Regular Acceleration for Number Decision Diagrams (Technical Report)

Jérôme Leroux[1]

LaBRI (CNRS UMR 5800), MVTSI Team, Talence, France
leroux@labri.fr

Abstract. The verification of an infinite state system often reduce to the computation of the reflexive and transitive closure of some binary relations over infinite set of states. Sébastien Bardin and Alain Finkel shown in 2004 that up to our knowledge, all the algorithms computing exactly such a reflexive and transitive closure, compute in fact a richer structure, called acceleration, as an intermediate step. In this paper, we prove that if the acceleration of a binary relation can be represented by a Number Decision Diagram (an automata-based symbolic representation for manipulating infinite sets of integer vectors), then it is effectively computable.

1 Introduction

The verification of a system can often be reduced to the computation of an enough precise over approximation of its reachability set. In general, systems are infinite (either they manipulate unbounded variables: integers, clocks, reals, queue contents, on so on, or they are parameterized), and the computation of such an approximation is a difficult problem.

At least two large famillies of methods have been developped for performing this computation: (1) In the field of abstract interpretation, an over approximation of the reachability set, is successively refined until an enough precise approximation is computed. This is for instance the case of predicate abstraction. (2) In the field of symbolic model checking, the system is approximated to an enough simple one (that remains infinite in general in order to keep for instance the important variables that cannot be abstracted into a finite domain) such that the exact computation of the reachability set is possible.

This paper contribute to the second approach (2) for the class of counter systems.

Informally, a counter system is a finite graph extended with integer variables such that the semantics of each transition is defined by a Presburger-definable binary relation over integer vectors (recall that a set is Presburger-definable if it can be defined in the linear first order logic of the integers). The class of counter systems is expressive: for instance it contains the reset/transfer Petri Nets [DJS99], generalized broadcast protocols [EN98,Del00], and Minsky machines.

As the class of counter systems is very general and powerful, the price to pay is the undecidability of reachability properties, and we can hope to develop only "good semi-algorithms" (good means that the semi-algorithm often terminates in practice). Naturally, a semi-algorithm that just computes inductively the set of reachable states, is not a "good one" because it cannot converge in practice since counter systems have an infinite number of reachable states. Good semi-algorithms, are based on a symbolic representation for manipulating subsets of integer vectors, enough rich to be able to compute the reflexive and transitive closure $\mathcal{R}^*$ of the binary relations $\mathcal{R}$ corresponding to some loops of the system's graph.

In this paper, we study this computation for the Number Decision Diagrams (NDD) symbolic representation [WB00], the automata-based symbolic representation for manipulating sets of integer vectors encoded as strings of digit vectors (least or most significant digit first). This representation is expressive because any Presburger-definable sets can be represented and satisfies interesting properties (canonical representations, efficient algorithms for boolean operations and variable eliminations) that are well adapted for applications that require a lot of boolean manipulations such as model-checking.

As the binary relations corresponding to any loops of the system's graph, are Presburger-definable and in particular can be represented by a NDD, we have reduced our problem to the computation of a NDD representing $\mathcal{R}^*$ from a NDD representing $\mathcal{R}$. A lot of algorithms have been developed for this computation. In [Boi98], Boigelot provided an algorithm for relations $\mathcal{R}$ equal to a function satisfying some algebraic conditions. This result was extended in [FL02] and effectively implemented in the tool FAST [BFLP03] with very efficient algorithms described in [BFL04]. In [BLW03] and [BHV04], some algorithms for computing a NDD that over-approximate $\mathcal{R}^*$, have been provided.

In this paper, we are interested in unified these algorithms. Recently, Sébastien Bardin and Alain Finkel shown in [BF04] that up to our knowledge, all the algorithms computing exactly $\mathcal{R}^*$, compute in fact the richer set $\bigcup_{k\in\mathbb{N}} \mathcal{R}^k \times \{k\}$, called *acceleration* of $\mathcal{R}$ as an intermediate step, and next obtain $\mathcal{R}^*$ by eliminating the variable k.

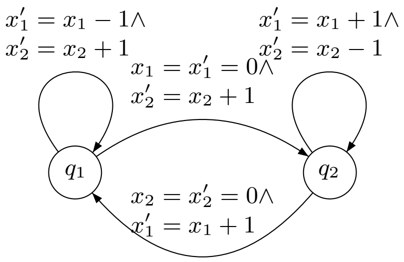
Our contributions:

- we prove that if the acceleration of a binary relation is representable in a symbolic representation that own some natural properties, then it is effectively computable. From [BHV04], we deduce an algorithm that effectivelly computes a NDD representing the acceleration of a binary relation, if such a NDD exists.
- we show that this algorithm in fact terminates on a lot of cases: all the examples studied in [BLW03], all the functions f manipulated by the tool FAST.
- we prove that the acceleration allow the computation of the reachability relations corresponding to nested loops.
- we explain why intuitively in [BLW03], they compute the NDD representing the relations $\mathcal{R}^{\leq 2^n}$ in order to approximate $\mathcal{R}^*$ rather than the NDD representing the relation $\mathcal{R}^k$.

Plan: In section 4, we provide the notations used in the sequel. Infinite state systems are defined in section 5. In this section, we also define the reachability set of a system, and recall the link between acceleration and fix point computation. In section 6, we recall the definition of a NDD, and finally in section 8, we present the class of counter systems, and some examples for which the acceleration can be successfully applied.

2 The Accelerated Symbolic Model-Checker FAST

FAST is a tool for analysing counter automata manipulating unbounded integer variables (see Fig. 1). The tool checks safety property by computing a formula in the decidable Presburger arithmetic FO $(\mathbb{N}, +, \leq)$ encoding the reachability set. Even if the reachability set is not recursive in general, FAST performs well on practical examples. Intuitively, the reachability set is computed inductivelly by a Kleene iteration. The tool starts from a Presburger formula encoding the initial configuration, and it collects with Presburger formulae reachable configurations discovered by exploring counter automaton transitions. The convergence of this iteration is improved by additing to the counter automata *shortcut-transitions* and *meta-transitions*. Intuitively *shortcut-transitions* correspond to sumerizes of paths with many transitions and *meta-transitions* correspond to the exact effect of exploring an arbitrary number of times a cycle in the counter automaton control-flow graph.



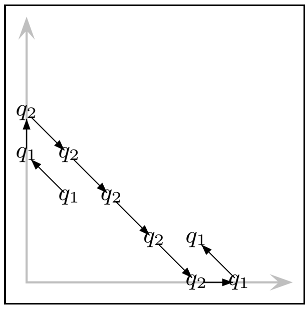


Fig. 1. On the left a counter automaton. On the right an example of execution based on the semantics following the configurations $(q_1, (1, 2))$,$(q_1, (0, 3))$,$(q_2, (0, 3))$,$(q_2, (1, 2))$, etc.

More formally, a m-dim *counter automata* is a finite graph $G = (Q, \Sigma, \Delta)$ where Q is the non-empty finite set of *states*, Σ is a finite set of Presburger formulae ϕ such that $\overrightarrow{x} = (x_1, \dots, x_m)$ and $\overrightarrow{x'} = (x'_1, \dots, x'_m)$ are the m-dim free variables of ϕ, and where $\Delta \subseteq Q \times \Sigma \times Q$ is the set of transitions. As expected, the set $S = Q \times \mathbb{N}^m$ is called the set of *configurations* and the *semantics* of a transition $t \in T$ is given by the binary relation $\xrightarrow{t}$ over the set of configurations by $(q, \overrightarrow{v}) \xrightarrow{t} (q', \overrightarrow{v'})$ if and only if there exists $\phi \in \Sigma$ such that $t = (q, \phi, q')$ and $(\overrightarrow{v}, \overrightarrow{v'}) \models \phi(\overrightarrow{x}, \overrightarrow{x'})$. A *path* from a state q to a state q' is a word of transitions $\pi = t_1 \dots t_l$ such that there exists $q_0, \dots, q_l \in Q$ and $\phi_1, \dots, \phi_l \in \Sigma$ satisfying $t_i = (q_{i-1}, \phi_i, q_i)$ for any $1 \leq i \leq l$. The semantics is naturally extended to paths by considering the binary relation $\xrightarrow{\pi}$ over the set of configurations defined by composition $\xrightarrow{\pi}=\xrightarrow{t_1} \cdots \xrightarrow{t_l}$. When $l \geq 1$ and $q_l = q_0$ the path π is denoted by θ and called a *cycle* on q_0.

We know recall the Kleene fix-point iteration algorithm for computing reachability sets of counter automata. Naturally, the *reachability set* of a counter automaton from an initial set $S_0 \subseteq S$ is the set of configurations s such that there exists $s_0 \in S_0$ and a path π satisfying $s \xrightarrow{\pi} s_0$. The Kleene algorithm computes the begining of the sequence $(S_k)_{k \geq 0}$ of configurations inductivelly defined by $S_k = S_{k-1} \cup \{s' \in S \mid \exists s \in S_{k-1}\ \exists t \in T\ s \xrightarrow{t} s'\}$ until an integer $k \geq 0$ such that $S_{k+1} = S_k$ is discovered. At this moment, the algorithm terminates at it retuns S_k. Note that the set S_k returned by the Kleene algorithm is equal to the reachability set from S_0. A Kleene fix-point algorithm can be effectivelly implemented with the Presburger arithmetic. Actually, assume that S_0 is encoded by a Presburger formula and observe that we can effectivelly compute inductively a Presburger formula encoding S_k for any $k \geq 0$. In particular we can effectivelly decide if $S_{k+1} = S_k$ and thus if such an integer exists we can effectivelly computed a Presburger formula encoding the reachability set from S_0.

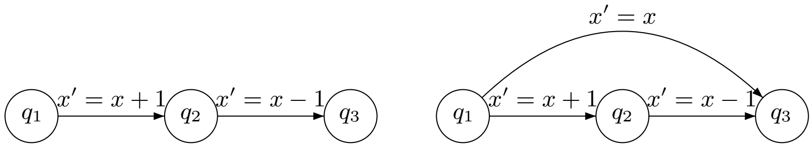


Fig. 2. On the left a counter automaton. On the right the counter automaton abtained by adding one shortcut-transition.

The first integer $k \geq 0$ such that $S_{k+1} = S_k$ can be reduced by introducing *shortcut-transitions*. Intuitively *shortcut-transitions* correspond to sumerizes of paths with many transitions (see Fig. 2). Given a path $\pi = t_1 \dots t_l$ where $t_i = (q_{i-1}, \phi_i, q_i)$ for any $1 \leq i \leq l$, observe that $(q_0, \overrightarrow{v}) \xrightarrow{\pi} (q_l, \overrightarrow{v'})$ if and only if $(\overrightarrow{v}, \overrightarrow{v'}) \models \phi_\pi(\overrightarrow{x}, \overrightarrow{x}')$ where ϕ_π is the following Presburger formula. Thus, *shortcut-transitions* can be added to counter automata without modify reachability properties. In practice, shortcut-transitions improve a Klenne fix-point iteration since intuitively the Presburger formula ϕ_π can be simplified (for instance quantification elimination) before starting the iteration.

$$\phi_\pi(\overrightarrow{x}, \overrightarrow{x'}) := \exists \overrightarrow{x_0}, \dots, \overrightarrow{x_k}\ \overrightarrow{x} = \overrightarrow{x_0} \wedge \bigwedge_{i=1}^{l} \phi_i(\overrightarrow{x_{i-1}}, \overrightarrow{x_i}) \wedge \overrightarrow{x_l} = \overrightarrow{x'}$$

Unfortunatelly, even if shortcut-transitions reduce the minimal integer $k \geq 0$ such that $S_{k+1} = S_k$, if such an integer does not exists, the Klenne fix-point algorithm never terminates whatever the shortcut-transitions added. In order to improve the termination of a Klenne fix-point iteration, we consider *meta-transitions*. Intuitivelly *meta-transitions* correspond to the exact effect of exploring an arbitrary number of times a cycle in the counter automaton control-flow graph (see Fig. 3). A *meta-transition* is a tuple (q, ϕ, q) where $\phi(\overrightarrow{x}, \overrightarrow{x'})$ is a Presburger formula such that there exists a cycle θ on a state q satisfying $(\overrightarrow{v}, \overrightarrow{v'}) \models \phi(\overrightarrow{x}, \overrightarrow{x'})$ if and only if there exists an integer $n \geq 0$ such that $(\overrightarrow{v}, \overrightarrow{v'}) \models \phi_{\theta^n}(\overrightarrow{x}, \overrightarrow{x'})$. This meta-transition is called the *acceleration* of θ. Note

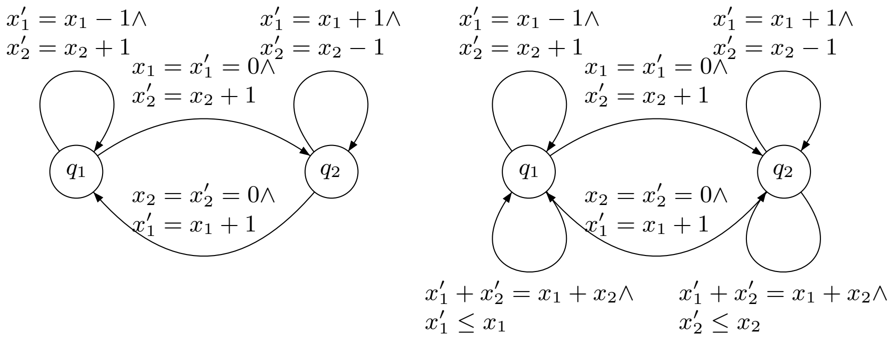


Fig. 3. On the left a counter automaton. On the right the counter automaton obtained by adding two meta-transitions.

that *meta-transitions* can be added to counter automata without modifing reachability properties. Observe that there exists cycles that cannot be accelerated. Moreover, since the one-step reachability relation of a Minsky machine can be encoded by a Presburger fomrula, we cannot even decide if a cycle can be accelerated.

