Convex Hull of Arithmetic Automata

Jérôme Leroux

LaBRI, Université de Bordeaux, CNRS
Domaine Universitaire, 351, cours de la Libération, 33405 Talence, France
leroux@labri.fr

Abstract. Arithmetic automata recognize infinite words of digits denoting decompositions of real and integer vectors. These automata are known expressive and efficient enough to represent the whole set of solutions of complex linear constraints combining both integral and real variables. In this paper, the closed convex hull of arithmetic automata is proved rational polyhedral. Moreover an algorithm computing the linear constraints defining these convex set is provided. Such an algorithm is useful for effectively extracting geometrical properties of the whole set of solutions of complex constraints symbolically represented by arithmetic automata.

1 Introduction

The *most significant digit first decomposition* provides a natural way to associate finite words of digits to any integer. Naturally, such a decomposition can be extended to real values just by considering *infinite* words rather than *finite* ones. Intuitively, an infinite word denotes the potentially infinite decimal part of a real number. Last but not least, the most significant digit first decomposition can be extended to real vectors just by interleaving the decomposition of each component into a single infinite word.

Arithmetic automata are Muller automata that recognize infinite words of most significant digit first decompositions of real vectors in a fixed basis of decomposition $r \geq 2$ (for instance $r = 2$ and $r = 10$ are two classical basis of decomposition). Sets symbolically representable by arithmetic automata in basis r are logically characterized [BRW98] as the sets definable in the first order theory $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_r)$ where X_r is an additional predicate depending on the basis of decomposition r. In practice, arithmetic automata are usually used for the first order additive theory $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq)$ where X_r is discarded. In fact this theory allows to express complex linear constraints combining both integral and real variables that can be represented by particular Muller automata called *deterministic weak Buchi automata* [BJW05]. This subclass of Muller automata has interesting algorithmic properties. In fact, compared to the general class, deterministic weak Buchi automata can be minimized (for the number of states) into a unique canonical form with roughly the same algorithm used for automata recognizing finite words. In particular, these arithmetic automata are well adapted to symbolically represent sets definable in $\mathrm{FO}(\mathbb{R},\mathbb{Z},+,\leq)$ obtained after many operations (boolean combinations, quantifications). In fact, since the obtained arithmetic automata only depends on the represented set and not on the potentially long sequence of operations used to compute this set, we avoid unduly complicated arithmetic automata. Intuitively, the automaton minimization algorithm performs like a simplification procedure for $\mathrm{FO}(\mathbb{R},\mathbb{Z},+,\leq)$. In particular arithmetic automata are adapted to the symbolic model checking approach computing inductively reachability sets of systems manipulating counters [BLP06] and/or clocks [BH06]. In practice algorithms for effectively computing an arithmetic automaton encoding the solutions of formulas in $\mathrm{FO}(\mathbb{R},\mathbb{Z},+,\leq)$ have been recently successfully implemented in tools Lash and Lira [BDEK07]. Unfortunately, interesting qualitative properties are difficult to extract from arithmetic automata. Actually, operations that can be performed on the arithmetic automata computed by tools Lash and Lira are limited to the universality and the emptiness checking (when the set symbolically represented is not empty these tools can also compute a real vector in this set).

Extracting geometrical properties from an arithmetic automaton representing a set $X \subseteq \mathbb{R}^m$ is a complex problem even if X is definable in $\mathrm{FO}(\mathbb{R},\mathbb{Z},+,\leq)$. Let us recall related works to this problem. Using a Karr based algorithm [Kar76], the affine hull of X has been proved efficiently computable in polynomial time [Ler04] (even if this result is limited to the special case $X \subseteq \mathbb{N}^m$, it can be easily extended to any arithmetic automata). When $X = \mathbb{Z}^m \cap C$ where C is a rational polyhedral convex set (intuitively when X is equal to the integral solutions of linear constraint systems), it has been proved in [Lat04] that we can effectively compute in exponential time a rational polyhedral convex set C' such that $X = \mathbb{Z}^m \cap C'$. Note that this worst case complexity in theory is not a real problem in practice since the algorithm presented in [Lat04] performs well on automata with more than 100 000 states. In [Lug04] this result was extended to sets $X = F + L$ where F is a finite set of integral vectors and L is a linear set. In [FL05], closed convex hulls of sets $X \subseteq \mathbb{Z}^m$ represented by arithmetic automata are proved rational polyhedral and effectively computable in exponential time. Note that compared to [Lat04], it is not clear that this result can be turn into an efficient algorithm. More recently [Ler05], we provided an algorithm for effectively computing in polynomial time a formula in the Presburger theory $\mathrm{FO}(\mathbb{Z},+,\leq)$ when $X \subseteq \mathbb{Z}^n$ is Presburger-definable. This algorithm has been successfully implemented in TaPAS [LP08] (The Talence Presburger Arithmetic Suite) and it can be applied on any arithmetic automata encoding a set $X \subseteq \mathbb{Z}^m$ with more than 100 000 states. Actually, the tool decides if an input arithmetic automaton denotes a Presburger-definable set and in this case it returns a formula denoting this set.

In this paper we prove that the closed convex hulls of sets symbolically represented by arithmetic automata are rational polyhedral and effectively computable in exponential time in the worst case. Note that whereas the closed convex hull of a set definable in $\mathrm{FO}(\mathbb{R},\mathbb{Z},+,\leq)$ can be easily proved rational polyhedral (thanks to quantification eliminations), it is difficult to prove that the closed convex hulls of arithmetic automata are rational polyhedral. We also provide an algorithm for computing this set. Our algorithm is based on the reduction of the closed convex hull computation to data-flow analysis problems. Note that widening operator is usually used in order to speed up the iterative computation of solutions of such a problem. However, the use of widening operators may lead to loss of precision in the analysis. Our algorithm is based on *acceleration* in convex data-flow analysis [LS07b,LS07a]. Recall that acceleration consists to compute the exact effect of some control-flow cycles in order to speed up the Kleene fix-point iteration.

Outline of the paper : In section 2 the most significant digit first decomposition is extended to any real vector and we introduce the arithmetic automata. In section 3 we provide the closed convex hull computation reduction to (1) a data-flow analysis problem and (2) the computation of the closed convex hull of arithmetic automata representing only decimal values and having a trivial accepting condition. In section 4 we provide an algorithm for computing the closed convex hull of such an arithmetic automaton. Finally in section 5 we prove that the data-flow analysis problem introduced by the reduction can be solved precisely with an accelerated Kleene fix-point iteration algorithm. Most proofs are only sketched in the paper, but detailed proofs are given in appendix. This paper is the long version of the SAS 2008 paper.

2 Arithmetic Automata

This section introduces arithmetic automata (see Fig. 1). These automata recognize infinite words of digits denoting *most significant digit first* decompositions of real and integer vectors.

As usual, we respectively denote by $\mathbb{Z}$, $\mathbb{Q}$ and $\mathbb{R}$ the sets of integers, rationals and real numbers and we denote by $\mathbb{N}, \mathbb{Q}_+, \mathbb{R}_+$ the restrictions of $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ to the non-negatives. The *components* of an m-dim vector x are denoted by $x[1], \dots, x[m]$.

We first provide some definitions about regular sets of infinite words. We denote by Σ a non-empty finite set called an *alphabet*. An *infinite word* w over Σ is a function $w \in \mathbb{N} \rightarrow \Sigma$ defined over $\mathbb{N} \backslash \{0\}$ and a *finite word* σ over Σ is a function $\sigma \in \mathbb{N} \rightarrow \Sigma$ defined over a set $\{1, \dots, k\}$ where $k \in \mathbb{N}$ is called the *length* of σ and denoted by $|\sigma|$. *In this paper, a finite word over Σ is denoted by σ with some subscript indices and an infinite word over Σ is denoted by w.* As usual Σ^* and Σ^ω respectively denote the set of finite words and the set of infinite words over Σ. The concatenation of two finite words $\sigma_1, \sigma_2 \in \Sigma^*$ and the concatenation of a finite word $\sigma \in \Sigma^*$ with an infinite word $w \in \Sigma^\omega$ are denoted by $\sigma_1\sigma_2$ and σw. A *graph labelled* by Σ is a tuple $G = (Q, \Sigma, T)$ where Q is a non empty finite set of *states* and $T \subseteq Q \times \Sigma \times Q$ is a set of *transitions*. A *finite path* π in a graph G is a finite word $\pi = t_1 \dots t_k$ of $k \geq 0$ transitions $t_i \in T$ such that there exists a sequence $q_0, \dots, q_k \in Q$ and a sequence $a_1, \dots, a_k \in \Sigma$ such that $t_i = (q_{i-1}, a_i, q_i)$ for any $1 \leq i \leq k$. The finite word $\sigma = a_1 \dots a_k$ is called the *label* of π and such a path π is also denoted by $q_0 \xrightarrow{\sigma} q_k$ or just $q_0 \rightarrow q_k$. We also say that π is a path *starting* from q_0 and *terminating* in q_k. When $q_0 = q_k$ and $k \geq 1$, the path π is called a cycle on q_0. Such a cycle is said *simple* if the states $q_0, \dots, q_{k-1}$ are distinct. Given an integer $m \geq 1$, a graph G is called an m-graph if m divides the length of any cycle in G. An *infinite path* θ is an infinite word of transitions such that any prefixes $\pi_k = \theta(1) \dots \theta(k)$ is a finite path. The unique infinite word $w \in \Sigma^\omega$ such that $\sigma_k = w(1) \dots w(k)$ is the label of the finite path π_k for any $k \in \mathbb{N}$ is called the *label* of θ. We say that θ is *starting* from q_0 if q_0 is the unique state such that any prefix of θ is starting from q_0. *In the sequel, a finite path is denoted by* π *and an infinite path is denoted by* θ. The *set of infinite paths starting from* q_0 is naturally denoted with the capital letter $\Theta_G(q_0)$. The set F of states $q \in Q$ such that there exists an infinite number of prefix of θ terminating in q is called the set of *states visited infinitely often* by θ. Such a path is denoted by $q_0 \xrightarrow{w} F$ or just $q_0 \rightarrow F$. A *Muller automaton* A is a tuple $A = (Q, \Sigma, T, Q_0, \mathcal{F})$ where (Q, Σ, T) is a graph, $Q_0 \subseteq Q$ is the *initial condition* and $\mathcal{F} \subseteq \mathcal{P}(Q)$ is the *accepting condition*. The language $L(A) \subseteq \Sigma^\omega$ *recognized* by a Muller automaton A is the set of infinite words $w \in \Sigma^\omega$ such that there exists an infinite path $q_0 \xrightarrow{w} F$ with $q_0 \in Q_0$ and $F \in \mathcal{F}$.

