Decomposition of Decidable First-Order Logics over Integers and Reals*

Florent Bouchy, Alain Finkel
LSV, ENS Cachan, CNRS
CNRS UMR 8643, Cachan, France
{bouchy,finkel}@lsv.ens-cachan.fr

Jérôme Leroux
Laboratoire Bordelais de Recherche en Informatique
CNRS UMR 5800, Talence, France
leroux@labri.fr

Abstract

We tackle the issue of representing infinite sets of real-valued vectors. This paper introduces an operator for combining integer and real sets. Using this operator, we decompose three well-known logics extending Presburger with reals. Our decomposition splits a logic into two parts : one integer, and one decimal (i.e. on the interval $[0, 1[$*). We also give a basis for an implementation of our representation.*

1 Introduction

Verification (and model-checking in particular) of infinite systems like timed automata [1] (and hybrid systems) and counter systems [5] need good symbolic representation classes ; by *good*, we mean having closure properties (by first-order logic operators) and decidability results (for testing inclusion and emptiness). Presburger arithmetic [27, 23] enjoys such good properties, and some efficient implementations (using finite automata) have been intensively used for the analysis of counter systems [6, 20, 14, 15].

Despite the fact that the complete arithmetic on reals is decidable [28], only some restricted classes of the first-order additive logic of reals (DBM, CPDBM, finite unions of convex polyhedra) have been used for the analysis of timed automata. This is mainly due to the fact that the algorithmic complexity of DBM is polynomial, which is the basis of efficient verification algorithms for timed automata in UPPAAL [11, 25].

However, we would like to be able to use both integers and reals, for at least two reasons. First, we want to analyse timed counter systems [2, 3, 13] in which the reachability sets contain vectors with both integers and reals. Second, we want to be able to use integers as parameters for a concise representation of pure reals : for instance, reals are used for the values of clocks and integers for expressing the parameters in CPDBM.

Fortunately, the first-order additive logic over integers and reals is decidable. Nevertheless, the algorithmic of sets combining integers and reals does not seem simple, even when it is based on finite automata like Real Vector Automata [13, 16] or weak RVA [8], or based on quantifier elimination [29].

For that matter, the algorithmic of Presburger (using finite automata) and variations of DBM are quite efficient. Hence, our idea is to reduce the algorithmic difficulty of the first-order additive logic of integers and reals (and of some subclasses and decidable extensions) by decomposing a complex set of integers and reals into a finite union of sums of integer sets and decimal sets. By decimal, we mean numbers in the dense inteval $[0, 1[$; then, we define a new class of sets as follows. Given n sets of integers $(Z_i)_{0\leq i\leq n}$ and n sets of decimals $(D_i)_{0\leq i\leq n}$, we introduce the operator *finite union of sums*, which builds the finite unions of the sums $Z_i + D_i$. This class is shown stable under boolean operations, cartesian product, quantification and reordering if both of the two initial classes are also stable.

One of our aims is then to re-use, in combining the best representations of these two initial sets $(Z_i)_{0\leq i\leq n}$ and $(D_i)_{0\leq i\leq n}$, the best libraries dealing with them to efficiently handle finite unions of $(Z_i + D_i)_{0\leq i\leq n}$ (for instance : PRESTAF [7] for the integers and PPL [4] for the reals).

We show that three of the main classes of mixed integer and real sets are in fact finite unions of sums of well-known classes. We prove that finite unions of sums of Presburger set of integers, and sets definable in the first-order additive logic of decimals are exactly the sets definable in the

*Work supported by the Agence Nationale de la Recherche, grant ANR-06-SETIN-001.

first-order logic of integers and reals. The finite unions of CPDBM are expressible as the finite unions of sums of Presburger-definable sets and DBM-definable decimal sets. Moreover, when we go beyond Presburger by considering RVA, we show that the class of sets representable by RVA in basis b is the finite unions of sums of Presburger extended with a predicate V_b (which gives integer powers in base b) and the additive logic of decimals extended with a predicate W_b (which, similarly to V_b, gives negative powers in base b).

2 Representations mixing integers and reals

In this section, we motivate our work with a small example of timed automaton. We show that extracting integers from reals can yield more concise formula than pure reals. Then we introduce an operator combining integer and real sets of vectors.

2.1 Timed Automata and DBM

In order to study real-life systems involving behaviours that depend on time elapsing, timed automata are probably the most used and well-known model for such systems. As described in [1], the basic idea of timed automata is to add real-valued variables (called clocks) to finite automata. These clocks model temporal behaviours of the system, flowing at a universal constant rate ; each clock can be compared to an integer constant, and possibly reset to 0. The only other guard allowed is called a diagonal constraint, consisting in comparing the difference of two clocks to an integer constant. As the clocks' values are unbounded, the state-space generated by a timed automaton is infinite ; therefore, regions are used to model a finite abstraction of the system's behaviour. Practically intractable because of its size, the region graph is then implemented as zones in most verification tools [11, 25, 18, 24] modelling such real-time systems.

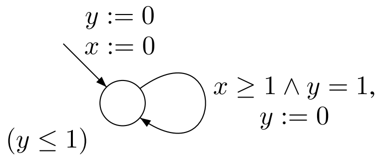
Technically, zones are represented by Difference Bound Matrices (DBM) [12, 21] in these tools. A DBM is a square matrix representing the constraints between n clocks defining a zone. Here, we see a DBM as a tuple $(\mathbf{c}, \prec)$, where $\mathbf{c} = (c_{i,j})_{0\leq i,j\leq n}$, $\prec = (\prec_{i,j})_{0\leq i,j\leq n}$, $c_{i,j} \in \mathbb{Z} \cup \{+\infty\}$, and $\prec_{i,j} \in \{\leq, <\}$. Each element of this tuple is an element of the square matrix, defining a DBM set as follows :

$$R_{\mathbf{c},\prec} = \{\mathbf{r} \in \mathbb{R}^n \mid \bigwedge_{0\leq i,j\leq n} r_i - r_j \prec_{i,j} c_{i,j}\}$$

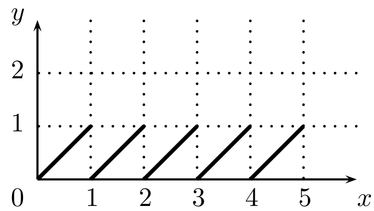
In order to deal with constraints involving only one clock, the fictive clock r_0 is always set to the value 0. An element $(c_{i,j}, \prec_{i,j})$ means that $r_i - r_j \prec_{i,j} c_{i,j}$, where r_i, r_j are clocks. Thus, each element of a DBM represents a diagonal constraint (i.e. a bounded difference). Finally, terms that do not represent any actual constraint are symbolized by $c_{i,j} = +\infty$.

2.2 About extensions of DBM

On the following example taken from [9], the timed automaton features 2 clocks x and y, and a unique location. The automaton's behaviour is very simple : y is reset to 0 as soon as it reaches 1, while x flows continually. In the initial state, the clocks are both set to 0. Moreover, an invariant in the location ensures that y never exceeds 1.



