Computing the Integer Points of a Polyhedron, I: Algorithm

Rui-Juan Jing[1,2] and Marc Moreno Maza[2]

[1] KLMM, UCAS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, rjing8@uwo.ca,
[2] University of Western Ontario, moreno@csd.uwo.ca.

Abstract. Let K be a polyhedron in $\mathbb{R}^d$, given by a system of m linear inequalities, with rational number coefficients bounded over in absolute value by L. In this series of two papers, we propose an algorithm for computing an irredundant representation of the integer points of K, in terms of "simpler" polyhedra, each of them having at least one integer point. Using the terminology of W. Pugh: for any such polyhedron P, no integer point of its grey shadow extends to an integer point of P. We show that, under mild assumptions, our algorithm runs in exponential time w.r.t. d and in polynomial w.r.t m and L. We report on a software experimentation. In this series of two papers, the first one presents our algorithm and the second one discusses our complexity estimates.

1 Introduction

The integer points of polyhedral sets are of interest in many areas of mathematical sciences, see for instance the landmark textbooks of A. Schrijver [19] and A. Barvinok [3], as well as the compilation of articles [4]. One of these areas is the analysis and transformation of computer programs. For instance, integer programming [7] is used by P. Feautrier in the scheduling of for-loop nests [8] and Barvinok's algorithm [2] for counting integer points in polyhedra is adapted by M. Köppe and S. Verdoolaege in [16] to answer questions like how many memory locations are touched by a for-loop nest. In [17], W. Pugh proposes an algorithm, called the *Omega Test*, for testing whether a polyhedron has integer points. In the same paper, W. Pugh shows how to use the Omega Test for performing dependence analysis [17] in for-loop nests. Then, in [18], he uses the Omega Test for deciding Presburger arithmetic formulas.

In [18], W. Pugh also suggests, without stating a formal algorithm, that the Omega Test could be used for quantifier elimination on Presburger formulas. This observation is a first motivation for the work presented in this series of two papers: we adapt the Omega Test so as to describe the integer points of a polyhedron via a *projection* scheme, thus performing elimination of existential quantifiers on Presburger formulas. Projections of polyhedra and parametric programming are tightly related problems, see [13]. Since the latter is essential to the parallelization of for-loop nests [7], which is of interest to the authors [5], we had here a second motivation for developing the proposed algorithm.

In [9] M. J. Fischer and M. O. Rabin show that any algorithm for deciding Presburger arithmetic formulas has a worst case running time which is a doubly exponential in the length of the input formula. However, this worst case scenario is based on a formula alternating existential and universal quantifiers. Meanwhile, in practice, the original *Omega Test* (for testing whether a polyhedron has integer points) can solve "difficult problems" as shown by W. Pugh in [18] and others, e.g. D. Wonnacott in [22]. This observation brings our third motivation: determining realistic assumptions under which our algorithm, based on the Omega Test, could run in a single exponential time.

Our algorithm takes as input a system of linear inequalities $\mathbf{Ax} \leq \mathbf{b}$ where $\mathbf{A}$ is a matrix over $\mathbb{Z}$ with m rows and d columns, $\mathbf{x}$ is the unknown vector and $\mathbf{b}$ is a vector of m coefficients in $\mathbb{Z}$. The points $\mathbf{x} \in \mathbb{R}^d$ satisfying $\mathbf{Ax} \leq \mathbf{b}$ form a polyhedron K and our algorithm decomposes its integer points (that is, $K \cap \mathbb{Z}^d$) into a disjoint union $\left(K_1 \cap \mathbb{Z}^{d_1}\right) \uplus \cdots \uplus \left(K_e \cap \mathbb{Z}^{d_e}\right)$, where $K_1, \ldots, K_e$ are "simpler" polyhedra such that $K_i \cap \mathbb{Z}^d \neq \varnothing$ holds and d_i is the dimentions of K_i, for $1 \leq i \leq e$. To use the terminology introduced by W. Pugh for the Omega test, no integer point of the grey shadow of any polyhedron K_i extends to an integer point of K_i. As a consequence, applying our algorithm to K_i would return K_i itself, for $1 \leq i \leq e$. Let us present the key principles and features of our algorithm through an example. Consider the polyhedron K of $\mathbb{R}^4$ given below:

$$\begin{cases} 2x + 3y - 4z + 3w \leq 1 \\ -2x - 3y + 4z - 3w \leq -1 \\ -13x - 18y + 24z - 20w \leq -1 \\ -26x - 40y + 54z - 39w \leq 0 \\ -24x - 38y + 49z - 31w \leq 5 \\ 54x + 81y - 109z + 81w \leq 2 \end{cases}.$$

A first procedure, called IntegerNormalize, detect implicit equations and solve them using techniques based on *Hermite normal form*, see Sect. 3 and 4.1. In our example $2x+3y-4z+3w = 1$ is an implicit equation and IntegerNormalize($\mathbf{Ax} \leq \mathbf{b}$) returns a triple $(\mathbf{t}, \mathbf{x} = \mathbf{Pt} + \mathbf{q}, \mathbf{Mt} \leq \mathbf{v})$ where $\mathbf{t}$ is a new unknown vector, the linear system $\mathbf{x} = \mathbf{Pt} + \mathbf{q}$ gives the general form of an integer solution of the implicit equation(s) and $\mathbf{Mt} \leq \mathbf{v}$ is obtained by substituting $\mathbf{x} = \mathbf{Pt} + \mathbf{q}$ into $\mathbf{Ax} \leq \mathbf{b}$. In our example, the systems $\mathbf{x} = \mathbf{Pt} + \mathbf{q}$ and $\mathbf{Mt} \leq \mathbf{v}$ are given by:

$$\begin{cases} x = -3t_1 + 2t_2 - 3t_3 + 2 \\ y = 2t_1 + t_3 - 1 \\ z = t_2 \\ w = t_3 \end{cases} \quad \text{and} \quad \begin{cases} 3t_1 - 2t_2 + t_3 \leq 7 \\ -2t_1 + 2t_2 - t_3 \leq 12 \\ -4t_1 + t_2 + 3t_3 \leq 15 \\ -t_2 \leq -25 \end{cases}.$$

A second procedure, called DarkShadow, takes $\mathbf{Mt} \leq \mathbf{v}$ as input and returns a couple $(\mathbf{t}', \Theta)$ where $\mathbf{t}'$ stands for all $\mathbf{t}$-variables except t_1, and Θ is a linear system in the $\mathbf{t}'$-variables such that any integer point solving of Θ extends to an integer point solving $\mathbf{Mt} \leq \mathbf{v}$. In our example, $\mathbf{t}' = \{t_2, t_3\}$ and Θ is given by:

$$\begin{cases} 2t_2 - t_3 \leq 48 \\ -5t_2 + 13t_3 \leq 67 \\ -t_2 \leq -25 \end{cases}$$

The polyhedron D of $\mathbb{R}^2$ defined by Θ, and the inequalities of $\mathbf{Mt} \leq \mathbf{v}$ not involving t_1, is called the *dark shadow* of the polyhedron defined by $\mathbf{Mt} \leq \mathbf{v}$. On

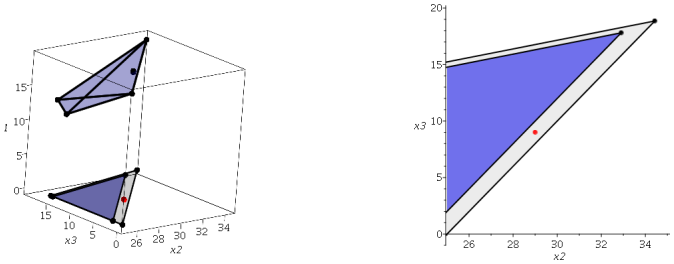

Fig. 1: The real, the dark and the grey shadows of a polyhedron.

the left-hand side of Fig. 1, one can see the polyhedron defined in $\mathbb{R}^3$ by $\mathbf{Mt} \leq \mathbf{v}$ together with its dark shadow D (shown in dark blue) as well as its projection on the (t_2, t_3)-plane, denoted by R and called *real shadow* by W. Pugh. The right-hand side of Fig. 1 gives a planar view of D and R. As we will see in Sect. 4.4, if $\mathbf{M't'} \leq \mathbf{v'}$ is the linear system generated by applying *Fourier-Motzkin elimination* (without removing redundant inequalities) to $\mathbf{Mt} \leq \mathbf{v}$ (in order to eliminate t_1) then Θ is given by a linear system of the form $\mathbf{M't'} \leq \mathbf{w'}$. This explains why, on the right-hand side of Fig. 1, each facet of the dark shadow D is parallel to a facet of the real shadow R. While this property is observed on almost all practical problems, in particular in the area of analysis and transformation of computer programs, it is possible to build examples where this property does not hold. We have examples in Section 5 of the second paper.