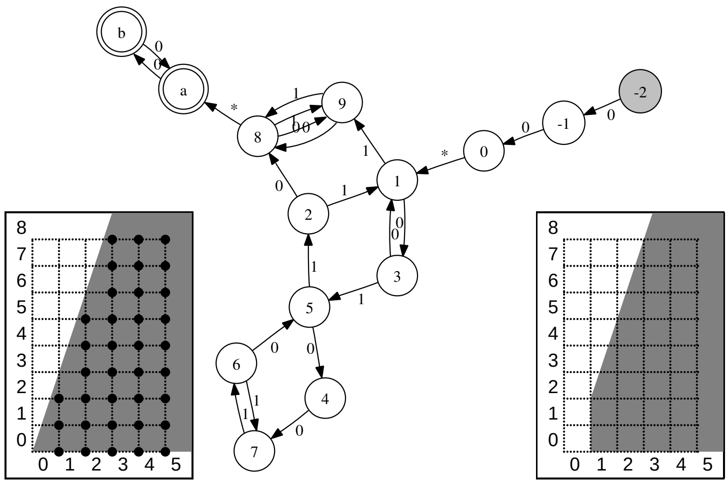

Fig. 1. On the left, the rational polyhedral convex set $C = \{x \in \mathbb{R}^2 \mid 3x[1] > x[2] \wedge x[2] \geq 0\}$ in gray and the set $X = \mathbb{Z}^2 \cap C$ of integers depicted by black bullets. On the center, an arithmetic automaton symbolically representing X in basis 2. On the right, the closed convex hull of X equals to $\mathrm{cl} \circ \mathrm{conv}(X) = \{x \in \mathbb{R}^2 \mid 3x[1] \geq x[2] + 1 \wedge x[2] \geq 0 \wedge x[1] \geq 1\}$ represented in gray.

Now, we introduce the *most significant digit first decomposition* of real vectors. In the sequel $m \geq 1$ is an integer called *the dimension*, $r \geq 2$ is an integer called the *basis of decomposition*, $\Sigma_r = \{0, \dots, r-1\}$ is called the alphabet of

r-*digits*, and $S_r = \{0, r-1\}$ is called the alphabet of *sign* r-*digits*. The most significant r-digit first decomposition provides a natural way to associate to any real vector $x \in \mathbb{R}^m$ a tuple $(s, \sigma, w) \in S_r^m \times (\Sigma_r^m)^* \times \Sigma_r^\omega$. Intuitively (s, σ) and w are respectively associated to an integer vector $z \in \mathbb{Z}^m$ and a decimal vector $d \in [0,1]^m$ satisfying $x = z + d$. Moreover, $s[i] = 0$ corresponds to $z[i] \geq 0$ and $s[i] = r-1$ corresponds to $z[i] < 0$. More formally, a *most significant* r-*digit first decomposition* of a real vector $x \in \mathbb{R}^m$ is a tuple $(s, \sigma, w) \in S_r^m \times (\Sigma_r^m)^* \times \Sigma_r^\omega$ such that for any $1 \leq i \leq m$, we have:

$$x[i] = r^{\frac{|\sigma|}{m}} \frac{s(i)}{1-r} + \sum_{j=0}^{\frac{|\sigma|}{m}-1} r^j \sigma(mj+i) + \sum_{j=0}^{+\infty} \frac{w(mj+i)}{r^{j+1}}$$

The previous equality is divided in two parts by introducing the functions $\lambda_{r,m} \in \Sigma_r^\omega \to [-1,0]^m$ and $\gamma_{r,m} \in S_r^m \times (\Sigma_r^m)^* \to \mathbb{Z}^m$ defined for any $1 \leq i \leq m$ by the following equalities. Note the sign in front of the definition of $\lambda_{r,m}$. This sign simplifies the presentation of this paper and it is motivated in the sequel.

$$-\lambda_{r,m}(w)[i] = \sum_{j=0}^{+\infty} \frac{w(mj+i)}{r^{j+1}}$$

$$\gamma_{r,m}(s,\sigma)[i] = r^{\frac{|\sigma|}{m}} \frac{s(i)}{1-r} + \sum_{j=0}^{\frac{|\sigma|}{m}-1} r^j \sigma(mj+i)$$

Definition 2.1 ([BRW98]). *An* arithmetic automaton A *in basis* r *and in dimension* m *is a Muller automaton over the alphabet* $\Sigma_r \cup \{\star\}$ *that recognizes a language* $L \subseteq S_r^m \star (\Sigma_r^m)^* \star \Sigma_r^\omega$. *The following set* $X \subseteq \mathbb{R}^m$ *is called the set symbolically represented by* A*:*

$$X = \{\gamma_{r,m}(s,\sigma) - \lambda_{r,m}(w) \mid s \star \sigma \star w \in L\}$$

Example 2.2. The arithmetic automaton depicted in Fig. 1 symbolically represents $X = \{x \in \mathbb{N}^2 \mid 3x[1] > x[2]\}$. This automaton has been obtained automatically from the tool LASH through the tool-suite TAPAS[LP08].

We observe that *Real Vector Automata (RVA)* and *Number Decision Diagrams (NDD)* [BRW98] are particular classes of arithmetic automata. In fact, RVA and NDD are arithmetic automata A that symbolically represent sets X included respectively in $\mathbb{R}^m$ and $\mathbb{Z}^m$ and such that the accepted languages $L(A)$ satisfy:

$$L(A) = \{s \star \sigma \star w \mid \gamma_{r,m}(s,\sigma) - \lambda_{r,m}(w) \in X\} \qquad \text{if } A \text{ is a RVA}$$
$$L(A) = \{s \star \sigma \star 0^\omega \mid \gamma_{r,m}(s,\sigma) \in X\} \qquad \text{if } A \text{ is a NDD}$$

Since in general a NDD is not a RVA and conversely a RVA is not a NDD, we consider arithmetic automata in order to solve the closed convex hull computation uniformly for these two classes. Note that simple (even if computationally expensive) automata transformations show that sets symbolically representable by arithmetic automata in basis r are exactly the sets symbolically representable by RVA in basis r. In particular [BRW98], sets symbolically representable by arithmetic automata in basis r are exactly the sets definable in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_r)$ where $X_r \subseteq \mathbb{R}^3$ is a basis dependant predicate defined in [BRW98]. This characterization shows that arithmetic automata can symbolically represent sets of solutions of complex linear constraints combining both integral and real values. Recall that the construction of arithmetic automata from formulae in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_r)$ is effective and tools LASH and LIRA [BDEK07] implement efficient algorithms for the restricted logic $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq)$. The predicate X_r is discarded in these tools in order to obtain arithmetic automata that are deterministic weak Buchi automata [BJW05]. In fact these automata have interesting algorithmic properties (minimization and deterministic form).

3 Reduction to Data-Flow Analysis Problems

In this section we reduce the computation of the closed convex hull of sets symbolically represented by arithmetic automata to data-flow analysis problems.

We first recall some general notions about complete lattices. Recall that a *complete lattice* is any partially ordered set $(A, \sqsubseteq)$ such that every subset $X \subseteq A$ has a *least upper bound* $\bigsqcup X$ and a *greatest lower bound* $\bigsqcap X$. The *supremum* $\bigsqcup A$ and the *infimum* $\bigsqcap A$ are respectively denoted by $\top$ and $\bot$. A function $f \in A \to A$ is *monotonic* if $f(x) \sqsubseteq f(y)$ for all $x \sqsubseteq y$ in A. For any complete lattice $(A, \sqsubseteq)$ and any set Q, we also denote by $\sqsubseteq$ the partial order on $Q \to A$ defined as the point-wise extension of $\sqsubseteq$, i.e. $f \sqsubseteq g$ iff $f(q) \sqsubseteq g(q)$ for all $q \in Q$. The partially ordered set $(Q \to A, \sqsubseteq)$ is also a complete lattice, with lub $\bigsqcup$ and glb $\bigsqcap$ satisfying $(\bigsqcup F)(s) = \bigsqcup \{f(s) \mid f \in F\}$ and $(\bigsqcap F)(s) = \bigsqcap \{f(s) \mid f \in F\}$ for any subset $F \subseteq Q \to A$.