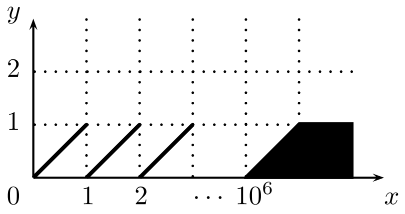
The clock diagram associated to the automaton explicitely shows this behaviour :



A classical forward analysis [17] is considered here, by computing the reachable states (i.e. $location \times clock\ values$) from the initial one (where $x = y = 0$). Then, we build the corresponding zones, each zone being represented by a DBM ; here, we have an infinite yet countable set of DBM as follows. Note that in this example $\prec$ is always $\leq$; therefore, we will omit it in the matrices.

$$\left\{ \begin{array}{cc} & \begin{array}{ccc} 0 & x & y \end{array} \\ \begin{array}{c} 0 \\ x \\ y \end{array} & \begin{pmatrix} 0 & -i & 0 \\ i+1 & 0 & i \\ 1 & -i & 0 \end{pmatrix} \end{array} \right\}_{i\geq 0}$$

In order to make the state-space computable, abstraction techniques are used to get a finite number of zones. The abstraction being used in most model-checkers is based on maximum constants : a clock c's valuation is considered equal to ∞ as soon as it exceeds the maximal constant to which c is ever compared. On the example, if a guarded transition $x \geq 10^6$ leads to another state, then the clock diagram becomes as follows :



More formally, this abstraction yields the following set of DBM :

$$\left\{ \left\{ \begin{array}{c|ccc} & 0 & x & y \\ \hline 0 & 0 & -i & 0 \\ x & i+1 & 0 & i \\ y & 1 & -i & 0 \end{array} \right\}_{0 \le i \le 10^6}, \begin{array}{c|ccc} & 0 & x & y \\ \hline 0 & 0 & \infty & 0 \\ x & \infty & 0 & \infty \\ y & 1 & -10^6 & 0 \end{array} \right\}$$

This set of DBM is finite, but remains huge : $10^6 + 2$ matrices need to be computed and memorized, which seems exaggerated, a fortiori for such a simple example. In [9], a more elaborate abstraction is proposed : the clocks' maximal constants are no more global to the system, but location-dependent. Another abstraction technique is proposed in [10], distinguishing between upper and lower bounds within maximal constants. To the best of our knowledge, these are the only zone-based abstraction techniques ; in each of them, the number of DBM still heavily depends on maximal constants.

Writing here such an infinite or huge number of DBM would have been impossible ; therefore, we naturally used a parametric representation of these DBM. Actually, this idea is also used by Constrained Parametric DBM (CPDBM) [2], which is the data structure implemented in the TREX [3] model-checker. CPDBM are indeed a more expressive version of DBM, extended in two steps. First, we consider PDBM, in which $c_{i,j}$ constants become $t_{i,j}$ arithmetical terms (the parameters). Such arithmetical terms t are given by the grammar $t ::= 0 \mid 1 \mid x \mid t - t \mid t + t \mid t * t$, where x belongs to a set $\mathcal{X}$ of real variables. Second, a PDBM becomes a CPDBM as terms are constrained by quantifier-free first-order formulas ϕ. Such formulas are defined by $\phi ::= t \le t \mid \neg\phi \mid \phi \vee \phi \mid Is\_int(t)$ (where the predicate $Is\_int(t)$ is true iff t is an integer). Each of the two sets of matrices hereinabove is in fact a single CPDBM.

Consider now another way to represent the set of reachable clock values. On the second diagram showing the abstraction, we can see an obvious regular pattern along x, defined by three shapes : ⋰, ◢, and ■. We define each shape as follows : ⋰ $= \{(x, y) \in [0, 1]^2 \mid x = y\}$, ◢ $= \{(x, y) \in [0, 1]^2 \mid x \ge y\}$, and ■ $= \{(x, y) \in [0, 1]^2\}$. If we want to represent the same set as the previous abstracted zones, but without DBM, we can express the periodicity of each pattern with integers. To formalize it, taking the union of the following three sums suffices :

$$\begin{aligned} &\Big(\{0, \ldots, 10^6 - 1\} \times \{0\} + ⋰\Big) \\ \bigcup &\Big(\{10^6\} \times \{0\} + ◢\Big) \\ \bigcup &\Big(\{10^6 + 1, \ldots, \infty\} \times \{0\} + ■\Big) \end{aligned}$$

This latter symbolic representation of such a reachability set is much smaller than DBM. Indeed, representing zones with DBM implies memorizing a possibly huge number of matrices, depending on the maximal constant for the clocks (one million, in this example). However, by introducing integers to express periodicity, we can reduce the representation to three small combinations of intervals. Moreover, we can even get rid of the abstraction, so as to get an exact representation for the same cost. CPDBM also have these advantages, but are undecidable because of the multiplication. Hence, let us specify a little more what is our representation : we take finite unions of reals, real numbers being decomposed as sums of integers and smaller reals (called decimals). These integers and reals can be defined using quantification, addition, and boolean operators.

Actually, our approach comes down to representing sets of real numbers by extracting their integer components ; the interesting point is that adding integers to real sets can simplify their representation and ease their handling. One might think that adding integers to such a first-order real logic would make it undecidable, but section 3 proves the opposite. Before that, we need to formalize our representation.

2.3 Composing integers and reals

Notations. The set $[0, 1[$ is denoted by $\mathbb{D}$ in the sequel. We also call a *decimal* (number) any $d \in \mathbb{D}$, and a *decimal set* any $D \subseteq \mathbb{D}$. We write $\mathbf{x}$ to denote a *vector* $(x_1, \ldots, x_n)$. Sometimes, in order to be concise, we use FO$(\ldots)$ to denote the sets represented by this first-order logic. However, it does not make our statements incorrect, because we mostly discuss the expressive power of such logics.

Let $\mathfrak{Z} \subseteq \mathsf{P}(\mathbb{Z}^n)$ and $\mathfrak{D} \subseteq \mathsf{P}(\mathbb{D}^n)$; we will assume in this paper that we are using $n-$dimensional vectors, with $n \in \mathbb{N}$. We denote by[1] $\mathfrak{Z} \uplus \mathfrak{D}$ the class of real vectors $R \subseteq \mathbb{R}^n$ s.t. $R = \bigcup_{i=1}^{p} (Z_i + D_i)$, with $(Z_i, D_i) \in \mathfrak{Z} \times \mathfrak{D}$

[1] The symbol $\uplus$ is sometimes used for the disjoint union, but we do not use such unions in this paper.

and $p \geq 1$.

