Decomposition of Decidable First-Order Logics over Integers and Reals
Florent Bouchy, Alain Finkel, Jérôme Leroux

To cite this version:

HAL Id: hal-00345998
https://hal.archives-ouvertes.fr/hal-00345998
Submitted on 10 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 \([3]\) 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 \([21, 23]\) enjoys such good properties, and some efficient implementations (using finite automata) have been intensively used for the analysis of counter systems \([1, 21, 14, 15]\).

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 \([1, 25]\).

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 \([13, 14]\) or weak RVA \([8]\), or based on quantifier elimination \([24]\).

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 interval \([0, 1]\); 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, 23, 12, 24] modelling such real-time systems.

Technically, zones are represented by Difference Bound Matrices (DBM) [13, 2] 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 $(c, \prec)$, where $c = (c_{i,j})_{0 \leq i,j \leq n}$, $\prec = (\prec_{i,j})_{0 \leq i,j \leq n}$, $c_{i,j} \in \mathbb{Z} \cup \{+\infty\}$, and $\prec_{i,j} \in \{\leq, <\}$. Each element of this tuple is an element of the square matrix, defining a DBM set as follows:

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

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.

$$ y := 0 \quad x := 0 \quad (y \leq 1) \quad x \geq 1 \land y = 1, \quad y := 0 $$

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

```
0 1 2 3 4 5
\_\_\_\_\_
|   |   |
|   | y |
| y |
```

A classical forward analysis [7] 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.

$$ \begin{pmatrix} 0 & x & y \\ 0 & 0 & -i \\ i+1 & 0 & i \\ y & 1 & -i \end{pmatrix} $$

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 $\infty$ 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:

\[
\begin{align*}
0 & \quad 0 \\
0 & \quad -i \\
i + 1 & \quad 0 \\
1 & \quad -i
\end{align*}
\middle| \begin{align*}
0 & \quad 0 \\
0 & \quad \infty \\
\infty & \quad 0 \\
1 & \quad -10^6
\end{align*}
\middle| \begin{align*}
x \quad y
\end{align*}
\text{for } 0 \leq i \leq 10^6
\]

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 \[3\], 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 \[4\], 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 incorrect; therefore, we naturally used a parametric representation of these DBM. Actually, this idea is also used by Constrained Parametric DBM (CPDBM) \[5\], which is the data structure implemented in the TReX 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 \(t ::= 0 \mid 1 \mid x \mid t - t \mid t + t \mid t \cdot t\), where \(x\) belongs to a set \(X\) of real variables. Second, a PDBM becomes a CPDBM as terms are constrained by quantifier-free first-order formulas \(\phi\). Such formulas are defined by \(\phi ::= t \leq t \mid \neg \phi \mid \phi \lor \phi \mid Is_{\text{int}}(t)\) (where the predicate \(Is_{\text{int}}(t)\) is true if \(t\) is an integer). Each of the two sets of matrices hereinafter 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: \(\mathcal{Z}\), \(\mathcal{A}\), and \(\mathcal{B}\). We define each shape as follows: \(\mathcal{Z} = \{(x, y) \in [0, 1]^2 \mid x = y\}\), \(\mathcal{A} = \{(x, y) \in [0, 1]^2 \mid x \geq y\}\), and \(\mathcal{B} = \{(x, y) \in [0, 1]^2 \mid x < y\}\). 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:

\[
\begin{align*}
&\bigcup \left( \{0, \ldots, 10^6 - 1\} \times \{0\} + \mathcal{Z} \right) \\
&\bigcup \left( \{10^6\} \times \{0\} + \mathcal{A} \right) \\
&\bigcup \left( \{10^6 + 1, \ldots, \infty\} \times \{0\} + \mathcal{B} \right)
\end{align*}
\]

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 \[6\] 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) \(d \in \mathbb{D}\), and a **decimal set** any \(D \subseteq \mathbb{D}\). We write \(x\) to denote a vector \((x_1, \ldots, x_n)\). Sometimes, in order to be concise, we use \(\text{FO}(...)\) 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 \(\mathcal{Z} \subseteq \mathcal{P}(\mathbb{Z}^n)\) and \(\mathbb{D} \subseteq \mathcal{P}(\mathbb{D}^n)\); we will assume in this paper that we are using \(n\)-dimensional vectors, with \(n \in \mathbb{N}\). We denote by \(\downarrow_1 \mathcal{Z} \uplus \mathbb{D}\) the class of real vectors \(R \subseteq \mathbb{R}^n\) s.t. \(R = \bigcup_{i=1}^{p} (Z_i + D_i)\), with \((Z_i, D_i) \in \mathcal{Z} \times \mathbb{D}\)

---

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 \( \mathbb{Z}^n + \{0\} \).

