Centre Fédéré en Vérification

Technical Report number 2007.83

Computing convex hulls by automata iteration

François Cantin, Axel Legay, Pierre Wolper

This work was partially supported by a FRFC grant: 2.4530.02

http://www.ulb.ac.be/di/ssd/cfv

Computing convex hulls by automata iteration

François Cantin, Axel Legay, and Pierre Wolper

Université de Liège,
Institut Montefiore, B28,
4000 Liège, Belgium
{cantin,legay,pw}@montefiore.ulg.ac.be,
http://www.montefiore.ulg.ac.be/~{cantin,legay,pw}/

Abstract. This paper considers the problem of computing the real convex hull of a finite set of n-dimensional integer vectors. The starting point is a finite-automaton representation of the initial set of vectors. The proposed method consists in computing a sequence of automata representing approximations of the convex hull and using extrapolation techniques to compute the limit of this sequence. The convex hull can then be directly computed from this limit in the form of an automaton-based representation of the corresponding set of real vectors. The technique is quite general and has been successfully implemented.

1 Introduction

Automata-based representations for sets of integer and real vectors have been a subject of growing interest in recent years [BB03,BH06,EK06,Ler04,LS05]. While usually not optimal for specific problems, they provide much stronger generality and canonicity than other representations. For instance, in this context, combining real and integer constraints is very simple once the right framework has been set up [BJW05]. The benefit of using automata-based representations for arithmetic sets could be even greater if one could, whenever appropriate, freely move between this and other representations such as explicit constraints. Going from constraints to automata has long been successfully studied [BC96,Boi99,BW02,Kla04,WB00], but going in the other direction is substantially more difficult. Nevertheless, it has been shown that it is possible [Ler05] to construct constraint formulas from automata representing sets of integer vectors and that, under some restrictions, this can be done quite effectively [Lat04,Lat06].

One case that is not well handled though is that of finite sets of integer vectors. Indeed, imagine that a finite set of integers is represented by constraints and that an automaton representing this set is built from these. Since the set is finite, this automaton is acyclic and lacks the structure needed to construct the corresponding constraints. One is thus stuck with the automaton or with an enumerative representation of the set it defines, which is far from satisfactory. The work presented here was motivated by this problem with the idea of solving it along the following lines. The first step is to compute, as an automaton, a minimal dense set of real-vectors that contains the finite set of integers. On this no-longer acyclic automaton, techniques similar to those of [Lat04,Lat06,Ler05] could then be applied to obtain constraints. This paper proposes a solution for the first step in the form of an automata-based technique for computing the real convex hull of a finite automaton-represented set of integers.

Computing convex hulls is of course a well studied problem of independent interest. We dot not claim to improve on existing algorithms when they apply, but independently of the intended application outlined above, our approach is of interesting generality in the sense that it does not impose any dimension restrictions and could also be extended to automata-represented infinite sets of reals and integers. It is also a nice application of automata extrapolation techniques, involves a delightful mix of concepts of techniques, and adds another stone to the cathedral of finite-automata applications.

In simple terms, our approach proceeds as follows. We start with an automata-based representation of a finite set of integer vectors. We then repeatedly apply a transformation to this automaton that adds to the set the vectors that are mid-way between those it includes. This yields an infinite sequence of automata-represented sets. The limit of this sequence is then computed as an automaton, using the techniques of [BLW03,BLW04]. This limit is not quite the convex closure since we prove that it will only contain convex combinations of the initial vectors with coefficients that are multiples of a negative power of 2. This limit thus needs to be "completed" in order to obtain the convex hull and we show that this can be done by computing its topological closure. Bar a technical point due to the fact that some reals have two encodings in our framework, the computation of the topological closure is quite an easy step. This being done, the closure is obtained.

For the procedure to work correctly, we are of course dependent on the result of the extrapolation being exact. This can be checked as described in [BLW03,BLW04], but one interest twist is that checking safety (enough is obtained) can be done much more easily (and just as correctly) after computing the topological closure. This is due to the fact that taking the topological closure yields an automaton that falls within an easier to handle class. Checking preciseness (nothing is added) with the techniques of [BLW03,BLW04] is probably not practical, but one could do this by arithmetic techniques if the extra step of converting the computed hull to constraints can be taken.

Our approach has been fully implemented and the implementation has actually served as a guide to hone our results. The implementation has been tested quite extensively and performs well on sets for which the automaton representation is efficient. We present a series of experimental results to illustrate how our technique performs.

2 Automata-theoretic background

In this section we recall some automata-theoretic concepts used in the paper and give a brief description of the use of automata to encode sets of integer and real vectors.

2.1 Automata on infinite words

An infinite word (or ω-word) w over an alphabet Σ is a mapping $w : \mathbb{N} \to \Sigma$ from the natural numbers to Σ. The length-k prefix of an infinite word w, i.e. the finite-word $w(0), w(1), \ldots, w(k-1)$, will be denoted by $\mathit{pref}_k(w)$.

A Büchi automaton on infinite words is a five-tuple $A = (Q, \Sigma, \delta, q_0, F)$, where

- Q is a finite set of states;
- Σ is the input alphabet;
- δ is the transition function and is of the form $\delta : Q \times \Sigma \to 2^Q$ if the automaton is nondeterministic and of the form $\delta : Q \times \Sigma \to Q$ if the automaton is deterministic;
- q_0 is the initial state;
- F is a set of accepting states.

A run π of a Büchi automaton $A = (Q, \Sigma, \delta, q_0, F)$ on an ω-word w is a mapping $\pi : \mathbb{N} \to Q$ such that $\pi(0) = q_0$ and for all $i \geq 0$, $\pi(i+1) \in \delta(\pi(i), w(i))$ (nondeterministic automata) or $\pi(i+1) = \delta(\pi(i), w(i))$ (deterministic automata).

Let $\mathit{inf}(\pi)$ be the set of states that occur infinitely often in a run π. A run π is said to be accepting if $\mathit{inf}(\pi) \cap F \neq \emptyset$. An ω-word w is accepted by a Büchi automaton if that automaton has some accepting run on w. The language $L_\omega(A)$ of infinite words defined by a Büchi automaton A is the set of ω-words it accepts.

A co-Büchi automaton is defined exactly as a Büchi automaton except that its accepting runs are those for which $\mathit{inf}(\pi) \cap F = \emptyset$.

We will also use the notion of *weak* automata [MSS86]. For a Büchi automaton $A = (Q, \Sigma, \delta, q_0, F)$ to be *weak*, there has to be a partition of its state set Q into disjoint subsets $Q_1, \ldots, Q_m$ such that

- for each of the Q_i either $Q_i \subseteq F$ or $Q_i \cap F = \emptyset$; and
- there is a partial order $\leq$ on the sets $Q_1, \ldots, Q_m$ such that for every $q \in Q_i$ and $q' \in Q_j$ for which, for some $a \in \Sigma$, $q' \in \delta(q, a)$ ($q' = \delta(q, a)$ in the deterministic case), $Q_j \leq Q_i$.

A weak automaton is thus a Büchi automaton such that each of the strongly connected components of its graph contains either only accepting or only non-accepting states.

