Polynomial Time Image Computation With Interval-definable Counters Systems

Alain Finkel and Jérôme Leroux

Laboratoire Spécification et Vérification,
CNRS UMR 8643 & ENS de Cachan,
61 av. du Président Wilson,
94235 Cachan cedex, France
Email:{`finkel,leroux`}`@lsv.ens-cachan.fr`

Abstract. The model checking of counters systems often reduces to the effective computation of the set of predecessors $\mathrm{Pre}^*(X')$ of a Presburger-definable set X'. Because there often exists an integer $k \geq 0$ such that $\mathrm{Pre}^{\leq k}(X') = \mathrm{Pre}^*(X')$ we will first look for an *efficient* algorithm to compute the set $\mathrm{Pre}(X)$ in function of X. In general, such a computation is exponential in the size of X. In [BB03], the computation is proved to be polynomial for a restrictive class of counters systems. In this article we show that for any counters systems, the computation is polynomial. Then we show that the computation of $\mathrm{Pre}^{\leq k}(X')$ is polynomial in k (and exponential in the dimension m) for effective counters systems with interval-definable sets.

1 Introduction.

Model checking infinite-state transition systems often reduces to the effective computation of the potentially infinite set of predecessors Pre^*. More precisely, the safety model checking can be expressed as the following problem.

Given as inputs: an infinite-state transition system, a set S_0 of initial states, and a set S_{bad} of non-safe states;
the question is $S_0 \cap \mathrm{Pre}^*(S_{bad}) = \emptyset$?

To succeed in computing the limit $\mathrm{Pre}^*(S_{bad})$ of the infinite non-decreasing sequence $(\mathrm{Pre}^{\leq i}(S_{bad}))_i$, we need three properties:

- The convergence of the sequence $\exists k;\ \mathrm{Pre}^*(S_{bad}) = \mathrm{Pre}^{\leq k}(S_{bad})$, and
- An efficient algorithm for computing $\mathrm{Pre}(X)$ from X,
- An efficient algorithm for computing $(\mathrm{Pre}^{\leq i}(X))$ from X,

The convergence of $(\mathrm{Pre}^{\leq i}(S_{bad}))_i$ with S_{bad} an upward closed is insured for Well Structured Transition Systems (WSTS) [FS01], but even for simple WSTS (for instance, lossy channel systems), the index k such that $\mathrm{Pre}^*(S_{bad}) = \mathrm{Pre}^{\leq k}(S_{bad})$ may be very large [Sch02]. However, even if the convergence is not guaranteed by the theory (for instance when the set of initial states is not upward-closed), in practice we observe that often, the sequence $(\mathrm{Pre}^{\leq i}(S_0))_i$ converge [Del00] [BB03] and it often converges quickly ([Bra] [Bab]). This explains that we will focus our attention on the second problem, to obtain an efficient algorithm for computing $\mathrm{Pre}(X)$ from X and then, on the third problem for efficiently computing $(\mathrm{Pre}^{\leq i}(X))$. We would like to understand what are the conditions insuring an efficient computation of the sequence $(\mathrm{Pre}^{\leq i}(S_{bad}))_i$ and then, may be, to better understand the good performances of some recent tools: BRAIN, BABYLON.

We first have to fix the model of our infinite-state transition systems, the type of infinite sets and the way to represent these sets.

Infinite-state transition systems. We will focus on programs with integer variables and more precisely on counters systems in which a transition is defined by a Presburger function relating the values of the counters before and after the transition is fired. This model is very general (our model restricts neither the number of counters nor the upward closed guards [FMP99]) and powerful ([FL02], [Ler03a]) so the price to pay is the undecidability of reachability properties. Generally, the transition relation will be effectively given by a saturated digit automaton [Ler03a].

Representations of Presburger-definable sets. In order to effectively compute $\mathrm{Pre}(X')$, one generally needs to find a class of infinite sets which has the following properties: (1) closure under union, (2) closure under Pre, (3) membership and inclusion are decidable with a good complexity, and (4) there exists a canonical representation.