Now, we recall notions about the complete lattice of closed convex sets. A function $f \in \mathbb{R}^n \to \mathbb{R}^m$ is said *linear* if there exists a sequence $(M_{i,j})_{i,j}$ of reals indexed by $1 \leq i \leq m$ and $1 \leq j \leq n$ and a sequence $(v_i)_i$ of reals indexed by $1 \leq i \leq m$ such that $f(x)[i] = \sum_{j=1}^{n} M_{i,j} x[j] + v_i$ for any $x \in \mathbb{R}^n$ and for any $1 \leq i \leq m$. When the coefficients $(M_{i,j})_{i,j}$ and $(v_i)_i$ are rational, the linear function f is said *rational.* The function $f' \in \mathbb{R}^m \to \mathbb{R}^n$ defined by $f'(x)[i] = \sum_{j=1}^{n} M_{i,j} x[j]$ for any $x \in \mathbb{R}^n$ and for any $1 \leq i \leq m$ is called the *uniform form* of f. A set $R \subseteq \mathbb{R}^m$ is said *closed* if the limit of any convergent sequence of vectors in R is in R. Recall that any set $X \subseteq \mathbb{R}^m$ is included in a minimal for the inclusion closed set. This closed set is called the *topological closure* of X and it is denoted by $\mathrm{cl}(X)$. Let us recall some notions about convex sets (for more details, see [Sch87]). A *convex combination* of $k \geq 1$ vectors $x_1, \ldots, x_k \in \mathbb{R}^m$ is a vector x such that there exists $r_1, \ldots, r_k \in \mathbb{R}_+$ satisfying $r_1 + \cdots + r_k = 1$ and $x = r_1 x_1 + \cdots + r_k x_k$. A set $C \subseteq \mathbb{R}^m$ is said *convex* if any convex combination of vectors in C is in C. Recall that any $X \subseteq \mathbb{R}^m$ is included in a minimal for the inclusion convex set. This convex set is called the *convex hull* of X and it is denoted by $\mathrm{conv}(X)$. A convex set $C \subseteq \mathbb{R}^m$ is said *rational polyhedral* if there exists a rational linear function $f \in \mathbb{R}^m \to \mathbb{R}^n$

such that C is the set of vectors $x \in \mathbb{R}^m$ such that $\bigwedge_{i=1}^{n} f(x)[i] \leq 0$. Recall that $\mathrm{cl}(\mathrm{conv}(X)) = \mathrm{conv}(\mathrm{cl}(X))$, $\mathrm{cl}(f(X)) = f(\mathrm{cl}(X))$ and $\mathrm{conv}(f(X)) = f(\mathrm{conv}(X))$ for any $X \subseteq \mathbb{R}^m$ and for any linear function $f \in \mathbb{R}^m \to \mathbb{R}^n$. The class of closed convex subsets of $\mathbb{R}^m$ is written $\mathcal{C}_m$. We denote by $\sqsubseteq$ the inclusion partial order on $\mathcal{C}_m$. Observe that $(\mathcal{C}_m, \sqsubseteq)$ is a complete lattice, with lub $\bigsqcup$ and glb $\bigsqcap$ satisfying $\bigsqcup \mathcal{C} = \mathrm{cl} \circ \mathrm{conv}(\bigcup \mathcal{C})$ and $\bigsqcap \mathcal{C} = \bigcap \mathcal{C}$ for any subset $\mathcal{C} \subseteq \mathcal{C}_m$.

Example 3.1. Let $X = \mathbb{Z}^2 \cap C$ where C is the convex set $C = \{x \in \mathbb{R}^2 \mid 3x[1] > x[2] \wedge x[2] \geq 0\}$ (see Fig. 1). Observe that $\mathrm{cl} \circ \mathrm{conv}(X) = \{x \in \mathbb{R}^2 \mid 3x[1] \geq x[2] + 1 \wedge x[2] \geq 0 \wedge x[1] \geq 1\}$ is strictly included in C.

In the previous section, we introduced two functions $\lambda_{r,m}$ and $\gamma_{r,m}$. Intuitively these functions "compute" respectively decimal vectors associated to infinite words and integer vectors associated to finite words equipped with sign vectors. We now introduce two functions $\Lambda_{r,m,\sigma}$ and $\Gamma_{r,m,\sigma}$ that "*partially compute*" the same vectors than $\lambda_{r,m}$ and $\gamma_{r,m}$. More formally, let us consider the unique sequences $(\Lambda_{r,m,\sigma})_{\sigma \in \Sigma_r^*}$ and $(\Gamma_{r,m,\sigma})_{\sigma \in \Sigma_r^*}$ of linear functions $\Lambda_{r,m,\sigma}, \Gamma_{r,m,\sigma} \in \mathbb{R}^m \to \mathbb{R}^m$ inverse of each other and satisfying $\Lambda_{r,m,\sigma_1\sigma_2} = \Lambda_{r,m,\sigma_1} \circ \Lambda_{r,m,\sigma_2}$, $\Gamma_{r,m,\sigma_1\sigma_2} = \Gamma_{r,m,\sigma_2} \circ \Gamma_{r,m,\sigma_1}$ for any $\sigma_1, \sigma_2 \in \Sigma_r^*$, such that $\Lambda_{r,m,\epsilon}$ and $\Gamma_{r,m,\epsilon}$ are the identity function and such that $\Lambda_{r,m,a}$ and $\Gamma_{r,m,a}$ with $a \in \Sigma_r$ satisfy the following equalities where $x \in \mathbb{R}^m$:

$$\Lambda_{r,m,a}(x) = (\frac{x[m] - a}{r}, x[1], \ldots, x[m-1])$$
$$\Gamma_{r,m,a}(x) = (x[2], \ldots, x[m], rx[1] + a)$$

We first prove the following two equalities (1) and (2) that explain the link between the notations $\lambda_{r,m}$ and $\gamma_{r,m}$ and their capital forms $\Lambda_{r,m,\sigma}$ and $\Gamma_{r,m,\sigma}$. Observe that $\Lambda_{r,m,a}(\lambda_{r,m}(w)) = \lambda_{r,m}(aw)$ for any $a \in \Sigma_r$ and for any $w \in \Sigma_r^\omega$. An immediate induction over the length of $\sigma \in \Sigma_r^*$ provides equality (1). Note also that $\Gamma_{r,m,a_1 \ldots a_m}(x) = rx + (a_1, \ldots, a_m)$ for any $a_1, \ldots, a_m \in \Sigma_r$. Thus an immediate induction provides equality (2).

$$\lambda_{r,m}(\sigma w) = \Lambda_{r,m,\sigma}(\lambda_{r,m}(w)) \qquad \forall \sigma \in \Sigma_r^* \quad \forall w \in \Sigma_r^\omega \tag{1}$$
$$\gamma_{r,m}(s, \sigma) = \Gamma_{r,m,\sigma}(\frac{s}{1-r}) \qquad \forall \sigma \in (\Sigma_r^m)^* \quad \forall s \in S_r^m \tag{2}$$

We now reduce the computation of the closed convex hull C of a set $X \subseteq \mathbb{R}^m$ represented by an arithmetic automaton $A = (Q, \Sigma, T, Q_0, \mathcal{F})$ in basis r to data-flow analysis problems. We can assume w.l.o.g that (Q, Σ, T) is a m-graph. As the language recognized by A is included in $S_r^m \star (\Sigma_r^m)^* \star \Sigma_r^\omega$, the set of states can be partitioned into sets depending intuitively on the number of occurrences $|\sigma|_\star$ of the $\star$ symbol in a word $\sigma \in \Sigma^*$. More formally, we consider the set Q_S of states reading *signs*, the set Q_I reading *integers*, and the set Q_D reading *decimals* defined by:

$$Q_S = \{q \in Q \mid \exists (q_0, \sigma, F) \in Q_0 \times \Sigma^* \times \mathcal{F} \quad |\sigma|_\star = 0 \quad \wedge \quad q_0 \xrightarrow{\sigma} q \rightarrow F\}$$
$$Q_I = \{q \in Q \mid \exists (q_0, \sigma, F) \in Q_0 \times \Sigma^* \times \mathcal{F} \quad |\sigma|_\star = 1 \quad \wedge \quad q_0 \xrightarrow{\sigma} q \rightarrow F\}$$
$$Q_D = \{q \in Q \mid \exists (q_0, \sigma, F) \in Q_0 \times \Sigma^* \times \mathcal{F} \quad |\sigma|_\star = 2 \quad \wedge \quad q_0 \xrightarrow{\sigma} q \rightarrow F\}$$

We also consider the m-graphs G_S, G_I and G_D obtained by restricting G respectively to the states Q_S, Q_I and Q_D and formally defined by:

$$\begin{aligned} G_S &= (Q_S, \Sigma_r, T_S) && \text{with } T_S = T \cap (Q_s \times \Sigma_r \times Q_S) \\ G_I &= (Q_I, \Sigma_r, T_I) && \text{with } T_I = T \cap (Q_I \times \Sigma_r \times Q_I) \\ G_D &= (Q_D, \Sigma_r, T_D) && \text{with } T_D = T \cap (Q_D \times \Sigma_r \times Q_D) \end{aligned}$$

Example 3.2. $Q_S = \{-2, -1, 0\}$, $Q_I = \{1, \ldots, 9\}$ and $Q_D = \{a, b\}$ in Fig. 1.