Here are some examples of simple sets that might be often used, written as finite unions of sums of integers and decimals :

Example 1. *The empty set $\emptyset$ is written $\emptyset + \emptyset$. The set $\mathbb{R}^n$ is written $\mathbb{Z}^n + \mathbb{D}^n$. The set $\mathbb{Z}^n$ is written $\mathbb{Z}^n + \{\mathbf{0}\}$.*

Example 2. *The set $R_= = \{\mathbf{r} \in \mathbb{R}^2 \mid r_1 = r_2\}$ is written $\{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 = z_2\} + \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 = d_2\}$*

Example 3. *The set $R_\leq = \{\mathbf{r} \in \mathbb{R}^2 \mid r_1 \leq r_2\}$ is written :*

$$\{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 \leq z_2\} + \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 \leq d_2\}$$
$$\bigcup \{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 < z_2\} + \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 > d_2\}$$

Example 4. *The set $R_+ = \{\mathbf{r} \in \mathbb{R}^3 \mid r_1 + r_2 = r_3\}$ is written $\bigcup_{c \in \{0,1\}} \{\mathbf{z} \in \mathbb{Z}^3 \mid z_1 + z_2 + c = z_3\} + \{\mathbf{d} \in \mathbb{D}^3 \mid d_1 + d_2 = d_3 + c\}$, where c denotes a carry.*

The limits of our representation can be seen with the following counter-example. Consider the set $R = \bigcup_{j=1}^{\infty} \left(\{j\} + \left\{ \frac{1}{j+1} \right\} \right)$; note that we use $j + 1$ (and not simply j) to avoid the case where the decimal part is $\frac{1}{j} = 1$ for $j = 1$ (because it would not be a decimal, i.e. in $[0, 1[$). Our representation can not deal with such a set ; indeed, despite the fact that it is a union of sums of integers and decimals, we can see that the union is inherently infinite. We insist on the finiteness of the union in our representation, mainly for implementability reasons ; this will be discussed in section 5.

Now, let us consider the stability of our representation. We prove[2] that if $\mathfrak{Z} \subseteq \bigcup_{n \in \mathbb{N}} \mathsf{P}(\mathbb{Z}^n)$ and $\mathfrak{D} \subseteq \bigcup_{n \in \mathbb{N}} \mathsf{P}(\mathbb{D}^n)$ are stable by the classical first order operations then the class $\mathfrak{Z} \uplus \mathfrak{D} = \bigcup_{n \in \mathbb{N}} \mathfrak{Z}_n \uplus \mathfrak{D}_n$ where $\mathfrak{Z}_n = \mathfrak{Z} \cap \mathsf{P}(\mathbb{Z}^n)$ and $\mathfrak{D}_n = \mathfrak{D} \cap \mathsf{P}(\mathbb{D}^n)$ is also stable by these operations. The operations we consider are : boolean combinations (union, intersection, difference), cartesian product, quantification, and reordering. We use the following definitions for these last two operations. First, quantification is done by projecting away variables from the considered vector : $\forall R \subseteq \mathbb{R}^n$, $\exists_i R = \{(r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n) \mid \exists r_i \ (r_1, \ldots, r_{i-1}, r_i, r_{i+1}, \ldots, r_n) \in R\}$. Second, a reordering is a mere permutation function π of the variables order in a vector : $\forall R \subseteq \mathbb{R}^n$, $\pi R = \{(r_{\pi(1)}, \ldots, r_{\pi(n)}) \mid (r_1, \ldots, r_n) \in R\}$. Then, we introduce a generic definition for stability :

[2] Here we have to take unions, depending on the number of dimensions, for a technical purpose : the projection of a component in the vector.

Definition 5. *A class $\mathfrak{R} \subseteq \bigcup_{n \in \mathbb{N}} \mathsf{P}(\mathbb{R}^n)$ is* stable *if it is closed under boolean operations, cartesian product, quantification, and reordering.*

Notice that taking the union of two such sets is trivial, as they are already unions of integer and decimal parts. Then, observe that $(Z_1 + D_1) \cap (Z_2 + D_2) = (Z_1 \cap Z_2) + (D_1 \cap D_2)$ for any $Z_1, Z_2 \subseteq \mathbb{Z}^n$ and for any $D_1, D_2 \subseteq \mathbb{D}^n$; thus, the stability by union of $\mathfrak{Z}_n \uplus \mathfrak{D}_n$ provides the stability by intersection. From the equality $(Z_1 + D_1) \backslash (Z_2 + D_2) = ((Z_1 \backslash Z_2) + D_1) \cup (Z_1 + (D_1 \backslash D_2))$ we get the stability by difference. The stability by cartesian product is provided by $(Z_1 + D_1) \times (Z_2 + D_2) = (Z_1 \times Z_2) + (D_1 \times D_2)$. The stability by projection comes from $\exists_i R = (\exists_i Z) + (\exists_i D)$, where $R = Z + D$. Finally, the stability by reordering is obtained thanks to $\pi(Z + D) = (\pi Z) + (\pi D)$. We have proved the following proposition, which is later used in the proofs of theorem 7 and proposition 10 :

Proposition 6 (Stability)**.** *The class $\mathfrak{Z} \uplus \mathfrak{D}$ is stable if $\mathfrak{Z}$ and $\mathfrak{D}$ are stable.*

3 First-order additive logic over integers and reals

Using at the same time integers and reals in the whole arithmetic is known to be undecidable. However, when multiplication is left apart, the first-order additive logic is decidable ; its decidability has been suggested by Büchi, then proved by [16] with automata and by [29] using quantifier elimination. Actually, it can be seen as the Presburger logic [27] extended to the reals. This first-order logic FO $(\mathbb{R}, \mathbb{Z}, +, \leq)$ can encode complex linear constraints combining both integral and real variables. In this section we prove that sets definable in this logic can be decomposed into finite unions of $Z + R$ where Z is definable in FO $(\mathbb{Z}, +, \leq)$ and R is definable in FO $(\mathbb{D}, +, \leq)$. This result proves that complex linear constraints combining integral and real variables can be decomposed into linear constraints over integers, and linear constraints over reals. More precisely, we prove the following decomposition :

Theorem 7. *FO $(\mathbb{R}, \mathbb{Z}, +, \leq)$ = FO $(\mathbb{Z}, +, \leq) \uplus$ FO $(\mathbb{D}, +, \leq)$.*

Proof. First of all, observe that any set definable in the logic FO $(\mathbb{Z}, +, \leq) \uplus$ FO $(\mathbb{D}, +, \leq)$ is also definable in FO $(\mathbb{R}, \mathbb{Z}, +, \leq)$. Conversely, the sets $\mathbb{R}$ and $\mathbb{Z}$, the function $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and the predicate $\leq$ are definable in FO $(\mathbb{Z}, +, \leq) \uplus$ FO $(\mathbb{D}, +, \leq)$ from examples 1, 2, 3, 4. Thus, stability by first order operations provides the inclusion FO $(\mathbb{R}, \mathbb{Z}, +, \leq) \subseteq$ FO $(\mathbb{Z}, +, \leq) \uplus$ FO $(\mathbb{D}, +, \leq)$. We deduce the equality. ☐