On the right-hand side of Fig. 1, one observes that the region $R \setminus D$, called *grey shadow*, contains integer points. Some of them, like $(t_2, t_3) = (29, 9)$, do not extend to an integer solution of $\mathbf{Mt} \leq \mathbf{v}$. Indeed, plugging $(t_2, t_3) = (29, 9)$ into $\mathbf{Mt} \leq \mathbf{v}$ yields $\frac{37}{2} \leq t_1 \leq \frac{56}{3}$, which has no integer solutions. However, other integer points of $R \setminus D$ may extend to integer solutions of $\mathbf{Mt} \leq \mathbf{v}$. In order to determine them, a third procedure, called Greyshadow, considers in turn the negation of each inequality θ of Θ. However, for each θ of Θ, instead of simply making a recursive call to the entire algorithm applied to $\mathbf{Mt} \leq \mathbf{v} \cup \{\theta\}$, simplifications (involving θ and the inequalities from which θ is derived) permit to replace this recursive call by several ones in lower dimension, thus guaranteeing termination of the whole algorithm. Details are given in Sect. 4.5 and 4.6.

Returning to our example, the negation of the inequality $2t_2 - t_3 \leq 48$ from Θ, combined with the system $\mathbf{Mt} \leq \mathbf{v}$, yields the following

$$\begin{cases} -2t_1 + 2t_2 - t_3 = 12 \\ 3t_1 - 2t_2 + t_3 \leq 7 \\ -4t_1 + t_2 + 3t_3 \leq 15 \\ -t_2 \leq -25 \end{cases},$$

which, by means of IntegerNormalize, rewrites to:

$$\begin{cases} t_1 = t_4 \\ t_2 = t_5 + 1 \\ t_3 = -2t_4 + 2t_5 + 1 \end{cases}, \text{ and } \begin{cases} t_4 \leq 8 \\ -10t_4 + 7t_5 \leq 11 \\ -t_5 \leq -24 \end{cases},$$

where t_4, t_5 are new variables. Continuing in this manner with the Greyshadow procedure, a decomposition of the integer points of $\mathbf{Mt} \leq \mathbf{v}$ is given by:

$$\begin{cases} 3t_1 - 2t_2 + t_3 \leq 7 \\ -2t_1 + 2t_2 - t_3 \leq 12 \\ -4t_1 + t_2 + 3t_3 \leq 15 \\ 2t_2 - t_3 \leq 48 \\ -5t_2 + 13t_3 \leq 67 \\ -t_2 \leq -25 \\ 2 \leq t_3 \leq 17 \end{cases}, \begin{cases} t_1 = 15 \\ t_2 = 27 \\ t_3 = 16 \end{cases}, \begin{cases} t_1 = 18 \\ t_2 = 33 \\ t_3 = 18 \end{cases}, \begin{cases} t_1 = 14 \\ t_2 = 25 \\ t_3 = 15 \end{cases}, \begin{cases} t_1 = 19 \\ t_2 = 50 + t_6 \\ t_3 = 50 + 2t_6 \\ -25 \leq t_6 \leq -16. \end{cases}.$$

Denoting these 5 systems respectively by $S_1, \ldots, S_5$ the integer points of K are finally given by the union of the integer points of the systems $\mathbf{x} = \mathbf{Pt} + \mathbf{q} \cup S_i$, for $1 \leq i \leq 5$. The systems $S_2, \ldots, S_5$ look simple enough to be considered as solution sets. What about S_1? The system S_1, as well as $S_2, \ldots, S_5$, satisfies a "back-substitution" property which is similar to that of a *regular chain* in the theory of polynomial system solving [1]. This property (formally stated in Sect. 4.2), when applied to S_1, says that for all $2 \leq i \leq 3$, every integer point of $\mathbb{R}^{4-i}$ solving all the inequalities of S_1 involving $t_i, \ldots, t_3$ only, extends to an integer point of $\mathbb{R}^{5-i}$ solving all the inequalities of S_1 involving $t_{i-1}, \ldots, t_3$.

With respect to the original Omega Test [17], our contributions are as follows.

1. We turn the decision procedure of the Omega Test into an algorithm decomposing all the integer points of a polyhedron.
2. Our decomposition is disjoint whereas the recursive calls in the original Omega Test may search for integer points in intersecting polyhedral regions.
3. The original Omega Test uses an ad-hoc routine for computing the integer solutions of linear equation systems, while we rely on Hermite normal form for this task. Consequently, we deduce complexity estimates for that task.
4. We also provide complexity estimates for the procedures Greyshadow and DarkShadow under realistic assumptions. From there, we derive complexity estimates for the entire algorithm, whereas no complexity estimates were known for the original Omega Test.

We report our work in a series of two papers. The present one describes and proves our algorithm. The second one establishes our complexity estimates.

2 Polyhedral Sets

This section is a review of the theory of polyhedral sets. It is based on the books of B. Grünbaum [10] and A. Schrijver [19], where proofs of the statements below can be found.

Given a positive integer d, we consider the d-dimensional Euclidean space $\mathbb{R}^d$ equipped with the Euclidean topology. Let K be a subset of $\mathbb{R}^d$. The *dimension* $\dim(K)$ of K is $a-1$ where a is the maximum number of affinely independent points in K. Let $\mathbf{a} \in \mathbb{R}^d$, let $b \in \mathbb{R}$ and denote by H the hyperplane defined by $H = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a}^T\mathbf{x} = b\}$. We say that the hyperplane H *supports* K if either $\sup\{\mathbf{a}^T\mathbf{x} \mid \mathbf{x} \in K\} = b$ or $\inf\{\mathbf{a}^T\mathbf{x} \mid \mathbf{x} \in K\} = b$ holds, but not both.

From now on, let us assume that K is convex. A set $F \subseteq K$ is a *face* if either $F = \varnothing$ or $F = K$, or if there exists a hyperplane H supporting K such that we have $F = K \cap H$. The set of all faces of K is denoted by $\mathcal{F}(K)$. We say that $F \in \mathcal{F}(K)$ is proper if we have $F \neq \varnothing$ or $F \neq K$. We note that the intersection of any family of faces of K is itself a face of K.

We say that K is a *polyhedral set* or a *polyhedron* if it is the intersection of finitely many closed half-spaces of $\mathbb{R}^d$. We say that K is *full-dimensional*, if we have $\dim(K) = d$, that is, if the interior of K is not empty. The proper faces of K that are $\subseteq$-maximal are called *facets* and those of dimension zero are called *vertices*. We observe that every face of K is also a polyhedral set.

Let $H_1, \ldots, H_m$ be closed half-spaces such that the intersection $\cap_{i=1}^{i=m} H_i$ is *irredundant*, that is, $\cap_{i=1}^{i=m} H_i \neq \cap_{i=1, j\neq i}^{i=m} H_i$ for all $1 \leq j \leq m$. We observe that this intersection is closed and convex. For each $i = 1 \cdots m$, let $\mathbf{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ such that H_i is defined by $\mathbf{a}_i^T\mathbf{x} \leq b_i$. We denote by $\mathbf{A}$ the $m \times d$ matrix $(\mathbf{a}_i^T, 1 \leq i \leq m)$ and by $\mathbf{b}$ the vector $(b_1, \ldots, b_d)^T$.

From now on, we assume that $K = \cap_{i=1}^{i=m} H_i$ holds. Such irredundant decomposition of a polyhedral set can be computed from an arbitrary intersection of finitely many closed half-spaces, in time polynomial in both d and m, using linear programming; see L. Khachian in [15]. The following property is essential. For every face F of K, there exists a subset I of $\{1, \ldots, m\}$ such that F corresponds to the set of solutions to the system of equations and inequalities

$$\mathbf{a}_i^T\mathbf{x} = b_i \quad \text{for } i \in I, \quad \text{and} \quad \mathbf{a}_i^T\mathbf{x} \leq b_i \quad \text{for } i \notin I \ .$$

This latter property has several important consequences. For each $i = 1 \cdots m$, the set $F_i = K \cap \{\mathbf{a}_i^T\mathbf{x} = b_i\}$ is a facet of K and the border of K equals $\cup_{i=1}^{i=m} F_i$. In particular, each proper face of K is contained in a facet of K. Each facet of a facet of K is the intersection of two facets of K. Moreover, if the $(m \times d)$-matrix A has full column rank, then the $\subseteq$-minimal faces are the vertices. The set $\mathcal{F}(K)$ is finite and has at most 2^m elements.