The algorithm implemented in FAST first adds *shortcut-transitions* and *meta-transitions* to the counter automata under study. Then it starts a Klenne fix-point iterations on this modified counter automata. Naturally, some heuristics (not presented in this paper) are implemented in FAST in order to select "interesting" paths and cycles to be respectively sumerized into shortcut-transitions and accelerated into meta-transitions. The acceleration algorithm implemented in FAST is limited to particular cycles θ. In fact FAST only accelerate cycles θ that corresponds to *finite-monoid linear transformation*. A cycle θ is called a *finite-monoid* linear transformation if there exists a square matrix $M \in \mathcal{M}_n(\mathbb{Z})$ such that the multiplicative monoid $M^* = \{M^i \mid i \in \mathbb{N}\}$ is finite, a vector $\overrightarrow{b} \in \mathbb{Z}^m$ and a Presburger formula $\phi(\overrightarrow{x})$ such that $(\overrightarrow{v}, \overrightarrow{v'}) \models \phi_\theta(\overrightarrow{x}, \overrightarrow{x'})$ if and only if $\overrightarrow{v'} = M\overrightarrow{v} + \overrightarrow{b}$ and $\overrightarrow{v} \models \phi(\overrightarrow{x})$. Intuitively θ is a linear transformation guarded by the Presburger formula ϕ. FAST implements an algorithm for accelerating these transformations. The algorithm is based on the following theorem.

Theorem 2.1. *Any finite-monoid linear transformation can be effectivelly accelerated.*

In this paper we present an algorithm for computing the acceleration of cycles that are not necessary finite-monoid linear transformations. Since in FAST, sets encoded by Presburger formula are symbolically manipulated by arithmetic automata (introduced in the next section), we present an acceleration algorithm that performs the acceleration of a cycle θ directly on arithmetic automata symbolically representing the set of vectors $(\overrightarrow{v}, \overrightarrow{v'})$ that models $\phi_\theta(\overrightarrow{x}, \overrightarrow{x'})$.

3 Arithmetic automata

3.1 Least Significant Digit First Decomposition

The usual *least significant digit first decomposition* provides a natural way to associate words of digits to non-negative integer values. In this section, we extend this decomposition to non-negative integer vectors. Intuitively this extension is obtained component wise. The least significant digit first decomposition is parameterized by an integer $r \geq 2$ called *basis of decomposition* and an integer $m \geq 1$ called the *dimension*. The set $\Sigma_{r,m} = \{0, \dots, r-1\}^m$ is called the set of (r,m)-*digit vectors*. A (r,m)-*decomposition* of a non-negative integer vector $x \in \mathbb{N}^m$ is a word $\sigma \in \Sigma_{r,m}^*$ such that $x = \rho_{r,m}(\sigma)$ where $\rho_{r,m} : \Sigma_{r,m}^* \to \mathbb{N}^m$ is defined by the following equality for any $b_1, \dots, b_k \in \Sigma_{r,m}$:

$$\rho_{r,m}(b_1 \dots b_k) = \sum_{i=1}^{k} r^{i-1} b_i$$

Example 3.1. $(011, 0)$ is a $(2,1)$-decomposition of $6 = 2^1 + 2^2$.

Example 3.2. $(\epsilon, 0)$, $(0, 0)$, ..., $(0 \dots 0, 0)$ are the $(2,1)$-decompositions of 0.

Following notations introduced in [Ler04], function $\rho_{r,m}$ can be expressed thanks to the unique sequence $(\gamma_{r,m,\sigma})_{\sigma \in \Sigma_{r,m}^*}$ of functions $\gamma_{r,m,\sigma} : \mathbb{N}^m \to \mathbb{N}^m$ such that $\gamma_{r,m,\sigma_1.\sigma_2} = \gamma_{r,m,\sigma_1} \circ \gamma_{r,m,\sigma_2}$ for $\sigma_1, \sigma_2 \in \Sigma_{r,m}^*$, $\gamma_{r,m,\epsilon}$ is the identity function, and such that $\gamma_{r,m,b}(x) = rx + b$ for any $(b, x) \in \Sigma_{r,m} \times \mathbb{Z}^m$. In fact, an immediate induction over the length of σ provides the following equality for any (r,m)-decomposition σ:

$$\rho_{r,m}(\sigma) = \gamma_{r,m,\sigma}(\overrightarrow{0})$$

Observe that any integer vector in $\mathbb{N}^m$ has at least one (r,m)-decomposition. Such a decomposition is not unique since two words $\sigma_1, \sigma_2 \in \sigma_{r,m}^*$ represents the same vector if and only if $\sigma_1 \overrightarrow{0}^* \cap \sigma_2 \overrightarrow{0}^* \neq \emptyset$.

3.2 Arithmetic Automata

Function $\rho_{r,m}$ provides a natural way to associate the language $\mathcal{L} = \rho_{r,m}^{-1}(X)$ to any set $X \subseteq \mathbb{N}^m$. Intuitively $\rho_{r,m}^{-1}(X)$ is the language of (r,m)-decompositions of vectors in X. Such a language $\mathcal{L}$ is said (r,m)-*saturated* since it satisfies $\sigma_1 \in \mathcal{L}$ if and only if $\sigma_2 \in \mathcal{L}$ for any (r,m)-decomposition σ_1, σ_2 of the same vector. In this section we introduce an automata-based representation for recognizing (r,m)-saturated languages stable by moving the initial state.

Definition 3.3. *A* (r,m)-arithmetic automaton *is a tuple* $\mathcal{A} = (Q, \Sigma_{r,m}, \delta, q_0, F)$ *where* Q *is a non-empty finite set of states,* $\delta : Q \times \Sigma_{r,m} \to Q$ *is the transition function,* $q_0 \in Q$ *is the initial state and* $F \subseteq Q$ *is a final function satisfying* $q \in F$ *if and only* $q' \in F$ *for any* $q, q' \in Q$ *such that* $q' = \delta(q, \overrightarrow{0})$.

The set $X = \rho_{r,m}(\{\sigma \in \Sigma_{r,m}^* \mid \delta(q_0, \sigma) \in F\})$ is called the set *represented* by the (r,m)-arithmetic automaton $\mathcal{A}$.

3.3 Expressiveness

Recall [BHMV94] that a set $X \subseteq \mathbb{N}^m$ is said *r-definable* if it can be denoted by a formula in the first order theory FO $(\mathbb{N}, +, \leq, V_r)$ where $V_r : \mathbb{N} \rightarrow \mathbb{N}$ is the *r-valuation function* defined by $V_r(0) = 0$ and $V_r(x)$ is the greatest power of r that divides $x \in \mathbb{N} \backslash \{0\}$ (a $(r, 2)$-arithmetic automaton that represents V_r is provided in Fig. 4). From [BHMV94] we get the following theorem,

Theorem 3.4 ([BHMV94]). *A set $X \subseteq \mathbb{N}^m$ can be represented by an (r, m)-arithmetic automaton if and only if it is r-definable.*

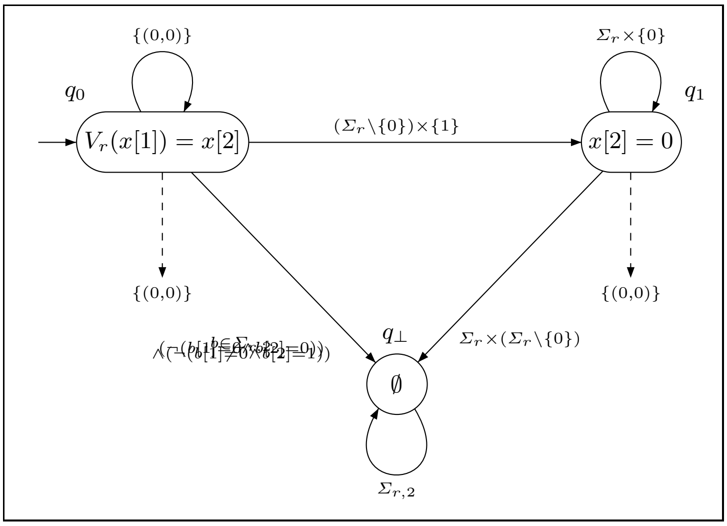


Fig. 4. $\mathcal{A}_{r,2}(\{x \in \mathbb{Z}^2 \mid V_r(x[1]) = x[2]\})$

Slightly different definitions of arithmetic automata are implemented in different tools. For instance, tool LASH implements *Number Decision Diagram (NDD)* an extension of arithmetic automata encoding a sign digit vector that allow to represents sets $X \subseteq \mathbb{Z}^m$. Tools LASH and LIRA also implements an extension with Buchi automata that allows to represents sets $X \subseteq \mathbb{R}^m$. Finally, the tool PRESTAF we developped implements roughly arithmetic automata. This last tool also includes an algorithm for effectivelly computing a Presburger formula from an arithmetic automaton represneting a Presburger-definable set (in few seconds a Presburger formula can be computed from large arithemetic automata with more than 100,000 states).

3.4 Moving The Initial State

The arithmetic automaton obtained from an arithmetic automaton $\mathcal{A}$ by replacing the initial state q_0 by another state q is denoted by $\mathcal{A}_q$. Given a set X implicitly represented by an arithmetic automaton $\mathcal{A}$, we denote by X_q the set represented by $\mathcal{A}_q$. Naturally the set X_q is r-definable for any state q. In this section we provide the following characterization.

Proposition 3.5. *$X_{q_2} = \gamma_{r,m,w}^{-1}(X_{q_1})$ for any path $(q_1, w, q_2) \in Q \times \Sigma_{r,m}^* \times Q$.*

Proof. Consider $x \in \gamma_{r,m,w}^{-1}(X_{q_1})$ and let us prove that $x \in X_{q_2}$. We denote by σ a (r, m)-decomposition of x. Note that $x_1 = \gamma_{r,m,w}(x)$ is in X_{q_1}. Thus there exists a (r, m)-decomposition σ_1 of x_1 such that $\delta(q_1, \sigma_1) \in F_0$. As $\mathcal{A}$ is an arithmetic automaton, by replacing σ_1 by a word in $\sigma_1 \overrightarrow{0}^*$ we can also assume that $|\sigma_1| \geq |w| + |\sigma|$. Now observe that $\gamma_{r,m,w}(x) = \rho_{r,m}(\sigma_1)$ and $x = \rho_{r,m}(\sigma)$ implies $\rho_{r,m}(w\sigma) = \rho_{r,m}(\sigma_1)$. Thus $w\sigma \overrightarrow{0}^* \cap \sigma_1 \overrightarrow{0}^* \neq \emptyset$. From $|\sigma_1| \geq |w| + |\sigma|$ and the previous non-empty intersection, there exists $k \in \mathbb{N}$ such that $\sigma_1 = w\sigma \overrightarrow{0}^k$. From $\delta(q_1, \sigma_1) \in F_0$ and $\delta(q_1, \sigma_1) = \delta(q_1, w\sigma \overrightarrow{0}^k) = \delta(q_2, \sigma \overrightarrow{0}^k)$ we deduce that $\delta(q_2, \sigma \overrightarrow{0}^k) \in F_0$. Therefore $\rho_{r,m}(\sigma \overrightarrow{0}^k) \in X_{q_2}$. As $\rho_{r,m}(\sigma \overrightarrow{0}^k, s) = \rho_{r,m}(\sigma) = x$, we have proved that $x \in X_{q_2}$.