For more details, a survey of automata on infinite words can be found in [Tho90].

2.2 Automata-based representations of sets of integers and reals

In this section, we briefly introduce the representation of sets of integer and real vectors by finite automata. Details are only given for the case of real vectors, the case of integer vectors being a simplification of the former where automata on finite words replace automata on infinite words. A survey on this topic can be found in [BW02].

In order to make a finite automaton recognize numbers, one needs to establish a mapping between these and words. Our encoding scheme corresponds to the usual notation for reals and relies on an arbitrary integer base $r > 1$. We encode a number x in base r, most significant digit first, by words of the form $w_I \star w_F$, where w_I encodes the integer part x_I of x as a finite word over $\{0, \ldots, r-1\}$, the special symbol "$\star$" is a separator, and w_F encodes the fractional part x_F of x as an infinite word over $\{0, \ldots, r-1\}$. Negative numbers are represented by their r's complement. The length p of $|w_I|$, which we refer to as the *integer-part length* of w, is not fixed but must be large enough for $-r^{p-1} \leq x_I < r^{p-1}$ to hold.

According to this scheme, each number has an infinite number of encodings, since their integer-part length can be increased unboundedly. In addition, the rational numbers whose denominator has only prime factors that are also factors of r have two distinct encodings with the same integer-part length. For example, in base 10, the number 11/2 has the encodings $005 \star 5(0)^\omega$ and $005 \star 4(9)^\omega$, "ω" denoting infinite repetition. We call these respectively the *high* and *low* encodings and refer collectively to them as *dual* encodings.

To encode a vector of real numbers, we represent each of its components by words of identical integer-part length. This length can be chosen arbitrarily, provided that it is sufficient for encoding the vector component with the highest magnitude. An encoding of a vector $\mathbf{x} \in \mathbb{R}^n$ can indifferently be viewed either as a n-tuple of words of identical integer-part length over the alphabet $\{0, \ldots, r-1, \star\}$, or as a single word w over the alphabet $\{0, \ldots, r-1\}^n \cup \{\star\}$.

Example 1. In base 2, the vector $(-2, 12.3)$ can be encoded as

$$(11110 \star 0^\omega, 01100 \star 0[1001]^\omega)$$

or as the word

$$(1,0)(1,1)(1,1)(1,0)(0,0) \star (0,0)[(0,1)(0,0)(0,0)(0,1)]^\omega.$$

Using an alphabet of size $r^n + 1$ is clearly going to be problematic as soon as n starts to grow. The solution proposed in [BJW05,WB00] is to read the digits of the various components of the vector serially, in a round robin way, thus reducing the alphabet size to the perfectly manageable $r + 1$. This scheme is referred as the *serial encoding* as opposed to the *simultaneous encoding* in which the alphabet consists of tuples of digits.

Example 2. Using the serial encoding, the vector $(-2, 12.3)$ can be encoded in base 2 as

$$1011111000 \star 0001[01000001]^\omega.$$

Implementations obviously use the serial encoding, but the simultaneous encoding is convenient for presentation and proof purposes. The set of all the encodings of a vector $\mathbf{v} \in \mathbb{R}^n$ is denoted by $W(\mathbf{v})$. This definition directly generalizes to sets of vectors.

Real vectors being encoded by infinite words, a set of vectors can be represented by an infinite-word automaton accepting the corresponding encodings. Since a real vector has an infinite number of possible encodings, we have to choose which of these the automata will recognize. A natural choice is to accept all encodings. This leads to the following definition.

Definition 1. *Let $n > 0$ and $r > 1$ be integers. A base-r n-dimension serial* Real Vector Automaton (RVA) *is a Büchi automaton $\mathcal{A} = (Q, \Sigma, \delta, Q_0, F)$ automaton over the alphabet $\Sigma = \{0, \dots, r-1\} \cup \{\star\}$, such that*

- *Every word accepted by $\mathcal{A}$ is a serial encoding in base r of a vector in $\mathbb{R}^n$, and*
- *For every vector $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{A}$ accepts either all the encodings of $\mathbf{x}$ in base r, or none of them.*

An RVA is said to *represent* the set of vectors encoded by the words that belong to its accepted language. The set of *fractional* states of a RVA A, denoted by Q_F^A, is the subset of Q that contains all the states of A that can be reached after having followed a transition labeled by $\star$. Efficient algorithms have been developed for constructing RVA representing the sets of solutions of systems of linear equations and inequations [BRW98]. Furthermore, in [BJW05], it is shown that if the represented set can be defined in the first-order theory of linear constraints, then one can work with RVAs that are weak deterministic Büchi automata. Weak deterministic Büchi automata are less expressive than general Büchi automata, but the algorithms for manipulating them are considerably simpler and they accept a unique normal form [Löd01]. If not explicitly mentioned, we assume that the RVAs we manipulate are minimal weak deterministic Büchi automaton. Also, since our implementation works with a base 2 representation, we will present all our results in this context, knowing that they can be generalized to other bases.

3 Convex hulls and topological concepts

We recall a few notations and definitions that are used throughout the paper.

Let $\mathbb{Z}$ and $\mathbb{R}$ be respectively the sets of integers and reals, and let $\mathbb{Z}^n$ and $\mathbb{R}^n$ denote the usual n-dimensional Euclidean vector spaces. Vectors are written in boldface, e.g. $\mathbf{x}$, and scalars without emphasis, e.g. a. The ith component of a vector $\mathbf{x} \in \mathbb{R}^n$ is denoted by $\mathbf{x}[i]$. We use the usual definition of a convex hull.

Definition 2. *Given a set $E \subset \mathbb{R}^n$, the* convex hull *of E is the set $Conv(E) \subset \mathbb{R}^n$ defined by*

$$Conv(E) = \{\mathbf{x} \mid \exists \mathbf{x}_1, \dots, \mathbf{x}_k \in E\ \exists \lambda_1, \dots, \lambda_k \in [0,1]\ \mathbf{x} = \sum_{i=1}^{k} \lambda_i \mathbf{x}_i\ \wedge \sum_{i=1}^{k} \lambda_i = 1\}$$

The *Euclidean distance* between two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$, denoted by $|\mathbf{x} - \mathbf{x}'|$ is the real number $\sqrt{\sum_{i=1}^{n}(\mathbf{x}[i] - \mathbf{x}'[i])^2}$. The *open ball* centered in $\mathbf{x} \in \mathbb{R}^n$ with a radius $\epsilon > 0$ is the subset $B_{(\mathbf{x},\epsilon)} = \{\mathbf{x}' \mid |\mathbf{x} - \mathbf{x}'| < \epsilon\}$. A set $E \subset \mathbb{R}^n$ is said to be *open* if for any $\mathbf{x} \in E$ there exists $\epsilon > 0$ such that $B_{(\mathbf{x},\epsilon)} \subset E$. A *closed* set E is a subset of $\mathbb{R}^n$ such that $\mathbb{R}^n \setminus E$ is an open set. We use the concept of *topological closure* of a set.