The closed convex hull $C = \mathrm{cl} \circ \mathrm{conv}(X)$ is obtained from the valuations $C_I \in Q_I \rightarrow \mathcal{C}_m$ and $C_D \in Q_D \rightarrow \mathcal{C}_m$ defined by $C_I = \mathrm{cl} \circ \mathrm{conv}(X_I)$ and $C_D = \mathrm{cl} \circ \mathrm{conv}(X_D)$ where X_I and X_D are given by:

$$X_I(q_I) = \{\Gamma_{r,m,\sigma}(\frac{s}{1-r}) \mid s \in S_r^m \quad \sigma \in \Sigma_r^* \quad \exists q_0 \in Q_0 \quad q_0 \xrightarrow{s \star \sigma} q_I\}$$
$$X_D(q_D) = \{\lambda_{r,m}(w) \mid w \in \Sigma_r^\omega \quad \exists F \in \mathcal{F} \quad q_D \xrightarrow{w} F\}$$

In fact from the definition of arithmetic automata we get:

$$C = \bigsqcup_{\substack{(q_I, q_D) \in Q_I \times Q_D \\ (q_I, \star, q_D) \in T}} C_I(q_I) - C_D(q_D)$$

We now provide data-flow analysis problems whose C_I and C_D are solutions. Observe that m-graphs naturally denote control-flow graphs. Before associating semantics to m-graph transitions, we first show that C_I and C_D are some fix-point solutions. As $\mathrm{cl} \circ \mathrm{conv}$ and $\Gamma_{r,m,a}$ are commutative, from the inclusion $\Gamma_{r,m,a}(X_I(q_1)) \subseteq X_I(q_2)$ we deduce that C_I satisfies the relation $\Gamma_{r,m,a}(C_I(q_1)) \sqsubseteq C_I(q_2)$ for any transition $(q_2, a, q_2) \in T_I$. Symmetrically, as $\mathrm{cl} \circ \mathrm{conv}$ and $\Lambda_{r,m,a}$ are commutative, from the inclusion $\Lambda_{r,m,a}(X_D(q_2)) \subseteq X_D(q_1)$, we deduce that $\Lambda_{r,m,a}(C_D(q_2)) \sqsubseteq C_D(q_1)$ for any transition $(q_1, a, q_2) \in T_D$. Intuitively C_I and C_D are two fix-point solutions of different systems. More formally, we associate two distinct semantics to a transition $t = (q_1, a, q_2)$ of a m-graph $G = (Q, \Sigma_r, T)$ by considering the monotonic functions $\Lambda_{G,m,t}$ and $\Gamma_{G,m,t}$ over the complete lattice $(Q \rightarrow \mathcal{C}_m, \sqsubseteq)$ defined for any $C \in Q \rightarrow \mathcal{C}_m$ and for any $q \in Q$ by the following equalities:

$$\Lambda_{G,m,t}(C)(q) = \begin{cases} \Lambda_{r,m,a}(C(q_2)) & \text{if } q = q_1 \\ C(q) & \text{if } q \neq q_1 \end{cases}$$

$$\Gamma_{G,m,t}(C)(q) = \begin{cases} \Gamma_{r,m,a}(C(q_1)) & \text{if } q = q_2 \\ C(q) & \text{if } q \neq q_2 \end{cases}$$

Observe that C_D is a fix-point solution of the data-flow problem $\Lambda_{G_D,m,t}(C_D) \sqsubseteq C_D$ for any transition $t \in T_D$ and C_I is a fix-point solution of the data-flow problem $\Gamma_{G_I,m,t}(C_I) \sqsubseteq C_I$ for any transition $t \in T_I$. In the next sections 3.1 and 3.2 we show that C_D and C_I can be characterized by these two data-flow analysis problems.

3.1 Reduction for C_D

The computation of C_D is reduced to a data-flow analysis problem for the m-graph G_D equipped with the semantics $(\Lambda_{G_D,m,t})_{t \in T_D}$.

Given an infinite path θ labelled by w, we denote by $\lambda_{r,m}(\theta)$ the vector $\lambda_{r,m}(w)$. Given a m-graph G labelled by Σ_r, we denote by $\Lambda_{G,m}$, the valuation $\mathrm{cl} \circ \mathrm{conv}(\lambda_{r,m}(\Theta_G))$ (recall that $\Theta_G(q)$ denotes the set of infinite paths starting from q). This notation is motivated by the following Proposition 3.3.

Proposition 3.3. *The valuation $\Lambda_{G,m}$ is the unique minimal valuation $C \in Q \to \mathcal{C}_m$ such that $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$ and such that $C(q) \neq \emptyset$ for any state $q \in Q$ satisfying $\Theta_G(q) \neq \emptyset$.*

The following Proposition 3.4 provides the reduction.

Proposition 3.4. $C_D = \Lambda_{G_D,m}$

Proof. We have previously proved that $\Lambda_{G_D,m,t}(C_D) \sqsubseteq C_D$ for any transition $t \in T_D$. Moreover, as $C_D(q_D) \neq \emptyset$ for any $q_D \in Q_D$, we deduce the relation $\Lambda_{G_D,m} \sqsubseteq C_D$ by minimality of $\Lambda_{G_D,m}$. For the other relation, just observe that $X_D \subseteq \lambda_{r,m}(\Theta_{G_D})$ and apply $\mathrm{cl} \circ \mathrm{conv}$. □

3.2 Reduction for C_I

The computation of C_I is reduced to data-flow analysis problems for the m-graphs G_S and G_I respectively equipped with the semantics $(\Gamma_{G_S,m,t})_{t \in T_S}$ and $(\Gamma_{G_I,m,t})_{t \in T_I}$.

Given a m-graph $G = (Q, \Sigma_r, T)$ and an *initial valuation* $C_0 \in Q \to \mathcal{C}_m$, it is well-known from Knaster-Tarski's theorem that there exists a unique minimal valuation $C \in Q \to \mathcal{C}_m$ such that $C_0 \sqsubseteq C$ and $\Gamma_{G,m,t}(C) \sqsubseteq C$ for any $t \in T$. We denote by $\Gamma_{G,m}(C_0)$ this unique valuation.

Symmetrically to the definitions of C_I and C_D we also consider the valuation $C_S \in Q_S \to \mathcal{C}_m$ defined by $C_S = \mathrm{cl} \circ \mathrm{conv}(X_S)$ where X_S is given by:

$$X_S(q_S) = \{\Gamma_{r,m,s}(0, \ldots, 0) \mid s \in S_r^* \quad \exists q_0 \in Q_0 \quad q_0 \xrightarrow{s} q_S\}$$

The reduction comes from the following Proposition 3.5 where $C_{S,0} \in Q_S \to \mathcal{C}_m$ and $C_{I,0} \in Q_I \to \mathcal{C}_m$ are the following two initial valuations:

$$C_{S,0}(q_S) = \begin{cases} \emptyset & \text{if } q_S \notin Q_0 \\ \{(0, \ldots, 0)\} & \text{if } q_S \in Q_0 \end{cases}$$

$$C_{I,0}(q_I) = \frac{1}{1-r} \bigsqcup_{\substack{q_S \in Q_S \\ (q_S, \star, q_I) \in T}} C_S(q_S)$$

Proposition 3.5. *$C_S = \Gamma_{G_S,m}(C_{S,0})$ and $C_I = \Gamma_{G_I,m}(C_{I,0})$.*

Proof. First observe that $X_S \subseteq \Gamma_{G_S,m}(C_{S,0})$ and $X_I \subseteq \Gamma_{G_I,m}(C_{I,0})$. Thus $C_S \sqsubseteq \Gamma_{G_S,m}(C_{S,0})$ and $C_I \sqsubseteq \Gamma_{G_I,m}(C_{I,0})$ by applying $\mathrm{cl} \circ \mathrm{conv}$. Finally, as $\Gamma_{r,m,a}$ and $\mathrm{cl} \circ \mathrm{conv}$ are commutative, we deduce that $\Gamma_{G_S,m,t}(C_S) \sqsubseteq C_S$ for any $t \in T_S$ and $\Gamma_{G_I,m,t}(C_I) \sqsubseteq C_I$ for any $t \in T_I$. The minimality of $\Gamma_{G_S,m}(C_{S,0})$ and $\Gamma_{G_I,m}(C_{I,0})$ provide $\Gamma_{G_S,m}(C_{S,0}) \sqsubseteq C_S$ and $\Gamma_{G_I,m}(C_{I,0}) \sqsubseteq C_I$. □

4 Infinite Paths Convex Hulls

In this section $G = (Q, \Sigma_r, T)$ is a m-graph. We prove that $\Lambda_{G,m}(q)$ is equal to the convex hull of a finite set of rational vectors. Moreover, we provide an algorithm for computing the minimal sets $\Lambda^0_{G,m}(q) \subseteq \mathbb{Q}^m$ for every $q \in Q$ such that $\Lambda_{G,m} = \mathrm{conv}(\Lambda^0_{G,m})$ in exponential time in the worst case.