Conversely, let us consider $x \in X_{q_2}$ and let us prove that $x \in \gamma_{r,m,w}^{-1}(X_{q_1})$. From $x \in X_{q_2}$ we deduce that there exists a (r, m)-decomposition σ of x such that $\delta(q_2, \sigma) \in F_0$. As $\delta(q_2, \sigma) = \delta(q_1, w\sigma)$ we deduce that $\rho_{r,m}(w\sigma) \in X_{q_1}$. As $\rho_{r,m}(w\sigma) = \gamma_{r,m,w}(\rho_{r,m}(\sigma))$, we have proved that $\gamma_{r,m,w}(x) \in X_{q_1}$. Therefore $x \in \gamma_{r,m,w}^{-1}(X_{q_1})$. $\square$

4 Notations

Throughout this paper, *intersection*, *union*, *difference* and *symmetric difference* of two sets A and B are written $A \cap B$, $A \cup B$, $A \backslash B$, and $A \Delta B = (A \backslash B) \cup (B \backslash A)$. We denote by $\mathbb{Q}$ and $\mathbb{N}$ respectively the set of rational numbers and non-negative integers. The *cardinality* of a finite set X is written $|X| \in \mathbb{N}$. Given a function $f : X \to Y$ and two sets A and B, we define $f(A) = \{f(a);\ a \in A\}$ and $f^{-1}(B) = \{x \in X;\ f(x) \in B\}$ (remark that A is not necessary a subset of X and B is not necessary a subset of Y).

A *relation* $\mathcal{R}$ is a subset of $S_1 \times S_2$ where S_1 and S_2 are two sets. We denote by $s_1 \mathcal{R} s_2$ if $(s_1, s_2) \in \mathcal{R}$. Such a relation is said *one-to-one* if there exists a unique $s_2 \in S_2$ such that $s_1 \mathcal{R} s_2$ for any $s_1 \in S_1$, and if there exists a unique $s_1 \in S_1$ such that $s_1 \mathcal{R} s_2$ for any $s_2 \in S_2$. A *binary relation* $\mathcal{R}$ over a set S is a relation $\mathcal{R} \subseteq S_1 \times S_2$ such that $S_1 = S = S_2$. Given such a binary relation $\mathcal{R}$, we denote by $\mathrm{Post}_{\mathcal{R}} : \mathcal{P}(S) \to \mathcal{P}(S)$ the function defined for any $X \subseteq S$ by $\mathrm{Post}_{\mathcal{R}}(X) = \{s' \in S;\ \exists x \in X;\ x \mathcal{R} s'\}$. The *concatenation* $\mathcal{R}_1.\mathcal{R}_2$ of two binary relations over a set S, is the binary relation over S defined by $s\mathcal{R}_1.\mathcal{R}_2 s'$ if and only if there exists $s_i \in S$ such that $s \mathcal{R}_1 s_i$ and $s_i \mathcal{R}_2 s'$. The *identity binary relation* I_S over S is defined by $s I_S s'$ if and only if $s = s'$. Given a binary relation $\mathcal{R}$ over a set S, the sequence $(\mathcal{R}^k)_{k \in \mathbb{N}}$ is uniquely defined by $\mathcal{R}^0$ is the identity binary relation over S, $\mathcal{R}^1 = \mathcal{R}$, and satisfying $\mathcal{R}^{k_1+k_2} = \mathcal{R}^{k_1}.\mathcal{R}^{k_2}$ for any $k_1, k_2 \in \mathbb{N}$. The binary relation $\mathcal{R}^*$ is defined by $\mathcal{R}^* = \bigcup_{k \in \mathbb{N}} \mathcal{R}^k$. Without ambiguity, a function $f : S \to S$ is also seen as the binary relation $\mathcal{R}$ over S defined by $s \mathcal{R} s'$ if and only if $s' = f(s)$. In particular $f^k(X)$, $f^*(X)$, $\mathrm{Post}_{\mathcal{R}}^k(X)$, and $\mathrm{Post}_{\mathcal{R}}^*(X)$ are well defined for any $k \in \mathbb{N}$ and $X \subseteq S$.

An *alphabet* Σ is a finite set. Given an alphabet Σ, we denote by Σ^+ the set of non-empty *words* over Σ. Given a non-empty word $\sigma = b_1 \dots b_k$ of $k \in \mathbb{N}\backslash\{0\}$ elements $b_i \in \Sigma$, and an integer $j \in \{1, \dots, k\}$, we denote by $\sigma[j] = b_j$. We denote by ϵ the empty word. As usual Σ^* denotes the set of words $\Sigma^+ \cup \{\epsilon\}$ and a *language* $\mathcal{L}$ is a subset of Σ^*. The *concatenation* of $\sigma_1 \in \Sigma^*$ and $\sigma_2 \in \Sigma^*$ is denoted by $\sigma_1.\sigma_2$. Given a word $\sigma \in \Sigma^*$, we define as usual σ^k where $k \in \mathbb{N}$ and $\sigma^* = \{\sigma^k;\ k \in \mathbb{N}\}$. The *length* of a word $\sigma \in \Sigma^*$ is denoted by $|\sigma| \in \mathbb{N}$. We also denote by $\Sigma^{\leq n}$ where $n \in \mathbb{N}$ the set of words $\sigma \in \Sigma^*$ such that $|\sigma| \leq n$.

A *graph* G is a tuple $G = (Q, T, \alpha, \beta)$ such that Q is a *finite set of states*, T is a *finite set of transitions*, $\alpha, \beta : T \to Q$ are the *source and target functions*. A *path* π in such a graph, is a word $\pi \in T^+$ such that there exists $n \in \mathbb{N}\backslash\{0\}$ and a sequence $q_0, t_1, q_1, \dots, t_n, q_n$ of $n + 1$ states $q_i \in Q$ and n transitions $t_i \in T$ such that $\pi = t_1 \dots t_n$ and such that $(\alpha(t_i), \beta(t_i)) = (q_{i-1}, q_i)$ for any i. The set of paths is denoted by $\Pi(G)$. Given such a path π, we denote by $\alpha(\pi)$ and $\beta(\pi)$ the states q_0 and q_n. A *loop* θ is a path such that $\alpha(\theta) = \beta(\theta)$.

An automaton $\mathcal{A}$ is a tuple $\mathcal{A} = (G, \Sigma, l, q_0, F)$ such that $G = (Q, T, \alpha, \beta)$ is a graph, Σ is an alphabet, $l : T \to \Sigma$ is a *labelling function*, $q_0 \in Q$ is the *initial state*, and $F \subseteq Q$ is the set of *final states*. As usual, the function l is extended into $l : T^* \to \Sigma^*$ as the unique function such that $l(\pi_1.\pi_2) = l(\pi_1).l(\pi_2)$ for any $\sigma_1, \sigma_2 \in T^*$. Given a state $q \in Q$, we denote by $\mathcal{A}_q$ the automaton obtained from $\mathcal{A}$ by replacing the initial state q_0 by q. The *number of states* of an automaton $\mathcal{A}$ is denoted by $|\mathcal{A}| = |Q|$. The *language* $\mathcal{L}(\mathcal{A})$ *recognized* by an automaton $\mathcal{A}$ is defined as the set of words $\sigma \in \Sigma^*$ such that either $\sigma = \epsilon$ and $q_0 \in F$, or there exists $\pi\Pi(G)$ such that $\alpha(\pi) = q_0$, $\beta(\pi) \in F$ and $\sigma = l(\pi)$. An automaton is said *deterministic and complete* if there exists a function $\delta : Q \times \Sigma \to Q$ such that for any $(q, b, q') \in Q \times \Sigma \times Q$, we have $q' = \delta(q, b)$ if and only if there exists $t \in T$ such that $(\alpha(t), l(t), \beta(t)) = (q, b, q')$. In this case, δ is unique and such an automaton is also denoted by $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. Two automata $\mathcal{A}_1 = ((Q_1, T_1, \alpha_1, \beta_1), \Sigma, l_1, q_{0,1}, F_1)$ and $\mathcal{A}_2 = ((Q_2, T_2, \alpha_2, \beta_2), \Sigma, l_2, q_{0,2}, F_2)$ are said *isomorph* by a one-to-one relation $\mathcal{R} \subseteq Q_1 \times Q_2$, if and only if (1) there exists $t_1 \in T_1$ such that $(\alpha_1(t_1), l_1(t_1), \beta_1(t_1)) = (q_1, b, q'_1)$ if and only if there exists $t_2 \in T_2$ such that $(\alpha_2(t_2), l_2(t_2), \beta_2(t_2)) = (q_2, b, q'_2)$, for any $(q_1, q_2) \in \mathcal{R}$, $(q'_1, q'_2) \in \mathcal{R}$ and $b \in \Sigma$, and (2) we have $(q_{0,1}, q_{0,2}) \in \mathcal{R}$, and (3) we have $q_1 \in F_1$ if and only if $q_2 \in F_2$ for any $(q_1, q_2) \in \mathcal{R}$. Two such automata $\mathcal{A}_1$ and $\mathcal{A}_2$, are denoted by $\mathcal{A}_1 = \mathcal{A}_2$.

The set X^m is called the set of vectors with $m \in \mathbb{N}$ components in a set X. Given an integer $i \in \{1, \dots, m\}$ and a vector $x \in X^m$, the i-th component of x is written $x[i] \in X$. Vectors $x + y$ and $t.x$ are defined by $(x + y)[i] = (x[i]) + (y[i])$ and $(t.x)[i] = t.(x[i])$ for any $i \in \{1, \dots, m\}$, $x, y \in \mathbb{Q}^m$, $t \in \mathbb{Q}$. The set of *square matrices* of dimension $m \in \mathbb{N}$ with coefficient in $\mathbb{Q}$ is denoted by $\mathcal{M}_m(\mathbb{Q})$. The *product* of two square matrices M_1 and M_2 in $\mathcal{M}_m(\mathbb{Q})$ is denoted by $M_1.M_2$ and as usual given a square matrix $M \in \mathcal{M}_m(\mathbb{Q})$, the sequence $(M^n)_{n \in \mathbb{N}}$ is uniquely defined by M^0 is the *identity matrix*, $M^1 = M$ and $M^{n_1+n_2} = M^{n_1}.M^{n_2}$ for any $n_1, n_2 \in \mathbb{N}$.

5 Flatenning and Acceleration

5.1 Systems

A *system* is a tuple $\mathcal{S} = (G, S, (\mathcal{R}_t)_{t \in T})$, where $G = (Q, T, \alpha, \beta)$ is a graph, S a (finite or infinite) set, and $(\mathcal{R}_t)_{t \in T}$ is a sequence of binary relations over S. A *configuration* is an element of $Q \times S$. The sequence $(\mathcal{R}_t)_{t \in T}$ is extended into the unique sequence $(\mathcal{R}_\pi)_{\pi \in \Pi(G)}$ satisfying $\mathcal{R}_{\pi_1.\pi_2} = \mathcal{R}_{\pi_1}.\mathcal{R}_{\pi_2}$ for any $\pi_1, \pi_2 \in \Pi(G)$ such that $\beta(\pi_1) = \alpha(\pi_2)$. The *reachability relation along a path* $\pi \in \Pi(G)$ is the binary relation $\xrightarrow{\pi}$ over the set of configurations defined by $(q, s) \xrightarrow{\pi} (q', s')$ if and only if $(\alpha(\pi), \beta(\pi)) = (q, q')$ and $s \mathcal{R}_\pi s'$. We also denote by $\xrightarrow{\epsilon}$ the identity binary relation over the set of configurations. Given a subset $\rho \subseteq \Pi(G) \cup \{\epsilon\}$, we also denote by $\xrightarrow{\rho}$ the binary relation $\xrightarrow{\rho} = \bigcup_{\pi \in \rho} \xrightarrow{\pi}$. The *global reachability relation* is the binary relation $\xrightarrow{\Pi(G) \cup \{\epsilon\}}$.