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

**Example 3.** The set \( R_{\leq} = \{ r \in \mathbb{R}^3 \mid r_1 + r_2 = r_3 \} \) is written \( \bigcup_{j=0}^{\infty} \{\{i\} \cup \left\{\frac{1}{j+1}\right\}\} \); 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 \( \mathcal{I} \subseteq \bigcup_{n \in \mathbb{N}} \mathcal{P}(\mathbb{Z}^n) \) and \( \mathcal{D} \subseteq \bigcup_{n \in \mathbb{N}} \mathcal{P}(\mathbb{D}^n) \) are stable by the classical first order operations then the class \( \mathcal{I} \cup \mathcal{D} = \bigcup_{n \in \mathbb{N}} (\mathcal{I} \cup \mathcal{D}) \subseteq \bigcup_{n \in \mathbb{N}} \mathcal{P}(\mathbb{D}^n) \) 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 : \( \forall R \subseteq \mathbb{R}^n, \exists \bar{R} = \{(r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_n) \mid \exists r_i (r_1,\ldots,r_{i-1},r_i,r_{i+1},\ldots,r_n) \in R\} \). Second, a reordering is a mere permutation function \( \pi \) 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:

**Definition 5.** A class \( \mathcal{I} \subseteq \bigcup_{n \in \mathbb{N}} \mathcal{P}(\mathbb{R}^n) \) 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 \( \mathcal{I} \) and \( \mathcal{D} \) provides the stability by intersection. From the equality \((Z_1 + D_1) \setminus (Z_2 + D_2) = ((Z_1 \setminus Z_2) + D_1) \cup (Z_1 + (D_1 \setminus D_2))\) 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 \( \exists R = (\exists Z) + (\exists D) \), 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 \( \mathcal{I} \cup \mathcal{D} \) is stable if \( \mathcal{I} \) and \( \mathcal{D} \) 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 \([14,21]\) with automata and by \([29]\) using quantifier elimination. Actually, it can be seen as the Presburger logic \([7]\) extended to the reals. This first-order logic \(\text{FO}(\mathbb{R}, Z, +, \leq)\) 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 \(\text{FO}(\mathbb{Z}, +, \leq)\) and \( R \) is definable in \(\text{FO}(\mathbb{D}, +, \leq)\). 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.** \(\text{FO}(\mathbb{R}, Z, +, \leq) = \text{FO}(\mathbb{Z}, +, \leq) \cup \text{FO}(\mathbb{D}, +, \leq).\)

**Proof.** First of all, observe that any set definable in the logic \(\text{FO}(\mathbb{Z}, +, \leq) \cup \text{FO}(\mathbb{D}, +, \leq)\) is also definable in \(\text{FO}(\mathbb{R}, Z, +, \leq)\). Conversely, the sets \( R \) and \( Z \), the function \( + : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) and the predicate \( \leq \) are definable in \(\text{FO}(\mathbb{Z}, +, \leq) \cup \text{FO}(\mathbb{D}, +, \leq)\) from examples 1, 2, 3, and 4. Thus, stability by first order operations provides the inclusion \(\text{FO}(\mathbb{R}, Z, +, \leq) \subseteq \text{FO}(\mathbb{Z}, +, \leq) \cup \text{FO}(\mathbb{D}, +, \leq)\). We deduce the equality. \(\square\)
Now, let us recall that sets definable in the Presburger logic $\text{FO}(\mathbb{Z}, +, \leq)$ can be characterized thanks to linear sets \cite{1}. 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 $B + P$ where $B \subseteq \mathbb{Z}^n$, $P$ is a finite subset of $\mathbb{Z}^n$, and $P$ is 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 $\text{FO}(\mathbb{Z}, +, \leq) \cup \text{FO}(\mathbb{D}, +, \leq)$ 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 $\langle \alpha, x \rangle < c$ where $\alpha \in \mathbb{Z}^n$, $c \in \mathbb{Z}$ and $G$. 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 \cite{2}, the authors have proved the following geometrical characterization: A set $R \subseteq \mathbb{R}^n$ is definable in $\text{FO}(\mathbb{R}, +, \leq)$ 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 $\bigcup \text{DBM}_L$ the finite unions of DBM sets which are included in $\mathbb{D}^n$. Notice that $\bigcup \text{DBM}_L$ is stable by first order operations, thanks to a Fourier-Motzkin quantifier elimination.