A *fry-pan* θ in a graph G is an infinite path $\theta = t_1 \ldots t_i (t_{i+1} \ldots t_k)^\omega$ where $0 \leq i < k$ and where $t_1 = (q_0 \to q_1), \ldots t_k = (q_{k-1} \to q_k)$ are transitions such that $q_k = q_i$. A fry-pan is said *simple* if $q_0, \ldots, q_{k-1}$ are distinct states. The *finite set of simple fry-pans* starting from q is denoted by $\Theta^S_G(q)$. As expected, we are going to prove that $\Lambda_{G,m} = \mathrm{conv}(\lambda_{r,m}(\Theta^S_G))$ and $\lambda_{r,m}(\Theta^S_G(q)) \subseteq \mathbb{Q}^m$.

We first prove that $\lambda_{r,m}(\theta)$ is rational for any fry-pan θ. Given $\sigma \in \Sigma_r^+$, the following Lemma 4.1 shows that $\lambda_{r,m}(\sigma^\omega)$ is the unique solution of the rational linear system $\Lambda_{r,m,\sigma}(x) = x$. In particular $\lambda_{r,m}(\sigma^\omega)$ is a rational vector. From equality (1) given in page 7, we deduce that the vector $\lambda_{r,m}(\theta)$ is rational for any fry-pan θ.

Lemma 4.1. *$\lambda_{r,m}(\sigma^\omega)$ is the unique fix-point of $\Lambda_{r,m,\sigma}$ for any $\sigma \in \Sigma_r^+$.*

The following Proposition 4.2 (see the graphical support given in Fig. 2) is used in the sequel for effectively computing $\Lambda_{G,m}$ thanks to a fix-point iteration algorithm.

Proposition 4.2. *Let $t = (q, a, q')$ be a transition and let θ' be a simple fry-pan starting from q' such that the fry-pan $t\theta'$ is not simple. In this case there exists a minimal non-empty prefix π of $t\theta'$ terminating in q. Moreover the fry-pan θ such that $t\theta' = \pi\theta$ and the fry-pan π^ω are simple and such that $\Lambda_{r,m,a}(\lambda_{r,m}(\theta')) \in \mathrm{conv}(\{\lambda_{r,m}(\theta), \lambda_{r,m}(\pi^\omega)\})$.*

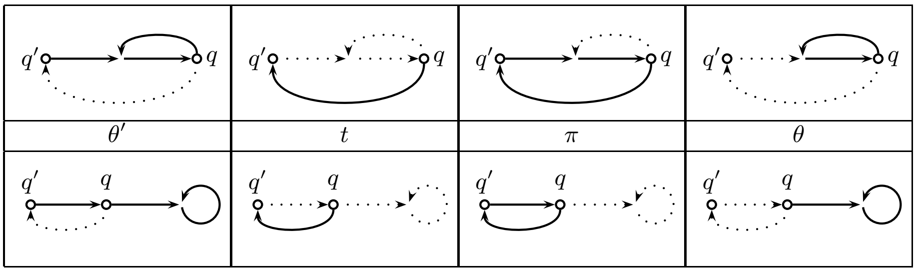

Fig. 2. A graphical support for Proposition 4.2 where θ' denotes a simple fry-pan starting from a state q' and $t = (q, a, q')$ is a transition such that the fry-pan $t\theta'$ is not simple. That means the state q is visited by θ'. Note that q is visited either once or infinitely often. These two situations are depicted respectively on the top line and the bottom line of the tabular.

Proof. As $t\theta'$ is not simple whereas θ' is simple we deduce that there exists a decomposition of $t\theta'$ into $\pi\theta$ where π is the minimal non-empty prefix of $t\theta'$ terminating in q. Let π be the non empty path with the minimal length. Observe that π is a simple cycle and thus π^ω is a simple fry-pan. Moreover, as θ is a suffix of the simple fry-pan θ', we also deduce that θ is a simple fry-pan. Observe that $\lambda_{r,m}(t\theta') = \lambda_{r,m}(\pi\theta)$. Moreover, as π is a cycle in a m-graph we deduce that m divides its length. Denoting by σ the label of π, we deduce that $\sigma \in (\Sigma_r^m)^+$. Now, observe that $\Lambda_{r,m,\sigma}(x) = (1 - r^{-\frac{|\sigma|}{m}})\lambda_{r,m}(\sigma^\omega) + r^{-\frac{|\sigma|}{m}} x$ for any $x \in \mathbb{R}^m$. We deduce that $\Lambda_{r,m,a}(\lambda_{r,m}(\theta')) = (1 - r^{-\frac{|\sigma|}{m}})\lambda_{r,m}(\pi^\omega) + r^{-\frac{|\sigma|}{m}}\lambda_{r,m}(\theta)$. Thus $\Lambda_{r,m,a}(\lambda_{r,m}(\theta')) \in \mathrm{conv}(\{\lambda_{r,m}(\theta), \lambda_{r,m}(\pi^\omega)\})$. □

From the previous Proposition 4.2 we deduce the following Proposition 4.3.

Proposition 4.3. *We have* $\Lambda_{G,m} = \mathrm{conv}(\lambda_{r,m}(\Theta_G^S))$.

We deduce that there exists a minimal finite set $\Lambda_{G,m}^0(q) \subseteq \mathbb{Q}^m$ such that $\Lambda_{G,m} = \mathrm{conv}(\Lambda_{G,m}^0)$. Note that an exhaustive computation of the whole set $\Theta_G^S(q)$ provides the set $\Lambda_{G,m}^0(q)$ by removing vectors that are convex combination of others. The efficiency of such an algorithm can be greatly improved by computing inductively subsets $\Theta(q) \subseteq \Theta_G^S(q)$ and get rid of any fry-pan $\theta \in \Theta(q)$ as soon as it becomes a convex combination of other fry-pans in $\Theta(q)\backslash\{\theta\}$. The algorithm Cycle is based on this idea.

Corollary 4.4. *The algorithm Cycle(G,m) terminates by iterating the main while loop at most* $|T|^{|Q|}$ *times and it returns* $\Lambda_{G,m}^0$.

Cycle($G = (Q, \Sigma_r, T)$ be a $m-$graph, $m \in \mathbb{N}\backslash\{0\}$)
  `for` each state $q \in Q$
    `if` $\Theta_G^S(q) \neq \emptyset$
      let $\theta \in \Theta_G^S(q)$
      let $\Theta(q) \leftarrow \{\theta\}$

    else
        let $\Theta(q) \leftarrow \emptyset$
while there exists $t = (q, a, q') \in T$ and $\theta' \in \Theta(q')$
    such that $\Lambda_{r,m,a}(\lambda_{r,m}(\theta')) \notin \mathrm{conv}(\lambda_{r,m}(\Theta(q)))$
    if $t\theta'$ is simple
        let $\Theta(q) \leftarrow \Theta(q) \cup \{t\theta'\}$
    else
        let π be the minimal strict prefix of $t\theta'$ terminating in q
        let θ be such that $t\theta' = \pi\theta$
        let $\Theta(q) \leftarrow \Theta(q) \cup \{\theta, \pi^\omega\}$
    while there exists $\theta_0 \in \Theta(q)$
        such that $\mathrm{conv}(\lambda_{r,m}(\Theta(q))) = \mathrm{conv}(\lambda_{r,m}(\Theta(q)\backslash\{\theta_0\}))$
        let $\Theta(q) \leftarrow \Theta(q)\backslash\{\theta_0\}$
return $\lambda_{r,m}(\Theta)$ $\quad //\Lambda^0_{G,m}$

5 Fix-point Computation

In this section we prove that the minimal post-fix-point $\Gamma_{G,m}(C_0)$ is effectively rational polyhedral for any m-graph $G = (Q, \Sigma_r, T)$ and for any rational polyhedral initial valuation $C_0 \in Q \to \mathcal{C}_m$. We deduce that the closed convex hull of sets symbolically represented by arithmetic automata are effectively rational polyhedral.

Example 5.1. Let $m = 1$ and $G = (\{q\}, \Sigma_r, \{t\})$ where $t = (q, r-1, q)$ and $C_0(q) = \{0\}$. Observe that the sequence $(C_i)_{i\in\mathbb{N}}$ where $C_{i+1} = C_i \sqcup \Gamma_{G,m,t}(C_i)$ satisfies $C_i(q) = \{x \in \mathbb{R} \mid 0 \leq x \leq r^i - 1\}$.