Definition 3. *Given a set $E \subset \mathbb{R}^n$, the* topological closure $TC(E)$ *of E is the smallest closed set that contains E.*

When dealing with infinite words, we will be working with the topology on words induced by the distance defined by

$$d(w, w') = \begin{cases} \frac{1}{|common(w,w')|+1} & \text{if } w \neq w' \\ 0 & \text{if } w = w', \end{cases}$$

where $common(w, w')$ denotes the longest common prefix of w and w'. Notice that, among words that validly encode vectors, words that are topologically close encode vectors that are close according to the Euclidean distance, the reverse also being true except for the cases where dual encodings can appear.

4 Computing Convex Hulls

In this section, we propose a technique to compute the convex hull over $\mathbb{R}^n$ of a *finite* set $E = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_k}\}$ defined over $\mathbb{Z}^n$.

We proceed by constructing a sequence of approximations of the convex hull by adding the vectors that are mid-way between those obtained so far. More formally, we use the following definitions.

Definition 4. *The median sequence of E is the* infinite *sequence $E_0, E_1, E_2, \dots$ such that (1) $E_0 = E$ and (2) $E_{i+1} = E_i \cup \{(\mathbf{x_1} + \mathbf{x_2})/2 \mid \mathbf{x_1}, \mathbf{x_2} \in E_i\}$ for each $i \in \mathbb{N}$.*

The *limit* of the median sequence of E, denoted by E^*, is defined by $\bigcup_{i=0}^{\infty} E_i$. It is easy to see that each vector $\mathbf{v}$ of E^* is also a vector of $Conv(E)$. However, E^* is not the complete convex hull, but can be characterized using the following definition.

Definition 5. *The 2-chopped convex hull of a finite subset $E = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_k}\}$ of $\mathbb{Z}^n$ is the* maximal *subset $Conv_{2^*}(E)$ of $Conv(E)$, where for each $\mathbf{v} \in Conv_{2^*}(E)$, $\mathbf{v} = \sum_{i=1}^{k} \lambda_i \mathbf{x_i}$ with $\lambda_i \in [0, 1]$, $\sum_{i=1}^{k} \lambda_i = 1$, and $\lambda_i = \frac{k_i}{2^{m_i}}$ for $k_i, m_i \in \mathbb{N}$ and $i \in [1, \dots, k]$.*

Theorem 1. *For any finite subset $E = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_k}\}$ of $\mathbb{Z}^n$, the limit of its median sequence and its 2-chopped convex hull coincide, i.e $E^* = Conv_{2^*}(E)$.*

Even though the 2-chopped convex hull of a set E is not quite its real convex hull, it contains vectors that are arbitrarily close to any element of the full convex closure.

Lemma 1. *For each* $\mathbf{v} \in Conv(E)$ *and* $\epsilon > 0$*, there exists* $\mathbf{v'} \in Conv_{2^*}(E)$ *such that* $|\mathbf{v} - \mathbf{v'}| \leq \epsilon$*.*

From Lemma 1 it follows that the convex hull of E is included in the topological closure of its 2-chopped hull. The following theorem states that these two sets coincide.

Theorem 2. *For any finite subset* $E = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}\}$ *of* $\mathbb{Z}^n$*, we have that* $TC(Conv_{2^*}(E)) = Conv(E)$*.*

Computing the real convex hull of a finite set of integer vectors can thus be reduced to compute the topological closure of the limit of its median sequence. We now investigate how to compute $Conv(E)$ and $TC(E)$ for a set E described by an *RVA*.

5 Algorithmic issues

We consider a finite subset $E = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_k}\}$ of $\mathbb{Z}^n$ that is represented by a (weak deterministic) RVA A_E. Our goal is to compute an RVA that represents the convex hull over $\mathbb{R}^n$ of E. According to the results in Section 4, this can be done by computing an RVA A_{E^*} representing the limit of the median sequence of E, and then computing an RVA representing the topological closure of E^*. We now show how these two problems can be tackled by automata-based algorithms.

5.1 Computing an RVA for the 2-chopped Hull

We consider the problem of computing A_{E^*}. We first notice that since E is represented by a weak deterministic RVA, each element in its median sequence can also be represented in the same way. Indeed, computing E_{i+1} from E_i can be done by simple operations on the automata representations that preserve their weak-deterministic nature (see [BJW05]). Consequently, computing A_{E^*} amounts to computing the limit of an infinite sequence of weak automata. To finitely compute this limit, we obviously need some form of "speed-up" technique. We will use the extrapolation-based technique proposed in [BLW04]. This technique computes the limit of an infinite sequence of minimal deterministic weak automata by extrapolating a finite sampled prefix of this sequence, i.e. selected automata from a prefix of the sequence. The technique does not guarantee that a result will be obtained, and correctness of the guessed extrapolation needs to be checked once it is obtained, but one should remember that the general problem of computing the limit of a sequence of automata is undecidable.

The extrapolation step proceeds by comparing successive automata in the sample sequence, trying to identify the difference between these in the form of an "increment", and extrapolating the repetition of this increment by adding loops to the last automaton of the sequence. Checking correctness is a more involved procedure whose description is, for technical reasons, postponed to Section 5.3.

Choosing the sample sequence is a rather tricky issue and there is no guarantee that this can be done in a way that ensures that the extrapolation step can be applied. However, there is a number of heuristics that are very effective for obtaining a sample sequence that can be extrapolated. For example, one should select a sequence of at least three automata whose number of states increase linearly. Also, the language of each selected automaton should include those of the previous ones (this will always be the case for the sequence we are considering here).

This extrapolation technique has been implemented in a tool called TORMC. The tool relies on the LASH package [LAS] for automata manipulation procedures, but implements the specific algorithms given in [BLW03,BLW04]. As our experimental results show (see Section 6) The technique performs well on the median sequences used in the present work.

5.2 Computing the Topological Closure of an RVA-represented set

Consider a set $E \subset \mathbb{R}^n$ represented by a weak deterministic RVA A_E. Our goal is to compute an RVA $A_{TC(E)}$ that represents the topological closure of E. The intuition behind the computation is that we need to add to the language accepted by A_E, all words that are arbitrarily close to words of this language. This is fairly straightforward to do since we only need to add words that have arbitrarily long common prefixes with accepted words. A simple step to do this is to make accepting all states of the fractional part of the automaton, i.e. the part that deals with the tail of the word found after the separator. Of course, this will compute the topological closure within the topology on infinite words, but this also almost computes the vector Euclidean topological closure as is shown by the following result.

Theorem 3. *Let A_E be a weak deterministic RVA representing a vector set E. Let $\overline{A}_E$ be A_E with all states of its fractional part made accepting. For each vector $\mathbf{v} \in \mathbb{R}^n$, $W(\mathbf{v}) \cap L(\overline{A}_E) \neq \emptyset$ if and only if $\mathbf{v} \in TC(E)$.*

Theorem 3 guarantees that $\overline{A}_E$ contains at least one encoding for each vector in $TC(E)$. However the automaton $\overline{A}_E$ is not necessarily $A_{TC(E)}$. Indeed, there is no guarantee that $\overline{A}_E$ will contain *all* the encodings of each vector included in the topological closure.