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

$$R_{\phi, \prec} = \bigcup_{c\in\phi} R_{c, \prec}$$

As introduced in \cite{3}, 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$_L$, which is CP-DBM$_L$ where $L$ is the decidable Presburger logic $\text{FO}(\mathbb{Z}, +, \leq)$. That is, CP-DBM$_L$ are CPDBM with parameters but without multiplication. We denote by $\bigcup \text{CP-DBM}_L$ the finite unions of $R_{\phi, \prec}$, i.e. finite unions of CP-DBM$_L$ sets.

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

**Proposition 8.** We have $\bigcup \text{CP-DBM}_L = \text{FO}(\mathbb{Z}, +, \leq) \cup \text{DBM}_D$.

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

For the converse inclusion, let us consider a CP-DBM$_L$ set $R_{\phi, \prec}$. Let $Z_d = X$ % $\bigcup \text{CP-DBM}_L$ set indexed by $d \in \mathbb{D}^n$. Observe that $Z_d$ is actually the following set of vectors:

$$Z_d = \bigcup_{c \in \phi} \left\{ z \in \mathbb{Z}^n \mid \bigwedge_{0 \leq i,j \leq n} z_i - z_j \prec c_{i,j} + (d_i - d_j) \right\}$$

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

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

$$D_m = \{ d \in \mathbb{D}^n \mid \bigwedge_{0 \leq i,j \leq n} (d_i - d_j \prec c_{i,j} 0 \iff m_{i,j} = 0) \}$$

Note that $D_m$ is a DBM set and $Z_d = I_m + D_m$. From $I_m = \bigcup_{d \in \mathbb{D}^n} D_m$ we deduce that $R_{\phi, \prec} = \bigcup_{d \in \mathbb{D}^n} Z_d + \{d\} = I_m + D_m$. We have proved that $R_{\phi, \prec}$ is definable in $\text{FO}(\mathbb{Z}, +, \leq) \cup \text{DBM}_D$.

### 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) \cite{4}, 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 \cite{5}. 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 $S_b = \{0, \ldots, b-1\}$ the finite set of digits and by $S_b^n = \{0, b-1\}$ the set of sign digits. An infinite word $\sigma = a_1 \ldots a_k \ast a_{k+1} \ldots a_{2}\ldots$ over the alphabet $S_b^n \cup \{\ast\}$ is said $b$-correct if $s \in S_b^n$ and $a_i \in S_b^n$ for any $i \geq 1$. In this case, $\sigma$ is called a most significant digit first decomposition of the following real vector $p_b(\sigma) \in \mathbb{R}^n$:

$$p_b(\sigma) = b^k \left( \frac{s}{1-b} + \sum_{i \geq 1} b^{-i} a_i \right)$$

5
A Real Vector Automaton (RVA) in basis $b$ is a Büchi automaton $A$ over the alphabet $\Sigma_b^* \cup \{\star\}$ such that the language $\text{Lan}(A)$ recognized by $A$ contains only $b$-correct words. The set $[A]$ represented by $A$ is defined by $[A] = \{p_0(\sigma) \mid \sigma \in \text{Lan}(A)\}$. A set $R \subseteq \mathbb{R}^n$ is said $b$-recognizable if there exists a RVA $A$ in basis $b$ such that $R = \lfloor A \rfloor$.

According to $[16]$, the class of $b$-recognizable sets can be logically characterized by $\text{FO}(R, Z, +, \leq, X_b)$ where $X_b$ is an additional predicate. The predicate $X_b$ over $R^3$ is such that $X_b(x, u, a)$ is true if and only if there exists a most significant digit first decomposition $\sigma = a_1 \ldots a_k \star a_{k+1} \ldots$ of $x$ and an integer $i \in \mathbb{N}$ such that $a_i = a$ and $u = b^{i+1}$.

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

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

- $V_b : Z \setminus \{0\} \rightarrow Z$ is the integer valuation function introduced in $[13]$ and defined by $V_b(z) = b^{j}$, where $j \in Z$ is the greatest integer such that $b^{j} \leq z \in Z$.
- $W_b : \mathbb{N} \setminus \{0\} \rightarrow \mathbb{N}$ is the decimal valuation function defined by $W_b(d) = b^{j}$, where $j \in Z$ is the least integer such that $b^{j} \leq d \bot \in \mathbb{N}$.

By expressing $X_b$ in $\text{FO}(R, Z, +, \leq, V_b, W_b)$ and $V_b, W_b$ in $\text{FO}(R, Z, +, \leq, X_b)$ we deduce that $\text{FO}(R, Z, +, \leq, X_b) = \text{FO}(R, Z, +, \leq, V_b, W_b)$. Finally, from proposition $[3]$ and theorem $[3]$ we get the following proposition.