For $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, we say that $\mathbf{a}^T\mathbf{x} \leq b$ is an *implicit equality* in $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ if for all $\mathbf{x} \in \mathbb{R}^d$ we have

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \implies \mathbf{a}\mathbf{x} = b \ . \tag{1}$$

Following [19], we denote by $\mathbf{A}^=$ (resp. $\mathbf{A}^+$) and $\mathbf{b}^=$ (resp. $\mathbf{b}^+$) the rows of $\mathbf{A}$ and $\mathbf{b}$ corresponding to the implicit (resp. non-implicit) equalities. The following

properties are easy to prove. If K is not empty, then there exists $\mathbf{x} \in K$ satisfying both

$$\mathbf{A}^{=}\mathbf{x} \;=\; b^{=} \quad \text{and} \quad \mathbf{A}^{+}\mathbf{x} \;<\; b^{+} \;\;.$$

The facets of K are in 1-to-1 correspondence with the inequalities of $\mathbf{A}^{+}\mathbf{x} \leq \mathbf{b}^{+}$. In addition, if K is full-dimensional, then $\mathbf{A}^{+} = \mathbf{A}$ and $\mathbf{b}^{+} = \mathbf{b}$ both hold; moreover the system of inequalities $\mathbf{A}\mathbf{x} \leq b$ is a unique representation of K, up to multiplication of inequalities by positive scalars.

From now on and in the sequel of this paper, we assume that variables are ordered as $x_1 > \cdots > x_d$. We call initial coefficient, or simply *initial*, of an inequality $\mathbf{a}_i^T\mathbf{x} \leq b_i$, for $1 \leq i \leq m$, the coefficient of $a_i^T\mathbf{x}$ in its largest variable. Following the terminology of W. Pugh in [17], if v is the largest variable of the inequality $\mathbf{a}_i^T\mathbf{x} \leq b_i$, we say that this inequality is an *upper* (resp. *lower*) *bound* of v whenever the initial c of $\mathbf{a}_i^T\mathbf{x} \leq b_i$ is positive (resp. negative); indeed, we have $v \leq \frac{\gamma}{c}$ (resp. $v \geq \frac{\gamma}{c}$) where $\gamma = b_i - \mathbf{a}_i^T\mathbf{x} - c\,v$.

Canonical representation. Recall that we assume that none of the inequalities of $Ax \leq b$ is redundant. If K is full-dimensional and if the initial of each inequality in $Ax \leq b$ is 1 or -1, then we call $Ax \leq b$ the *canonical representation* of K w.r.t. the variable ordering $x_1 > \cdots > x_d$ and we denote it by $\mathsf{can}(K; x_1, \ldots, x_d)$.

We observe that the notion of *canonical representation* can also be expressed in a more geometrical and less algebraic way, that is, independently of any coordinate system. Assume again that K is full-dimensional and that the intersection $\cap_{i=1}^{i=n} H_i = K$ of closed half-spaces $H_1, \ldots, H_n$ is irredundant. Since K is full-dimensional, the supporting hyperplane of each facet of K must be the frontier of one half-space among $H_1, \ldots, H_n$. Clearly, two (or more) half-spaces among $H_1, \ldots, H_n$ may not have the same frontier without contradicting one of our hypotheses (K is full-dimensional, $\cap_{i=1}^{i=n} H_i$ is irredundant). Therefore, the half-spaces $H_1, \ldots, H_n$ are in one-to-one correspondence with the facets of K. This implies that there is a unique irredundant intersection of closed half-spaces equaling K and we denote it by $\mathsf{can}(K)$.

Projected representation. Let again $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ be the *canonical representation* of the polyhedral set K w.r.t. the variable ordering $x_1 > \cdots > x_d$. We denote by $\mathbf{A}^{x_1}$ (resp. $\mathbf{A}^{<x_1}$) and $\mathbf{b}^{x_1}$ (resp. $\mathbf{b}^{<x_1}$) the rows of $\mathbf{A}$ and $\mathbf{b}$ corresponding to the inequalities whose largest variable is x_1 (resp. less than x_1). For each upper bound $cx_1 \leq \gamma$ of x_1 and each lower bound $-ax_1 \leq -\alpha$ of x_1 (where $c > 0$, $a > 0$, $\gamma \in \mathbb{R}[x_2, \ldots, x_d]$ and $\alpha \in \mathbb{R}[x_2, \ldots, x_d]$ hold), we have a new inequality $c\alpha - a\gamma \leq 0$. Augmenting $\mathbf{A}^{<x_1}$ with all inequalities obtained in this way, we obtain a new linear system which represents a polyhedral set which is the standard projection of K on the $d-1$ least coordinates of $\mathbb{R}^d$, namely $(x_2, \ldots, x_d)$; hence we denote this latter polyhedral set by $\mathit{\Pi}^{x_2,\ldots,x_d}K$ and we call it the *real shadow* of K, following the terminology of [17]. The procedure by which $\mathit{\Pi}^{x_2,\ldots,x_d}K$ is computed from K is the well-known Fourier-Motzkin elimination procedure, see [15]. We call *projected representation* of K w.r.t. the variable ordering $x_1 > \cdots > x_d$ and denote by $\mathsf{proj}(K; x_1, \ldots, x_d)$ the linear system given by $\mathbf{A}^{x_1}\mathbf{x} \leq \mathbf{b}^{x_1}$ if $d = 1$ and, by the conjunction of $\mathbf{A}^{x_1}\mathbf{x} \leq \mathbf{b}^{x_1}$ and $\mathsf{proj}(\mathit{\Pi}^{x_2,\ldots,x_d}K; x_2, \ldots, x_d)$, otherwise.

3 Integer Solutions of Linear Equation Systems

We review how Hermite normal forms [6, 19] can be used to represent the integer solutions of systems of linear equations. Let $\mathbf{A} = (a_{i,j})$ and $H = (h_{i,j})$ be two matrices over $\mathbb{Z}$ with m rows and d columns, and let $\mathbf{b}$ be a vector over $\mathbb{Z}$ with d coefficients. We denote by r the rank of $\mathbf{A}$ and by h the maximum bit size of coefficients in the matrix $[\mathbf{A}\ \mathbf{b}]$. Definition 1 is taken from [14], see also [12].

Definition 1. *The matrix H is called a* column Hermite normal form (abbr. column HNF) *if there exists a strictly increasing map f from $[d-r+1, d] \cap \mathbb{Z}$ to $[1, m] \cap \mathbb{Z}$ satisfying the following properties for all $j \in [d-r+1, d] \cap \mathbb{Z}$:*

1. *for all integer i such $1 \leq i \leq m$ and that $i > f(j)$ both hold, we have $h_{i,j} = 0$,*
2. *for all integer k such that $j < k \leq d$ holds, we have $h_{f(j),j} > h_{f(j),k} \geq 0$,*
3. *the first $d-r$ columns of H are equal to zero.*

We say that H is the column Hermite normal form *of $\mathbf{A}$ if H is a column Hermite normal form and there exists a uni-modular $d \times d$-matrix U over $\mathbb{Z}$ such that we have $H = \mathbf{A}U$. When those properties hold, we call $\{f(d-r+1), \ldots, f(d)\}$ the* pivot row set *of $\mathbf{A}$.*

Remark 1. The matrix $\mathbf{A}$ admits a unique column Hermite normal form. Let H be this column Hermite normal form and let U be the uni-modular $(d \times d)$-matrix given in Definition 1. Let us decompose U as $U = [U_L, U_R]$ where U_L(resp. U_R) consist of the first $d-r$ (resp. last r) columns of U. Then we define $H_L := \mathbf{A}U_L$ and $H_R := \mathbf{A}U_R$. We have $H_L = \mathbf{0}^{m,d-r}$, where $\mathbf{0}^{m,d-r}$ is the zero-matrix with m rows and $d-r$ columns. We observe that U_R is a full column-rank matrix. Moreover, if $\mathbf{A}$ is full row-rank, that is, if $r = m$ holds, then H_R is non-singular.

Lemma 2 shows how to compute the integer solutions of the system of linear equations $\mathbf{Ax} = \mathbf{b}$ when $\mathbf{A}$ is full row-rank. In the general case, one can use Lemma 1 to reduce to the hypothesis of Lemma 2. While the construction of this latter lemma relies on the HNF, alternative approaches are available. For instance, one can use the *equation elimination procedure* of the Omega Test [17], However, no running-time estimates are known for that procedure.