Example 3. Assume that A_E is the RVA representing the 2-chopped hull of the set $E = \{(0,0),(6,3)\}$. Here, $\overline{A}_E$ is not a proper RVA. Indeed, the vector $(2,1)$ belongs to the topological closure of A_E, but the infinite word $w = 001000\star(01)^\omega$ that corresponds to the high encoding of 2 and the low encoding of 1 is never added.

We thus need an extra step that adds all missing encodings. To do this, we use the fact that an automaton that recognizes words that are dual encodings of the same numbers can be built with simple automata-based operations. Using such an automaton for each dimension of the vectors being handled it is then fairly direct to perform the required computation.

5.3 Correctness Criterion

After having constructed the extrapolation A_E^* of a finite sequence $A_E^{i_1}, A_E^{i_2}, \ldots, A_E^{i_l}$ of automata representing elements in the median sequence of a set E, it remains to check whether it accurately corresponds to what we really intend to compute, i.e., A_{E^*}. This is done by first checking that the extrapolation is *safe*, in the sense that it captures all words accepted by A_{E^*}, and then checking that it is *precise*, i.e. that it accepts no more words than A_{E^*}.

We first investigate how to check whether A_E^* is safe.

The idea is simply to perform one more mid-point adding step on A_E^* and to check that this does not change the accepted language. Given a set E, let $C_2(E)$ be the set $\{\mathbf{y} \mid \mathbf{y} = (\mathbf{x_1} + \mathbf{x_2})/2 \text{ with } W(\mathbf{x_1}), W(\mathbf{x_2}) \subset E\}$. By extension we will also consider this operation on automata representing vector sets.

We then have the following sufficient lemma.

Lemma 2. *Let A_E^* and A_{E^*} be respectively the extrapolation of a median automata sequence for a set E and a representation of the actual limit of this sequence. We have that, if $L(C_2(A_E^*)) \subset L(A_E^*)$, then $L(A_{E^*}) \subset L(A_E^*)$.*

The required computation step is thus to check that $L(C_2(A_E^*)) \subset L(A_E^*)$. This is simple except for the fact that, the result of the extrapolation is representable by a weak automaton, but not necessarily a deterministic one, and hence testing inclusion becomes problematic. The problem can be solved by first applying the topological closure step to A_E^* and then performing the safety check given by Lemma 2. It is quite easy to prove that doing this has no impact on the result of the test.

Checking preciseness can be performed, under the assumption that we work with the topological closure of the result, by adapting the techniques proposed in [BLW04]. However, this solution is computationally demanding and not really practical. An alternative strategy would be to check preciseness by applying arithmetic techniques on the constraint-based representation of the convex hull. This, however, still requires further work on algorithms for transforming the automaton-based representation of the convex hull into its constraint-based representation.

6 Experimental Results

The approach presented in this paper has been widely tested with help of the TORMC tool. One of the functions of TORMC takes as input a finite sequence of automata and, if successful, outputs an extrapolation of this sequence. The user can then check according to his own criteria whether the extrapolation is safe and precise.

We computed the convex hull over $\mathbb{R}^n$ of finite convex sets in $\mathbb{Z}^n$, of the difference/union between finite convex sets in $\mathbb{Z}^n$ and of arbitrary finite sets of points in $\mathbb{Z}^n$. In Table 1 we give the vertices of some of the convex sets that were considered. We also give the number of states of the RVA that represents those sets, of the RVA that represents the largest element in the median sequence, and of the RVA that represents the convex hull. The same information is given for the difference/union of finite convex sets in Table 2 and for arbitrary finite sets of points in Table 3.

Finite convex in $\mathbb{Z}^n$			
Vertices	$\|A_E\|$	$\|A_E^i\|$	$\|A_{Conv(E)}\|$
(1), (2)	7	9	7
(-1,7), (5,-6)	28	290	104
(-13,1), (11,0)	40	354	142
(0,2), (0,4), (2,6), (4,4), (4,2)	54	78	58
(0,0,0), (3,3,2)	63	110	100
(1,1,1), (3,3,2), (2,2,4)	86	286	127
(-1,0,-1), (-1,2,-1), (0,1,-1), (0,1,1)	72	205	97

Table 1. Convex hull for finite convex sets.

Non Convex in $\mathbb{Z}^n$			
Description	$\|A_E\|$	$\|A_E^i\|$	$\|A_{Conv(E)}\|$
[(0,0), (4,4), (8,0)] \ [(4,0), (4,2), (6,0)]	65	97	61
[(0,0), (3,3), (6,3), (6,0)] ∪ [(6,0), (6,3), (9,6), (9,0)]	62	174	73
[(0,0,0), (0,2,0), (0,2,2), (3,0,0), (3,2,0), (3,2,2)] ∪ [(0,0,0), (0,2,0), (0,0,2), (3,0,0), (3,2,0), (3,0,2)]	170	283	160
[(-1,0,-1), (-1,2,-1), (0,1,-1), (0,1,1)] ∪ [(-1,0,3), (-1,2,3), (0,1,3), (0,1,1)]	96	337	134
[(0,0,0), (0,3,0), (3,0,0), (3,3,0), (0,0,5), (0,3,5), (3,0,5), (3,3,5)] \ [(1,1,0), (1,2,0), (2,1,0), (2,2,0), (1,1,5), (1,2,5), (2,2,5), (2,1,5)]	218	265	184
[(0,3,0), (0,4,0), (3,3,0), (3,4,0), (0,0,3), (3,0,3), (3,7,3), (0,7,3)] \ [(1,0,1), (1,0,2), (2,0,2), (2,0,1), (1,7,1), (2,7,1), (1,7,2), (2,7,2)]	227	334	219

Table 2. Convex hull for the difference/union between finite convex sets.

In all these examples, the sequence given to the tool is made of the 10 first automata of the median sequence. With such an input, the tool was always able to automatically select a sampled subsequence that can successfully be extrapolated. To perform its selection, the tool used a safety check that we implemented using the technique outlined in Section 5.3.

Sets of points			
Points of the set	$\lvert A_E \rvert$	$\lvert A_E^i \rvert$	$\lvert A_{Conv(E)} \rvert$
(0,0), (6,3)	27	97	39
(0,0), (3,3), (4,3)	31	314	61
(0,0), (3,3), (6,3), (9,6), (9,0)	42	686	73
(1,1,1), (3,2,1), (2,2,4)	64	370	137
(0,0,0), (0,2,0), (0,0,2), (0,2,2), (0,1,1), (3,0,0), (3,2,0), (3,0,2), (3,1,1), (3,2,2)	126	556	160

Table 3. Convex hull for finite sets of points.

We did not implement a criterion to assess the preciseness of our results, but we still have been able to positively check the quality of the solution obtained by a simple comparison with a directly computed RVA representing the convex hull of the initial set.

7 Conclusions

There are quite a few known techniques for computing convex hulls. Among these a long series of algorithms specialized to the 2D and 3D case and widely used and studied in computational geometry. There are also recent results given in an automata-theoretic setting [FL05]. Though we can claim originality with respect to the approach and techniques used, we cannot claim efficiency with respect to the algorithms used in computational geometry, or generality with respect to [FL05].

