{x}^{q_{is}})},$$

where $\alpha_i \in \mathbb{Q}$, $p_i, q_{ij} \in \mathbb{N}^n$, and $s = s(d, n, m)$ is a constant.

Quantifiers?

Example:

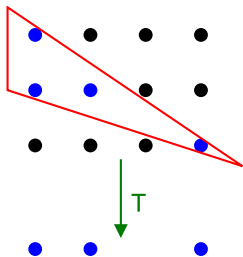
$$S = \{x \in \mathbb{N} \mid \exists \lambda_1 \in \mathbb{N}, \dots, \exists \lambda_d \in \mathbb{N} : \\ x = a_1 \lambda_1 + \dots + a_d \lambda_d\}.$$

Note: Theorem also holds when all of the quantifiers are \forall .

"Proof"

Projections

We need to compute generating functions for **projections** of $P \cap \mathbb{Z}^n$, where P is a polyhedron.



$$S = \{x \in \mathbb{N} \mid \exists y \in \mathbb{N} : (x, y) \in P\}.$$

$$T(x, y) = x, \text{ and } S = T(P \cap \mathbb{Z}^2).$$

“Proof”

1-d Kernel

Example: Frobenius Problem with $a_1 = 2$, $a_2 = 5$.

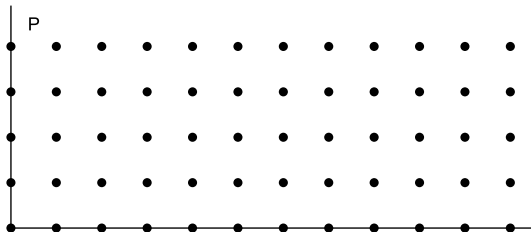
$$P = \{(x, y) : x, y \geq 0\}$$

$$T(x, y) = 2x + 5y. \text{ (1-d Kernel)}$$

$$\text{Then } S = T(P \cap \mathbb{Z}^2).$$

“Proof”

1-d Kernel



Compute

$$f(P \cap \mathbb{Z}^2; x, y) = \frac{1}{(1-x)(1-y)}$$

(Barvinok's algorithm).

“Proof”

1-d Kernel

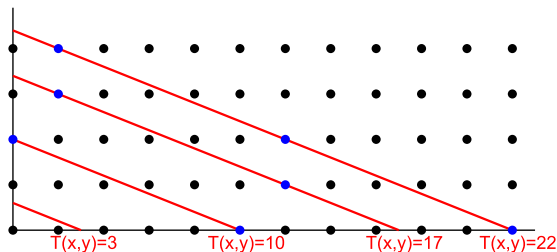
Compute

$$\begin{aligned} f(P \cap \mathbb{Z}^2; t^2, t^5) &= \frac{1}{(1-t^2)(1-t^5)} = (1+t^2+t^4+\dots)(1+t^5+\dots) \\ &= 1+t^2+t^4+t^5+t^6+t^7+t^8+t^9+2t^{10}+\dots \end{aligned}$$

Problem: T is not 1-1 on $P \cap \mathbb{Z}^2$.

"Proof"

1-d Kernel



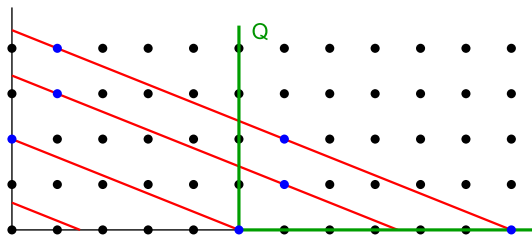
Compute

$$\begin{aligned} f(P \cap \mathbb{Z}^2; t^2, t^5) &= \frac{1}{(1-t^2)(1-t^5)} = (1+t^2+t^4+\dots)(1+t^5+\dots) \\ &= 1+t^2+t^4+t^5+t^6+t^7+t^8+t^9+2t^{10}+\dots \end{aligned}$$

Problem: T is **not 1-1** on $P \cap \mathbb{Z}^2$.

"Proof"

1-d Kernel



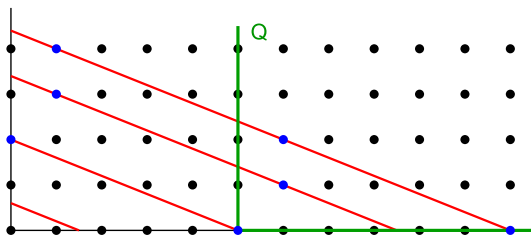
Let $Q = \{(x, y) : x \geq 5, y \geq 0\}$.

$$f(Q \cap \mathbb{Z}^2; x, y) = \frac{x^5}{(1-x)(1-y)}.$$

$$f(Q \cap \mathbb{Z}^2; t^2, t^5) = \frac{t^{10}}{(1-t^2)(1-t^5)}.$$

"Proof"

1-d Kernel

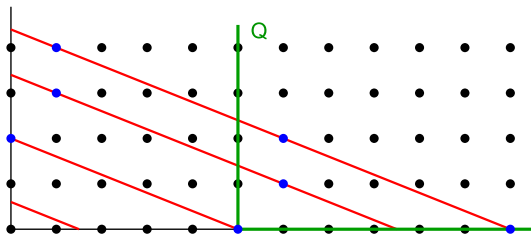


T is 1-1 on $(P - Q) \cap \mathbb{Z}^2$.