Notation 1 *For $I \subseteq \{1, \ldots, m\}$, we denote by $\mathbf{A}_I$ (resp. $\mathbf{b}_I$) the sub-matrix (resp. vector) of $\mathbf{A}$ (resp. $\mathbf{b}$) consisting of the rows of $\mathbf{A}$ (coefficients of $\mathbf{b}$) with indices in I.*

Lemma 1. *Let I be the pivot row set of $\mathbf{A}$, as given in Definition 1. Assume that $\mathbf{Ax} = \mathbf{b}$ admits at least one solution in $\mathbb{R}^d$. Then, for any $\mathbf{x} \in \mathbb{R}^d$, we have*

$$\mathbf{Ax} = \mathbf{b} \iff \mathbf{A}_I\mathbf{x} = \mathbf{b}_I \ .$$

Proof. We clearly have $\{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}\} \subseteq \{\mathbf{x} \mid \mathbf{A}_I\mathbf{x} = \mathbf{b}_I\}$. We prove the reversed inclusion. Since I is the pivot row set of $\mathbf{A}$, one can check that $\mathrm{rank}(\mathbf{A}) = \mathrm{rank}(\mathbf{A}_I)$ holds. Since $\mathbf{Ax} = \mathbf{b}$ admits solutions, we have $\mathrm{rank}(\mathbf{A}) = \mathrm{rank}([\mathbf{A}\ \mathbf{b}])$. Similarly, we have $\mathrm{rank}(\mathbf{A}_I) = \mathrm{rank}([\mathbf{A}_I\ \mathbf{b}_I])$. Therefore, we have $\mathrm{rank}([\mathbf{A}\ \mathbf{b}]) = \mathrm{rank}([\mathbf{A}_I\ \mathbf{b}_I])$. Hence, any equation $\mathbf{a}^T\mathbf{x} = b$ in $\mathbf{Ax} = \mathbf{b}$ is a linear combination of the equations of $\mathbf{A}_I\mathbf{x} = \mathbf{b}_I$, thus $\{\mathbf{x} \mid \mathbf{A}_I\mathbf{x} = \mathbf{b}_I\} \subseteq \{\mathbf{x} \mid \mathbf{Ax} = \mathbf{b}\}$ holds.

Lemma 2. *We use the same notations as in Definition 1 and Remark 1. We assume that H_R is non-singular. Then, the system $\mathbf{Ax} = \mathbf{b}$ has an integer solution if and only if $H_R^{-1}\mathbf{b}$ is integral. In this case, all integral solutions to $\mathbf{Ax} = \mathbf{b}$ are given by $\mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}$ where*

1. *the columns of $\mathbf{P}$ consist of a $\mathbb{Z}$-basis of the linear space $\{\mathbf{x} : \mathbf{Ax} = \mathbf{0}\}$,*
2. *$\mathbf{q}$ is a particular solution of $\mathbf{Ax} = \mathbf{b}$, and*
3. *$\mathbf{t} = (t_1, \ldots, t_{d-r})$ is a vector of $d - r$ unknowns.*

The maximum absolute value of any coefficient in $\mathbf{P}$ (resp. $\mathbf{q}$) can be bounded over by $r^{r+1}L^{2r}$ (resp. $r^{r+1}L^{2r}$), where L is the maximum absolute value of any coefficient in $\mathbf{A}$ (resp. in either $\mathbf{A}$ or $\mathbf{b}$). Moreover, $\mathbf{P}$ and $\mathbf{q}$ can be computed within $O(mdr^2(\log r + \log L)^2 + r^4(\log r + \log L)^3)$ bit operations.

Proof. Except for the coefficient bound and running time estimates, we refer to [11] for a proof of this lemma. The running time estimate follows from Theorem 19 of [20] whereas the coefficient bound estimates are taken from [21]. □

Example 1. Let $\mathbf{A}$, H and U be as follows:

$$\mathbf{A} = \begin{pmatrix} 3 & 4 & -4 & -1 \\ 2 & -2 & 8 & 4 \\ 5 & 2 & 4 & 3 \\ 3 & 5 & -5 & -2 \\ 2 & -3 & 9 & 5 \end{pmatrix}, \quad H = \begin{pmatrix} 0 & -18 & -1 & -15 \\ 0 & \mathbf{18} & \mathbf{2} & \mathbf{16} \\ 0 & 0 & 1 & 1 \\ 0 & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix}, \quad U = \begin{pmatrix} -1 & 30 & -3 & -25 \\ 1 & -37 & 4 & 31 \\ 0 & -19 & 2 & 16 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The matrix H is the column HNF of $\mathbf{A}$, with unimodular matrix U and pivot row set $[2, 4, 5]$. We denote by H_R the sub-matrix of H whose coefficients are in bold fonts. Applying Lemma 1, we deduce that for any vector $\mathbf{b}$ such that $\mathbf{Ax} = \mathbf{b}$ admits one rational solution, we have:

$$\begin{cases} 3x_1 + 4x_2 - 4x_3 - x_4 = b_1 \\ 2x_1 - 2x_2 + 8x_3 + 4x_4 = b_2 \\ 5x_1 + 2x_2 + 4x_3 + 3x_4 = b_3 \\ 3x_1 + 5x_2 - 5x_3 - 2x_4 = b_4 \\ 2x_1 - 3x_2 + 9x_3 + 5x_4 = b_5 \end{cases} \Leftrightarrow \begin{cases} 2x_1 - 2x_2 + 8x_3 + 4x_4 = b_2 \\ 3x_1 + 5x_2 - 5x_3 - 2x_4 = b_4 \\ 2x_1 - 3x_2 + 9x_3 + 5x_4 = b_5 \end{cases}. \tag{2}$$

We apply Lemma 2: if $\mathbf{Ax} = \mathbf{b}$ is consistent over $\mathbb{Q}$ and if $H_R^{-1}[b_2, b_4, b_5]^T$ is integral, then all the integer solutions of the second equation system in Relation (2) are given by $\mathbf{x} = \mathbf{Pt} + \mathbf{q}$, where $\mathbf{P} = [-1, 1, 0, 1]^T$, $\mathbf{q} = [\frac{5}{3}b_2 - \frac{19}{3}b_4 - \frac{155}{3}b_5, -\frac{37}{18}b_2 + \frac{73}{9}b_4 + \frac{575}{9}b_5, -\frac{19}{18}b_2 + \frac{37}{9}b_4 + \frac{296}{9}b_5]^T$, $\mathbf{t} = (t_1)$ and t_1 is a new variable.

4 Integer Solutions of Linear Inequality Systems

In this section, we present an algorithm for computing the integer points of a polyhedron $K \subseteq \mathbb{R}^d$, that is, the set $K \cap \mathbb{Z}^d$. To do so, we adapt the Omega Test invented by W. Pugh [17] for deciding whether or not a polyhedral set has an integer point. Our algorithm decomposes the set $K \cap \mathbb{Z}^d$ into a disjoint union

$(K_1 \cap \mathbb{Z}^d) \cup \cdots \cup (K_s \cap \mathbb{Z}^d)$, where $K_1, \ldots, K_s$ are polyhedral sets in $\mathbb{R}^d$, for which the integer points can be represented in a sense specified in Section 4.2. Sect. 4.3 states the specifications of the main procedure while Sect. 3, 4.1, 4.4 4.5, 4.6. describe its main subroutines and its proof. We use the same notations as in Sect. 2. However, from now on, we assume that all matrix and vector coefficients are integer numbers, that is, elements of $\mathbb{Z}$. To be precise, we have the following.

Notation 2 *We consider a polyhedral set $K \subseteq \mathbb{R}^d$ given by an irredundant intersection $K = \cap_{i=1}^{i=m} H_i$ of closed half-spaces $H_1, \ldots, H_m$ such that, for each $i = 1, \ldots, m$, the half-space H_i is defined by $\mathbf{a}_i^T \mathbf{x} \leq b_i$, with $\mathbf{a}_i \in \mathbb{Z}^d$ and $b_i \in \mathbb{Z}$. The conjunction of those inequalities forms a system of linear inequalities that we denote by $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, as well as Σ. We do not assume that K is full-dimensional.*

4.1 Normalization of Linear Inequality Systems

The purpose of the procedure IntegerNormalize, presented below, is to solve the system consisting of the equations of $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ and substitute its solutions into the system consisting of the inequalities of $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. This process is performed by Steps (S2) to (S6) and relies on Lemmas 1 and 2; this yields Proposition 1, which provides the output specification of IntegerNormalize. Step (S1) is an optimization: performing it is not needed, but improves performance in practice.

When applied to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, the IntegerNormalize procedure proceeds as follows.