Our contribution stands in a middle ground where it offers generality coming from the automata representation, but limited by the incompleteness of the extrapolation technique and, compared to more theoretical work a strong implementation backing that fits in a wider scheme. In other words, there is no measure on which we can claim to beat all other approaches, but what we propose has at least some strong point with respect to each of the alternatives.

One of the first limiting factors we have encountered in our experiments is the growing cost of dual encodings as the number of dimensions of the represented sets grows. This clearly should be fixed to make the approach more practical and, fortunately, [EK06] shows a promising way of doing this. This is just one of the implementation optimizations that could improve the efficiency of our approach. In conclusion, the road to an efficient automata-based implementation of arithmetic might still be long, but the scenery is fascinating and the drive exciting.

References

[BB03] C. Bartzis and T. Bultan. Construction of efficient bdds for bounded arithmetic constraints. In *Proc. 9th Int. Conference on Tools and Algorithms for* *the Construction and Analysis of Systems (TACAS)*, volume 2619 of *Lecture Notes in Computer Science*, pages 394–408. Springer, 2003.

[BC96] A. Boudet and H. Comon. Diophantine equations, presburger arithmetic and finite automata. In *Proc. 21th Int. Colloquium on Trees in Algebra and Programming (CAAP)*, volume 1059 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 1996.

[BH06] B. Boigelot and F. Herbreteau. The power of hybrid acceleration. In *Proc. 18th Int. Conference on Computer Aided Verification (CAV)*, volume 4144 of *Lecture Notes in Computer Science*, pages 438–451. Springer, 2006.

[BJW05] B. Boigelot, S. Jodogne, and P. Wolper. An effective decision procedure for linear arithmetic over the integers and reals. *ACM Transactions on Computational Logic*, 6(3):614–633, 2005.

[BLW03] B. Boigelot, A. Legay, and P. Wolper. Iterating transducers in the large (extended abstract). In *Proc. 15th Int. Conference on Computer Aided Verification (CAV)*, Lecture Notes in Computer Science, pages 223–235. Springer, 2003.

[BLW04] B. Boigelot, A. Legay, and P. Wolper. Omega-regular model checking. In *Proc. 10th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 2988 of *Lecture Notes in Computer Science*, pages 561–575. Springer, 2004.

[Boi99] B. Boigelot. *Symbolic Methods for Exploring Infinite State Spaces.* Collection des publications de la Faculté des Sciences Appliquées de l'Université de Liège, Liège, Belgium, 1999.

[BRW98] B. Boigelot, S. Rassart, and P. Wolper. On the expressiveness of real and integer arithmetic automata (extended abstract). In *Proc. 25th Int. Colloquium on Automata, Languages and Programming (ICALP)*, volume 1443 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 1998.

[BW02] B. Boigelot and P. Wolper. Representing arithmetic constraints with finite automata: An overview. In *Proc. 18th Int. Conference on logic Programming (ICLP)*, volume 2401 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2002.

[EK06] J. Eisinger and F. Klaedtke. Don't care words with an application to the automata-based approach for real addition. In *Proc. 18th Int. Conference on Computer Aided Verification (CAV)*, volume 4144 of *Lecture Notes in Computer Science*, pages 67–80. Springer, 2006.

[FL05] A. Finkel and J. Leroux. The convex hull of a regular set of integer vectors is polyhedral and effectively computable. *Information Processing Letter*, 96(1):30–35, 2005.

[Kla04] F. Klaedtke. On the automata size for presburger arithmetic. In *Proc. 19th IEEE Symposium on Logic in Computer Science (LICS)*. IEEE Computer Society, 2004.

[LAS] The Liège Automata-based Symbolic Handler (LASH). Available at `http://www.montefiore.ulg.ac.be/~boigelot/research/lash/`.

[Lat04] L. Latour. From automata to formulas: Convex integer polyhedra. In *Proc. 19th IEEE Symposium on Logic in Computer Science (LICS)*, pages 120–129. IEEE Computer Society, 2004.

[Lat06] L. Latour. *Presburger Arithmetic: From Automata to Formulas.* Collection des publications de la Faculté des Sciences Appliquées de l'Université de Liège, Liège, Belgium, 2006.

[Ler04] J. Leroux. *Algorithmique de la vérification des systèmes à compteurs. Approximation et accélération. Implémentation de l'outil FAST*. PhD thesis, LSV Cachan, 2004.

[Ler05] J. Leroux. A polynomial time presburger criterion and synthesis for number decision diagrams. In *Proc. 20th IEEE Symposium on Logic in Computer Science (LICS)*, pages 147–156. IEEE Computer Society, 2005.

[Löd01] C. Löding. Efficient minimization of deterministic weak $\omega-$automata. *Information Processing Letters*, 79(3):105–109, 2001.

[LS05] J. Leroux and G. Sutre. Flat counter automata almost everywhere! In *Proc. 3th Int. Conference on Automated Technology for Verification and Analysis (ATVA)*, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005.

[MSS86] D. E. Muller, A. Saoudi, and P. E. Schupp. Alternating automata, the weak monadic theory of the tree and its complexity. In *Proc. 13th Int. Colloquium on Automata, Languages and Programming*, pages 275–283, Rennes, 1986. Springer-Verlag.

[Tho90] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 133–192. Elsevier, 1990.

[WB00] P. Wolper and B. Boigelot. On the construction of automata from linear arithmetic constraints. In *Proc. 6th Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1785 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2000.

A Proof of Theorem 1

We use the following definition

Definition 6. *A 2-term t of $E \subset \mathbb{R}^n$ is either a vector of E, or an expression of the form $(t_1 + t_2)/2$, where t_1 and t_2 are two 2-term. The depth of t, denoted by $d(t)$, is 0 if $t \in E$, and $max(d(t_1), d(t_2)) + 1$ otherwise.*

We now give the intuition behind the proof of Theorem 1.

Proof. We consider the two directions of the equivalence.

- We have $E^* \subset Conv_{2^*}(E)$. Indeed, by construction, each vector $\mathbf{v} \in E^*$ can be expressed as a 2-term of E. Moreover, a 2-term t can be rewritten as an expression of the form

$$e = a_1\mathbf{x_1} + \cdots + a_k\mathbf{x_k}$$

  with (1) $\forall(1 \leq i \leq k)\, [(a_i{\geq}0){\wedge}(\exists(k_i, m_i \in \mathbb{N})\, (a_i = \frac{k_i}{2^{m_i}}))]$ and (2) $\sum_{i=1}^{k} a_i = 1$. (1) is obvious by construction and (2) can easily be shown by induction on the depth of t.
- We have $Conv_{2^*}(E) \subset E^*$. Indeed, it is easy to see that each vector of $Conv_{2^*}(E)$ can be rewritten as a 2-term of E. Moreover, a 2-term of depth i is, by construction, included in all E_j for $j \geq i$.

B Proof of Lemma 1

We use the following lemma.