Semi-linear basis/periods and Presburger formulas are not canonical representations of Presburger-definable sets (which are also semi-linear sets). Recall that Number Decision Diagrams (NDD) ([Boi98], [WB00], [BC96]), provide a m-digit by m-digit representation of vectors in $\mathbb{N}^m$ whereas Saturated Digit Automata (SDA) use a digit-by-digit representation of vectors in $\mathbb{N}^m$. We prove that NDD and SDA have the same expressiveness but the theory of SDA enjoys an useful characterization of the minimal SDA associated with a set X.

Our computation problems. Now we may precise the input of our two computation problems.

1. The first problem is to compute $\mathrm{Pre}_S(X')$:
   Input: a counters system S and a saturated digit automaton $\mathcal{A}'$ that represents a set $X' \subseteq \mathbb{N}^m$.
   Question: Can we compute in polynomial time in the size of $\mathcal{A}'$, a saturated digit automaton $\mathcal{A}$ representing $X = \mathrm{Pre}_S(X')$?

2. The second problem is to compute the k^{th} element $\mathrm{Pre}_S^{\leq k}(X)$:
   Input: a counters system S, a saturated digit automaton $\mathcal{A}'$ that represents a set $X' \subseteq \mathbb{N}^m$ and an integer k.
   Question: Can we compute in polynomial time in k, a saturated digit automaton $\mathcal{A}$ representing $X = \mathrm{Pre}_S^{\leq k}(X')$?

Related works.

We use the approach called the *regular model checking*: for channel systems, Semi-Linear Regular Expressions [FPS00] and Constrained Queue-content Decision Diagrams [BH99] have been proposed; for lossy channel systems [ABJ98], the tools LCS (in the more general tool TREX [ABS01] [Tre]) uses the downward-closed regular languages and the corresponding subset of Simple Regular Expressions for sets and it represents them by finite automata to compute Post^*; for stack automata, regular expressions or finite automata are sufficient to represent Pre^* and Post^* [BEF+00]; for Petri nets and parameterized rings, [FO97] uses regular languages and Presburger arithmetics (and acceleration) for sets. For Transfer and Reset Petri nets [DFS98], the tool BABYLON [Bab] utilizes the upward closed sets and represents them by Covering Sharing Trees [DRV01], a variant of BDD; for counters automata, the tool BRAIN [Bra] uses linear sets and represent them by their linear bases and periods; for extended counters automata, the tools FAST [Fas], [FL02], [BFLP03] and LASH [Las] utilize semi-linear sets and represents them by NDD, moreover, these two tools are able to accelerate loops [Boi] [FL02]; MONA [Mon] [KMS02] and FMONA [BF00] use formula in WS1S to represent sets; the tool CSL-ALV [BGP97] [Alv] uses linear arithmetic constraints for sets and manipulates formula with the OMEGA solver and the automata library of LASH.
In [BB03], the computation of $\text{Pre}_S(X')$ is proved to be polynomial for a complex subclass of counters systems.
Our results.

1. We introduce SDA as a canonical representation of set of vectors of integers. Even if SDA have the same expression power than NDD, there exists an elegant theoretical characterization of the minimal SDA associated with a set X which is useful for computing the size of the minimal SDA.
2. We show that for counters systems S, the set of immediate predecessors $\text{Pre}_S(X')$ is computable in polynomial time in the size of the SDA that represents X'. This result generalizes a recent result of [BB03].
3. We characterize the affine functions whose the inverse image of any interval-definable set remains interval-definable. Then we prove that the asymptotic size of the minimal SDA that represents $\text{Pre}_S^{\leq k}(X')$ is polynomial in k and exponential in m.

Plan of the paper. Saturated Digit Automata are introduced in section 3 and compared with NDD. In the next section 4, we define counters systems and prove that the computation of $\text{Pre}(X')$ is polynomial in X'. In the last section 5, the asymptotic size of the minimal SDA representing $\text{Pre}_S^{\leq k}(X')$ is proved to be polynomial in k.

2 Preliminaries

The cardinal of a finite set X is written $\text{card}(X)$.