An *initialized system* is a tuple $(\mathcal{S}, q_0, X_0)$ such that $\mathcal{S} = (G, S, (\mathcal{R}_t)_{t \in T})$ is a system, $q_0 \in Q_0$ and $X_0 \subseteq S$. The *reachability set of an initialized system* is the following subset:

$$\mathrm{Post}_{\xrightarrow{\Pi(G) \cup \{\epsilon\}}}(\{q_0\} \times X_0)$$

5.2 Flatenning

In order to compute the reachability set of an initialized system, in [BFLS05] and [LS05], algorithms based on *flatenning* have been proposed. Flatenning idea (see bellow the definition) is based on the fact that in practice $\mathrm{Post}^*_{\mathcal{R}_\theta}(X)$ is effectively computable for any loop θ and some subsets $X \subseteq S$.

A *Linear Regular Expression (LRE)* [LS05] ρ of a graph G, is a subset of $\Pi(G) \cup \{\epsilon\}$ such that there exists $n \in \mathbb{N} \backslash \{0\}$, a sequence $\pi_0, \theta_1, \pi_1, ..., \theta_n, \pi_n$ of $n+1$ paths π_i and n loops θ_i such that $\rho = \pi_0.\theta_1^*.\pi_1 \ldots \pi_{n-1}.\theta_n^*.\pi_n$. A *Semi Linear Regular Expression (SLRE)* ρ is a finite union of LRE.

A *flattening ρ of an initialized system* $(\mathcal{S}, q_0, X_0)$, is a SLRE ρ such that:

$$\mathrm{Post}_{\xrightarrow{\Pi(G) \cup \{\epsilon\}}}(\{q_0\} \times X_0) = \mathrm{Post}_{\xrightarrow{\rho}}(\{q_0\} \times X_0)$$

Remark 5.1. A *flattening ρ of a system* $\mathcal{S}$, is a SLRE ρ such that $\xrightarrow{\Pi(G) \cup \{\epsilon\}} = \xrightarrow{\rho}$. Naturally, a flattening ρ of a system $\mathcal{S}$ is also a flatenning of $(\mathcal{S}, q_0, X_0)$ for any $q_0 \in Q$ and $X_0 \subseteq S$.

Intuitively, algorithms for computing the reachability set of an initialized system thanks to a flatenning are based on the choice of a *symbolic representation* [BFLS05] for some subsets of S such that (1) the inclusion $X_1 \subseteq X_2$ of two symbolically represented subsets $X_1, X_2 \subseteq S$ is decidable, and (2) a symbolic representation of $X_1 \cup X_2$ is effectively computable from a symbolic representation of X_1 and X_2, and (3) the set X_0 can be symbolically represented, and (4) a symbolic representation of $\mathrm{Post}_{\mathcal{R}_t}(X)$ is effectively computable from any symbolically represented subset $X \subseteq S$ and for any transition $t \in T$, and (5) a symbolic representation of $\mathrm{Post}^*_{\mathcal{R}_\theta}(X)$ is effectively computable for any loop θ and for any symbolically represented subset $X \subseteq S$.

If a symbolic representation satisfies (1),(2),(3),(4), and (5), in [BFLS05,LS05], we have proved that if there exists a flatenning ρ of the initialized system $(\mathcal{S}, q_0, X_0)$, then a symbolic representation of the reachability set of this initialized system, is effectively computable.

In practice, the most difficult and technical part of this approach is to provide an algorithm performing (5).

5.3 Acceleration

Sébastien Bardin and Alain Finkel shown in [BF04] that up to our knowledge, all the algorithms computing $\mathrm{Post}^*_{\mathcal{R}}(X)$ where $\mathcal{R}$ is a binary relation over a set S and $X \subseteq S$, computes as an intermediate step, the set $\mathrm{acc}(\mathcal{R}, X) = \bigcup_{k\in\mathbb{N}}(\mathrm{Post}^k_{\mathcal{R}}(X) \times \{k\})$. In this article, this set $\mathrm{acc}(\mathcal{R}, X)$ is called the *acceleration* of $(\mathcal{R}, X)$.

Acceleration and fix-point computation are linked because (1) by projecting away the second component of $\mathrm{acc}(\mathcal{R}, X)$, we obtain the fix-point $\mathrm{Post}^*_{\mathcal{R}}(X)$, and (2) the acceleration of $(\mathcal{R}, X)$ is equal to $\mathrm{Post}^*_{[\mathcal{R}]}(X \times \{0\})$ where $[\mathcal{R}]$ is the binary relation over $S \times \mathbb{N}$ defined by $(s, k)[\mathcal{R}](s', k')$ if and only if $s\mathcal{R}s'$ and $k' = k + 1$.

However, whereas there does not exist a "simple (from a decidability point of view) property" in order to check if a set $Z \subseteq S$ is equal to $\mathrm{Post}^*_{\mathcal{R}}(X)$ (we can just check that Z is a fix-point $\mathrm{Post}_{\mathcal{R}}(Z) \subseteq Z$ that contains $X \subseteq Z$), the following lemma shows that we can easily check if a set $Z \subseteq S \times \mathbb{N}$ is equal to $\mathrm{acc}(\mathcal{R}, X)$.

Lemma 5.2. *Given a couple $(\mathcal{R}, X)$ where $\mathcal{R}$ is a binary relation over a set S and X is a subset of S, a subset $Z \subseteq S \times \mathbb{N}$ is equal to acc$(\mathcal{R}, X)$ if and only if $Z = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$.*

Proof. Naturally, the set $Z = \mathrm{acc}(\mathcal{R}, X)$ satisfies $Z = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$. For the converse, let us consider a subset $Z \subseteq S \times \mathbb{N}$ such that $Z = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$ and let us prove by induction over $k \in \mathbb{N}$ that $Z \cap (S \times \{k\}) = \mathrm{Post}^k_{\mathcal{R}}(X) \times \{k\}$. As $Z = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$, we get $Z \cap (S \times \{0\}) = X \times \{0\} = \mathrm{Post}^0_{\mathcal{R}}(X) \times \{0\}$. Assume the induction true for an integer $k \in \mathbb{N}$. From $Z = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$, we get $Z \cap (S \times \{k + 1\}) = \mathrm{Post}_{[\mathcal{R}]}(Z) \cap (S \times \{k + 1\}) = \mathrm{Post}_{[\mathcal{R}]}(Z \cap (S \times \{k\}))$. Induction hypothesis proves that $Z \cap (S \times \{k\}) = \mathrm{Post}^k_{\mathcal{R}}(X) \times \{k\}$. Therefore $\mathrm{Post}_{[\mathcal{R}]}(Z \cap (S \times \{k\})) = \mathrm{Post}^{k+1}_{\mathcal{R}}(X) \times \{k + 1\}$ and we have proved the induction. From $Z = \bigcup_{k\in\mathbb{N}}(Z \cap (S \times \{k\}))$, we get $Z = \mathrm{acc}(\mathcal{R}, X)$. □

The previous lemma proves that if $\mathrm{acc}(\mathcal{R}, X)$ can be represented by a symbolic representation, such that (1) the inclusion $Z_1 \subseteq Z_2$ of two symbolically represented subsets $Z_1, Z_2 \subseteq S \times \mathbb{N}$ is decidable, and (2) a symbolic representation of $\mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$ is effectively computable from any symbolically represented subset $Z \subseteq S \times \mathbb{N}$, then a symbolic representation of $\mathrm{acc}(\mathcal{R}, X)$ is effectively computable just by enumerating symbolic representations of subsets of $S \times \mathbb{N}$ and check if $Z = \mathrm{acc}(\mathcal{R}, X)$ thanks to lemma 5.2 .

In practice, such an enumerative algorithm is not possible. Efficient algorithms performing acceleration computation are based on three points (1) the inclusion $Z_1 \subseteq Z_2$ of two symbolically represented subsets $Z_1, Z_2 \subseteq S \times \mathbb{N}$ is decidable, and (2) a symbolic representation of $\mathrm{Post}_{\mathcal{R}}^k(X) \times \{k\}$ is computable for any bounded integer $k \in \mathbb{N}$, and (3) a symbolic representation of the acceleration of $(\mathcal{R}, X)$ can be *extrapolated* from the $k \in \mathbb{N}$ symbolic representations of the sets $\mathrm{Post}_{\mathcal{R}}^0(X)$, ..., $\mathrm{Post}_{\mathcal{R}}^{k-1}(X)$, and (4) the correctness of the extrapolation is decided thanks to lemma 5.2

Remark 5.3. The tool TREX [Tre] implements the previous points (1), (2), (3) and (4) for the *Parametric Difference Bound Matrices (PDBM)* [AAB00,ABLS01].

In the following section, we present the class of r-definable subsets of integers vectors, that will be used as a symbolic representation in the sequel.

Remark 5.4. The *acceleration* $\mathrm{acc}(\mathcal{R})$ of a binary relation $\mathcal{R}$ over a set S, is the subset of $(S \times S) \times \mathbb{N}$ defined by $\mathrm{acc}(\mathcal{R}) = \bigcup_{k\in\mathbb{N}}(\mathcal{R}^k \times \{k\})$. Naturally, the acceleration of a relation can be reduced to the acceleration of a particular couple $(\mathcal{R}_2, I_2)$. In fact, let us consider the binary relation $\mathcal{R}_2$ defined over $S \times S$ by $(s_1, s_2)\mathcal{R}_2(s_1', s_2')$ if and only if $s_1 = s_1'$ and $s_2\mathcal{R}s_2'$, and let us consider the subset I_2 of $S \times S$ defined by $I_2 = \{(s, s);\ s \in S\}$. Now just remark that $\mathrm{acc}(\mathcal{R}) = \mathrm{acc}(\mathcal{R}_2, I_2)$.

6 Number Decision Diagrams

Sets of integer vectors that can be represented by automata called *Number decision Diagrams (NDD)* are related to the notion of *r-definability* [BHMV94] where $r \in \mathbb{N}\backslash\{0, 1\}$ is an integer called *the basis of decomposition*. In this section, we recall the definition of r-definable sets, NDD, and recall the characterization of the minimal NDD.

Given an integer $r \in \mathbb{N}\backslash\{0, 1\}$ and a dimension $m \in \mathbb{N}\backslash\{0\}$, a subset $X \subseteq \mathbb{N}^m$, is said *r-definable* if it is definable in the first order logic FO $(\mathbb{N}, +, \leq, V_r)$ where $V_r : \mathbb{N} \rightarrow \mathbb{N}$ is the *r-valuation function* defined by $V_r(0) = 1$ and $V_r(x)$ is the greatest power of r that divides $x \in \mathbb{N}\backslash\{0\}$.

A subset $X \subseteq \mathbb{N}^m$ is said *Presburger-definable* if X is definable in the first order logic FO $(\mathbb{N}, +, \leq)$. Remark that if X is Presburger-definable then X is r-definable in any basis of decomposition $r \in \mathbb{N}\backslash\{0, 1\}$. Moreover, recall [BHMV94] that given two basis of decomposition $r_1, r_2 \in \mathbb{N}\backslash\{0, 1\}$ such that $r_1^* \cap r_2^* = \emptyset$ where $r^* = \{r^n;\ n \in \mathbb{N}\}$, then a set $X \subseteq \mathbb{N}^m$ both r_1-definable and r_2-definable is in fact Presburger-definable.

Recall that the first order logic FO $(\mathbb{N}, +, \leq, V_r)$ is *decidable*. The proof of this well known result is based on the decomposition of an integer vector into a *word of digit vectors* $\Sigma_{r,m} = \{0, \ldots, r-1\}^m$. A word $\sigma \in \Sigma_{r,m}^*$ is called a *least significant digit first decomposition* of an integer vector $x \in \mathbb{N}^m$, if $\rho_{r,m}(\sigma) = x$ where $\rho_{r,m} : \Sigma_{r,m}^* \rightarrow \mathbb{N}^m$ is defined by $\rho_{r,m}(\sigma) = \sum_{i=1}^{|\sigma|} r^{i-1}.\sigma[i]$.