Lemma 3. *Consider $E = \{\mathbf{x_1}, \ldots, \mathbf{x_k}\}$, a finite set of vectors of $\mathbb{R}^n$. Let $\mathbf{v} = \sum_{i=1}^{k} \lambda_i \mathbf{x_i}$, $\mathbf{v'} = \sum_{i=1}^{k} \lambda'_i \mathbf{x_i}$, and $x_{max} = max_{i,j}(|\mathbf{x_i}[j]|)$ with $i \in [1, k]$ and $j \in [1, n]$. For each $\epsilon > 0$, if $\forall(1 \leq i \leq k)\, \exists(\epsilon_i > 0)\, |\lambda_i - \lambda'_i| \leq \epsilon_i$ such that $\sum_{i=1}^{k} \epsilon_i \leq \frac{\epsilon}{\sqrt{n} x_{max}}$, then $|\mathbf{v} - \mathbf{v'}| \leq \epsilon$.*

Proof.

$$\forall i\, \exists \epsilon_i\, ((|\lambda_i - \lambda'_i| \leq \epsilon_i) \wedge (\sum_{i=1}^{k} \epsilon_i \leq \frac{\epsilon}{\sqrt{n}\, x_{max}}))$$

$$\Leftrightarrow \sum_{i=1}^{k} |\lambda_i - \lambda'_i| \leq \frac{\epsilon}{\sqrt{n}\, x_{max}}$$

$$\Leftrightarrow \sqrt{\left(\sum_{i=1}^{k} |\lambda_i - \lambda'_i|\right)^2} \leq \frac{\epsilon}{\sqrt{n}\, x_{max}}$$

$$\Leftrightarrow \sqrt{n}\, x_{max} \sqrt{(|\lambda_1 - \lambda'_1| + \cdots + |\lambda_k - \lambda'_k|)^2} \leq \epsilon$$

$$\Leftrightarrow \sqrt{n\, x_{max}^2\, (|\lambda_1 - \lambda'_1| + \cdots + |\lambda_k - \lambda'_k|)^2} \leq \epsilon$$

$$\Leftrightarrow \sqrt{n\, (|\lambda_1 - \lambda'_1|\, x_{max} + \cdots + |\lambda_k - \lambda'_k|\, x_{max})^2} \leq \epsilon$$

$$\begin{aligned} \Leftrightarrow \ [\ & (|\lambda_1 - \lambda'_1|\, x_{max} + \cdots + |\lambda_k - \lambda'_k|\, x_{max})^2 + \cdots \\ & + (|\lambda_1 - \lambda'_1|\, x_{max} + \cdots + |\lambda_k - \lambda'_k|\, x_{max})^2\ ]^{\frac{1}{2}} \leq \epsilon \end{aligned} \quad (1)$$

by Minkowski, for all $1 \leq i \leq n$,

$$|(\lambda_1 - \lambda'_1)\mathbf{x_1}[i] + \ldots + (\lambda_k - \lambda'_k)\mathbf{x_k}[i]| \ \leq\ |\lambda_1 - \lambda'_1| x_{max} + \ldots + |\lambda_k - \lambda'_k| x_{max}$$

Therefore, we have

$$\begin{aligned} (1) \ \Rightarrow\ [\ & ((\lambda_1 - \lambda'_1)\, \mathbf{x_1}[1] + \cdots + (\lambda_k - \lambda'_k)\, \mathbf{x_k}[1])^2 + \cdots \\ & + ((\lambda_1 - \lambda'_1)\, \mathbf{x_1}[n] + \cdots + (\lambda_k - \lambda'_k)\, \mathbf{x_k}[n])^2\ ]^{\frac{1}{2}} \leq \epsilon \end{aligned}$$

$$\begin{aligned} \Leftrightarrow\ [\ & ((\lambda_1 \mathbf{x_1}[1] + \cdots + \lambda_k \mathbf{x_k}[1]) - (\lambda'_1 \mathbf{x_1}[1] + \cdots + \lambda'_k \mathbf{x_k}[1]))^2 + \cdots \\ & + ((\lambda_1 \mathbf{x_1}[n] + \cdots + \lambda_k \mathbf{x_k}[n]) - (\lambda'_1 \mathbf{x_1}[n] + \cdots + \lambda'_k \mathbf{x_k}[n]))^2\ ]^{\frac{1}{2}} \leq \epsilon \end{aligned}$$

$$\Leftrightarrow \sqrt{(\mathbf{v}[1] - \mathbf{v}[1]')^2 + \cdots + (\mathbf{v}[n] - \mathbf{v}[n]')^2} \leq \epsilon$$

$$\Leftrightarrow |\mathbf{v} - \mathbf{v'}| \leq \epsilon$$

We now prove Lemma 1.

Proof. We recall that $E = \{\mathbf{x_1}, \ldots, \mathbf{x_k}\}$. We define $x_{max} = max_{i,j}(|\mathbf{x_i}[j]|)$ with $i \in [1, k]$ and $j \in [1, n]$. For each $\mathbf{v} \in Conv(E)$ and each $\epsilon > 0$, we build a vector $\mathbf{v'} \in Conv_{2^*}(E)$ with $|\mathbf{v} - \mathbf{v'}| \leq \epsilon$. This amounts to assign a value to each λ'_i. This assignation is direct if $\mathbf{v} \in Conv_{2^*}(E)$.

Assume now that $\mathbf{v} \notin Conv_{2^*}(E)$. By hypothesis, we have

- $\mathbf{v} = \sum_{i=1}^{k} \lambda_i \mathbf{x_i}$, where $\sum_{i=1}^{k} \lambda_i = 1$ and $\forall (1 \leq i \leq k)\, \lambda_i \geq 0$.
- $\mathbf{v'} = \sum_{i=1}^{k} \lambda'_i \mathbf{x_i}$, where $\sum_{i=1}^{k} \lambda'_i = 1$ and $\forall (1 \leq i \leq k)\, [\lambda'_i \geq 0 \wedge \exists (k_i, m_i \in \mathbb{N})(\lambda'_i = \frac{k_i}{2^{m_i}})]$.

By Lemma 3, if $\forall (1 \leq i \leq k)\, \exists (\epsilon_i > 0)\, |\lambda_i - \lambda'_i| \leq \epsilon_i$ where $\sum_{i=1}^{k} \epsilon_i \leq \frac{\epsilon}{\sqrt{n} x_{max}}$, then $|\mathbf{v} - \mathbf{v'}| \leq \epsilon$.

Consider $l \in \mathbb{N}$ with $k\, 2^{-l} \leq \frac{\epsilon}{\sqrt{n} x_{max}}$. For each $1 \leq i \leq k$, we define λ_{i_1} by truncating the 2's encoding of λ_i after the l first bits of its fractional part. It is easy to see that $\forall(1 \le i \le k)\, |\lambda_i - \lambda_{i_1}| \le 2^{-l}$. For each $1 \le i \le k$, let $\lambda_{i_2} = \lambda_i - \lambda_{i_1} \le 2^{-l}$. Since $\sum_{i=1}^{k} \lambda_{i_1}$ is a multiple of 2^{-l}, $\sum_{i=1}^{k} \lambda_{i_2} = 1 - \sum_{i=1}^{k} \lambda_{i_1}$ is also a multiple of 2^{-l}.