(S1) It computes $\mathsf{proj}(K; x_1, \ldots, x_d)$, obtaining a new system of linear inequalities that we denote again by Σ; if this proves that K has no rational points, then the procedure stops and returns $(\varnothing, \varnothing, \varnothing)$ implying that $K \cap \mathbb{Z}^d$ is empty,

(S2) for every inequality $\mathbf{a}\mathbf{x} \leq b$, let g be the absolute value of the GCD of coefficients in $\mathbf{a}$: if $g > 1$, replace $\mathbf{a}\mathbf{x} \leq b$ by $\frac{\mathbf{a}}{g}\mathbf{x} \leq \lfloor \frac{b}{g} \rfloor$.

(S3) Every pair of inequalities of the form $(\mathbf{a}_i^T \mathbf{x} \leq b_i, -\mathbf{a}_i^T \mathbf{x} \leq -b_i)$ is replaced by the equivalent equation, that is, $\mathbf{a}_i^T \mathbf{x} = b_i$; Every pair of inequalities of the form $(\mathbf{a}_i^T \mathbf{x} \leq b_i, \mathbf{a}_i^T \mathbf{x} \leq b_j)$ is replaced by $\mathbf{a}_i^T \mathbf{x} \leq \min(b_i, b_j)$.

(S4) Equations and inequalities form, respectively, a system of linear equations $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ and a system of linear inequalities $\mathbf{A}^{\leq}\mathbf{x} \leq \mathbf{b}^{\leq}$, as specified in Notation 3, so that the conjunction of these two systems is equivalent to Σ.

(S5) If $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ is empty, that is, if Σ has no equations, then the procedure stops returning $(\mathbf{x}, \varnothing, \mathbf{A}^{\leq}\mathbf{x} \leq \mathbf{b}^{\leq})$.

(S6) Proposition 1 is applied to $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$; if this proves that this latter system has no integer solutions, then the procedure stops returning $(\varnothing, \varnothing, \varnothing)$, otherwise the change of variables given by (3) is applied to $\mathbf{A}^{\leq}\mathbf{x} \leq \mathbf{b}^{\leq}$; as a result, the output of the IntegerNormalize procedure is the triple $(\mathbf{t}, \mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}, \mathbf{M}\mathbf{t} \leq \mathbf{v})$, where $\mathbf{t}, \mathbf{P}, \mathbf{q}, \mathbf{M}, \mathbf{v}$ are defined in Proposition 1.

Notation 3 *From now we consider an equation system $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ and an inequality system $\mathbf{A}^{\leq}\mathbf{x} \leq \mathbf{b}^{\leq}$. The matrices $\mathbf{A}^{=}, \mathbf{A}^{\leq}$ as well as the vectors $\mathbf{b}^{=}, \mathbf{b}^{\leq}$ have integer coefficients. The total number of rows in both $\mathbf{A}^{=}$ and $\mathbf{A}^{\leq}$ is m, each of $\mathbf{A}^{=}, \mathbf{A}^{\leq}$ has d columns, and $\mathbf{A}^{\leq}$ has e rows. We denote by L and h the maximum absolute value and maximal bit size of any coefficient in the matrix in either $[\mathbf{A}^{=}\ \mathbf{b}^{=}]$ or $[\mathbf{A}^{\leq}\ \mathbf{b}^{\leq}]$ respectively. We define $r := \mathrm{rank}(\mathbf{A}^{=})$.*

Proposition 1. *One can decide whether or not* $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ *has integer solutions. If this system has integer solutions, then, for any* $\varepsilon > 0$*, one can compute*

1. *a matrix* $\mathbf{P} \in \mathbb{Z}^{d \times (d-r)}$ *within* $O(m\,d\,r^{2+\varepsilon}\,h^3)$ *bit operations,*
2. *a vector* $\mathbf{q} \in \mathbb{Z}^d$ *within* $O(mdr^{2+\varepsilon}\,h^3)$ *bit operations,*
3. *a matrix* $\mathbf{M} \in \mathbb{Z}^{e \times (d-r)}$*, whose coefficients can be bounded over by* $dr^{r+1}L^{2r+1}$*, within* $O(m\,d^2\,r^{1+\varepsilon}h^3)$ *bit operations,*
4. *a vector* $\mathbf{v} \in \mathbb{Z}^e$*, whose coefficients can be bounded over by* $2dr^{r+1}L^{2r+1}$*, within* $O(m\,d^2\,r^{1+\varepsilon}h^3)$ *bit operations,*

such that an integer point $(x_1, \ldots, x_d) \in \mathbb{Z}^d$ *solves* $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ *and* $\mathbf{A}^{\leq}\mathbf{x} \leq \mathbf{b}^{\leq}$ *if and only if there exists an integer point* $(t_1, \ldots, t_{d-r}) \in \mathbb{Z}^{d-r}$ *such that we have*

$$\begin{cases} (x_1, \ldots, x_d)^T = \mathbf{P}(t_1, \ldots, t_{d-r})^T + (q_1, \ldots, q_d)^T \\ \mathbf{M}(t_1, \ldots, t_{d-r})^T \leq (v_1, \ldots, v_e)^T \end{cases} . \tag{3}$$

That is, one can perform the IntegerNormalize *procedure within* $O(m\,d^2\,r^{1+\varepsilon}\,h^3)$ *bit operations.*

Proof. We first observe that one can decide whether or not $\mathbf{A}^{=}\mathbf{x} = \mathbf{b}^{=}$ has solutions in $\mathbb{R}^d$, using standard techniques, say Gaussian elimination. If $\mathbf{A}^{=}$ is not full row-rank, this observation allows us to apply Lemma 1 and thus to reduce to the case where $\mathbf{A}^{=}$ is full row-rank, via the computation of the column HNF of $\mathbf{A}^{=}$. Hence, from now on, we assume that $\mathbf{A}^{=}$ is full row-rank. We apply Lemma 2 which yields the matrix $\mathbf{P}$ and the vector $\mathbf{q}$. Next, we compute $\mathbf{M}$ and $\mathbf{v}$ as follows: $\mathbf{M} := \mathbf{A}^{\leq}\mathbf{P}$ and $\mathbf{v} := -\mathbf{A}^{\leq}\mathbf{q} + \mathbf{b}$. The coefficient bounds and cost estimates for $\mathbf{M}$ and $\mathbf{v}$ follow easily from Lemma 2 and the inequality $r \leq d$. □

4.2 Representing the Integer Points

Applying IntegerNormalize to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$, produces a triple $(\mathbf{t}, \mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}, \mathbf{M}\mathbf{t} \leq \mathbf{v})$, with $\mathbf{P}, \mathbf{q}, \mathbf{M}, \mathbf{v}$ as in Proposition 1. Assume $\mathbf{t} \neq \varnothing$. Since the system $\mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}$ solves the $\mathbf{x}$-variables as functions of the $\mathbf{t}$-variables, we turn our attention to $\mathbf{M}\mathbf{t} \leq \mathbf{v}$. Definition 2 states conditions on $\mathbf{M}$ under which we view $(\mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}, \mathbf{M}\mathbf{t} \leq \mathbf{v})$ as a "solved system", that is, a system describing its integer solutions.

Definition 2. *Let* $\widehat{K}$ *be the polyhedron of* $\mathbb{Z}^{2d-r}$ *defined by the system of linear equations and inequalities given by* $\mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}$ *and* $\mathbf{M}\mathbf{t} \leq \mathbf{v}$*, in Relation (3). We say that this system is a* representation of the integer points *of the polyhedron* $\widehat{K}$ *whenever* $\mathbf{M}$ *has the following form:*

$$\begin{pmatrix} M_{11} & M_{12} & \cdots & M_{1,\ell-1} & M_{1,\ell} \\ & M_{22} & \cdots & M_{2,\ell-1} & M_{2,\ell} \\ & & \ddots & \vdots & \vdots \\ & & & M_{\ell-1,\ell-1} & M_{\ell-1,\ell} \\ & & & & M_{\ell,\ell} \end{pmatrix} , \tag{4}$$

where for each i, j *with* $1 \leq i, j \leq \ell$*, the block* $M_{i,j}$ *has* m_i *rows and* k_j *columns such that the following six assertions hold:*

(*i*) $k_1, \ldots, k_{\ell-1} \geq 1$, $k_\ell \geq 0$ *and* $k_1 + \cdots + k_\ell = d - r$;
(*ii*) $m_1, \ldots, m_{\ell-1} \geq 2$ *and* $m_\ell \geq 0$;
(*iii*) *for* $1 \leq i < \ell$, *each column in* $M_{i,i}$ *has both positive coefficients and negative coefficients, but no null coefficients;*
(*iv*) *if* $m_\ell > 0$ *holds, then in each column of* $M_{\ell,\ell}$, *all coefficients are non-zero and have the same sign;*
(*v*) (Consistency) *the system* $\mathbf{Mt} \leq \mathbf{v}$ *admits at least one integer point in* $\mathbb{Z}^{d-r}$;
(*vi*) (Extensibility) *for all* $1 < i < d - r$, *every integer point of* $\mathbb{R}^{d-r-i}$ *solving all the inequalities of* $\mathbf{Mt} \leq \mathbf{v}$ *involving* $t_{i+1}, \ldots, t_{d-r}$ *only extends to an integer point of* $\mathbb{R}^{d-r-i+1}$ *solving all the inequalities of* $\mathbf{Mt} \leq \mathbf{v}$ *involving* $t_i, \ldots, t_{d-r}$.