Recall that a Kleene iteration algorithm applied on the computation of $\Gamma_{G,m}(C_0)$ consists in computing the beginning of the sequence $(C_i)_{i\in\mathbb{N}}$ defined by the induction $C_{i+1} = C_i \bigsqcup_{t\in T} \Gamma_{G,m,t}(C_i)$ until an integer i such that $C_{i+1} = C_i$ is discovered. Then the algorithm terminates and it returns C_i. In fact, in this case we have $C_i = \Gamma_{G,m}(C_0)$. However, as proved by the previous Example 5.1 the Kleene iteration does not terminate in general. Nevertheless we are going to compute $\Gamma_{G,m}(C_0)$ by a Kleene iteration such that each C_i is *safely* enlarged into a C_i' satisfying $C_i \sqsubseteq C_i' \sqsubseteq \Gamma_{G,m}(C_0)$. This enlargement follows the acceleration framework introduced in [LS07b,LS07a] that roughly consists to compute the precise effect of iterating some cycles. This framework motivate the introduction of the monotonic function $\Gamma^W_{G,m}$ defined over the complete lattice $(Q \to \mathcal{C}_m, \sqsubseteq)$ for any $C \in Q \to \mathcal{C}_m$ and for any $q \in Q$ by the following equality:

$$\Gamma^W_{G,m}(C)(q) = \bigsqcup_{q \xrightarrow{\sigma} q} \Gamma_{r,m,\sigma}(C(q))$$

The following Proposition 5.2 shows that $\Gamma^W_{G,m}(C)$ is effectively computable from C and the function $\Lambda_{G,m}$ introduced in section 3. In this proposition, G_q denotes the graph G reduced to the strongly connected components of q.

q	$C_{I,0}(q)$	$\Gamma_{G_I,2}(C_{I,0})(q)$
1	$\{(0,0)\}$	$\mathbb{R}_+(1,3)$
2	$\emptyset$	$(1,1)+\mathbb{R}_+(3,2)$
3	$\emptyset$	$\mathbb{R}_+(3,2)$
4	$\emptyset$	$(1,0)+\mathbb{R}_+(3,2)$
5	$\emptyset$	$(0,1)+\mathbb{R}_+(1,3)$
6	$\emptyset$	$(2,1)+\mathbb{R}_+(3,2)$
7	$\emptyset$	$(0,2)+\mathbb{R}_+(1,3)$
8	$\emptyset$	$\mathrm{conv}(\{(1,0),(1,2)\})+\mathbb{R}_+(1,0)+\mathbb{R}_+(1,3)$
9	$\emptyset$	$(0,1)+\mathbb{R}_+(0,1)+\mathbb{R}_+(3,2)$

Table 1. The values of $C_{I,0}$ and $C_I = \Gamma_{G_I,2}(C_{I,0})$.

Proposition 5.2. *For any $C \in Q \to \mathcal{C}_m$, and for any $q \in Q$, we have:*

$$\Gamma^W_{G,m}(C)(q) = C(q) + \mathbb{R}_+(C(q) - \Lambda_{G_q,m}(q))$$

We now prove that the enlargement is sufficient to enforce the convergence of a Kleene iteration.

Proposition 5.3. *Let $C_0 \sqsubseteq C'_0 \sqsubseteq C_1 \sqsubseteq C'_1 \sqsubseteq \ldots$ be the sequence defined by the induction $C_{i+1} = C'_i \bigsqcup_{t\in T} \Gamma_{G,m,t}(C'_i)$ and $C'_i = \Gamma^W_{G,m}(C_i)$. There exists $i < |Q|$ satisfying $C_{i+1} = C_i$. Moreover, for such an integer i we have $C_i = \Gamma_{G,m}(C_0)$.*

Proof. Observe that $C_i \sqsubseteq C'_i \sqsubseteq \Gamma_{G,m}(C_0)$ for any $i \in \mathbb{N}$. Thus, if there exists $i \in \mathbb{N}$ such that $C_{i+1} = C_i$ we deduce that $C_i = \Gamma_{G,m}(C_0)$. Finally, in order to get the equality $C_{|Q|} = C_{|Q|-1}$, just observe by induction over i that we have following equality for any $q_2 \in Q$:

$$C'_i(q_2) = \bigsqcup_{\substack{q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma} q_1 \xrightarrow{\sigma_2} q_2 \\ |\sigma_1|+|\sigma_2| \leq i}} \Gamma_{G,m,\sigma_1\sigma\sigma_2}(C_0(q_1))$$

□

Example 5.4. Let us consider the 2-graph G_I obtained from the 2-graph depicted in the center of Fig. 1 and restricted to the set of states $Q_I = \{1,\ldots,9\}$. Let us also consider the function $C_{I,0} \in Q_I \to \mathcal{C}_2$ defined by $C_{I,0}(1) = \{(0,0)\}$ and $C_{I,0}(q) = \emptyset$ for $q \in \{2,\ldots,9\}$. Computing inductively the sequence $C_0 \sqsubseteq C'_0 \sqsubseteq C_1 \sqsubseteq C'_1 \sqsubseteq \ldots$ defined in Proposition 5.3 from $C_0 = C_{I,0}$ shows that $C_6 = C_5$ (see section G in appendix). Moreover, this computation provides the value of $C_I = \Gamma_{G_I,2}(C_{I,0})$ (see Table 1).

```
FixPoint(G = (Q, Σ_r, T) a m−graph, m ∈ ℕ\{0}, C_0 ∈ Q → 𝒞_m)
   let C ← C_0
   while there exists t ∈ T such that Γ_{G,m,t}(C) ⋢ C
      C ← Γ^W_{G,m}(C)
      let C ← C ⊔ ⨆_{t∈T} Γ_{G,m,t}(C)
   return C
```

Corollary 5.5. *The algorithm FixPoint(G,m,C_0) terminates by iterating the main while loop at most $|Q|-1$ times. Moreover, the algorithm returns $\Gamma_{G,m}(C_0)$.*

From Propositions 3.4 and 3.5 and corollaries 4.4 and 5.5 we get:

Theorem 5.6. *The closed convex hull of sets symbolically represented by arithmetic automata are rational polyhedral and computable in exponential time.*

Example 5.7. We follow notations introduced in Examples 3.1, 3.2 and 5.4. Observe that $C_I(8) - C_D(a) = \mathrm{conv}(\{(1,0),(1,2)\}) + \mathbb{R}_+(1,0) + \mathbb{R}_+(1,3)$ is exactly the closed convex hull of $X = \{x \in \mathbb{N}^2 \mid 3x[1] > x[2]\}$.

6 Conclusion

We have proved that the closed convex hull of sets symbolically represented by arithmetic automata are rational polyhedral. Our approach is based on acceleration in convex data-flow analysis. It provides a simple algorithm for computing this set. Compare to [Lat04] (1) our algorithm has the same worst case exponential time complexity, (2) it is not limited to sets of the form $\mathbb{Z}^m \cap C$ where C is a rational polyhedral convex set, (3) it can be applied to any set definable in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_r)$, (4) it can be easily implemented, and (5) it is not restricted to the most significant digit first decomposition. This last advantage directly comes from the class of arithmetic automata we consider. In fact, since the arithmetic automata can be non deterministic, our algorithm can be applied to least significant digit first arithmetic automata just by flipping the direction of the transitions. Finally, from a practical point of view, as the arithmetic automata representing sets in the restricted logic $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq)$ (where X_r is discarded) have a very particular structure, we are confident that the exponential time complexity algorithm can be applied on automata with many states like the one presented in [Lat04]. The algorithm will be implemented in TaPAS [LP08] (The Talence Presburger Arithmetic Suite) as soon as possible.

References

[BDEK07] Bernd Becker, Christian Dax, Jochen Eisinger, and Felix Klaedtke. Lira: Handling constraints of linear arithmetics over the integers and the reals. In *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 307–310. Springer, 2007.

[BH06] Bernard Boigelot and Frédéric Herbreteau. The power of hybrid acceleration. In *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 438–451. Springer, 2006.

[BJW05] Bernard Boigelot, Sébastien Jodogne, and Pierre Wolper. An effective decision procedure for linear arithmetic over the integers and reals. *ACM Trans. Comput. Log.*, 6(3):614–633, 2005.

[BLP06] Sébastien Bardin, Jérôme Leroux, and Gérald Point. Fast extended release. In *Computer Aided Verification, 18th International Conference, CAV 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, volume 4144 of *Lecture Notes in Computer Science*, pages 63–66. Springer, 2006.

[BRW98] Bernard Boigelot, Stéphane Rassart, and Pierre Wolper. On the expressiveness of real and integer arithmetic automata (extended abstract). In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 1998.

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

[Kar76] Michael Karr. Affine relationships among variables of a program. *Acta Informatica*, 6:133–151, 1976.

[Lat04] Louis Latour. From automata to formulas: Convex integer polyhedra. In *19th IEEE Symposium on Logic in Computer Science (LICS 2004), 14-17 July 2004, Turku, Finland, Proceedings*, pages 120–129. IEEE Computer Society, 2004.

[Ler04] Jérôme Leroux. The affine hull of a binary automaton is computable in polynomial time. In *Verification of Infinite State Systems, 5th International Workshop, INFINITY 2003, Marseille, France, September 2, 2003, Proceedings*, volume 98, pages 89–104. Elsevier, 2004.

[Ler05] Jérôme Leroux. A polynomial time presburger criterion and synthesis for number decision diagrams. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 147–156. IEEE Computer Society, 2005.

[LP08] Jérôme Leroux and Gérald Point. TaPAS : The Talence Presburger Arithmetic Suite. In *Submited*, 2008.

[LS07a] Jérôme Leroux and Grégoire Sutre. Accelerated data-flow analysis. In *Static Analysis, 14th International Symposium, SAS 2007, Kongens Lyngby, Denmark, August 22-24, 2007, Proceedings*, volume 4634 of *Lecture Notes in Computer Science*, pages 184–199. Springer, 2007.