Define $d \in \mathbb{N}$ as follows

$$d = \frac{\sum_{i=1}^{k} \lambda_{i_2}}{2^{-l}}$$

Since $\forall(1 \le i \le k)\, \lambda_{i_2} \le 2^{-l}$, we have $d \le k$. For each $(1 \le i \le k)$, we define λ'_i as follows:

$$\lambda'_i = \begin{cases} \lambda_{i_1} + 2^{-l} & \text{if } 1 \le i \le d \\ \lambda_{i_1} & \text{otherwise.} \end{cases}$$

We have

- $\forall(d < i \le k)\ \ |\lambda_i - \lambda'_i| \le 2^{-l}$, and
- $\forall(1 \le i \le d)\ |\lambda_i - \lambda'_i| \le |\lambda_i - (\lambda_i + 2^{-l})| = 2^{-l}$.

Consequently, $\forall(1 \le i \le k)\, |\lambda_i - \lambda'_i| \le 2^{-l}$, and by Lemma 3, $|\mathbf{v} - \mathbf{v}'| \le \epsilon$.

To conclude, observe that

- $\forall(1 \le i \le k)\ \exists(k_i, m_i \in \mathbb{N})\ \lambda'_i = \frac{k_i}{2^{m_i}}$,
- $\sum_{i=1}^{k} \lambda'_i = \sum_{i=1}^{k} \lambda_{i_1} + d\, 2^{-l} = \sum_{i=1}^{k} \lambda_{i_1} + \sum_{i=1}^{k} \lambda_{i_2} = 1$, and
- $\forall(1 \le i \le k)\ \lambda'_i \ge \lambda_{i_1} \ge 0$.

C Proof of Theorem 2

The proof requires the next intermediary results.

Definition 7. *A* compact *set in* $\mathbb{R}^n$ *is a bounded and closed set.*

Theorem 4. *Consider K a compact subset of $\mathbb{R}^n$ and f a continuous application, $f(K)$ is a compact set.*

Lemma 4. *Let $E = \{\mathbf{x_1}, \ldots, \mathbf{x_k}\}$ be a finite subset of $\mathbb{R}^n$. The convex hull $Conv(E)$ of E is a closed set.*

Proof. Let

$$K = \{(\lambda_1, \ldots, \lambda_k) \mid \forall i\, (\lambda_i \in [0,1]) \wedge \sum_{i=1}^{k} \lambda_i = 1\} \subset \mathbb{R}^k$$

and

$$f : (\lambda_1, \ldots, \lambda_k) \mapsto \sum_{i=1}^{k} \lambda_i \mathbf{x_i}$$

Since K is a compact set and f is a continuous application, $f(K)$ is a closed set. We conclude by noticing that $Conv(E) = Conv(\{\mathbf{x_1}, \ldots, \mathbf{x_k}\}) = f(K)$.

We now prove Theorem 2.

Proof. We prove the two directions of the equivalence.

- We have $TC(Conv_{2^*}(E)) \subset Conv(E)$. Indeed, we have $TC(Conv_{2^*}(E)) \subset TC(Conv(E)) \subset Conv(E)$. The first inclusion holds because $Conv_{2^*}(E) \subset Conv(E)$ and, for any $E_1, E_2 \in \mathbb{R}^n$, $E_1 \subset E_2 \Rightarrow TC(E_1) \subset TC(E_2)$. The second inclusion holds because of Lemma 4.
- By Lemma 1, we have $Conv(E) \subset TC(Conv_{2^*}(E))$.

D Proof of Theorem 3

We first state and prove a series of theorems and lemmas. We denote by $W^{-1}(w, n)$ the unique vector $\mathbf{v}$ of $\mathbb{R}^n$ such that $w \in W(\mathbf{v})$.

Lemma 5. *Let A be a RVA that represents a set $E \subset \mathbb{R}^n$. Let A' be A with all states of its fractional part made accepting. We have*

$$\forall(w' \in L(A'))\forall(k \in \mathbb{N})\exists(w \in L(A))(pref_k(w) = pref_k(w')).$$

Lemma 6. *Consider $E \subset \mathbb{R}^n$ and $\mathbf{v} \in \mathbb{R}^n$. If for each $\epsilon > 0$ there exists $\mathbf{v}' \in E$ such that $|\mathbf{v} - \mathbf{v}'| \leq \epsilon$, then there exists $w \in W(\mathbf{v})$ such that*

$$\forall(k \in \mathbb{N})\exists(\mathbf{v}' \in E, w' \in W(\mathbf{v}'))\,(pref_k(w) = pref_k(w')).$$

Lemma 7. *Let A be a RVA that represents a set $E \subset \mathbb{R}^n$. Let A' be A with all states of its fractional part made accepting. Consider w, an encoding of a vector $\mathbf{v}$ of $\mathbb{R}^n$. If for each $k \in \mathbb{N}$ there exists $w' \in L(A)$ such that $pref_k(w) = pref_k(w')$, then $w \in L(A')$.*

Theorem 5. *Let A be a RVA that represents a set $E \in \mathbb{R}^n$. Let A' be A with all states of its fractional part made accepting. For each $w' \in L(A')$, the vector $W^{-1}(w', n)$ is in $TC(E)$.*

Proof. The result is obvious if $w' \in L(A)$. Assume now that $w' \notin L(A)$. By Lemma 5, we have

$$\forall(w' \in L(A'))\forall(k \in \mathbb{N})\exists(w \in L(A))(pref_k(w) = pref_k(w')).$$

Which implies

$$\forall(\mathbf{v}' \in W^{-1}(L(A'), n))\forall(\epsilon > 0)\exists(\mathbf{v} \in W^{-1}(L(A), n))(|\mathbf{v} - \mathbf{v}'| \leq \epsilon).$$

This concludes the result.

Theorem 6. *Let A be a RVA that represents a set $E \subset \mathbb{R}^n$. Let A' be A with all states of its fractional part made accepting. For each $\mathbf{v}$ of $TC(E)$, there exists $w \in W(\mathbf{v})$ such that $w \in L(A')$.*

Proof. Let $\mathbf{v}$ be a vector of $\mathbb{R}^n$. If $\mathbf{v} \in E$, then the result is obvious. Consider now that $\mathbf{v}$ does not belong to E.

By definition of the topological closure, we have

$$\forall(\epsilon > 0)\exists(\mathbf{v}' \in E)\,(|\mathbf{v} - \mathbf{v}'| \leq \epsilon).$$

By Lemma 6, there exists $w \in W(\mathbf{v})$ such that

$$\forall(k \in \mathbb{N})\exists(\mathbf{v}' \in E, w' \in W(\mathbf{v}'))\,(\mathit{pref}_k(w) = \mathit{pref}_k(w')).$$

Using Lemma 7, we conclude that $w \in L(A')$.

We now observe that Theorem 3 is a direct consequence of the two theorems above.