More generally, we say that $\mathbf{x} = \mathbf{P}\,\mathbf{t} + \mathbf{q}$ *and* $\mathbf{Mt} \leq \mathbf{v}$ *form a representation of the integer points of* $\widehat{K}$ *if* $\mathbf{M}$ *satisfies* (*i*) *to* (*vi*) *up to a permutation of its columns.*

Remark 2. Assume that the above matrix $\mathbf{M}$ satisfies the properties (*i*) to (*vi*) of Definition 2. Then, the values of the first $k_1 + \cdots + k_{\ell-1}$ (resp. last k_ℓ) variables of $\mathbf{t}$ are bounded (resp. unbounded) in the polyhedron given by $\mathbf{Mt} \leq \mathbf{v}$. For these reasons, we call those variables *bounded* and *unbounded* in $\mathbf{Mt} \leq \mathbf{v}$, respectively. Clearly, the original polyhedron $\mathbf{Ax} \leq \mathbf{b}$ is bounded if and only if $m_\ell = k_\ell = 0$.

4.3 The IntegerSolve Procedure: Specifications

We are ready to specify the main algorithm presented in this paper. This procedure, called IntegerSolve will be formally stated in Sect. 4.6. When applied to $\mathbf{Ax} \leq \mathbf{b}$, with the assumptions of Notation 2, IntegerSolve produces a decomposition of the integer points of the polyhedron K in the sense of the following.

Definition 3. *Let* $\mathbf{A}, \mathbf{x}, \mathbf{b}, K$ *be as in Notation 2. A sequence of pairs* $(\mathbf{y}_1, \Sigma_1)$, …, $(\mathbf{y}_s, \Sigma_s)$ *is called a* decomposition of the integer points of the polyhedron K *whenever the following conditions hold:*
(*i*) $\mathbf{y}_i$ *is a sequence of* $d_i \geq d$ *independent variables* $x_1, \ldots, x_d, x_{d+1}, \ldots, x_{d_i}$ *thus starting with* $\mathbf{x}$,
(*ii*) Σ_i *is a system of linear inequalities with* $\mathbf{y}_i$ *as unknown,*
(*iii*) Σ_i *is a representation of the integer points of a polyhedral set* K_i,

and we have $V_{\mathbb{Z}}(\Sigma) = V_{\mathbb{Z}}(\Sigma_1, \mathbf{x}) \cup \cdots \cup V_{\mathbb{Z}}(\Sigma_s, \mathbf{x})$, *where* $V_{\mathbb{Z}}(\Sigma)$ *denotes the set of the integer points of* Σ *and where* $V_{\mathbb{Z}}(\Sigma_i, \mathbf{x})$ *is defined as the set of the points* $(x_1, \ldots, x_d) \in \mathbb{Z}^d$ *such that there exists a point* $(x_{d+1}, \ldots, x_{d_i}) \in \mathbb{Z}^{d_i - d}$ *such that* $(x_1, \ldots, x_d, x_{d+1}, \ldots, x_{d_i})$ *solves* Σ_i.

In the sequel of Sect. 4, we shall propose and prove an algorithm satisfying the above specifications. The construction is by induction on $d \geq 1$. We observe that the case $d = 1$ is trivial. Indeed, in this case, K is necessarily an interval of the real line. Then, either $K \cap \mathbb{Z}$ is empty and IntegerSolve(Σ) returns the empty set, or $K \cap \mathbb{Z}$ is not empty and the system Σ is clearly a representation of the integer points of K in the sense of Definition 2. The case $d > 1$ will be treated in Sect. 4.6, after presenting the main subroutines of the IntegerSolve procedure.

4.4 The DarkShadow Procedure

Let $\mathbf{M}, \mathbf{v}$ be as in Proposition 1. Recall that we write $\mathbf{t} = (t_1, \ldots, t_{d-r})$ and assume $0 \leq r < d$. The system $\mathbf{Mt} \leq \mathbf{v}$ represents a polyhedral set that we denote by $K_{\mathbf{t}}$. We order the variables as $t_1 > \cdots > t_{d-r}$. We call DarkShadow the procedure stated by Algorithm 1, for which Proposition 2 serves as output specification. In Algorithm 1, the polyhedral set represented by $\mathbf{M}^{<t_1}\mathbf{t} \leq \mathbf{v}^{<t_1}$ (resp. Θ) is called the *dark shadow* of $K_{\mathbf{t}}$, denoted as $\mathbf{D}_{t_1}$ when **case** 1 (resp. **case** 2) holds.

Algorithm 1 DarkShadow($\mathbf{Mt} \leq \mathbf{v}$)

1: **case** 1: for all $1 \leq i \leq d-r$, the inequalities in t_i are either all lower bounds of t_i or all upper bounds of t_i
2: **return** $((t_2, \ldots, t_{d-r}), \mathbf{M}^{<t_1}\mathbf{t} \leq \mathbf{v}^{<t_1})$.
3: **case** 2: otherwise
4: re-order the variables, such that t_1 has both lower bounds and upper bounds.
5: initialize Δ to the empty set.
6: **for** each upper bound $c\,t_1 \leq \gamma$ of t_1, where $c > 0$, $\gamma \in \mathbb{Z}[t_2, \ldots, t_{d-r}]$ **do**
7: **for** each lower bound $-a\,t_1 \leq -\alpha$ of t_1, where $a > 0$, $\alpha \in \mathbb{Z}[t_2, \ldots, t_{d-r}]$ **do**
8: let $\Delta := \Delta \cup \{c\,\alpha - a\,\gamma \leq -(c-1)(a-1)\}$.
9: **end for**
10: **end for**
11: Let $\Theta_0 := \Delta \cup \mathbf{M}^{<t_1}\mathbf{t} \leq \mathbf{v}^{<t_1}$
12: Let Θ be the system obtained by removing from Θ_0 all redundant inequalities.
13: **return** $((t_2, \ldots, t_{d-r}), \Theta)$.

For the inequalities in the set Δ in Algorithm 1, we have the following.

Lemma 3 (Pugh [17]). *Let $c\,t_1 \leq \gamma$ be an upper bound of t_1 and $-a\,t_1 \leq -\alpha$ be an lower bound of t_1, where $c > 0$, $a > 0$, $\gamma \in \mathbb{Z}[t_2, \ldots, t_{d-r}]$ and $\alpha \in \mathbb{Z}[t_2, \ldots, t_{d-r}]$ hold. Then, every integer point $(t_2, \ldots, t_{d-r})$ satisfying $c\,\alpha - a\,\gamma \leq -(c-1)(a-1)$ extends to an integer point $(t_1, t_2, \ldots, t_{d-r})$ satisfying both $c\,t_1 \leq \gamma$ and $-a\,t_1 \leq \alpha$.*

Proposition 2. *Let $((t_2, \ldots, t_{d-r}), \Theta)$ be the output of the* DarkShadow *procedure. Then, every integer point of $V_{\mathbb{Z}}(\Theta, (t_2, \ldots, t_{d-r}))$ extends to an integer point solving $\mathbf{Ax} \leq \mathbf{b}$.*

Proof. If the DarkShadow procedure returns at Line 2 of Algorithm 1, the claim holds easily. Lemma 3 shows that any integer point $(t_2, \ldots, t_{d-r})$ solving Δ can be extended to an integer point solving $\mathbf{Mt} \leq \mathbf{v}$, thus with Proposition 1, to an integer point solving $\mathbf{Ax} \leq \mathbf{b}$. Therefore, if the DarkShadow procedure returns at Line 13, the claim also holds.