Now, let us recall that sets definable in the Presburger logic $\text{FO}(\mathbb{Z},+,\leq)$ can be characterized thanks to *linear sets* [23]. In fact, a set $Z \subseteq \mathbb{Z}^n$ is definable in this logic if and only if it is equal to a finite union of linear sets $\mathbf{b}+P^*$ where $\mathbf{b} \in \mathbb{Z}^n$, P is a finite subset of $\mathbb{Z}^n$, and P^* denotes the set of finite sums $\sum_{i=1}^{k} p_i$ with $p_1,\ldots,p_k \in P$ and $k \in \mathbb{N}$. This geometrical characterization can be extended to the class of sets definable in $\text{FO}(\mathbb{Z},+,\leq) \uplus \text{FO}(\mathbb{D},+,\leq)$ by introducing the class of *polyhedral convex sets*. A set $C \subseteq \mathbb{R}^n$ is said *polyhedral convex* if C is defined by a finite conjunction of formulas $\langle\alpha,\mathbf{x}\rangle \prec c$ where $\alpha \in \mathbb{Z}^n$, $\prec \in \{\leq,<\}$ and $c \in \mathbb{Z}$. Recall that a *Fourier-Motzkin quantification elimination* proves that a set $C \subseteq \mathbb{R}^n$ is definable in $\text{FO}(\mathbb{R},+,\leq)$ if and only if it is equal to a finite union of polyhedral convex sets. In [22], the authors have proved the following geometrical characterization :
A set $R \subseteq \mathbb{R}^n$ is definable in $FO(\mathbb{R},\mathbb{Z},+,\leq)$ if and only if it is equal to a finite union of sets of the form $C+P^*$ where $C \subseteq \mathbb{R}^n$ is a polyhedral convex set and P is a finite subset of $\mathbb{Z}^n$.

3.1 Decomposing DBM-based representations

In this section, we characterize an extension of DBM. We denote by $\bigcup \text{DBM}_{\mathbb{D}}$ the finite unions of DBM sets which are included in $\mathbb{D}^n$. Notice that $\bigcup \text{DBM}_{\mathbb{D}}$ is stable by first order operations, thanks to a Fourier-Motzkin quantifier elimination.

A CP-DBM_L is a DBM where the vector $\mathbf{c}$ is no longer a constant, but a vector of parameters constrained by a formula $\phi(\mathbf{c})$ defined in a logic L. More precisely, a CP-DBM_L is a tuple $(\phi,\prec)$ representing a set $R_{\phi,\prec}$ s.t. :

$$R_{\phi,\prec} = \bigcup_{\mathbf{c} \models \phi} R_{\mathbf{c},\prec}$$

As introduced in [2], CPDBM correspond to CP-DBM_L where L is the first-order arithmetic without quantifiers ; in particular, multiplication is allowed in this formalism. In this section, we study another variation of DBM : CP-DBM_+, which is CP-DBM_L where L is the decidable Presburger logic $\text{FO}(\mathbb{Z},+,\leq)$. That is, CP-DBM_+ are CPDBM with quantifiers but without multiplication. We denote by $\bigcup \text{CP-DBM}_+$ the finite unions of $R_{\phi,\prec}$, i.e. finite unions of CP-DBM_+ sets.

We show that finite unions of CP-DBM_+ sets are in fact a combination of Presburger-definable sets and finite unions of DBM decimal sets :

Proposition 8. *We have $\bigcup \mathit{CP\text{-}DBM}_+ = \mathit{FO}(\mathbb{Z},+,\leq) \uplus \bigcup \mathit{DBM}_{\mathbb{D}}$.*

Proof. Let us first prove the inclusion $\supseteq$. Let us consider a DBM $(\mathbf{c},\prec)$ denoting a set $D \subseteq \mathbb{D}^n$ and a Presburger formula $\psi(\mathbf{x})$ denoting a set $Z \subseteq \mathbb{Z}^n$ and let us prove that $Z+D$ is a $\bigcup \text{CP-DBM}_+$ set. Observe that $\mathbf{r} \in Z+D$ if and only if there exists $\mathbf{z} \in Z$ such that $\mathbf{r}-\mathbf{z} \in D$. The condition $\mathbf{r}-\mathbf{z} \in D$ is equivalent to $\bigwedge_{0\leq i,j\leq n} r_i - r_j \prec_{i,j} c_{i,j}+z_i-z_j$. Let us consider the Presburger formula $\psi(\mathbf{p}) := \exists \mathbf{z} \in \mathbb{Z}^n\ p_{i,j} = c_{i,j}+z_i-z_j$ and observe that $R_{\psi,\prec} = Z+D$. We have proved the inclusion $\supseteq$.

For the converse inclusion, let us consider a CP-DBM_+ set $R_{\phi,\prec}$. Let $Z_{\mathbf{d}} = \mathbb{Z}^n \cap (R_{\phi,\prec} - \mathbf{d})$ indexed by $\mathbf{d} \in \mathbb{D}^n$. Observe that $Z_{\mathbf{d}}$ is actually the following set of vectors :

$$Z_{\mathbf{d}} = \bigcup_{\mathbf{c} \models \phi} \left\{ \mathbf{z} \in \mathbb{Z}^n | \bigwedge_{0\leq i,j\leq n} z_i - z_j \prec_{i,j} c_{i,j} + (d_j - d_i) \right\}$$

Since $d_j - d_i \in]-1,1[$ and $z_i - z_j, c_{i,j} \in \mathbb{Z}$ we deduce that $z_i - z_j \prec_{i,j} c_{i,j} + (d_j - d_i)$ is equivalent to $z_i - z_j \leq c_{i,j}$ if $d_i - d_j \prec_{i,j} 0$ and it is equivalent to $z_i - z_j \leq c_{i,j} - 1$ otherwise. Given a matrix $\mathbf{m} = (m_{i,j})_{0\leq i,j\leq n}$ such that $m_{i,j} \in \{0,1\}$ for any $0 \leq i,j \leq n$, we denote by $I_{\mathbf{m}}$ and $D_{\mathbf{m}}$ the following sets:

$$I_{\mathbf{m}} = \{\mathbf{z} \in \mathbb{Z}^n \mid \exists \mathbf{c}\ \phi(\mathbf{c}) \wedge \bigwedge_{0\leq i,j\leq n} z_i - z_j \leq c_{i,j} - m_{i,j}\}$$

$$D_{\mathbf{m}} = \{\mathbf{d} \in \mathbb{D}^n \mid \bigwedge_{0\leq i,j\leq n} (d_i - d_j \prec_{i,j} 0 \Longleftrightarrow m_{i,j} = 0)\}$$