$$\begin{aligned} f(S; t) &= f(P \cap \mathbb{Z}^2; t^2, t^5) - f(Q \cap \mathbb{Z}^2; t^2, t^5) \\ &= \frac{1 - t^{10}}{(1 - t^2)(1 - t^5)}. \end{aligned}$$

"Proof"

1-d Kernel

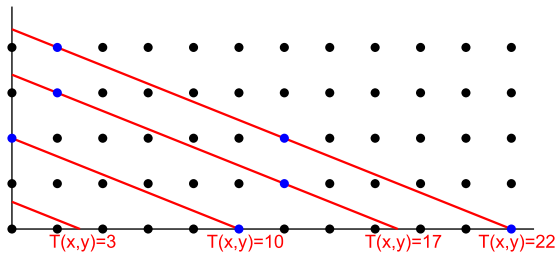


Why This Works: There are no **gaps** in the fibers of T .

Only works for **1-d** kernel.

"Proof"

1-d Kernel

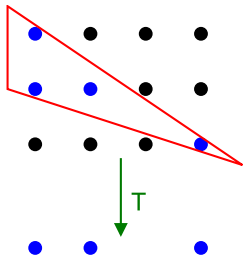


Note: This picture also shows us why $\mathbb{N} \setminus S$ has finite cardinality.

“Proof”

Higher-d Kernel

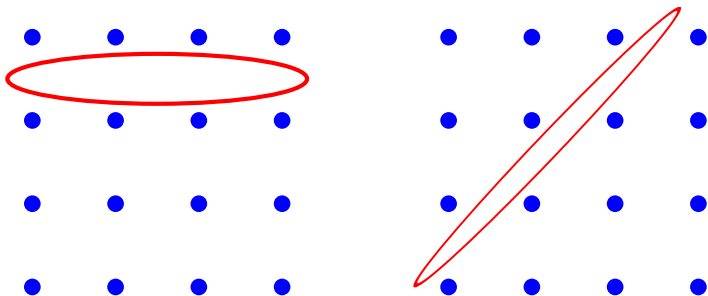
General situation: Use induction on the dimension of the kernel.



Must control the **gaps**.

“Proof”

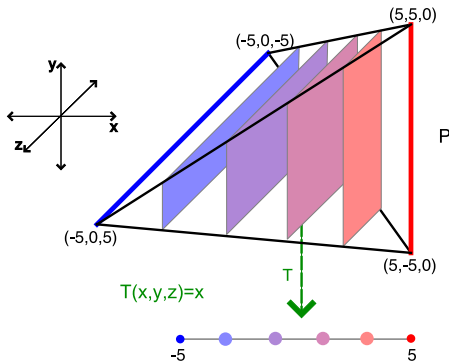
Higher-d Kernel



Flatness Theorem (Khinchin): Convex objects that contain **no** integer points are **thin** in some direction.

“Proof”

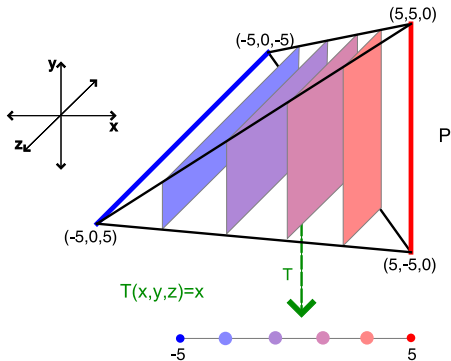
Higher-d Kernel



Look at the fibers of $T(P)$, and pick the **thinnest** direction. That direction gets projected out last.

"Proof"

Higher-d Kernel



Complication: Different fibers have different thin directions.

Solution: Break things up into pieces (Kannan).

Applications

- ▶ Frobenius problem (Barvinok-W)
- ▶ Minimal Hilbert Bases (Barvinok-W)
- ▶ Hilbert series of rings generated by monomials (Barvinok-W)
- ▶ Test sets for integer programming (Barvinok-W)
- ▶ Integer programming gaps (Hoşten-Sturmfels)
- ▶ Reduced Gröbner bases for toric ideals, and some related computations (De Loera, et al.)
- ▶ Standard pairs and arithmetic degree of order ideals in integer programming (Thomas-W)
- ▶ Ehrhart quasi-polynomials (and their period) (W)

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

Bad, even with a single quantifier (W; Schöning)

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

Bad, even with a single quantifier (W; Schöning)

- ▶ Fix number of variables and inequalities, only \exists quantifiers.

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

Bad, even with a single quantifier (W; Schöning)

- ▶ Fix number of variables and inequalities, only \exists quantifiers.

Good (W)

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

Bad, even with a single quantifier (W; Schöning)

- ▶ Fix number of variables and inequalities, only \exists quantifiers.

Good (W)

- ▶ Fix number of variables and inequalities, mixed quantifiers.

The Good, the Bad, and the _____

Presburger sentences from an algorithmic perspective:

- ▶ General sentences.

Bad

- ▶ Fix number of variables, no quantifiers.

Good (Barvinok)

- ▶ Fix number of variables, quantifiers allowed.

Bad, even with a single quantifier (W; Schöning)

- ▶ Fix number of variables and inequalities, only \exists quantifiers.

Good (W)

- ▶ Fix number of variables and inequalities, mixed quantifiers.

?????

Summary

- ▶ We can often use hidden **structure** in seemingly complicated sets to encode them compactly as generating functions.
- ▶ We can **manipulate** the generating functions to answer questions about the sets.
- ▶ We can do many of these things **quickly**.