[LS07b] Jérôme Leroux and Grégoire Sutre. Acceleration in convex data-flow analysis. In *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 27th International Conference, New Delhi, India, December 12-14, 2007, Proceedings*, volume 4855 of *Lecture Notes in Computer Science*, pages 520–531. Springer, 2007.

[Lug04] Denis Lugiez. From automata to semilinear sets: A logical solution for sets L(C, P). In *Implementation and Application of Automata, 9th International Conference, CIAA 2004, Kingston, Canada, July 22-24, 2004, Revised Selected Papers*, volume 3317 of *Lecture Notes in Computer Science*, pages 321–322. Springer, 2004.

[Sch87] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, New York, 1987.

A Proof of Proposition 3.3

Proposition 3.3. *The valuation $\Lambda_{G,m}$ is the unique minimal valuation $C \in Q \rightarrow \mathcal{C}_m$ such that $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$ and such that $C(q) \neq \emptyset$ for any state $q \in Q$ satisfying $\Theta_G(q) \neq \emptyset$.*

Proof. Let us first prove that $C = \mathrm{cl} \circ \mathrm{conv}(\lambda_{r,m}(\Theta_G))$ is a valuation in $Q \rightarrow \mathcal{C}_m$ such that $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$. We have the inclusion $\Lambda_{r,m,a}(\lambda_{r,m}(\Theta_G(q_2))) \subseteq \lambda_{r,m}(\Theta_G(q_1))$ for any transition $(q_1, a, q_2) \in T$. As $\mathrm{cl} \circ \mathrm{conv}$ and $\Lambda_{r,m,a}$ are commutative, the valuation $C = \mathrm{cl} \circ \mathrm{conv}(\lambda_{r,m}(\Theta_G))$ satisfies $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$.

Now, let us consider a valuation $C \in Q \rightarrow \mathcal{C}_m$ such that $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$ and such that $C(q) \neq \emptyset$ for any state $q \in Q$ satisfying $\Theta_G(q) \neq \emptyset$. Let us prove that $\mathrm{cl} \circ \mathrm{conv}(\lambda_{r,m}(\Theta_G)) \sqsubseteq C$. As $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$ an immediate induction shows that $\Lambda_{r,m,\sigma}(C(q)) \sqsubseteq C(q')$ for any finite path $\pi = (q \xrightarrow{\sigma} q')$. Let us consider an infinite path $\theta = (q \xrightarrow{w} F)$. As F is non empty, there exists a state $q' \in F$. Recall that F is the set of states visited infinitely often by the path θ. We deduce that there exists a cycle on q' and in particular $\Theta_G(q') \neq \emptyset$. This condition implies $C(q') \neq \emptyset$. Thus there exists $x' \in C(q')$. Moreover, as q' is visited infinitely often by θ, there exists a strictly increasing sequence $0 \leq i_0 < i_1 < \cdots$ of integers such that $q \xrightarrow{w(1)\dots w(i_j)} q'$. This path shows that the vector $x_j = \Lambda_{r,m,w(1)\dots w(i_j)}(x')$ is in $C(q)$. As $\lim_{j \rightarrow +\infty} x_j = \lambda_{r,m}(w)$ and $C(q)$ is closed we deduce that $\lambda_{r,m}(w) \in C(q)$. We have proved that $\lambda_{r,m}(\Theta_G) \subseteq C$. Therefore $\mathrm{cl} \circ \mathrm{conv}(\lambda_{r,m}(\Theta_G)) \sqsubseteq C$. $\square$

B Proof of Lemma 4.1

Lemma 4.1. $\lambda_{r,m}(\sigma^\omega)$ *is the unique fix-point of* $\Lambda_{r,m,\sigma}$ *for any* $\sigma \in \Sigma_r^+$.

Proof. As $\sigma\sigma^\omega$ and σ^ω are equal, equality (1) page 7 shows that $\lambda_{r,m}(\sigma^\omega)$ is a fix-point of $\Lambda_{r,m,\sigma}$. Moreover as the uniform form of the linear function $\Lambda_{r,m,a}$ is equal to $\Lambda_{r,m,0}$ we deduce that the uniform form of $\Lambda^m_{r,m,\sigma}$ is equal to $\Lambda^{m|\sigma|}_{r,m,0}$. Since $\Lambda^m_{r,m,0}(x) = r^{-1}x$ we have proved that the uniform form of $\Lambda^m_{r,m,\sigma}$ is $x \to r^{-|\sigma|}x$ for any $x \in \mathbb{R}^m$. Moreover, as $\lambda_{r,m}(\sigma^\omega)$ is a fix-point of $\Lambda^m_{r,m,\sigma}$ we deduce that $\Lambda^m_{r,m,\sigma}(x) = \lambda_{r,m}(\sigma^\omega) + r^{-|\sigma|}(x - \lambda_{r,m}(\sigma^\omega))$ for any $x \in \mathbb{R}^m$. In particular, if x is a fix-point of $\Lambda_{r,m,\sigma}$, we get $x = \lambda_{r,m}(\sigma^\omega) + r^{-|\sigma|}(x - \lambda_{r,m}(\sigma^\omega))$. As $r^{-|\sigma|} \neq 1$ we obtain $x = \lambda_{r,m}(\sigma^\omega)$. □

C Proof of Proposition 4.3

Proposition 4.3. *We have* $\Lambda_{G,m} = \mathrm{conv}(\lambda_{r,m}(\Theta_G^S))$.

Proof. From $\Theta_G^S(q) \subseteq \Theta_G(q)$ we deduce the inclusion $\mathrm{conv}(\lambda_{r,m}(\Theta_G^S)) \subseteq \Lambda_{G,m}$. Let us prove the other inclusion. Observe that $\Theta_G^S(q)$ is a finite set and in particular $\mathrm{conv}(\Theta_G^S(q))$ is a closed convex set for any $q \in Q$. Let us consider the function $C \in Q \to \mathcal{C}_m$ defined by $C = \mathrm{conv}(\lambda_{r,m}(\Theta_G^S))$. From Proposition 4.2, we deduce that $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$. Note also that $\mathcal{C}(q) \neq \emptyset$ for any state $q \in Q$ such that $\Theta_G(q) \neq \emptyset$. By minimality of $\Lambda_{G,m}$ we get the other inclusion $\Lambda_{G,m} \sqsubseteq C$. $\square$

D An Additional Example For Section 4

q	$\Theta^S_{G_1}(q)$	$-\Lambda^0_{G_1,2}(q)$
1	$(00)^\omega, (0111)^\omega, 01(0010)^\omega, 0100(11)^\omega$	$\{(0,0), (\frac{1}{3}, 1))\}$
2	$1(00)^\omega, (1011)^\omega, 101(0010)^\omega, 10100(11)^\omega$	$\{(\frac{1}{2}, 0), (\frac{7}{8}, \frac{1}{4}))\}$
3	$(1110)^\omega, 100(11)^\omega, 1(0010)^\omega$	$\{(1, \frac{2}{3}), (\frac{1}{2}, \frac{1}{3})\}$
4	$0(11)^\omega, (0100)^\omega, 01011(00)^\omega, 010(1101)^\omega$	$\{(\frac{1}{2}, 1), \{(0, \frac{2}{3})\}$
5	$11(00)^\omega, (1101)^\omega, (0010)^\omega, 00(11)^\omega$	$\{(\frac{2}{3}, 1), (\frac{1}{3}, 0)\}$
6	$(11)^\omega, (0001)^\omega, 0(1101)^\omega, 011(00)^\omega$	$\{(1,1), (0, \frac{1}{3})\}$
7	$(11)^\omega, (1000)^\omega, 10(1101)^\omega, 1011(00)^\omega$	$\{(\frac{2}{3}, 0), (1,1)\}$

Table 2. Some values computed

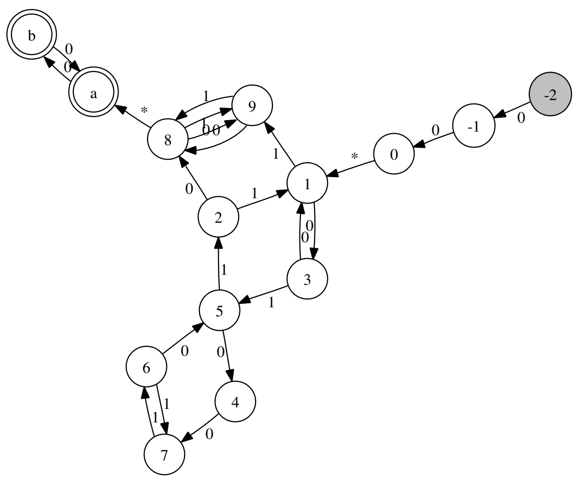

Fig. 3. An arithmetic automaton in basis 2 and in dimension 2.

Example D.1. Let us consider the 2-graph G labelled by Σ_2 and depicted in Fig. 3. We denote by G_1 the graph G restricted to the strongly connected component $\{1, \ldots, 7\}$. By enumerating all the possible simple fry-pans $\Theta^S_{G_1}(q)$ starting from a state q, observe that we get the values given in the Table 2. This table only provides the labels of the fry-pans in order to simplify the presentation. However, the fry-pans can be recovered from their labels since the graph is deterministic.