Note that $D_{\mathbf{m}}$ is a DBM set and $Z_{\mathbf{d}} = I_{\mathbf{m}}$ for any $\mathbf{d} \in D_{\mathbf{m}}$. From $\bigcup_{\mathbf{m}} D_{\mathbf{m}} = \mathbb{D}^n$ we deduce that $R_{\phi,\prec} = \bigcup_{\mathbf{d}\in\mathbb{D}^n} Z_{\mathbf{d}} + \{\mathbf{d}\} = \bigcup_{\mathbf{m}} I_{\mathbf{m}} + D_{\mathbf{m}}$. We have proved that $R_{\phi,\prec}$ is definable in $\text{FO}(\mathbb{Z},+,\leq) \uplus \bigcup \text{DBM}_{\mathbb{D}}$. □

4 Beyond Presburger

We have just shown our decomposition to be working on $\text{FO}(\mathbb{R},\mathbb{Z},+,\leq)$ and below. Now, we prove that it can also be used on more expressive logics. We take the example of Real Vector Automata (RVA) [16], which is, to the best of our knowledge, the most expressive decidable implemented representation for sets of real and integer vectors. RVA are used in the tool LASH [14, 15]. In this section, the class of sets representable by RVA is proved decomposable into our formalism.

Let $b \geq 2$ be an integer called the *basis of decomposition*. We denote by $\Sigma_b = \{0,\ldots,b-1\}$ the finite set of *digits* and by $S_b = \{0,b-1\}$ the set of *sign digits*. An infinite word $\sigma = \mathbf{s}\mathbf{a}_1\ldots\mathbf{a}_k\star\mathbf{a}_{k+1}\mathbf{a}_{k+2}\ldots$ over the alphabet $\Sigma_b^n \cup \{\star\}$ is said *b-correct* if $\mathbf{s} \in S_b^n$ and $\mathbf{a}_i \in \Sigma_b^n$ for any $i \geq 1$. In this case, σ is called a *most significant digit first decomposition* of the following real vector $\rho_b(\sigma) \in \mathbb{R}^n$:

$$\rho_b(\sigma) = b^k \left(\frac{\mathbf{s}}{1-b} + \sum_{i\geq 1} b^{-i}\mathbf{a}_i \right)$$

A *Real Vector Automaton (RVA)* in basis b is a Büchi automaton A over the alphabet $\Sigma_b^n \cup \{\star\}$ such that the language $\mathsf{Lan}(A)$ recognized by A contains only b-correct words. The set $[\![A]\!]$ represented by A is defined by $[\![A]\!] = \{\rho_b(\sigma) \mid \sigma \in \mathsf{Lan}(A)\}$. A set $R \subseteq \mathbb{R}^n$ is said *b-recognizable* if there exists a RVA A in basis b such that $R = [\![A]\!]$.

According to [16], the class of b-recognizable sets can be logically characterized by $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_b)$ where X_b is an additional predicate. The predicate X_b over $\mathbb{R}^3$ is such that $X_b(x, u, a)$ is true if and only if there exists a most significant digit first decomposition $\sigma = sa_1 \dots a_k \star a_{k+1} \dots$ of x and an integer $i \in \mathbb{N}$ such that $a_i = a$ and $u = b^{k-i}$.

Theorem 9. *[16] A set $R \subseteq \mathbb{R}^n$ is b-recognizable if and only if it is definable in FO $(\mathbb{R}, \mathbb{Z}, +, \leq, X_b)$.*

In order to provide a decompostion of $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_b)$, the predicate X_b is proved expressible by two valuation functions V_b and W_b where :

- $V_b : \mathbb{Z}\backslash\{0\} \to \mathbb{Z}$ is the *integer valuation function* introduced in [19] and defined by $V_b(z) = b^j$, where $j \in \mathbb{Z}$ is the greatest integer such that $b^{-j}z \in \mathbb{Z}$.
- $W_b : \mathbb{D}\backslash\{0\} \to \mathbb{D}$ is the *decimal valuation function* defined by $W_b(d) = b^j$, where $j \in \mathbb{Z}$ is the least integer such that $b^{-j}d \notin \mathbb{D}$.

By expressing X_b in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, V_b, W_b)$ and V_b, W_b in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_b)$ we deduce that $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, X_b) = \mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, V_b, W_b)$. Finally, from proposition 6 and theorem 7, we get the following proposition.

Proposition 10. *$FO(\mathbb{R}, \mathbb{Z}, +, \leq, X_b) = FO(\mathbb{Z}, +, \leq, V_b) \uplus FO(\mathbb{D}, +, \leq, W_b)$.*

Moreover, it is clear that the logic $\mathrm{FO}(\mathbb{Z}, +, \leq, V_b) \uplus \mathrm{FO}(\mathbb{D}, +, \leq, W_b)$ extends $\mathrm{FO}(\mathbb{Z}, +, \leq) \uplus \mathrm{FO}(\mathbb{D}, +, \leq)$. However, even if the function W_b is crucial to logically characterize the class of b-recognizable sets, this predicate is not used in practice. In fact, in order to get efficient algorithms for manipulating Büchi automata (more precisely, minimization and determinization), we only consider sets $R \subseteq \mathbb{R}^n$ that can be represented by a *weak RVA* [14]. Recall that a Büchi automaton A is said *weak* if any strongly connected component S satisfies $S \subseteq F$ or $S \cap F = \emptyset$, where F is the set of accepting states. Unfortunately, the class of sets $R \subseteq \mathbb{R}^n$ representable by a weak RVA is not logically characterized since this class is not stable by first order operations (because of projection). In practice, since any set $R \subseteq \mathbb{R}^n$ definable in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, V_b)$ can be represented by a weak RVA, the RVA symbolic representation is only used for representing sets in this logic (i.e. without W_b). Just remark that $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq, V_b) = \mathrm{FO}(\mathbb{Z}, +, \leq, V_b) \uplus \mathrm{FO}(\mathbb{D}, +, \leq)$. Finally, note that weak RVA are used in the tool LIRA [8], whose benchmarks show very efficient computation times for sets defined in $\mathrm{FO}(\mathbb{R}, \mathbb{Z}, +, \leq)$.

5 Towards an implementation

From an implementation perspective, our decomposition has been designed to fit GENEPI's requirements. GENEPI [26] is a modular framework supporting Presburger-based solvers and model-checkers, distributed under GNU Public License. Its core consists of a plugin manager, which computes generic operations (such as boolean operations, quantification, satisfiability) on sets encoded as the solutions of Presburger-like formulas. Different implementations of these operations can be used as plugins ; existing ones include PRESTAF, LIRA, LASH, MONA, OMEGA, and PPL. We have begun to design a plugin for our decomposition, which uses two existing plugins : one for the integer part, and one for the decimal part.