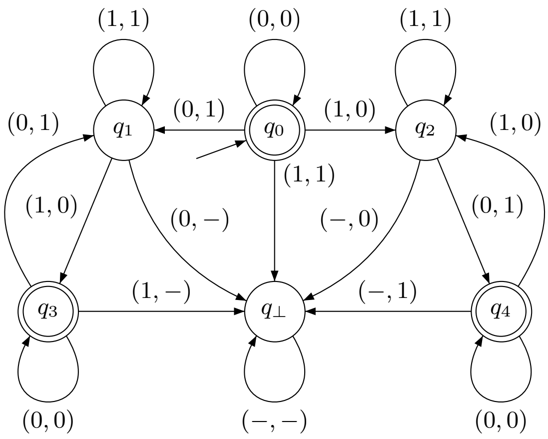


Fig. 5. A NDD representing in basis of decomposition 2 the set $\{x \in \mathbb{N}^2;\ (x[1] = 2.x[2]) \vee (x[2] = 2.x[1])\}$ where $- \in \{0, 1\}$

Definition 6.1 ([WB00,BC96]). *A* Number Decision Diagram (NDD) $\mathcal{A}$ *in basis of decomposition* $r \in \mathbb{N}\backslash\{0, 1\}$ *and dimension* $m \in \mathbb{N}\backslash\{0\}$*, is a deterministic and complete automaton* $\mathcal{A} = (Q, \Sigma_{r,m}, \delta, q_0, F)$ *such that for any transition* $q \xrightarrow{0} q'$ *we have* $q \in F$ *if and only if* $q' \in F$.

The set $X = \rho_{r,m}(\mathcal{L}(\mathcal{A}))$ is called the set represented by $\mathcal{A}$ (see figure [**?**] for an example of NDD). Recall that a set $X \subseteq \mathbb{N}^m$ can be represented by a NDD in basis of decomposition r if and only if it is r-definable [BHMV94].

When a set $X \subseteq \mathbb{N}^m$ is implicitly represented by a NDD $\mathcal{A}$, we denote by X_q the set represented by the NDD $\mathcal{A}_q$ obtained by replacing the initial state q_0 by an other state $q \in Q$. The set $X_{q'}$ can be easily characterized in function of X_q for any path $q \xrightarrow{\sigma} q'$ by introducing [Ler04] the unique sequence $(\gamma_{r,m,\sigma})_{\sigma \in \Sigma_{r,m}^*}$ of functions $\gamma_{r,m,\sigma} : \mathbb{N}^m \rightarrow \mathbb{N}^m$ that satisfies $\gamma_{r,m,\sigma_1.\sigma_2} = \gamma_{r,m,\sigma_1} \circ \gamma_{r,m,\sigma_2}$ for any word $\sigma_1, \sigma_2 \in \Sigma_{r,m}^*$, and such that $\gamma_{r,m,b}(x) = r.x + b$ for any $(b, x) \in \Sigma_{r,m} \times \mathbb{N}^m$.

Proposition 6.2 ([FL04]). *Let* $\mathcal{A}$ *be a NDD that represents a set* $X \subseteq \mathbb{N}^m$ *in basis of decomposition* r*. We have* $X_{q'} = \gamma_{r,m,\sigma}^{-1}(X_q)$ *for any path* $q \xrightarrow{\sigma} q'$.

Example 6.3. Let us consider the NDD $\mathcal{A}$ (figure 5) that represents the set $X = \{x \in \mathbb{N}^2;\ (x[1] = 2.x[2]) \vee (x[2] = 2.x[1])\}$ in basis of decomposition 2. We have $X_{q_0} = X$, $X_{q_\perp} = \emptyset$, $X_{q_1} = \{x \in \mathbb{N}^2; x[1] = 2.x[2] + 1\}$, $X_{q_2} = \{x \in \mathbb{N}^2; x[2] = 2.x[1] + 1\}$, $X_{q_3} = \{x \in \mathbb{N}^2; x[1] = 2.x[2]\}$, and $X_{q_4} = \{x \in \mathbb{N}^2; x[2] = 2.x[1]\}$.

Remark that if a set $X \subseteq \mathbb{N}^m$ is r-definable, then it can be represented by at least one NDD in basis of decomposition r and from the previous proposition 6.2, we deduce that the set $Q_{r,m}(X) = \{\gamma_{r,m,\sigma}^{-1}(X);\ \sigma \in \Sigma_{r,m}^*\}$ is finite. Thanks to this set, we can easily characterize the unique (up to isomorphism) minimal (for the number of states) NDD in basis of decomposition r that represents a r-definable set $X \subseteq \mathbb{N}^m$. We denote by $\mathcal{A}_{r,m}(X) = (Q_{r,m}(X), \Sigma_{r,m}, \delta_{r,m}, X, F_{r,m})$ the deterministic and complete automaton defined by the transition function $\delta_{r,m}(Y,b) = \gamma_{r,m,b}^{-1}(Y)$ for any $(Y,b) \in Q_{r,m}(X) \times \Sigma_{r,m}$, and defined by the set of final states $F_{r,m} = \{Y \in Q_{r,m}(X);\ 0 \in Y\}$. Remark that $X \in Q_{r,m}(X)$ is the initial state of $\mathcal{A}_{r,m}(X)$.

Proposition 6.4 ([FL04]). $\mathcal{A}_{r,m}(X)$ *is the unique (up to isomorphism) minimal (for the number of states) NDD that represents in basis of decomposition r, a r-definable set $X \subseteq \mathbb{N}^m$.*

7 Regular acceleration

The following theorem shows that it is possible to compute the acceleration of a couple as soon as it is r-definable.

Theorem 7.1. *Let $(\mathcal{R}, X)$ be a r-definable couple such that its acceleration acc$(\mathcal{R}, X)$ is r-definable. We can effectively compute a NDD representing in basis of decomposition r the acceleration acc$(\mathcal{R}, X)$.*

Proof. In fact just enumerate the NDD $\mathcal{A}$ in basis of decomposition r that represent sets $Z \subseteq \mathbb{N}^m \times \mathbb{N}$ and remark that a NDD $\mathcal{A}'$ representing $Z' = \mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$ in basis of decomposition r is effectively computable. As we can decide if $\mathcal{A}$ and $\mathcal{A}'$ represents the same set, lemma 5.2, shows that we can effectively computes a NDD $\mathcal{A}$ representing in basis of decomposition r the set $\mathrm{acc}(\mathcal{R}, X)$. $\square$

However, as explained in section 5, the proof of the previous theorem does not provide an efficient way for computing the NDD representing $\mathrm{acc}(\mathcal{R}, X)$.

From [FL04,WB00,BB03] efficient algorithms for computing the NDD representing in basis of decomposition r the set $\mathrm{Post}_{[\mathcal{R}]}(Z) \cup (X \times \{0\})$, from a NDD representing in basis of decomposition r, a set $Z \subseteq \mathbb{N}^m \times \mathbb{N}$, can be extracted. That means the set Y^{k_0} defined by $Y^0 = \emptyset$ and $Y^{k+1} = \mathrm{Post}_{[\mathcal{R}]}(Y^k) \cup (X \times \{0\})$ can be computed inductivelly. Now just remark that $Y^{k_0} = \bigcup_{0 \leq k < k_0} (\mathrm{Post}_{\mathcal{R}}^k(X) \times \{k\})$.

Our *extrapolation* comes from [BHV04], and it is obtained by merging some "interesting states" of the minimal NDD representing in basis of decomposition r the set Y^{k_0}, from which the recognized languages are "similar", in order to obtain the minimal NDD representing in basis of decomposition r the set $Y^\infty = \mathrm{acc}(\mathcal{R}, X)$.

Let us consider two integers $d, n \in \mathbb{N}$, and a deterministic and complete automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. The "interesting set of states" Q_d of $\mathcal{A}$ is defined by $Q_d = \delta(q_0, \Sigma^{\leq d})$. Given an interesting state $q \in Q_d$, we denote by $[q]_{d,n}$ the *equivalence class* of interesting states $q' \in Q_d$ that "recognize similar languages": $\mathcal{L}(\mathcal{A}_q) \cap \Sigma^{\leq n} = \mathcal{L}(\mathcal{A}_{q'}) \cap \Sigma^{\leq n}$. Finally, we denote by $[\mathcal{A}]_{d,n}$ the following automaton

$$[\mathcal{A}]_{d,n} = ((Q_{d,n}, T_{d,n}, \alpha_{d,n}, \beta_{d,n}), \Sigma, l_{d,n}, [q_0]_{d,n}, F_{d,n})$$

where the set of states is defined by $Q_{d,n} = \{[q]_{d,n};\ q \in Q_d\}$, the set of transitions $T_{d,n} = \{(q, b, q') \in Q_{d-1} \times \Sigma \times Q_d;\ q' = \delta(q, b)\}$, the source and target functions by $\alpha_{d,n}(q, b, q') = [q]_{d,n}$ and $\beta_{d,n}(q, b, q') = [q']_{d,n}$, the labelling function $l((q, b, q')) = b$, the set of final states $F_{d,n} = \{[q]_{d,n};\ q \in Q_d \cap F\}$.

The following theorem is the main contribution of this paper.

Theorem 7.2 (proof in appendix). *Let us consider a sequence $(X_k)_{k\in\mathbb{N}}$ of subsets of $\mathbb{N}^m$, and let us consider $Y^{k_0} = \bigcup_{0\leq k<k_0}(X_k \times \{k\})$ for any $k_0 \in \mathbb{N}\cup\{\infty\}$. Assume that Y^∞ is r-definable. Then Y^{k_0} is r-definable and its minimal NDD denoted by $\mathcal{A}^{k_0}$ in basis of decomposition r, is well defined for any $k_0 \in \mathbb{N}\cup\{\infty\}$. Moreover, there exists $d, n, k_0 \in \mathbb{N}$ such that:*

$$[\mathcal{A}^{k_0}]_{d,n} = \mathcal{A}^\infty$$

Finally, the previous equality is in fact obtained for any d, n, k_0 such that $d \geq |\mathcal{A}^\infty|$, $n \geq |\mathcal{A}^\infty|^2 - 1$, and $k_0 \geq r^{d+n}$.

Corollary 7.3. *Let $(\mathcal{R}, X)$ be a r-definable couple such that its acceleration is r-definable and let d_∞ be the number of states of a NDD representing acc$(\mathcal{R}, X)$ in basis of decomposition r. The following automaton is the minimal NDD representing in basis of decomposition r the set acc$(\mathcal{R}, X)$ for any $d, n, k_0 \in \mathbb{N}$ such that $d \geq d_\infty$, $n \geq d_\infty^2 - 1$ and $k_0 \geq r^{d+n}$:*

$$[\mathcal{A}_{r,m}(\bigcup_{0\leq k<k_0}(\mathrm{Post}^k_{\mathcal{R}}(X)\times\{k\}))]_{d,n}$$

Remark that the previous corollary can be applied for integers k_0 exponentially larger than the naturally unknown size of $\mathcal{A}^\infty$. Hence, it seems that the algorithm corresponding to the previous theorem is as inefficient as the one that just enumerates NDD in basis of decomposition r representing some subsets $Z \subseteq \mathbb{N}^m \times \mathbb{N}$ and checks if $Z = \mathrm{acc}(\mathcal{R}, X)$ thanks to lemma 5.2. However, from a practical point of view, this condition is not a true limitation because (1) the exponential size of k_0 is reached in the worst case and we can hope that in practice the couple $(\mathcal{R}, X)$ is enough regular to extrapolate $\mathcal{A}^\infty$ from $\mathcal{A}^{k_0}$ for smaller values of k_0 and (2) The set Y^{2^n+1} can be computed in n steps by remarking that $Y^{2^n+1} = \mathrm{Post}_{[\mathcal{R}]^{\leq 2^n}}(X\times\{0\})$ where $([\mathcal{R}]^{\leq 2^n})_{n\in\mathbb{N}}$ is given by the $[\mathcal{R}]^{\leq 2^0} = I_{\mathbb{N}^{2.m+1}} \cup [\mathcal{R}]$ and the induction $[\mathcal{R}]^{\leq 2^{n+1}} = [\mathcal{R}]^{\leq 2^n}.[\mathcal{R}]^{\leq 2^n}$.