4.5 The GreyShadow Procedure

Let $\mathbf{M}$, $\mathbf{t}$, $\mathbf{v}, K_{\mathbf{t}}, \mathbf{D}_{t_1}$ be as in Sect. 4.4. We call *grey shadow* of $K_{\mathbf{t}}$, denoted by $\mathbf{G}_{t_1}$, the set-theoretic difference $\left(\Pi^{t_2,\ldots,t_{d-r}} K_{\mathbf{t}}\right) \setminus \mathbf{D}_{t_1}$. Algorithm 2 states the GreyShadow procedure, for which Lemma 4 serves as output specification.

Lemma 4. *Let* $\mathcal{G} = \{(\mathbf{u}_1, \mathbf{t} = \mathbf{P}_1\mathbf{u}_1+\mathbf{q}_1, \mathbf{M}_1\mathbf{u}_1 \le \mathbf{v}_1), \ldots, (\mathbf{u}_s, \mathbf{t} = \mathbf{P}_s\mathbf{u}_s+\mathbf{q}_s, \mathbf{M}_s\mathbf{u}_s \le \mathbf{v}_s)\}$ *be the output of Algorithm 2. Then, the disjoint union* $\biguplus_{1\le i\le s} V_{\mathbb{Z}}(\mathbf{t} = \mathbf{P}_i\mathbf{u}_i + \mathbf{q}_i \cup \mathbf{M}_i\mathbf{u}_i \le \mathbf{v}_i,\ \mathbf{t})$ *forms the set of the integer points of the grey shadow* $\mathbf{G}_{t_1}$.

Proof. The correctness of **case** 1 follows from the fact that $\mathbf{G}_{t_1}$ is empty when all $\mathbf{t}$-variables are unbounded. From now on, we consider **case** 2. At Line 12, all the $\mathbf{t}$-variables are solved by IntegerNormalize as functions of new variables $\mathbf{u}_i$. The fact that $\bigcup_{1\le i\le s} V_{\mathbb{Z}}(\mathbf{t} = \mathbf{P}_i\mathbf{u}_i + \mathbf{q}_i \cup \mathbf{M}_i\mathbf{u}_i \le \mathbf{v}_i,\ \mathbf{t})$ equals $\mathbf{G}_{t_1}$ follows from Section 2.3.1. of [17]. Now, at Line 8 of Algorithm 2, we add the constraint $c\alpha - a\gamma > -(c-1)(a-1)$ to Θ_2, while at Line 14, we use $c\alpha - a\gamma \le -(c-1)(a-1)$ to construct Υ in the next loop iteration. From that construction of Θ_2 and Υ, we easily deduce that the above union is disjoint.

Algorithm 2 GreyShadow($\mathbf{Mt} \le \mathbf{v}$)

1: **case** 1: for all $1 \le i \le d-r$, the inequalities in t_i are either all lower bounds of t_i or all upper bounds of t_i
2: **return** $(\varnothing, \varnothing, \varnothing)$
3: **case** 2: otherwise
4: Re-order the variables, such that t_1 has both lower bounds and upper bounds.
5: Initialize both Υ and $\mathcal{G}$ to the empty set; the former set will be a set of linear inequalities while the latter will form the result of the procedure.
6: **for** each upper bound $c\,t_1 \le \gamma$ of t_1, where $c > 0$, $\gamma \in \mathbb{Z}[t_2, \ldots, t_h]$ **do**
7: **for** each lower bound $-a\,t_1 \le -\alpha$ of t_1, where $a > 0$, $\alpha \in \mathbb{Z}[t_2, \ldots, t_h]$ **do**
8: let $\Theta_2 := \Upsilon \cup \mathbf{Mt} \le \mathbf{v} \cup \{c\alpha - a\gamma > -(c-1)(a-1)\}$,
9: **for** each non-negative integer $i \le \frac{ca-c-a}{c}$ **do**
10: check whether $at_1 = \alpha + i$ is consistent over $\mathbb{Z}$ using Lemma 2,
11: **case** no: move to the next iteration,
12: **case** yes: let $\mathcal{G} := \mathcal{G} \cup$ IntegerNormalize$(\{at_1 = \alpha + i\} \cup \Theta_2)$,
13: **end for**
14: let $\Upsilon := \Upsilon \cup \{c\alpha - a\gamma \le -(c-1)(a-1)\}$.
15: **end for**
16: **return** $\mathcal{G}$.
17: **end for**

4.6 The IntegerSolve Procedure: Algorithm

We are ready to state an algorithm satisfying the specifications of IntegerSolve introduced in Sect. 4.3. The recursive nature of this algorithm leads us to define an "inner procedure", called $\mathsf{IntegerSolve}_0$, of which IntegerSolve is a wrapper function. The procedure $\mathsf{IntegerSolve}_0$ takes as input the system to be solved, namely $\mathbf{Ax} \le \mathbf{b}$, together with another system of linear equations and inequalities, denoted by E, see Notation 4. This second system E keeps track of the

relations between those variables that have already been solved and those that remain to be solved. To be more precise, the procedure $\mathsf{IntegerSolve}_0$, see Algorithm 3, relies on IntegerNormalize and thus introduces new variables when solving systems of linear equations over $\mathbb{Z}$. For this reason, variables appearing in E may not be present in $\mathbf{x}$ and we need another vector of variables, namely $\mathbf{y} = (y_1, \ldots, y_{d'})$, to denote the unknowns of E that are regarded as "solved".

Notation 4 *We denote by E a second system of linear equations and inequalities, with coefficients in $\mathbb{Z}$ and with $\mathbf{y} \oplus \mathbf{x}$ as "unknown" vector, where $\mathbf{y} \oplus \mathbf{x}$ denotes the concatenated vector $(y_1, \ldots, y_{d'}, x_1, \ldots, x_d)$. In fact, the variables of $\mathbf{y}$ are regarded as solved by the equations and inequalities of E, meanwhile those of $\mathbf{x}$ remain to be solved. Hence, we can view the conjunction of the systems $\mathbf{Ax} \leq \mathbf{b}$ and E as a system of linear equations and inequalities with $\mathbf{y} \oplus \mathbf{x}$ as unknown vector, defining a polyhedron K^E in $\mathbb{R}^{d'+d}$.*

Theorem 1 states, that Algorithm 3 returns a decomposition (in the sense of Definition 3) of the integer points of the polyhedron K^E, defined in Notation 4. From Algorithm 3, we easily implement the IntegerSolve procedure (as specified in Sect. 4.3) with the call $\mathsf{IntegerSolve}_0(\{\,\}, \{\,\}, \mathbf{x}, \mathbf{Ax} \leq \mathbf{b})$.

Theorem 1. *Algorithm 3 terminates and returns a decomposition of the integer points of the polyhedron K^E.*

Proof. We first prove termination. Lines 1 to 21 in Algorithm 3 handle the case where $\mathbf{Ax} \leq \mathbf{b}$ has a single unknown. This is simply done by case inspection. Consider now the case where $\mathbf{Ax} \leq \mathbf{b}$ has more than one variable. The calls to the procedures DarkShadow and Greyshadow at Lines 29 and 32 generate the input to the recursive calls. From Lines 2 and 13 of Algorithm 1, and Lines 2 and 12 of Algorithm 2, we deduce that the number of unknowns decreases at least by one after each recursive call. Therefore, Algorithm 3 terminates.

Next we prove that Algorithm 3 is correct. Let $(\mathbf{y}_1, \Sigma_1)$, …, $(\mathbf{y}_s, \Sigma_s)$ be the output of Algorithm 3 where each Σ_i is a system of linear inequalities with $\mathbf{y}_i$ as unknown. The fact that each Σ_i is a representation of the integer points of the polyhedron it defines, can be established by induction on the length of $\mathbf{y}_i$. To give more details, the properties required by Definition 2 are easy to check in the case $d = 1$. For the cases $d > 1$, these properties, in particular the consistency and the extensibility, follow from the way the set E is incremented at Lines 27 and 33, as well as from Proposition 2. Finally, the fact that the integer points of the input system of the initial call to Algorithm 3 are given by the integer points of $\Sigma_1, \ldots, \Sigma_s$ can be established by induction on the length of $\mathbf{y}_i$, thanks to Lemma 4.