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

Moreover, it is clear that the logic $\text{FO}(Z, +, \leq, V_b) \uplus \text{FO}(D, +, \leq, W_b)$ extends $\text{FO}(Z, +, \leq) \uplus \text{FO}(D, +, \leq)$. 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 $[3]$. 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 $\text{FO}(R, Z, +, \leq, V_b)$ 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 $\text{FO}(R, Z, +, \leq, V_b) = \text{FO}(Z, +, \leq, V_b) \uplus \text{FO}(D, +, \leq)$. Finally, note that weak RVA are used in the tool LIRA $[9]$, whose benchmarks show very efficient computation times for sets defined in $\text{FO}(R, Z, +, \leq)$.

## 5 Towards an implementation

From an implementation perspective, our decomposition has been designed to fit GENEPi’s requirements. GENEPi $[24]$ 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 $\mathcal{Z} \subseteq \mathbb{P}(\mathbb{Z}^n)$ and $\mathcal{D} \subseteq \mathbb{P}(\mathbb{D}^n)$. 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 $\mathcal{D}$ 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:

\[
\begin{align*}
f_R : \mathcal{Z} & \longrightarrow \mathcal{D} \\
Z_i & \longmapsto D_i
\end{align*}
\]

This function’s interpretation is defined as $\lfloor f_R \rfloor = \bigcup_{i=1}^{p} (Z_i + f_R(Z_i))$, which matches the natu-
For technical reasons, we extend $f_R$ to a totally defined function $\overline{f_R}$ s.t. $\overline{f_R}(Z) = \emptyset$ if $Z \notin \text{dom}(f_R)$ and $\overline{f_R}(Z) = f_R(Z)$ otherwise. Moreover, we define the support of $f_R$ as $\text{supp}(\overline{f_R}) = \{ Z \mid \overline{f_R}(Z) \neq \emptyset \}$. In the reminder 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 reminder of this section, we formalize this reasoning.

Let $\mathcal{F}_{3 \rightarrow \mathbb{D}} = \{ f : \mathbb{Z} \rightarrow \mathbb{D} \mid \text{supp}(f) \text{ is finite} \}$.

**Definition 11.** The interpretation function $[\cdot]$ associates to every $f \in \mathcal{F}_{3 \rightarrow \mathbb{D}}$ a set of real vectors $\{ Z \mid f(Z) \}$ defined by $[f] = \bigcup_{Z \in \text{supp}(f)} (Z + f(Z))$.

Notice that since $\text{supp}(f)$ is finite, $\mathcal{F}_{3 \rightarrow \mathbb{D}}$ do not suffice to represent every set of real vectors, as shown in the counter-example on page 23. Let us now restrict ourselves to the functions we handle :

**Definition 12.** An IDF (Integer-Decimal Function) is a function $f \in \mathcal{F}_{3 \rightarrow \mathbb{D}}$ 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 $\text{IDF}_{3 \rightarrow \mathbb{D}} = \{ f \in \mathcal{F}_{3 \rightarrow \mathbb{D}} \mid f \text{ is an IDF} \}$. We also write $\llbracket \text{IDF}_{3 \rightarrow \mathbb{D}} \rrbracket = \{ [f] \mid f \in \text{IDF}_{3 \rightarrow \mathbb{D}} \}$.

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

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

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

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

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

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

$Z_c = \{ z \in \mathbb{Z}^3 \mid z_1 + z_2 + c = z_3 \}$

$D_c = \{ d \in \mathbb{D}^3 \mid d_1 + d_2 + d_3 = d + c \}$

Observe that any set in $\llbracket \text{IDF}_{3 \rightarrow \mathbb{D}} \rrbracket$ in $3_\mathbb{D} \cup \mathbb{D}_n$. The converse is obtained by proving the following proposition :

**Proposition 17 (Closure by union).** Let $R \in \llbracket \text{IDF}_{3 \rightarrow \mathbb{D}} \rrbracket$. Then, for any $Z \in 3_\mathbb{D}$ and $D \in \mathbb{D}_n$, we also have $R \cup (Z + D) \in \llbracket \text{IDF}_{3 \rightarrow \mathbb{D}} \rrbracket$.