The set of rationals, integers and positive integers are respectively written $\mathbb{Q}$, $\mathbb{Z}$ and $\mathbb{N}$. The set of vectors with m components in a set X is written X^m. The

i-th component of a vector $x \in X^m$ is written $x_i \in X$; we have $x = (x_1, \dots, x_m)$. For any vector $v, v' \in \mathbb{Q}^m$ and for any $t \in \mathbb{Q}$, we define $t.v$ and $v + v'$ in $\mathbb{Q}^m$ by $(t.v)_i = t.v_i$ and $(v + v')_i = v_i + v'_i$. The vector $\mathbf{e}_i \in \mathbb{N}^m$ is defined as $(\mathbf{e}_i)_j = 1$ if $j = i$ and $(\mathbf{e}_i)_j = 0$ otherwise.

The set of square matrices of size m in $\mathbb{Q}$ is written $\mathcal{M}_m(\mathbb{Q})$. A function $f : D \to \mathbb{N}^m$ with $D \subseteq \mathbb{N}^m$ is affine if there exists a square matrix $M \in \mathcal{M}_m(\mathbb{Q})$ and a vector $v \in \mathbb{Q}^m$ such that $f(x) = M.x + v$ for every $x \in D$. Remark that rational matrices are needed for representing some affine functions like $f : 2.\mathbb{N} \to \mathbb{N}$ defined by $f(x) = \frac{x}{2}$.

The set of words over a finite alphabet Σ is written Σ^*. The concatenation of two words σ and σ' in Σ^* is written $\sigma\sigma'$. The empty word in Σ^* is written ϵ.

A finite automaton $\mathcal{A}$ is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, Q_0, F)$; Q is the finite set of states, Σ is the finite alphabet, $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation, $Q_0 \subseteq Q$ is the set of initial states and $F \subseteq Q$ is the set of final states. The size of a finite automaton $\mathcal{A}$ is $|\mathcal{A}| = \text{card}(Q)$. A finite automaton $\mathcal{A}$ is said deterministic if the set Q_0 is reduced to one element $Q_0 = \{q_0\}$ and if there exists a function δ defined over a subset of $Q \times \Sigma$ into Q such that $\Delta = \{(q, \delta(q, a));\ q \in Q;\ a \in \Sigma\}$. A deterministic automaton is said complete if the function δ is defined over the whole set $Q \times \Sigma$. A path P in a finite automaton $\mathcal{A}$ from a state q to a state q' is a finite sequence $q = q_0, (q_0, a_1, q_1), q_1, \dots, (q_{n-1}, a_n, q_n), q_n = q'$ with $n \geq 0$ such that (q_{i-1}, a_i, q_i) is a transition in Δ. The label of P is the word $\sigma = a_1 \dots a_n \in \Sigma^*$. Such a path is also written $q \xrightarrow{\sigma} q'$. The state q' is said reachable from q and q is said co-reachable from q'. The language accepted by a finite automaton $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) = \{\sigma \in \Sigma^*;\ \exists q_0 \in Q_0;\ \exists q_f \in F;\ q_0 \xrightarrow{\sigma} q_f\}$.

Let us recall the two considered logics:

- The Presburger logic ([Ber77]) is built with the following formulas:

$$\phi := \sum_{i \in I} c_i.v_i = c | \exists v\ \phi | \forall v\ \phi | \phi \vee \phi | \phi \wedge \phi | \neg\phi | true | false$$

  where $(c_i)_{i \in I}$ is a finite sequence of $\mathbb{N}$, $c \in \mathbb{N}$ and $(v_i)_{i \in I}$, v are in a finite set V of variables.
- The interval logic ([Str98] (a.k.a simple constraint [AAB00]) is defined by the following formulas:

$$\phi := v_i = c | \phi \vee \phi | \phi \wedge \phi | \neg\phi | true | false$$

A set $X \subseteq \mathbb{N}^m$ is said Presburger-definable (resp. interval-definable) if it can be defined by a Presburger formula (resp. by a formula in the interval logic).