Remark 7.4. The previous induction explains why in [BLW03], authors compute inductively Y^{2^n+1} rather than Y^k.

Finally, let us remark that even if we have provided algorithms for computing r-definable acceleration, deciding if a r-definable couple own this property is undecidable.

Proposition 7.5 (proof in appendix). *We cannot decide if the acceleration of a r-definable couple $(\mathcal{R}, X)$ is r-definable.*

8 Examples

A m-dim *counter system [FL02]* $\mathcal{S}$ is a system $\mathcal{S} = (G, S, (\mathcal{R}_t)_{t\in T})$ such that $S = \mathbb{N}^m$ and such that $\mathcal{R}_t$ is Presburger-definable.

8.1 Presburger-definable linear functions with a finite monoid

In our symbolic model-checker FAST [Fas], the exact computation of the *reachability sets of a counter system* is based on a flattening algorithm (see section 5.2) and the computation of f^* where f is a *Presburger-definable linear functions f with a finite monoid*: that means a function $f : D \rightarrow \mathbb{N}^m$ where $D \subseteq \mathbb{N}^m$ is Presburger-definable and such that there exists $(M, v) \in \mathcal{M}_m(\mathbb{Q}) \times \mathbb{Q}^m$ such that the monoid $M^* = \{M^i;\ i \in \mathbb{N}\}$ is finite and $f(x) = M.x + v$ for any $x \in D$. By remarking that $[f]$ is a Presburger-definable linear function over $\mathbb{N}^{m+1}$ with a finite monoid, we deduce from [FL02] that $[f]^*$ is Presburger-definable. In particular $\mathrm{acc}(f)$ is Presburger-definable and can be computed by the corollary 7.3.

8.2 Examples from [BLW03]

In [BLW03], authors presented a method for computing an over-approximation of the NDD in basis of decomposition r representing $\mathcal{R}^*$ from a NDD in basis of decomposition r representing a binary relation $\mathcal{R}$. All the binary relations considered expect one, are in fact some Presburger-definable linear functions with a finite monoid (studied in the previous section). The other binary relation $\mathcal{R}$ is defined over $\mathbb{N}^2$ by $x\mathcal{R}x'$ if and only if $(x[1]' = x[1] + 1 \wedge x[2]' = x[2] + 1) \vee (x[1]' = x[1] + 2 \wedge x[2]' = x[2] - 1) \vee (x[1]' = x[1] - 1 \wedge x[2]' = [2] + 2)$. In order to prove that the acceleration $\mathrm{acc}(\mathcal{R}) \subseteq \mathbb{N}^5$ is Presburger-definable, we have used our symbolic model-checker FAST that computes (the file used for this computation is provided in appendix) in few seconds a NDD in basis of decomposition 2 representing $\mathrm{acc}(\mathcal{R})$ (it has 572 states). Recall that FAST only computes Presburger-definable sets. In particular $\mathrm{acc}(\mathcal{R})$ is Presburger-definable. Recall also that FAST and its documentation are freely available on the FAST web-page [Fas].

8.3 Example from [LS05]

Consider the 2-dim *counter system* $\mathcal{S} = ((Q, T, \alpha, \beta), \mathbb{N}^2, (f_t)_{t \in T})$ depicted in figure 6. Intuitively, the loop t_1 on *state* q_1 transfers the contents of the first counter into the second counter, while the loop t_2 on *state* q_2 does the converse. *Intermediate configurations* along $(q_1, (1, 2)) \xrightarrow{t_1 t_3 t_2^4 t_4 t_1} (q_1, (4, 1))$ are also depicted in this figure.

More formally, we have $Q = \{q_1, q_2\}, T = \{t_1, t_2, t_3, t_4\}, (\alpha(t_1), \alpha(t_2), \alpha(t_3), \alpha(t_4)) = (q_1, q_2, q_1, q_2), (\beta(t_1), \beta(t_2), \beta(t_3), \beta(t_4)) = (q_2, q_1, q_2, q_1)$, and the functions $f_{t_1}, f_{t_2}, f_{t_3}, f_{t_4}$ are defined by $f_{t_1}(x) = x + (-1, 1)$ if $x[1] \geq 1$, $f_{t_2}(x) = x + (1, -1)$ if $x[2] \geq 1$, $f_{t_3}(x) = x + (0, 1)$ if $x[1] = 0$, and $f_{t_4}(x) = x + (1, 0)$ if $x[2] = 0$.

This counter system exhibits a simple reachability relation, since it is readily seen that $(q_1, x)\mathcal{R}(q_1, x')$ if and only if: $(x[1]' + x'[2]) - (x[1] + x[2])$ is even, and $x[1]' + x[2]' = x[1] + x[2]$ implies $x[1]' \leq x[1]$. Similarly $(q_i, x)\mathcal{R}(q_j, x')$ can be described for any $i, j \in \{1, 2\}$. Than means the reachability relation $\mathcal{R}$ is Presburger-definable. Recall that this reachability relation cannot be computed using flatenning [LS05], because they does not exists a flatenning of $\mathcal{S}$. In particular FAST cannot compute automatically this relation.

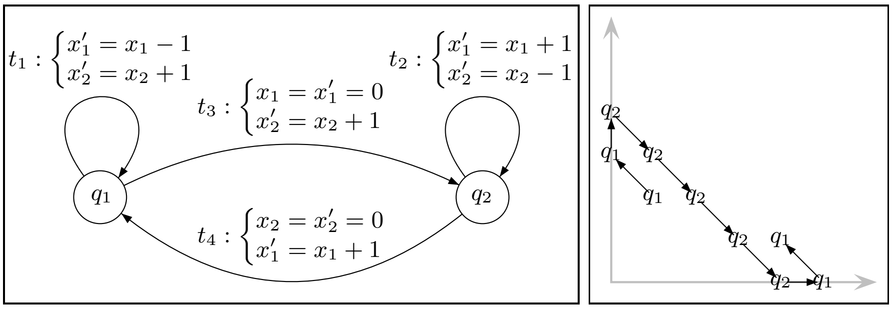


Fig. 6. A counter automaton without any flatenning.

However, we can prove that $\mathcal{R}$ can be automatically computed thanks to some accelerations. Remark that f_{t_1} and f_{t_2} are Presburger-definable linear functions with a finite monoid. Hence $\text{acc}(f_{t_1})$ and $\text{acc}(f_{t_2})$ are Presburger-definable and hence they can be computed automatically. We deduce automatically the binary relations $f_{t_1}^*$ and $f_{t_2}^*$ by projecting away the last component of $\text{acc}(f_{t_1})$ and $\text{acc}(f_{t_2})$. We deduce the binary relations $\xrightarrow{t_1^*}$ and $\xrightarrow{t_2^*}$. Next, remark that $(x, x', k) \in \text{acc}(f_{t_1}^* \circ f_{t_3} \circ f_{t_2}^* \circ f_{t_4})$ if and only if either (1) $k = 0$ and $x = x'$, or (2) $k \geq 1$ and $x'[1]+x'[2] = x[1]+x[2]+2.k$. Therefore $\text{acc}(f_{t_1}^* \circ f_{t_3} \circ f_{t_2}^* \circ f_{t_4})$ is Presburger-definable and hence effectively computable. We get $(f_{t_1}^* \circ f_{t_3} \circ f_{t_2}^* \circ f_{t_4})^*$ by projecting away the last component of the set. We deduce the binary relation $\xrightarrow{(t_1^*.t_3.t_2^*.t_4)^*}$. Symmetrically, we effectively compute $\xrightarrow{(t_2^*.t_4.t_1^*.t_3)^*}$. In particular we can automatically compute the binary relation $\mathcal{R}'$ equal to the union of $\xrightarrow{t_1^*}$, $\xrightarrow{(t_1^*.t_3.t_2^*.t_4)^*}$, $\xrightarrow{(t_2^*.t_4.t_1^*.t_3)^*}$, and $\xrightarrow{t_2^*}$. As $\mathcal{R}'.\mathcal{R}' \subseteq \mathcal{R}'$ (this inclusion can be checked automatically), we deduce that $\mathcal{R}' = \mathcal{R}$ is the reachability relation.

Remark 8.1. We have accelerated nested loops.

References

[AAB00] Aurore Annichini, Eugene Asarin, and Ahmed Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In *Proc. 12th Int. Conf. Computer Aided Verification (CAV'2000), Chicago, IL, USA, July 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 419–434. Springer, 2000.

[ABLS01] Aurore Annichini, Ahmed Bouajjani, Yassine Lakhnech, and Mihaela Sighireanu. Languages, rewriting systems, and verification of infinite-state systems. In *Proc. 8th Int. Symp. Static Analysis (SAS'2001), Paris, France, July 2001*, volume 2126 of *Lecture Notes in Computer Science*, pages 335–355. Springer, 2001.

[BB03] Constantinos Bartzis and Tevfik Bultan. Efficient image computation in infinite state model checking. In *Proc. 15th Int. Conf. Computer Aided Verification (CAV'2003), Boulder, CO, USA, July 2003*, volume 2725 of *Lecture Notes in Computer Science*, pages 249–261. Springer, 2003.

[BC96] Alexandre Boudet and Hubert Comon. Diophantine equations, Presburger arithmetic and finite automata. In *Proc. 21st Int. Coll. on Trees in Algebra and Programming (CAAP'96), Linköping, Sweden, Apr. 1996*, volume 1059 of *Lecture Notes in Computer Science*, pages 30–43. Springer, 1996.

[BF04] Sébastien Bardin and Alain Finkel. Composition of accelerations to verify infinite heterogeneous systems. In *Proc. 2nd Int. Symp. Automated Technology for Verification and Analysis (ATVA'2004), Taipei, Taiwan, Nov. 2004*, volume 3299 of *Lecture Notes in Computer Science*, pages 248–262. Springer, 2004.

[BFL04] Sébastien Bardin, Alain Finkel, and Jérôme Leroux. Faster acceleration of counter automata. In *Proc. 10th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2004) Barcelona, Spain, Mar. 2004*, volume 2988 of *Lecture Notes in Computer Science*, pages 576–590. Springer, 2004.

[BFLP03] Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Laure Petrucci. FAST: Fast Acceleration of Symbolic Transition systems. In *Proc. 15th Int. Conf. Computer Aided Verification (CAV'2003), Boulder, CO, USA, July 2003*, volume 2725 of *Lecture Notes in Computer Science*, pages 118–121. Springer, 2003.

[BFLS05] Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Philippe Schnoebelen. Flat acceleration in symbolic model checking. In *Proc. 3rd Int. Symp. on Automated Technology for Verification and Analysis (ATVA'2005), Taipei, Taiwan, Oct. 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2005.

[BHMV94] Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, 1(2):191–238, March 1994.

[BHV04] A. Bouajjani, P. Habermehl, and T. Vojnar. Abstract regular model checking. In *Proc. 16th Int. Conf. on Computer Aided Verification (CAV'04), Boston, Ma, USA, Jul. 2004.*, volume 3114 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 2004.

[BLW03] Bernard Boigelot, Axel Legay, and Pierre Wolper. Iterating transducers in the large. In *Proc. 15th Int. Conf. Computer Aided Verification (CAV'2003), Boulder, CO, USA, July 2003*, volume 2725 of *Lecture Notes in Computer Science*, pages 223–235. Springer, 2003.

[Boi98] Bernard Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. PhD thesis, Université de Liège, 1998.

[Del00] Gorgio Delzanno. Automatic verification of parameterized cache coherence protocols. In *Proc. 12th Int. Conf. Computer Aided Verification (CAV'2000), Chicago, IL, USA, July 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 53–68. Springer, 2000.

[DJS99] Catherine Dufourd, Petr Jančar, and Philippe Schnoebelen. Boundedness of Reset P/T nets. In *Proc. 26th Int. Coll. Automata, Languages, and Programming (ICALP'99), Prague, Czech Republic, July 1999*, volume 1644 of *Lecture Notes in Computer Science*, pages 301–310. Springer, 1999.

[EN98] E. Allen Emerson and Kedar S. Namjoshi. On model checking for non-deterministic infinite-state systems. In *Proc. 13th IEEE Symp. Logic in Computer Science (LICS'98), Indianapolis, IN, USA, June 1998*, pages 70–80. IEEE Comp. Soc. Press, 1998.