**Proof.** We consider an IDF $f : 3_\mathbb{D} \rightarrow \mathbb{D}_n$ such that $[f] = R$ and two sets $Z \in 3_\mathbb{D}$ and $D \in \mathbb{D}_n$. We must prove that there exists an IDF $f' : 3_\mathbb{D} \rightarrow \mathbb{D}_n$ such that $[f'] = R'$ with $R' = R \cup (Z + D)$. We consider the following function:

$f' : 3_\mathbb{D} \rightarrow \mathbb{D}_n$

$Z' \rightarrow (f(Z') \setminus D) \bigcup (f(Z'' \cap D))$

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 $Z'_1$, $Z'_2 \in 3_\mathbb{D}$ such that $f'(Z'_1) \cap f'(Z'_2) \neq \emptyset$ then either $(f(Z'_1) \cap D) \cap (f(Z'_2) \setminus D) \neq \emptyset$ or there exists $Z'_1$, $Z'_2$ such that $Z'_1 \cup Z = Z'_1$ and $Z'_2 \cup Z = Z'_2$ and $(f(Z'_1) \cap D) \cap (f(Z'_2) \setminus D) \neq \emptyset$ since the other cases are not possible. But $(f(Z'_1) \setminus D) \cap (f(Z'_2) \setminus D) \neq \emptyset$ implies $f(Z'_1) \cap f(Z'_2) \neq \emptyset$ and since $f$ is an IDF we get $Z'_1 = Z'_2$. And $(f(Z'_1) \cap D) \cap (f(Z'_2) \setminus D) \neq \emptyset$ 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:

$[f'] = \bigcup_{Z'} (Z' + f'(Z'))$

$= \bigcup_{Z'} (f(Z') \setminus D) \bigcup (f(Z' \cap D))$

$= \bigcup_{Z' \cap D} (f(Z') \setminus D) \cup (f(Z' \cap D))$

$= \bigcup_{Z''} (f(Z' \cap D))$

$= \bigcup_{Z''} (f(Z' \cap D))$

$= \bigcup_{Z'} (Z' + D)$

$= [f] \cup (Z + D)$
Hence, we have just proved the following proposition:

**Proposition 18.** $\exists n \in \mathbb{D}_n = [\mathsf{IDF}_{3n} - \mathcal{D}]$

Let us prove that this new representation is canonical:

**Proposition 19.** For any $f_1, f_2 \in \mathsf{IDF}_{3n} - \mathcal{D}$. $[f_1] = [f_2] \implies f_1 = f_2$.

**Proof.** Consider $Z_1 \subseteq Z^n$ 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 $d \in f_1(Z_1)$. As $(f_2(Z))z$ forms a sharing of $\mathcal{D}^n$, there exists $Z_2$ such that $d \in f_2(Z_2)$. Let us prove that $Z_1 \subseteq Z_2$. We can assume that $Z_1 \neq \emptyset$. Let $z_1 \in Z_1$ and observe that $r_1 = z_1 + d \in [f_1]$ and from $[f_1] = [f_2]$ we get $r_1 \in [f_2]$. Thus, there exists $Z_2'$ such that $r_1 \in Z_2' + f_2(Z_2')$. Since $Z_2' \subseteq Z^n$ and $f_2(Z_2') \subseteq \mathcal{D}^n$ we get $z_1 \in Z_2'$ and $d \subseteq f_2(Z_2')$. As $(f_2(Z))z$ forms a sharing of $\mathcal{D}^n$ and $d \subseteq f_2(Z_2') \cap f_2(Z_2')$ we get $Z_2 = Z_2'$. In particular $z_1 \in Z_2$ and we have proved that $Z_1 \subseteq Z_2$. The other inclusion $Z_2 \subseteq Z_1$ is obtained symmetrically. 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 $f_1 = f_2$.

Notice that in practice, this canonicity depends on how the sets in $\mathcal{D}$ and $\mathcal{D}$ are represented. Indeed, if any of these representations are not canonical, then we can not guarantee that an $\mathsf{IDF}_{3n} - \mathcal{D}$ 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 [5] 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 $\mathsf{FO}(\mathbb{R}, Z, +, \leq)$; but thanks to our decomposition, the problem reduces to the computation of the convex hull of Presburger-definable sets (as automata [15] or as semi-linear sets [23]), and the convex hull of sets definable in $\mathsf{FO}(\mathcal{D}, +, \leq)$ (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**


B. Boigelot, S. Jodogne, and P. Wolper. An effective de-

B. Boigelot, S. Jodogne, and P. Wolper. On the use of weak


B. Boigelot, L. Bronne, and S. Rassart. An improved reach-

B. Berthomieu and M. Menasche. An enumerative ap-

G. Behrmann, P. Bouyer, E. Fleury, and K. G. Larsen. Static