3 Saturated Digit Automata

Recall that there exist two natural ways in order to associate to a word σ a vector in $\mathbb{N}^m$ following that the first letter of σ is considered as an "high bit" or a "low bit". In this article, we consider the "low bit" representation (even if the other one, just seems to be symmetrical, results proved in the paper cannot be easily extended to the other one).

Let us consider an integer $r \geq 2$ called the *basis of decomposition* and an integer $m \geq 1$ called the *dimension of the represented vectors*. A *digit* b is an element of the finite alphabet $\Sigma_r = \{0, \dots, r-1\}$. In general ([Boi98] [WB00], [BC96]), a vector in $\mathbb{N}^m$ is only associated to words of digits whose the length is multiple of m. However, as shown in this article, an extension to any word of any length can be useful.

Like in [Ler03a,Ler03b], function $\gamma_\sigma : \mathbb{N}^m \to \mathbb{N}^m$ is defined by the induction $\gamma_{\sigma.\sigma'} = \gamma_\sigma \circ \gamma'_\sigma$ and $\gamma_b((x_1, \dots, x_m)) = (r.x_m + b, x_1, \dots, x_{m-1})$ for any digit b. Let us remark that if m divides the length of $\sigma = (b_{1,1} \dots b_{1,m}) \dots (b_{n,1} \dots b_{n,m})$, then the following equality holds:

$$\gamma_\sigma((0, \dots, 0)) = \sum_{i=1}^{n} r^{i-1}(b_{i,1}, \dots, b_{i,m})$$

Naturally, the vector $\rho_m(\sigma) = \gamma_\sigma((0, \dots, 0))$ is called the *vector associated* to σ. Thanks to the function $\rho_m : \Sigma_r^* \to \mathbb{N}^m$, we can now define the Saturated Digit Automata and the Number Decision Diagrams.

Definition 1. *A* Saturated Digit Automaton (SDA) $\mathcal{A}$ *that represents a set* $X \subseteq \mathbb{N}^m$ *is a deterministic and complete automaton over* Σ_r *such that* $\mathcal{L}(\mathcal{A}) = \rho_m^{-1}(X)$. *Such a set* X *is called SDA-definable.*

Definition 2. *A* Number Decision Diagram (NDD) $\mathcal{A}$ *([Boi98] [WB00]) that represents a set* $X \subseteq \mathbb{N}^m$ *is a deterministic and complete automaton over* Σ_r *such that* $\mathcal{L}(\mathcal{A}) = \rho_m^{-1}(X) \cap (\Sigma_r^m)^*$.

Remark 1. NDD also allow to represent vectors in $\mathbb{Z}^m$ with "high" or "low" bit first representation. Whereas the results proved in this article can be extended to $\mathbb{Z}^m$, an extension to "high" bit first representation seems difficult.

The following proposition shows that SDA and NDD represent the same sets of $\mathbb{N}^m$.

Proposition 1.
- *From any NDD* $\mathcal{A}$*, we can effectively compute in time* $O(r.|\mathcal{A}|)$ *a SDA* $\mathcal{A}'$ *that represents the same subset, such that* $|\mathcal{A}'| \leq |\mathcal{A}|$.
- *From any SDA* $\mathcal{A}$*, we can effectively compute in time* $O(r.m.|\mathcal{A}|)$ *an NDD* $\mathcal{A}'$ *that represents the same subset, such that* $|\mathcal{A}'| \leq m.|\mathcal{A}|$

Proof. (Sketch). Let us consider a NDD $\mathcal{A}$ that represents a set X. By replacing the set of final states F of $\mathcal{A}$ by the set $F' = \{q \in Q;\ \exists q_f \in F;\ q \xrightarrow{0^*} q_f\}$, we deduce a SDA $\mathcal{A}'$ that represents X.

Now, let us consider a SDA $\mathcal{A}$ that represents a set X. As $\mathcal{L}(\mathcal{A}) = \rho^{-1}(X)$, the "synchronized product" of $\mathcal{A}$ and the automaton with m states that recognizes the language $(\Sigma_r^m)^*$ provides a NDD $\mathcal{A}'$ that also represents X.