Algorithm 3 $\mathsf{IntegerSolve}_0(\mathbf{y},\ E,\ \mathbf{x},\ \mathbf{Ax} \le \mathbf{b})$

Let d be the cardinality of $\mathbf{x}$;
case $d = 1$
  let $\mathbf{x} = \{x\}$, solve $\mathbf{Ax} \le \mathbf{b}$ over $\mathbb{R}$,
  case only lower bounds of x exist in $\mathbf{Ax} \le \mathbf{b}$
    the solution to $\mathbf{Ax} \le \mathbf{b}$ over $\mathbb{R}$ is $\{x : -x \le q_1\}$ for some $q_1 \in \mathbb{R}$,
    $\mathbf{y} := \mathbf{y} \oplus \mathbf{x}$ and $E := E \cup \{-x \le \lfloor q_1 \rfloor\}$;
    return $\{(\mathbf{y},\ E)\}$
  case only upper bounds of x exist in $\mathbf{Ax} \le \mathbf{b}$
    the solution to $\mathbf{Ax} \le \mathbf{b}$ over $\mathbb{R}$ is $\{x : x \le q_2\}$ for some $q_2 \in \mathbb{R}$,
    $\mathbf{y} := \mathbf{y} \oplus \mathbf{x}$ and $E := E \cup \{x \le \lfloor q_2 \rfloor\}$;
    return $\{(\mathbf{y},\ E)\}$
  case both lower bounds and upper bounds of x exist in $\mathbf{Ax} \le \mathbf{b}$
    the solution to $\mathbf{Ax} \le \mathbf{b}$ over $\mathbb{R}$ is $\{x : x \le q_3$ and $-x \le q_4\}$ for some $q_3,\ q_4 \in \mathbb{R}$,
    case $\lfloor q_3 \rfloor > -\lfloor q_4 \rfloor$
      $\mathbf{y} := \mathbf{y} \oplus \mathbf{x}$ and $E := E \cup \{x \le \lfloor q_3 \rfloor,\ -x \le \lfloor q_4 \rfloor\}$;
      return $\{(\mathbf{y},\ E)\}$
    case $\lfloor q_3 \rfloor = -\lfloor q_4 \rfloor$
      $\mathbf{y} := \mathbf{y} \oplus \mathbf{x}$, $E := \mathrm{eval}(E, x = \lfloor q_3 \rfloor) \cup \{x = \lfloor q_3 \rfloor\}$,
      return $\{(\mathbf{y},\ E)\}$
    case $\lfloor q_3 \rfloor < -\lfloor q_4 \rfloor$
      return $\{(\varnothing, \varnothing)\}$
case $d > 1$
  $(\mathbf{t}, \mathbf{x} = \mathbf{Pt} + \mathbf{q},\ \mathbf{Mt} \le \mathbf{v}) := \mathsf{IntegerNormalize}(\mathbf{Ax} \le \mathbf{b})$,
  case $(\mathbf{t}, \mathbf{x} = \mathbf{Pt} + \mathbf{q},\ \mathbf{Mt} \le \mathbf{v}) = (\varnothing, \varnothing, \varnothing)$
    return $\{(\varnothing, \varnothing)\}$
  case $(\mathbf{t}, \mathbf{x} = \mathbf{Pt} + \mathbf{q},\ \mathbf{Mt} \le \mathbf{v}) \ne (\varnothing, \varnothing, \varnothing)$
    $\mathbf{y} := \mathbf{y} \oplus \mathbf{x}$, $E := \mathrm{eval}(E,\ \mathbf{x} = \mathbf{Pt} + \mathbf{q}) \cup \mathbf{x} = \mathbf{Pt} + \mathbf{q} \cup \mathbf{M}^{t_1}\mathbf{t} \le \mathbf{v}^{t_1}$,
    $\mathcal{G} := \varnothing$,
    $(\mathbf{t}', \Theta) := \mathsf{DarkShadow}(\mathbf{Mt} \le \mathbf{v})$,
    $\mathbf{y} := \mathbf{y} \oplus \{t_1\}$,
    $\mathcal{G} := \mathcal{G} \cup \mathsf{IntegerSolve}_0(\mathbf{y}, E, \mathbf{t}', \Theta)$;
    for $(\mathbf{u}, E_u, \mathbf{M}_u\mathbf{u} \le \mathbf{v_u}) \in \mathsf{Greyshadow}(\mathbf{Mt} \le \mathbf{v})$ **do**
      $\mathcal{G} := \mathcal{G} \cup \mathsf{IntegerSolve}_0(\mathbf{y} \cup \mathbf{t}, E \cup E_{\mathbf{u}}, \mathbf{u}, \mathbf{M_u u} \le \mathbf{v_u})$
    end for
    return $\mathcal{G}$

Bibliography

[1] Philippe Aubry, Daniel Lazard, and Marc Moreno Maza. On the theories of triangular sets. *J. Symb. Comput.*, 28:105–124, July 1999.

[2] Alexander I. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19(4):769–779, 1994.

[3] Alexander I. Barvinok. *Integer Points in Polyhedra*. Contemporary mathematics. European Mathematical Society, 2008.

[4] Matthias Beck. *Integer Points in Polyhedra–Geometry, Number Theory, Representation Theory, Algebra, Optimization, Statistics: AMS-IMS-SIAM Joint Summer Research Conference, June 11-15, 2006, Snowbird, Utah*. Contemporary mathematics - American Mathematical Society. American Mathematical Society, 2008.

[5] Changbo Chen, Xiaohui Chen, Abdoul-Kader Keita, Marc Moreno Maza, and Ning Xie. MetaFork: A compilation framework for concurrency models targeting hardware accelerators and its application to the generation of parametric CUDA kernels. In *Proceedings of CASCON 2015*, pages 70–79, 2015.

[6] Henri Cohen. *A course in computational algebraic number theory*, volume 138. Springer Science & Business Media, 2013.

[7] Paul Feautrier. Parametric integer programming. *RAIRO Recherche Opérationnelle*, 22, 1988. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.9957&rep=rep1&type=pdf`.

[8] Paul Feautrier. Automatic parallelization in the polytope model. In *The Data Parallel Programming Model: Foundations, HPF Realization, and Scientific Applications*, pages 79–103, London, UK, UK, 1996. Springer-Verlag. `http://dl.acm.org/citation.cfm?id=647429.723579`.

[9] Michael Jo Fischer, Michael J Fischer, and Michael O Rabin. Super-exponential complexity of presburger arithmetic. Technical report, Cambridge, MA, USA, 1974.

[10] Branko Grünbaum. *Convex Polytops*. Springer, New York, NY, USA, 2003.

[11] Ming S. Hung and Walter O. Rom. An application of the hermite normal form in integer programming. *Linear Algebra and its Applications*, 140:163 – 179, 1990.

[12] Rui-Juan Jing, Chun-Ming Yuan, and Xiao-Shan Gao. A polynomial-time algorithm to compute generalized hermite normal form of matrices over $\mathbb{Z}[x]$. *CoRR*, abs/1601.01067, 2016.

[13] C.N. Jones, E.C. Kerrigan, and J.M. Maciejowski. On polyhedral projection and parametric programming. *Journal of Optimization Theory and Applications*, 138(2):207–220, 2008.

[14] Ravindran Kannan and Achim Bachem. Polynomial algorithms for computing the smith and hermite normal forms of an integer matrix. *siam Journal on Computing*, 8(4):499–507, 1979.

[15] Leonid Khachiyan. Fourier-motzkin elimination method. In Christodoulos A. Floudas and Panos M. Pardalos, editors, *Encyclopedia of Optimization, Second Edition*, pages 1074–1077. Springer, 2009.

[16] Matthias Köppe and Sven Verdoolaege. Computing parametric rational generating functions with a primal barvinok algorithm. *Electr. J. Comb.*, 15(1), 2008.

[17] William Pugh. The omega test: a fast and practical integer programming algorithm for dependence analysis. In Joanne L. Martin, editor, *Proceedings Super-*

computing '91, Albuquerque, NM, USA, November 18-22, 1991, pages 4–13. ACM, 1991.

[18] William Pugh. Counting solutions to presburger formulas: How and why. In Vivek Sarkar, Barbara G. Ryder, and Mary Lou Soffa, editors, *Proceedings of the ACM SIGPLAN'94 Conference on Programming Language Design and Implementation (PLDI), Orlando, Florida, USA, June 20-24, 1994*, pages 121–134. ACM, 1994.

[19] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[20] Arne Storjohann. *A fast practical deterministic algorithm for triangularizing integer matrices*. Citeseer, 1996.

[21] Arne Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2000.

[22] David Wonnacott. Omega test. In *Encyclopedia of Parallel Computing*, pages 1355–1365. 2011.

Software

We have implemented the algorithm presented in the first paper with in the `Polyhedra` library in MAPLE. This library is publicly available in source on the download page of the `RegularChains` library at `www.regularchains.org`

Acknowledgements

The authors would like to thank IBM Canada Ltd (CAS project 880) and NSERC of Canada (CRD grant CRDPJ500717-16), as well as the University of Chinese Academy of Sciences, UCAS Joint PhD Training Program, for supporting their work.