[Fas] FAST homepage. `http://www.lsv.ens-cachan.fr/fast/`.

[FL02] Alain Finkel and Jérôme Leroux. How to compose Presburger-accelerations: Applications to broadcast protocols. In *Proc. 22nd Conf. Found. of Software Technology and Theor. Comp. Sci. (FST&TCS'2002), Kanpur, India, Dec. 2002*, volume 2556 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2002.

[FL04] Alain Finkel and Jérôme Leroux. Image computation in infinite state model checking. In *Proc. 16th Int. Conf. Computer Aided Verification (CAV'2004), Omni Parker House Hotel, Boston, USA, July 2004*, volume 3114 of *Lecture Notes in Computer Science*, pages 361–371. Springer, 2004.

[Ler04] Jérôme Leroux. The affine hull of a binary automaton is computable in polynomial time. In Elsevier Science, editor, *Proc. 5th Int. Workshop on Verification of Infinite State Systems (INFINITY 2003), Marseille, France, Sep. 2003*, volume 98 of *Electronic Notes in Theor. Comp. Sci.*, pages 89–104, 2004.

[LS05] Jérôme Leroux and Grégoire Sutre. Flat counter automata almost everywhere! In *Prococeedings of the 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'2005), Taipei, Taiwan, Oct. 2005*, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005.

[Tre] TREX homepage. `http://www.liafa.jussieu.fr/~sighirea/trex/`.

[WB00] Pierre Wolper and Bernard Boigelot. On the construction of automata from linear arithmetic constraints. In *Proc. 6th Int. Conf. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2000), Berlin, Germany, Mar.-Apr. 2000*, volume 1785 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2000.

A A FAST model

```
model BoigelotLegayWolperCAV03 {
  var  x,xp,y,yp,k;

  states normal;

  transition f1 := {
    from := normal;
    to := normal;
    guard := true;
    action :=
      xp'=xp+1,
      yp'=yp+1,
      k'=k+1;
  };

  transition f2 := {
    from :=normal;
    to := normal;
    guard := yp>=1;
    action :=
      xp'=xp+2,
      yp'=yp-1,
      k'=k+1;
  };

  transition f3   := {
    from := normal;
    to := normal;
    guard := xp>=1;
      action:=
      xp'=xp-1,
      yp'=yp+2,
      k'=k+1;
  };
}

strategy s1 {
  setMaxState(0);
  setMaxAcc(100);
  Region I := {state=normal && x=xp && y=yp && k=0};

  Transitions set1 := {f1};
  Region R1 := post*(I, set1,1);
```

```
  Transitions set23 := {f2,f3};
  Region R2 := post*(R1, set23,2);

  Transitions set123 := {f1,f2,f3};
  Region R3 := post*(R2, set123,1);
}
```

B Proof of theorem 7.2=B.1

Theorem B.1. *Let us consider a sequence $(X_k)_{k\in\mathbb{N}}$ of subsets of $\mathbb{N}^m$, and let us consider $Y^{k_0} = \bigcup_{0\leq k<k_0}(X_k \times \{k\})$ for any $k_0 \in \mathbb{N}\cup\{\infty\}$. Assume that Y^∞ is r-definable. Then Y^{k_0} is r-definable and its minimal NDD denoted by $\mathcal{A}^{k_0}$ in basis of decomposition r, is well defined for any $k_0\in\mathbb{N}\cup\{\infty\}$. Moreover, there exists $d, n, k_0 \in \mathbb{N}$ such that:*

$$[\mathcal{A}^{k_0}]_{d,n} = \mathcal{A}^\infty$$

Finally, the previous equality is in fact obtained for any d, n, k_0 such that $d \geq |\mathcal{A}^\infty|$, $n \geq |\mathcal{A}^\infty|^2 - 1$, and $k_0 \geq r^{d+n}$.

Proof. Given $\lambda \in \mathbb{Q}\cup\{\infty\}$, let us consider the Presburger-definable set $T_\lambda \subseteq \mathbb{N}^m\times\mathbb{N}$ defined by $T_\lambda = \bigcup_{0\leq k<\lambda}(\mathbb{N}^m\times\{k\})$.

As $Y^{k_0} = Y^\infty\cap T_{k_0}$, we deduce that Y^{k_0} is r-definable for any $k_0\in\mathbb{N}\cup\{\infty\}$.

Components of the minimal NDD $\mathcal{A}^{k_0}$ representing in basis of decomposition r the set Y^{k_0}, are denoted by $(Q^{k_0}, \Sigma_{r,m+1}, \delta^{k_0}, q_0^{k_0}, F^{k_0})$. Naturally, the set $Y_q^{k_0}$ is well defined for any state $q\in Q^{k_0}$.

For any word $\sigma\in\Sigma_{r,m+1}^*$, we denote by $(\sigma', k_\sigma)\in\Sigma_{r,m}^*\times\mathbb{N}$ the unique couple such that $|\sigma'| = |\sigma|$ and $\rho_{r,m+1}(\sigma) = (\rho_{r,m}(\sigma'), k_\sigma)$.

Remark that for any word $\sigma\in\Sigma_{r,m+1}^*$ and for any $k_0\in\mathbb{N}\cup\{\infty\}$, we have the following equality:

$$Y^{k_0}_{\delta^{k_0}(q_0^{k_0},\sigma)} = \bigcup_{0\leq k<\frac{k_0-k_\sigma}{r^{|\sigma|}}}(\gamma^{-1}_{r,m,\sigma'}(X_{r^{|\sigma|}.k+k_\sigma})\times\{k\}) \tag{1}$$

Let us define $Z^{k_0}_{\sigma_1,\sigma_2}$ and $\lambda_{\sigma_1,\sigma_2}$ for any $\sigma_1,\sigma_2\in\Sigma^*_{r,m+1}$ and any $k_0\in\mathbb{N}\cup\{\infty\}$ by the following equality:

$$\begin{cases} Z^{k_0}_{\sigma_1,\sigma_2} = Y^{k_0}_{\delta^{k_0}(q_0^{k_0},\sigma_1)}\Delta Y^{k_0}_{\delta^{k_0}(q_0^{k_0},\sigma_2)} \\ \lambda_{\sigma_1,\sigma_2} = \min_{i\in\{1,2\}}(\frac{k_0-k_{\sigma_i}}{r^{|\sigma_i|}}) \end{cases}$$

From the equality (1), we deduce:

$$Z^{k_0}_{\sigma_1,\sigma_2}\cap T_{\lambda_{\sigma_1,\sigma_2}} = \bigcup_{0\leq k<\lambda_{\sigma_1,\sigma_2}}((\gamma^{-1}_{r,m,\sigma_1'}(X_{r^{|\sigma_1|}.k+k_{\sigma_1}})\Delta\gamma^{-1}_{r,m,\sigma_2'}(X_{r^{|\sigma_2|}.k+k_{\sigma_2}}))\times\{k\})$$

In particular, we have proved the following equality:

$$Z^{k_0}_{\sigma_1,\sigma_2}\cap T_{\lambda_{\sigma_1,\sigma_2}} = Z^\infty_{\sigma_1,\sigma_2}\cap T_{\lambda_{\sigma_1,\sigma_2}} \tag{2}$$

Let us consider $d, n, k_0\in\mathbb{N}$ such that $d\geq|\mathcal{A}^\infty|$, $n\geq|\mathcal{A}^\infty|^2-1$, and $k_0\geq r^{d+n}$.

Let us show that $\lambda_{\sigma_1,\sigma_2} > r^n - 1$ for any $\sigma_1,\sigma_2\in\Sigma^{\leq d}_{r,m+1}$. Consider $\sigma\in\Sigma^{\leq d}_{r,m+1}$. We have $k_\sigma\leq r^{|\sigma|}-1 < r^d$, $r^{|\sigma|}\leq r^d$, and $k_0\geq r^{d+n}$. Hence $\frac{k_0-k_\sigma}{r^{|\sigma|}} > \frac{r^{d+n}-r^d}{r^d} = r^n-1$. We deduce that $\lambda_{\sigma_1,\sigma_2} > r^n-1$ for any $\sigma_1,\sigma_2\in\Sigma^{\leq d}_{r,m+1}$.

Now, assume that $\delta^\infty(q_0^\infty,\sigma_1) \neq \delta^\infty(q_0^\infty,\sigma_2)$ where $\sigma_1,\sigma_2 \in \Sigma_{r,m+1}^{\leq d}$, and let us prove that $[\delta^{k_0}(q_0^{k_0},\sigma_1)]_{d,n} \neq [\delta^{k_0}(q_0^{k_0},\sigma_2)]_{d,n}$. As $\mathcal{A}^\infty$ is minimal, the language $\mathcal{L}(\mathcal{A}^\infty_{\delta^\infty(q_0^\infty,\sigma_1)})\Delta\mathcal{L}(\mathcal{A}^\infty_{\delta^\infty(q_0^\infty,\sigma_2)})$ is not empty and there exists at least one word σ in this language. As the number of states of the Cartesian product of $\mathcal{A}^\infty_{q_1}$ by $\mathcal{A}^\infty_{q_2}$ is equal to $|\mathcal{A}^\infty|^2$, we can assume that $|\sigma| \leq |\mathcal{A}^\infty|^2-1$. By definition of n, we get $|\sigma| \leq n$. By construction of σ, we have $\rho_{r,m+1}(\sigma) \in Z^\infty_{\sigma_1,\sigma_2}$. Moreover, as $k_\sigma \leq r^n-1 < \lambda_{\sigma_1,\sigma_2}$, we get $\rho_{r,m+1}(\sigma) \in T_{\lambda_{\sigma_1,\sigma_2}}$. Hence $\rho_{r,m+1}(\sigma) \in Z^\infty_{\sigma_1,\sigma_2} \cap T_{\lambda_{\sigma_1,\sigma_2}}$ and from equality (2), we get $\rho_{r,m+1}(\sigma) \in Z^{k_0}_{\sigma_1,\sigma_2} \cap T_{\lambda_{\sigma_1,\sigma_2}}$. We deduce that $\sigma \in \mathcal{L}(\mathcal{A}^{k_0}_{\delta^{k_0}(q_0^{k_0},\sigma_1)})\Delta\mathcal{L}(\mathcal{A}^{k_0}_{\delta^{k_0}(q_0^{k_0},\sigma_2)})$. Therefore $[\delta^{k_0}(q_0^{k_0},\sigma_1)]_{d,n} \neq [\delta^{k_0}(q_0^{k_0},\sigma_2)]_{d,n}$.

Next, assume that $[\delta^{k_0}(q_0^{k_0},\sigma_1)]_{d,n} \neq [\delta^{k_0}(q_0^{k_0},\sigma_2)]_{d,n}$ where $\sigma_1,\sigma_2 \in \Sigma_{r,m+1}^{\leq d}$, and let us prove that $\delta^\infty(q_0^\infty,\sigma_1) \neq \delta^\infty(q_0^\infty,\sigma_2)$. There exists a word $\sigma \in \Sigma_{r,m+1}^{\leq n}$ such that $\rho_{r,m+1}(\sigma) \in Z^{k_0}_{\sigma_1,\sigma_2}$. Moreover, as $k_\sigma \leq r^{|\sigma|}-1 \leq r^n-1 < \lambda_{\sigma_1,\sigma_2}$, we get $\rho_{r,m}(\sigma) \in T_{\lambda_{\sigma_1,\sigma_2}}$. Hence $\rho_{r,m+1}(\sigma)$ is in $Z^{k_0}_{\sigma_1,\sigma_2} \cap T_{\lambda_{\sigma_1,\sigma_2}}$, and from equality (2), we get $\rho_{r,m+1}(\sigma) \in Z^\infty_{\sigma_1,\sigma_2} \cap T_{k_\sigma}$. In particular $Z^\infty_{\sigma_1,\sigma_2} \neq \emptyset$. We get $\delta^\infty(q_0^\infty,\sigma_1) \neq \delta^\infty(q_0^\infty,\sigma_2)$.