Once this plugin is ready, any combination of two other plugins is possible : for example, one could try PRESTAF over integers and PPL over decimals. One could even be curious and study the efficiency of two instances of LIRA plugins, each one working on its own part (integer or decimal). Another benefit, coming from the new decomposition of RVA, would be to use the LASH plugin only on one part, and manage the other one differently : this might improve the effectiveness of RVA, which are very expressive but not really efficient in practice. So far, our first tests on small conjunctions of linear constraints show execution times close to the ones of LIRA.

What we need now for an implementation is a unique way to represent sets. Indeed, in order to avoid unduly complicated representations of sets, we have to make our representation canonical. Therefore, let us set the theoretical framework we use in practice.

Let $\mathfrak{Z} \subseteq \mathsf{P}(\mathbb{Z}^n)$ and $\mathfrak{D} \subseteq \mathsf{P}(\mathbb{D}^n)$. Notice that if $R = (Z + D_1) \cup (Z + D_2)$, then $R = Z + D$ with $D = D_1 \cup D_2$; we will always suppose that $\mathfrak{D}$ is closed under union wlog. Then, notice that $R \subseteq \mathbb{R}^n$ can be represented by a partially defined function f_R such that :

$$\begin{aligned} f_R : \mathfrak{Z} &\longrightarrow \mathfrak{D} \\ Z_i &\longmapsto D_i \end{aligned}$$

This function's interpretation is defined as $[\![f_R]\!] = \bigcup_{i=1}^{p} \Big(Z_i + f_R(Z_i) \Big)$, which matches the natural writing of R introduced in section 2.3. Note that this representation f_R is not unique.

For technical reasons, we extend f_R to a totally defined function $\overline{f_R}$ s.t. $\overline{f_R}(Z) = \emptyset$ if $Z \notin \mathsf{dom}(f_R)$ and $\overline{f_R}(Z) = f_R(Z)$ otherwise. Moreover, we define the support of $\overline{f_R}$ as $\mathsf{supp}(\overline{f_R}) = \{Z \mid \overline{f_R}(Z) \neq \emptyset\}$. In the remainder of this paper, we will use without ambiguity the notation f_R instead of $\overline{f_R}$.

We are now able to represent the set R with a function we wish to handle. Therefore, we want to identify f_R and $[\![f_R]\!]$: in order to do so, this latter interpretation has to be an injection. Generally, this is not the case : using the previous definitions, we could have different writings of $[\![f_R]\!]$. However, if the images by f_R are disjoint, then the interpretation $[\![f_R]\!]$ is an injection. Finally, for effectivity reasons, we will only consider functions whose support is finite. In the remainder of this section, we formalize this reasoning. Let $\mathcal{F}_{\mathfrak{Z}\to\mathfrak{D}} = \{f : \mathfrak{Z} \longrightarrow \mathfrak{D} \mid \mathsf{supp}(f) \text{ is finite}\}$.

Definition 11. *The* interpretation function $[\![.]\!]$ *associates to every* $f \in \mathcal{F}_{\mathfrak{Z}\to\mathfrak{D}}$ *a set of real vectors defined by* $[\![f]\!] = \bigcup_{Z\in \mathit{supp}(f)} \Big(Z + f(Z)\Big)$.

Notice that since $\mathsf{supp}(f)$ is finite, $\mathcal{F}_{\mathfrak{Z}\to\mathfrak{D}}$ do not suffice to represent every set of real vectors, as shown in the counter-example on page 4. Let us now restrict ourselves to the functions we handle :

Definition 12. *An* IDF (Integer-Decimal Function) *is a function* $f \in \mathcal{F}_{\mathfrak{Z}\to\mathfrak{D}}$ *such that* $\bigcup_Z f(Z) = \mathbb{D}^n$ *and such that* $Z \neq Z' \implies f(Z) \cap f(Z') = \emptyset$. *We denote them all by* $IDF_{\mathfrak{Z}\to\mathfrak{D}} = \{f \in \mathcal{F}_{\mathfrak{Z}\to\mathfrak{D}} \mid f \text{ is an IDF}\}$. *We also write* $[\![IDF_{\mathfrak{Z}\to\mathfrak{D}}]\!] = \{[\![f]\!] \mid f \in IDF_{\mathfrak{Z}\to\mathfrak{D}}\}$.

The sets from examples 1, 2, 3, 4 are represented by the following IDF :

Example 13. *The empty set $\emptyset$ is represented by the IDF $f_\perp$ defined by $f_\perp(Z) = \emptyset$ for any $Z \neq \emptyset$ and by $f_\perp(\emptyset) = \mathbb{D}^n$. The set $\mathbb{R}^n$ is represented by the IDF $f_\top$ (also noted $f_{\mathbb{R}^n}$) defined by $f_\top(\mathbb{Z}^n) = \mathbb{D}^n$ and $f_\top(Z) = \emptyset$ otherwise. The set $\mathbb{Z}^n$ is represented by the IDF $f_{\mathbb{Z}^n}$ defined by $f_{\mathbb{Z}^n}(\mathbb{Z}^n) = \{\mathbf{0}\}$ and $f_{\mathbb{Z}^n}(Z) = \emptyset$ otherwise.*

Example 14. *The set $R_= = \{\mathbf{r} \in \mathbb{R}^2 \mid r_1 = r_2\}$ is represented by the IDF $f_=$ defined by $f_=(Z_=) = D_=$, $f_=(\emptyset) = \mathbb{D}^2 \backslash D_=$ and $f_=(Z) = \emptyset$ otherwise, where:*

$$Z_= = \{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 = z_2\} \quad D_= = \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 = d_2\}$$

Example 15. *The set $R_\leq = \{\mathbf{r} \in \mathbb{R}^2 \mid r_1 \leq r_2\}$ is represented by the IDF $f_\leq$ defined by $f_\leq(Z_<) = D_>$, $f_\leq(Z_\leq) = D_\leq$ and $f_\leq(Z) = \emptyset$ otherwise where:*

$$Z_< = \{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 < z_2\} \quad D_> = \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 > d_2\}$$
$$Z_\leq = \{\mathbf{z} \in \mathbb{Z}^2 \mid z_1 \leq z_2\} \quad D_\leq = \{\mathbf{d} \in \mathbb{D}^2 \mid d_1 \leq d_2\}$$

Example 16. *The set $R_+ = \{\mathbf{r} \in \mathbb{R}^3 \mid r_1 + r_2 = r_3\}$ is represented by the IDF f_+ defined by $f_+(Z_0) = D_0$, $f_+(Z_1) = D_1$, $f_+(\emptyset) = \mathbb{D}^3 \backslash (D_1 \cup D_2)$ and $f_+(Z) = \emptyset$ otherwise where (intuitively $c \in \{0,1\}$ denotes a carry) :*

$$Z_c = \{\mathbf{z} \in \mathbb{Z}^3 \mid z_1 + z_2 + c = z_3\}$$
$$D_c = \{\mathbf{d} \in \mathbb{D}^3 \mid d_1 + d_2 = d_3 + c\}$$