E All the Examples

Convex in $\mathbb{Z}^n$				
	Vertices	$\|A_E\|$	$\|A_{E_i}\|$	$\|A_{Conv(E)}\|$
1	(1), (2)	7	9	7
2	(1), (3)	8	10	8
3	(1), (4)	9	11	9
4	(-1), (2)	9	11	9
5	(0), (1)	7	9	7
6	(1,1), (3,2)	19	35	25
7	(1,1), (4,4)	27	33	29
8	(1,1), (5,3)	22	39	29
9	(1,1), (5,2)	21	60	36
10	(1,1), (5,4)	23	96	42
11	(1,1), (64,32)	54	2363	444
12	(1,1), (4,2)	22	54	31
13	(-7,1), (-2,14)	31	249	111
14	(-1,7), (5,-6)	28	290	104
15	(-13,1), (11,0)	40	354	142
16	(-3,-9), (4,8)	40	326	114
17	(4,8), (15,9)	36	189	98
18	(4,8), (9,2)	36	128	75
19	(9,2), (15,9)	29	232	90
20	(1,1), (1,2), (2,1)	22	42	24
21	(2,1), (2,3), (6,1)	37	57	41
22	(4,8), (15,9), (9,2)	80	620	200
23	(-4,1), (-2,1), (5,4)	41	395	103
24	(-6,-1), (-1,-4), (1,1)	55	350	107
25	(0,0), (2,5), (5,-1)	53	253	81
26	(-2,0), (0,1), (1,-3)	36	92	47
27	(4,8), (15,9), (10,2)	79	445	173
28	(4,8), (14,9), (9,2)	81	594	190
29	(4,7), (15,9), (9,2)	81	422	163
30	(0,0), (5,7), (6,3)	59	258	98
31	(0,0), (3,3), (4,3)	34	170	61
32	(1,1), (2,1), (1,2), (2,2)	26	38	26
33	(2,2), (2,12), (12,2), (12,12)	73	85	73
34	(-16,16), (16,16), (16,-16), (-16,-16)	102	114	102
35	(1,2), (3,1), (3,-2), (-1,-1)	48	201	72
36	(2,64), (16,64), (16,-8), (-16,-8)	128	175	135
37	(0,2), (0,4), (2,6), (4,4), (4,2)	54	78	58
38	(-2,-1), (-1,2), (2,4), (3,1), (3,-1), (1,-2)	62	217	98
39	(2,0), (0,2), (0,4), (2,6), (4,4), (4,2)	60	114	68

	Vertices	$\lvert A_E\rvert$	$\lvert A_{E_i}\rvert$	$\lvert A_{Conv(E)}\rvert$
40	(10,30), (10,20), (20,10), (30,10), (40,20), (40,30), (30,40), (20,40)	163	217	171
41	(-32,16), (-32,-16), (-16,-32), (16,-32), (32,-16), (32,16), (16,32), (-16,32)	139	178	144
42	(1,1,1), (2,3,4)	47	106	66
43	(1,2,3), (-1,3,0)	57	126	82
44	(0,0,0), (3,3,2)	63	110	100
45	(1,1,1), (3,3,2), (3,2,1)	51	191	97
46	(1,1,1), (3,2,1), (2,2,4)	64	370	137
47	(3,2,1), (3,3,2), (2,2,4)	72	225	109
48	(1,1,1), (3,3,2), (2,2,4)	86	286	127
49	(-1,0,-1), (-1,2,-1), (0,1,-1), (0,1,1)	72	205	97
50	(-1,0,3), (-1,2,3), (0,1,3), (0,1,1)	77	258	106
51	(0,0,0), (2,0,0), (0,2,0), (0,0,2)	94	183	102
52	(0,0,0), (0,0,1), (0,2,0), (1,0,0), (1,0,1), (1,2,0)	100	188	125
53	(0,0,0), (0,2,0), (0,2,2), (3,0,0), (3,2,0), (3,2,2)	146	218	157
54	(0,0,0), (0,2,0), (0,0,2), (3,0,0), (3,2,0), (3,0,2)	141	213	152
55	(1,1,1), (2,1,1), (2,0,1), (1,0,1), (1,1,2), (2,1,2), (1,0,2), (2,0,2)	88	142	88
56	Dimension 4 between 1 and 2 on each axis	280	496	280
57	Dimension 5 between 1 and 2 on each axis	881	1691	881

Not Convex in $\mathbb{Z}^n$				
	Description	$\lvert A_E\rvert$	$\lvert A_{E_i}\rvert$	$\lvert A_{Conv(E)}\rvert$
58	[(0,0), (4,4), (8,0)] $\setminus$ [(4,0), (4,2), (6,0)]	65	97	61
59	[(0,0), (3,3), (6,3), (6,0)] $\cup$ [(6,0), (6,3), (9,6), (9,0)]	62	174	73
60	[(0,0,0), (0,2,0), (0,2,2), (3,0,0), (3,2,0), (3,2,2)] $\cup$ [(0,0,0), (0,2,0), (0,0,2), (3,0,0), (3,2,0), (3,0,2)]	170	283	160
61	[(-1,0,-1), (-1,2,-1), (0,1,-1), (0,1,1)] $\cup$ [(-1,0,3), (-1,2,3), (0,1,3), (0,1,1)]	96	337	134
62	[(0,0,0), (0,3,0), (3,0,0), (3,3,0), (0,0,5), (0,3,5), (3,0,5), (3,3,5)] $\setminus$ [(1,1,0), (1,2,0), (2,1,0), (2,2,0), (1,1,5), (1,2,5), (2,2,5), (2,1,5)]	218	265	184
63	[(0,3,0), (0,4,0), (3,3,0), (3,4,0), (0,0,3), (3,0,3), (3,7,3), (0,7,3)] $\setminus$ [(1,0,1), (1,0,2), (2,0,2), (2,0,1), (1,7,1), (2,7,1), (1,7,2), (2,7,2)]	227	334	219

Sets of points				
	Points of the set	$\lvert A_E\rvert$	$\lvert A_{E_i}\rvert$	$\lvert A_{Conv(E)}\rvert$
64	(0), (3)	8	17	9
65	(0,0), (6,3)	27	97	39
66	(0,0), (0,4), (4,4)	36	66	51
67	(0,0), (5,7), (6,3)	30	726	98
68	(0,0), (3,3), (4,3)	31	314	61
69	(0,2), (0,4), (2,6), (4,4), (4,2)	38	117	58
70	(-2,-1), (-1,2), (2,4), (3,1), (3,-1), (1,-2)	38	918	98
71	(2,0), (0,2), (0,4), (2,6), (4,4), (4,2)	42	142	68
72	(0,0), (3,3), (6,3), (9,6), (9,0)	42	686	73
73	(1,1,1), (3,3,2), (3,2,1)	51	191	97
74	(1,1,1), (3,2,1), (2,2,4)	64	370	137
75	(0,0,0), (2,0,0), (0,2,0), (0,0,2)	84	183	102
76	(0,0,0), (0,2,0), (0,0,2), (0,2,2), (0,1,1), (3,0,0), (3,2,0), (3,0,2), (3,1,1), (3,2,2)	126	556	160