E Proof of Corollary 4.4

Corollary 4.4. *The algorithm* Cycle(G,m) *terminates by iterating the main while loop at most* $|T|^{|Q|}$ *times and it returns* $\Lambda^0_{G,m}$.

Proof. Observe that $\Theta(q) \subseteq \Theta^S_G(q)$ for any state q at any step of the algorithm. Moreover, each time the while loop is executed, the set $C(q) = \mathrm{conv}(\lambda_{r,m}(\Theta(q)))$ strictly increases. Thus, the set $\{\theta \in \Theta^S_G(q) \mid \lambda_{r,m}(\theta) \in C(q)\}$ strictly increase each time the while loop is executed. Observe that a simple fry-pan θ is uniquelly determined from its $|Q|$ first transitions. Thus $\sum_{q \in Q} |\Theta^S_G(q)| \leq |T|^{|Q|}$. We deduce that the algorithm terminates after executing at most $|T|^{|Q|}$ times the while loop. Finally, let us prove that when the algorithm terminates it returns $\Lambda^0_{G,m}$. It is sufficient to show that $C = \Lambda_{G,m}$ when it terminates. Note that the while loop condition is no longer valid. Thus $\Lambda_{G,m,t}(C) \sqsubseteq C$ for any transition $t \in T$. As $C(q) \neq \emptyset$ for any state $q \in Q$ such that $\Theta_G(q) \neq \emptyset$, by minimality of $\Lambda_{G,m}$ we deduce that $\Lambda_{G,m} \sqsubseteq C$. Thus $C = \Lambda_{G,m}$ when the algorithm terminates. We deduce that the algorithm returns $\Lambda^0_{G,m}$. □

F Proof of Proposition 5.2

We first prove the following two technical lemmas.

Lemma F.1. *For any $\sigma \in (\Sigma_r^m)^+$ and for any $x \in \mathbb{R}^m$ we have:*

$$\mathrm{cl} \circ \mathrm{conv}(\{\Gamma_{r,m,\sigma^i}(x) \mid i \in \mathbb{N}\}) = x + \mathbb{R}_+(x - \lambda_{r,m}(\sigma^\omega))$$

Proof. As $\lambda_{r,m}(\sigma^\omega)$ is a fix-point of the linear function Γ_{r,m,σ^i} and as the uniform form of the linear function Γ_{r,m,σ^i} is $\Gamma^{i|\sigma|}_{r,m,0}$, we deduce that $\Gamma_{r,m,\sigma^i}(x) = x + (r^{i\frac{|\sigma|}{m}} - 1)(x - \lambda_{r,m}(\sigma^\omega))$ for any $i \in \mathbb{N}$. As $\mathrm{cl} \circ \mathrm{conv}(\{r^{i\frac{|\sigma|}{m}} - 1 \mid i \in \mathbb{N}\}) = \mathbb{R}_+$ we deduce the lemma. □

Lemma F.2. *For any strongly connected m-graph $G = (Q, \Sigma_r, T)$ and for any state $q \in Q$, we have :*

$$\Lambda_{G,m}(q) = \mathrm{cl} \circ \mathrm{conv}(\{\lambda_{r,m}(\sigma^\omega) \mid q \xrightarrow{\sigma \in (\Sigma_r^m)^+} q\})$$

Proof. Let $C(q) = \mathrm{cl} \circ \mathrm{conv}(\{\lambda_{r,m}(\sigma^\omega) \mid q \xrightarrow{\sigma \in (\Sigma_r^m)^+} q\})$ be defined for any $q \in Q$. Note that for any cycle $\pi = (q \xrightarrow{\sigma \in (\Sigma_r^m)^+} q)$ we have $\pi^\omega \in \Theta_{G_q}(q)$. In particular $\lambda_{r,m}(\sigma^\omega) \in \Lambda_{G,m}(q)$. We deduce the inclusion $C(q) \sqsubseteq \Lambda_{G,m}(q)$. For the other inclusion, let us consider an infinite path $q \xrightarrow{w} F$ and let $q' \in F$. Since F is the set of states visited infinitely often, there exists a strictly increasing sequence of integers $0 < i_0 < i_1 < \cdots$ such that $q \xrightarrow{w(1)\dots w(i_j)} q'$ for any integer $j \geq 0$. As G is strongly connected, there exists a path $q' \xrightarrow{\sigma} q$. The cycle $q \xrightarrow{w(1)\dots w(i_j)\sigma} q$ shows that the vector $x_j = \lambda_{r,m}((w(1)\dots w(i_j)\sigma)^\omega)$ is in $C(q)$. As $\lim_{j\to+\infty} x_j = \lambda_{r,m}(w)$ and $C(q)$ is closed we have proved that $\lambda_{r,m}(w) \in C(q)$. Thus $\Lambda_{G,m}(q) \sqsubseteq C(q)$. □

Proposition 5.2. *For any $C \in Q \to \mathcal{C}_m$, and for any $q \in Q$, we have :*

$$\Gamma^W_{G,m}(C)(q) = C(q) + \mathbb{R}_+(C(q) - \Lambda_{G_q,m}(q))$$

Proof. Note that if there does not exist a $q \xrightarrow{\sigma} q$ then $\Lambda_{G_q,m}(q) = \emptyset$ and the previous equality is immediate. Otherwise, from Lemmas F.1 and F.2 we get the following equalities:

$$\begin{aligned}
\Gamma^W_{G,m}(C)(q) &= \bigsqcup_{q \xrightarrow{\sigma} q} \Gamma_{r,m,\sigma}(C(q)) \\
&= \bigsqcup_{x \in C(q)} \bigsqcup_{q \xrightarrow{\sigma \in (\Sigma_r^m)^+} q} \mathrm{cl} \circ \mathrm{conv}(\{\Gamma_{r,m,\sigma^i}(x) \mid i \in \mathbb{N}\}) \\
&= \bigsqcup_{x \in C(q)} \bigsqcup_{q \xrightarrow{\sigma \in (\Sigma_r^m)^+} q} x + \mathbb{R}_+(x - \lambda_{r,m}(\sigma^\omega)) \\
&= \bigsqcup_{x \in C(q)} x + \mathbb{R}_+(x - \Lambda_{G_q,m}(q))
\end{aligned}$$

In particular we deduce that $\Gamma_{G,m}^{W}(C)(q) \sqsubseteq C(q) + \mathbb{R}_{+}(C(q) - \Lambda_{G_q.m}(q))$. Conversely, let us consider $x \in C(q) + \mathbb{R}_{+}(C(q) - \Lambda_{G_q,m}(q))$. The vector x can be decomposed into $x = c_1 + h(c_2 - z)$ where $c_1, c_2 \in C(q)$, $z \in \Lambda_{G_q,m}(q)$ and $h \in \mathbb{R}_{+}$. Let us denote by $c = \frac{1}{1+h}(c_1 + hc_2)$. As $C(q)$ is convex we deduce that $c \in C(q)$. From $x = c + h(c - z)$ we deduce that $x \in \Gamma_{G,m}^{W}(C)(q)$. $\square$

G An Execution of Algorithm FixPoint

	1	2	3	4	5	6	7	8	9
C_0	$\{(0,0)\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
C'_0	L	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
C_1	L	$\emptyset$	L'	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$(0,1)+\mathbb{R}_+(0,1)$
C'_1	L	$\emptyset$	L'	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$(0,1)+D'$
C_2	L	$\emptyset$	L'	$\emptyset$	$(0,1)+L$	$\emptyset$	$\emptyset$	$(1,1)+D$	$(0,1)+D'$
C'_2	L	$\emptyset$	L'	$\emptyset$	$(0,1)+L$	$\emptyset$	$\emptyset$	$(1,1)+D$	$(0,1)+D'$
C_3	L	$(1,1)+L'$	L'	$(1,0)+L'$	$(0,1)+L$	$\emptyset$	$\emptyset$	$\mathrm{conv}(\{(1,0),(1,1)\})+D$	$(0,1)+D'$
C'_3	L	$(1,1)+L'$	L'	$(1,0)+L'$	$(0,1)+L$	$\emptyset$	$\emptyset$	$\mathrm{conv}(\{(1,0),(1,1)\})+D$	$(0,1)+D'$
C_4	L	$(1,1)+L'$	L'	$(1,0)+L'$	$(0,1)+L$	$\emptyset$	$(0,2)+L$	$\mathrm{conv}(\{(1,0),(1,2)\})+D$	$(0,1)+D'$
C'_4	L	$(1,1)+L'$	L'	$(1,0)+L'$	$(0,1)+L$	$\emptyset$	$(0,2)+L$	$\mathrm{conv}(\{(1,0),(1,2)\})+D$	$(0,1)+D'$
C_5	L	$(1,1)+L'$	L'	$(1,0)+L'$	$(0,1)+L$	$(2,1)+L'$	$(0,2)+L$	$\mathrm{conv}(\{(1,0),(1,2)\})+D$	$(0,1)+D'$

Where $L=\mathbb{R}_+(1,3)$, $L'=\mathbb{R}_+(3,2)$, $D=\mathbb{R}_+(1,0)+L$ and $D'=\mathbb{R}_+(0,1)+L'$.