Observe that any set in $[\![IDF_{\mathfrak{Z}_n\to\mathfrak{D}_n}]\!]$ is in $\mathfrak{Z}_n \uplus \mathfrak{D}_n$. The converse is obtained by proving the following proposition :

Proposition 17 (Closure by union)**.** *Let $R \in [\![IDF_{\mathfrak{Z}_n\longrightarrow\mathfrak{D}_n}]\!]$. Then, for any $Z \in \mathfrak{Z}_n$ and $D \in \mathfrak{D}_n$, we also have $R \cup (Z + D) \in [\![IDF_{\mathfrak{Z}_n\to\mathfrak{D}_n}]\!]$.*

Proof. We consider an IDF $f : \mathfrak{Z}_n \longrightarrow \mathfrak{D}_n$ such that $[\![f]\!] = R$ and two sets $Z \in \mathfrak{Z}_n$ and $D \in \mathfrak{D}_n$. We must prove that there exists an IDF $f' : \mathfrak{Z}_n \longrightarrow \mathfrak{D}_n$ such that $[\![f']\!] = R'$ with $R' = R \cup (Z + D)$. We consider the following function:

$$\begin{array}{llll} f' : & \mathfrak{Z}_n & \longrightarrow & \mathfrak{D}_n \\ & Z' & \longrightarrow & \Big(f(Z') \backslash D\Big) \displaystyle\bigcup_{Z'' \mid Z'' \cup Z = Z'} \Big(f(Z'') \cap D\Big) \end{array}$$

As expected we are going to prove that f' is an IDF such that $[\![f']\!] = R'$. We first show that f' is an IDF. First of all observe that $\bigcup_{Z'} f'(Z') = \mathbb{D}^n$. Next, let $Z'_1, Z'_2 \in \mathfrak{Z}_n$ such that $f'(Z'_1) \cap f'(Z'_2) \neq \emptyset$ then either $(f(Z'_1)\backslash D) \cap (f(Z'_2)\backslash D) \neq \emptyset$ or there exists Z''_1, Z''_2 such that $Z''_1 \cup Z = Z'_1$ and $Z''_2 \cup Z = Z'_2$ and $(f(Z''_1) \cap D) \cap (f(Z''_2) \cap D) \neq \emptyset$ since the other cases are not possible. But $(f(Z'_1)\backslash D) \cap (f(Z'_2)\backslash D) \neq \emptyset$ implies $f(Z'_1) \cap f(Z'_2) \neq \emptyset$ and since f is an IDF we get $Z'_1 = Z'_2$. And $(f(Z''_1) \cap D) \cap (f(Z''_2) \cap D) \neq \emptyset$ implies $Z''_1 = Z''_2$ and in particular $Z'_1 = Z'_2$. We have proved that f' is an IDF. Finally, equality $[\![f']\!] = R'$ comes from:

$$\begin{aligned}
[\![f']\!] &= \bigcup_{Z'} (Z' + f'(Z')) \\
&= \bigcup_{Z'} \Big((Z' + (f(Z')\backslash D)) \\
&\qquad \bigcup_{Z'' \mid Z'' \cup Z = Z'} (Z' + (f(Z'') \cap D))\Big) \\
&= \bigcup_{Z'} (Z' + (f(Z')\backslash D)) \bigcup_{Z''} ((Z'' \cup Z) + (f(Z'') \cap D)) \\
&= \bigcup_{Z''} (Z'' + ((f(Z'')\backslash D) \cup (f(Z'') \cap D))) \\
&\quad \cup (Z + D \cap (\bigcup_{Z''} f(Z''))) \\
&= [\![f]\!] \cup (Z + D)
\end{aligned}$$

$\square$

Hence, we have just proved the following proposition :

Proposition 18. $\mathfrak{Z}_n \uplus \mathfrak{D}_n = [\![IDF_{\mathfrak{Z}_n \to \mathfrak{D}_n}]\!]$

Let us prove that this new representation is canonical :

Proposition 19. *For any* $f_1, f_2 \in IDF_{\mathfrak{Z} \to \mathfrak{D}}$, $[\![f_1]\!] = [\![f_2]\!] \implies f_1 = f_2$ *.*

Proof. Consider $Z_1 \subseteq \mathbb{Z}^n$ and let us prove that $f_1(Z_1) \subseteq f_2(Z_1)$. Naturally, we can assume that $f_1(Z_1) \neq \emptyset$ since otherwise the inclusion is immediate. In this case, there exists $\mathbf{d} \in f_1(Z_1)$. As $(f_2(Z))_Z$ forms a sharing of $\mathbb{D}^n$, there exists Z_2 such that $\mathbf{d} \in f_2(Z_2)$. Let us prove that $Z_1 \subseteq Z_2$. We can assume that $Z_1 \neq \emptyset$. Let $\mathbf{z}_1 \in Z_1$ and observe that $\mathbf{r}_1 = \mathbf{z}_1 + \mathbf{d} \in [\![f_1]\!]$ and from $[\![f_1]\!] = [\![f_2]\!]$ we get $\mathbf{r}_1 \in [\![f_2]\!]$. Thus, there exists Z'_2 such that $\mathbf{r}_1 \in Z'_2 + f_2(Z'_2)$. Since $Z'_2 \subseteq \mathbb{Z}^n$ and $f_2(Z'_2) \subseteq \mathbb{D}^n$ we get $\mathbf{z}_1 \in Z'_2$ and $\mathbf{d} \in f_2(Z'_2)$. As $(f_2(Z))_Z$ forms a sharing of $\mathbb{D}^n$ and $\mathbf{d} \in f_2(Z_2) \cap f_2(Z'_2)$ we get $Z_2 = Z'_2$. In particular $\mathbf{z}_1 \in Z_2$ and we have proved that $Z_1 \subseteq Z_2$. The other inclusion $Z_2 \subseteq Z_1$ is obtained symetrically. We have proved that $Z_1 = Z_2$. Therefore, $f_1(Z_1) \subseteq f_2(Z_1)$ for any Z_1. By symmetry we deduce that $f_1(Z) = f_2(Z)$ for any Z. Therefore $f_1 = f_2$. $\square$

Notice that in practice, this canonicity depends on how the sets in $\mathfrak{Z}$ and $\mathfrak{D}$ are represented. Indeed, if any of these representations are not canonical, then we can not guarantee that an $IDF_{\mathfrak{Z} \to \mathfrak{D}}$ will be canonical.

6 Conclusion

We have proposed a decomposition of three known classes into finite unions of sums of integers and decimals, providing a new characterization. This decomposition can be applied to other subsets of real vectors, and possibly yield an interest in the exploration of decidable subclasses of the full arithmetic.

Our main goal is to use this representation of real vectors to verify infinite systems involving counters and clocks. Indeed, we wish to extend the abilities of the tool FAST [6] to the reals, so that it can compute exact reachability sets using acceleration techniques. A first step in such an implementation is the framework GENEPI, allowing to solve mixed integer and real constraints defined in first-order theories. Thus, our decomposition would allow working separately on integers and reals.