As $\mathcal{A}^\infty$ is minimal, any state $q \in Q^\infty$ is reachable from q_0^∞. Hence, for any $q \in Q^\infty$, there exists a word $\sigma_q \in \Sigma_{r,m+1}^{<|\mathcal{A}^\infty|}$ such that $\delta^\infty(q_0^\infty,\sigma_q)=q$. As $d \geq |\mathcal{A}^\infty|$, we deduce that $\delta^{k_0}(q_0^{k_0},\sigma_q) \in Q^{k_0}_{d-1} \subseteq Q^{k_0}_d$ for any $q \in Q^\infty$. In particular the following function f is well defined:

$$\begin{array}{rl} f : Q^\infty \longrightarrow & [Q^{k_0}]_{d,n} \\ q \longrightarrow & [\delta^{k_0}(q_0,\sigma_q)]_{d,n} \end{array}$$

Let us prove that f is a surjective function. Let us consider $C \in [Q^{k_0}]_{d,n}$. There exists $q \in Q^{k_0}_d$ such that $C=[q]_{d,n}$. There exists a word $\sigma \in \Sigma_{r,m+1}^{\leq d}$ such that $q=\delta^{k_0}(q_0^{k_0},\sigma)$. Let $q_\infty=\delta^\infty(q_0^\infty,\sigma)$. As $\delta(q_0^\infty,\sigma)=\delta^\infty(q_0^\infty,\sigma_{q_\infty})$, and $\sigma,\sigma_{q_\infty} \in \Sigma_{r,m+1}^{\leq d}$, we deduce that $[\delta^{k_0}(q_0^{k_0},\sigma)]_{d,n}=[\delta^{k_0}(q_0^{k_0},\sigma_{q_\infty})]_{d,n}$. By definition of f, we get $[\delta^{k_0}(q_0^{k_0},\sigma_{q_\infty})]_{d,n}=f(q_\infty)$. Moreover, as $[\delta^{k_0}(q_0^{k_0},\sigma)]_{d,n}=[q]_{d,n}=C$, we have proved that $C \in f(Q^\infty)$. Therefore f is surjective.

Let us prove that f is an injective function. Consider $q_1,q_2 \in Q^\infty$ such that $f(q_1)=f(q_2)$. We deduce that $[\delta^{k_0}(q_0^{k_0},\sigma_{q_1})]_{d,n}=[\delta^{k_0}(q_0^{k_0},\sigma_{q_2})]_{d,n}$. As $\sigma_{q_1},\sigma_{q_2} \in \Sigma_{r,m+1}^{\leq d}$, we get $\delta^\infty(q_0^\infty,\sigma_{q_1})=\delta^\infty(q_0^\infty,\sigma_{q_2})$. Therefore, by definition of σ_{q_1} and σ_{q_2}, we get $q_1=q_2$. Hence f is injective.

We have proved that f is a one-to-one function.

We denote by $((Q_{d,n},T_{d,n},\alpha_{d,n},\beta_{d,n}),\Sigma,l_{d,n},[q_0^{k_0}]_{d,n},F_{d,n})$ the elements of $[\mathcal{A}^{k_0}]_{d,n}$.

Let us show that for any $(q_1,b,q_2) \in Q^\infty \times \Sigma_{r,m+1} \times Q^\infty$, we have $q_2=\delta^\infty(q_1,b)$ if and only if there exists a transition $t \in T_{d,n}$ such that $(\alpha_{d,n}(t),l_{d,n}(t)\beta_{d,n}(t))=(f(q_1),b,f(q_2))$. Assume first that $q_2=\delta^\infty(q_1,b)$. From $\delta^\infty(q_0^\infty,\sigma_{q_1}.b)=\delta^\infty(q_0^\infty,\sigma_{q_2})$, and $\sigma_{q_1}.b,\sigma_{q_2} \in \Sigma_{r,m+1}^{\leq d}$ (recall that $d \geq |\mathcal{A}^\infty| > |\sigma_{q_1}|$), we get $[\delta^{k_0}(q_0^{k_0},\sigma_{q_1}.b)]_{d,n}=[\delta^{k_0}(q_0^{k_0},\sigma_{q_2})]_{d,n}=f(q_2)$. Let $t=(\delta^{k_0}(q_0^{k_0},\sigma_{q_1}),b,\delta^{k_0}(q_0^{k_0},\sigma_{q_1}))$. As $|\sigma_{q_1}| \leq d-1$, we deduce that $t \in T_{d,n}$. Moreover, remark that $\alpha_{d,n}(t)=[\delta^{k_0}(q_0^{k_0},\sigma_{q_1})]_{d,n}=f(q_1)$, $\beta_{d,n}(t)=[\delta^{k_0}(q_0^{k_0},\sigma_{q_1}.b)]_{d,n}=[\delta^{k_0}(q_0^{k_0},\sigma_{q_2})]_{d,n}=f(q_2)$. Therefore, we have proved $(\alpha_{d,n}(t),l_{d,n}(t),\beta_{d,n}(t))=(f(q_1),b,f(q_2))$. Now, assume that there exists $t \in$

$T_{d,n}$ such that $(\alpha_{d,n}(t), l_{d,n}(t), \beta_{d,n}(t)) = (f(q_1), b, f(q_2))$, and let us prove that $q_2 = \delta^\infty(q_1, b)$. By definition of $T_{d,n}$, there exists $q_1^{k_0} \in Q_{d-1}$ such that $t = (q_1^{k_0}, b, q_2^{k_0})$ where $q_2^{k_0} = \delta^{k_0}(q_1^{k_0}, b)$. As $q_1^{k_0} \in Q_{d-1}$, there exists a word $\sigma \in \Sigma_{r,m+1}^{\leq d-1}$ such that $q_1^{k_0} = \delta^{k_0}(q_0^{k_0}, \sigma)$. From $[\delta^{k_0}(q_0^{k_0}, \sigma)]_{d,n} = \alpha(t) = f(q_1) = [\delta^{k_0}(q_0^{k_0}, \sigma_{q_1})]_{d,n}$, and $\sigma, \sigma_{q_1} \in \Sigma_{r,m+1}^{\leq d}$, we deduce that $\delta^\infty(q_0^\infty, \sigma) = \delta^\infty(q_0^\infty, \sigma_{q_1})$. Moreover, by definition of σ_{q_1}, we have $\delta^\infty(q_0^\infty, \sigma_{q_1}) = q_1$. Therefore $\delta^\infty(q_0^\infty, \sigma) = q_1$. Now remark that we also have $[\delta^{k_0}(q_0^{k_0}, \sigma.b)]_{d,n} = \beta(t) = f(q_2) = [\delta^{k_0}(q_0^{k_0}, \sigma_{q_2})]_{d,n}$, and $\sigma.b, \sigma_{q_2} \in \Sigma_{r,m+1}^{\leq d}$. Hence $\delta^\infty(q_0^\infty, \sigma.b) = \delta^\infty(q_0^\infty, \sigma_{q_2})$. Once again, we have by definition of σ_{q_2}, the equality $\delta^\infty(q_0^\infty, \sigma_{q_2}) = q_2$. So $\delta^\infty(q_0^\infty, \sigma.b) = q_2$ and from $\delta^\infty(q_0^\infty, \sigma) = q_1$, we deduce that $\delta^\infty(q_1, b) = q_2$.

Finally, let us prove that for any $q \in Q^\infty$, we have $q \in F^\infty$ if and only if $f(q) \in F_{d,n}$. Assume first that $q \in F^\infty$ and let us prove that $f(q) \in F_{d,n}$. As $q \in F^\infty$, we get $0 \in Y^\infty_{\delta^\infty(q_0^\infty, \sigma_q)}$. From the equality (1), as $\frac{k_0 - k_{\sigma_q}}{r^{|\sigma_q|}} > r^n - 1 \geq 0$, we get $0 \in Y^{k_0}_{\delta^{k_0}(q_0^{k_0}, \sigma_q)}$. Hence $\delta^{k_0}(q_0^{k_0}, \sigma_q) \in F^{k_0}$. As $|\sigma_q| \leq d$, we deduce that $\delta^{k_0}(q_0^{k_0}, \sigma_q) \in Q_d^{k_0}$ and so $[\delta^{k_0}(q_0^{k_0}, \sigma_q)]_{d,n} \in F_{d,n}$. We have proved that $f(q) \in F_{d,n}$. For the converse, assume now that $f(q) \in F_{d,n}$ and let us prove that $q \in F^\infty$. There exists $q^{k_0} \in f(q)$ such that $q^{k_0} \in F^{k_0} \cap Q_d$. From $[q^{k_0}]_{d,n} = [\delta^{k_0}(q_0^{k_0}, \sigma_q)]_{d,n}$, $\epsilon \in \mathcal{L}(\mathcal{A}^{k_0}_{q^{k_0}})$ and $|\epsilon| = 0 \leq d$, we deduce that $\epsilon \in \mathcal{L}(\mathcal{A}^{k_0}_{\delta^{k_0}(q_0^{k_0}, \sigma_q)})$. Hence $0 \in Y^{k_0}_{\delta^{k_0}(q_0^{k_0}, \sigma_q)}$. From proposition 6.2 and the inclusion $Y^{k_0} \subseteq Y^\infty$, we deduce that $0 \in Y^\infty_{\delta^\infty(q_0^\infty, \sigma_q)}$. Recall that $\delta^\infty(q_0^\infty, \sigma_q) = q$. Hence $0 \in Y_q^\infty$ and we have proved that $q \in F^\infty$.

Therefore, $[\mathcal{A}^{k_0}]_{d,n}$ and $\mathcal{A}^\infty$ are isomorph by f. □

C Proof of proposition 7.5=C.1

Proposition C.1. *We cannot decide if the acceleration of a r-definable couple $(\mathcal{R}, X)$ is r-definable.*

Proof. We are going to reduce the reachability problem for r-definable binary relations to our problem.

The reachability problem for r-definable binary relations consists of deciding from a tuple $(x_0, \mathcal{R}, x_1)$ where $x_0, x_1 \in \mathbb{N}^m$ and $\mathcal{R}$ is a r-definable binary relation over $\mathbb{N}^m$, if $x_0\mathcal{R}^*.x_1$. Naturally, this problem is undecidable because the reachability problem for Minsky machines can be easily reduced to this problem.

Let us provide our reduction by considering an input $(x_0, \mathcal{R}, x_1)$ of the reachability problem for r-definable binary relations. Let us consider the Presburger-definable binary relation $\mathcal{R}_{x_1} \subseteq \mathbb{N}^m \times \mathbb{N}^m$ defined by $\mathcal{R}_{x_1} = \{x_1\} \times \mathbb{N}^m$. Let us prove that $x_0\mathcal{R}^*x_1$ if and only if either (1) acc$(\mathcal{R}, \{x_0\})$ is r-definable and $x_1 \in \mathrm{Post}^*_{\mathcal{R}}(\{x_0\})$, or (2) acc$(\mathcal{R}, \{x_0\})$ is not r-definable and acc$(\mathcal{R} \cup \mathcal{R}_{x_1}, \{x_0\})$ is r-definable. Assume first that $x_0\mathcal{R}^*x_1$. As $x_1 \in \mathrm{Post}^*_{\mathcal{R}}(\{x_0\})$, if acc$(\mathcal{R}, \{x_0\})$ is r-definable then (1) is satisfied. Assume that acc$(\mathcal{R}, \{x_0\})$ is not r-definable and let us prove that (2) is satisfied. There exists $k_0 \in \mathbb{N}$ such that $x_0\mathcal{R}^{k_0}x_1$. In particular for any $k > k_0$, we have $\mathrm{Post}^k_{\mathcal{R}\cup\mathcal{R}_{x_1}}(\{x_0\}) = \mathbb{N}^m$. In particular acc$(\mathcal{R} \cup \mathcal{R}_{x_1}, \{x_0\})$ is r-definable and (2) is satisfied. Let us prove the converse. Assume that we do not have $x_0\mathcal{R}^*x_1$. In particular (1) is not true. An immediate induction over $k \in \mathbb{N}$ shows that acc$(\mathcal{R}, \{x_0\}) =$ acc$(\mathcal{R} \cup \mathcal{R}_{x_1}, \{x_0\})$. Hence (2) is not true. We have proved the converse.

Finally, assume by contraction that we can decide if the acceleration of a r-definable couple $(\mathcal{R}, X)$ is r-definable, then we can decide (2), and from theorem 7.1, we can decide (1). In particular, the reachability problem for r-definable binary relations is decidable and we get a contradiction. □