Another advantage of our decomposition is that we can now compute operations that we did not know how to perform on certain logics. For example, there is currently no algorithm computing directly the convex hull of a set defined in FO $(\mathbb{R}, \mathbb{Z}, +, \leq)$; but thanks to our decomposition, the problem reduces to the computation of the convex hull of Presburger-definable sets (as automata [19] or as semi-linear sets [23]), and the convex hull of sets definable in FO $(\mathbb{D}, +, \leq)$ (as finite unions of convex sets, using Fourier-Motzkin). We can push this reasoning to other symbolic representations and to other operations, such as upward or downward closure.

Globally, this method of separating integers and reals would speed up the software development process, because of the ease of using already existing plugins. As mentioned above, one can test the combination of any pair of plugins (provided there's at least one working on reals and another one on integers). Furthermore, a very interesting point is that a programmer can test his new plugin for real sets directly in GENEPI, and then extend its expressivity by coupling it with PRESTAF or another plugin handling integer sets. Obviously, the converse (extending an integer plugin to the reals) is also possible in the same fashion.

References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, volume 1855 of *Lecture Notes in Computer Science*, pages 419–434. Springer, 2000.

[3] A. Annichini, A. Bouajjani, and M. Sighireanu. TReX: A tool for reachability analysis of complex systems. In *Computer Aided Verification, 13th International Conference, CAV 2001, Paris, France, July 18-22, 2001, Proceedings*, volume 2102 of *Lecture Notes in Computer Science*, pages 368–372. Springer, 2001.

[4] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 2008. To appear.

[5] S. Bardin, A. Finkel, and J. Leroux. FASTer acceleration of counter automata in practice. In *Tools and Algorithms for the Construction and Analysis of Systems, 10th International Conference, TACAS 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*, volume 2988 of *Lecture Notes in Computer Science*, pages 576–590. Springer, 2004.

[6] S. Bardin, A. Finkel, J. Leroux, and L. Petrucci. FAST: Fast Acceleration of Symbolic Transition systems. In *Computer Aided Verification, 15th International Conference,*

CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings, volume 2725 of *Lecture Notes in Computer Science*, pages 118–121. Springer, 2003.

[7] S. Bardin, J. Leroux, and G. 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.

[8] B. Becker, C. Dax, J. Eisinger, and F. 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.

[9] G. Behrmann, P. Bouyer, E. Fleury, and K. G. Larsen. Static guard analysis in timed automata verification. In *Tools and Algorithms for the Construction and Analysis of Systems, 9th International Conference, TACAS 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, volume 2619 of *Lecture Notes in Computer Science*, pages 254–277. Springer, 2003.

[10] G. Behrmann, P. Bouyer, and K. G. Larsen. Lower and upper bounds in zone-based abstractions of timed automata. *International Journal on Software Tools for Technology Transfer (STTT)*, 8(3):204–215, 2006.

[11] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi. UPPAAL - a tool suite for automatic verification of real-time systems. In *Hybrid Systems III: Verification and Control, Proceedings of the DIMACS/SYCON Workshop, October 22-25, 1995, Ruttgers University, New Brunswick, NJ, USA*, volume 1066 of *Lecture Notes in Computer Science*, pages 232–243. Springer, 1995.

[12] B. Berthomieu and M. Menasche. An enumerative approach for analyzing time Petri nets. In *Information Processing, 9th World Computer Congress, IFIP'83, Paris, France, September 19-23, 1983, Proceedings*, pages 41–46. North-Holland/IFIP, 1983.

[13] B. Boigelot, L. Bronne, and S. Rassart. An improved reachability analysis method for strongly linear hybrid systems (extended abstract). In *Computer Aided Verification, 9th International Conference, CAV '97, Haifa, Israel, June 22-25, 1997, Proceedings*, volume 1254 of *Lecture Notes in Computer Science*, pages 167–178. Springer, 1997.

[14] B. Boigelot, S. Jodogne, and P. Wolper. On the use of weak automata for deciding linear arithmetic with integer and real variables. In *Automated Reasoning, First International Joint Conference, IJCAR 2001, Siena, Italy, June 18-23, 2001, Proceedings*, volume 2083 of *Lecture Notes in Computer Science*, pages 611–625. Springer, 2001.

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

[16] B. Boigelot, S. Rassart, and P. 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.

[17] P. Bouyer, F. Laroussinie, and P.-A. Reynier. Diagonal constraints in timed automata: Forward analysis of timed systems. In *Formal Modeling and Analysis of Timed Systems, Third International Conference, FORMATS 2005, Uppsala, Sweden, September 26-28, 2005, Proceedings*, volume 3829 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 2005.

[18] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine. Kronos: A model-checking tool for real-time systems. In *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*, pages 546–550. Springer, 1998.

[19] V. Bruyère, G. Hansel, C. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bulletin of the Belgian Mathematical Society*, 1(2):191–238, 1994.

[20] C. Darlot, A. Finkel, and L. van Begin. About FAST and TReX accelerations. *Electronic Notes in Theoretical Computer Science*, 128(6):87–103, 2005.

[21] D. L. Dill. Timing assumptions and verification of finite-state concurrent systems. In *Automatic Verification Methods for Finite State Systems, International Workshop, Grenoble, France, June 12-14, 1989, Proceedings*, volume 407 of *Lecture Notes in Computer Science*, pages 197–212. Springer, 1989.

[22] A. Finkel and J. Leroux. Presburger functions are piecewise linear. Research Report LSV-08-08, Laboratoire Spécification et Vérification, ENS Cachan, France, Mar. 2008. 9 pages.

[23] S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

[24] F. Laroussinie and K. G. Larsen. CMC: A tool for compositional model-checking of real-time systems. In *Formal Description Techniques and Protocol Specification, Testing and Verification, FORTE XI / PSTV XVIII'98, IFIP TC6 WG6.1 Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE XI) and Protocol Specification, Testing and Verification (PSTV XVIII), 3-6 November, 1998, Paris, France*, volume 135 of *IFIP Conference Proceedings*, pages 439–456. Kluwer, 1998.

[25] K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1-2):134–152, 1997.

[26] J. Leroux and G. Point. The GENEPI Framework, 2006. http://altarica.labri.fr/wiki/tools:tapas:genepi.

[27] M. Presburger. On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation. *Hist. Philos. Logic*, 12(2):225–233, 1991. Translated from the German and with commentaries by Dale Jacquette.

[28] A. Tarski. A problem concerning the notion of definability. *J. Symb. Log.*, 13(2):107–111, 1948.

[29] V. Weispfenning. Mixed real-integer linear quantifier elimination. In *Symbolic and Algebraic Computation, International Symposium, ISSAC'99, Vancouver BC, Canada, July 28-31, 1999, Proceedings*, pages 129–136. ACM, 1999.