Remark 2. As any Presburger-definable set can be effectively represented by a NDD [WB00], the same result holds for SDA.

We have introduced the class of SDA rather than using the NDD because the minimal SDA that represents a set X is given by the "residues" of X.

Definition 3. *The set* $\sigma^{-1}.X = \gamma_\sigma^{-1}(X)$ *is called* the residue *of* $X \subseteq \mathbb{N}^m$ *by* $\sigma \in \Sigma_r^*$.

From $\gamma_{\sigma_1.\sigma_2} = \gamma_{\sigma_1} \circ \gamma_{\sigma_2}$, we deduce the equality $\sigma_2^{-1}.\sigma_1^{-1}.X = (\sigma_1.\sigma_2)^{-1}.X$ that enables us to give the following definition.

Definition 4. *Let* $X \subseteq \mathbb{N}^m$ *be such that its set of residues* $Q(X) = \{\sigma^{-1}.X;\ \sigma \in \Sigma_r^*\}$ *is finite. The deterministic and complete automaton* $\mathcal{A}(X)$ *is defined by:*

$$\begin{cases} \mathcal{A}(X) = (Q(X), \Sigma_r, \delta, q_0, F) \\ \delta(q,b) = b^{-1}.q \\ q_0 = X \\ F = \{q \in Q(X);\ (0,\ldots,0) \in q\} \end{cases}$$

Lemma 1. *For any* $X \subseteq \mathbb{N}^m$ *and* $\sigma \in \Sigma_r^*$, *we have* $\sigma^{-1}.\rho^{-1}(X) = \rho^{-1}(\sigma^{-1}.X)$.

Proof. We have $w \in \sigma^{-1}.\rho^{-1}(X)$ iff $\sigma.w \in \rho^{-1}(X)$ iff $\rho(\sigma.w) \in X$ iff $\gamma_\sigma(\rho(w)) \in X$ iff $\rho(w) \in \sigma^{-1}.x$ iff $w \in \rho^{-1}(\sigma^{-1}.X)$. □

The following theorem is really important because it proves that the structure of the minimal SDA that represents a set X can be obtained just by studying the set of residues of X.

Theorem 1. *A set* $X \subseteq \mathbb{N}^m$ *is SDA-definable if and only if its set of residues is finite. Moreover, in this case,* $\mathcal{A}(X)$ *is the unique minimal SDA that represents* X.

Proof. Assume that $Q(X)$ is a finite set. We are going to show that $\mathcal{A}(X)$ is a SDA that represents X by proving that $\mathcal{L}(\mathcal{A}(X)) = \rho^{-1}(X)$. We have $\sigma \in \mathcal{L}(\mathcal{A}(X))$ iff $(0,\ldots,0) \in \sigma^{-1}.X = \gamma_\sigma^{-1}(X)$. Therefore $\sigma \in \mathcal{L}(\mathcal{A}(X))$ iff $\rho(\sigma) = \gamma_\sigma((0,\ldots,0)) \in X$. Hence, we have proved that $\mathcal{L}(\mathcal{A}(X)) = \rho^{-1}(X)$. In particular $\rho(\mathcal{L}(\mathcal{A}(X))) = X$ and $\rho^{-1}(\rho(\mathcal{L}(\mathcal{A}(X)))) = \mathcal{L}(\mathcal{A}(X))$. We have proved that $\mathcal{A}(X)$ is a SDA that represents X.

Now, assume that X is SDA-definable and let us prove that $Q(X)$ is finite. The language $\mathcal{L} = \rho^{-1}(X)$ is regular. As the minimal deterministic and complete automaton that recognizes $\mathcal{L}$ is unique, there exists a unique minimal SDA that represents X. Recall that the set of states of this minimal automaton is given by $\{\sigma^{-1}.\mathcal{L}\}$. From lemma 1, we deduce that $Q(X) = \{\rho(\sigma^{-1}.\mathcal{L})\}$. Therefore, $Q(X)$ is finite and by uniqueness of the minimal automaton, $\mathcal{A}(X)$ is the unique minimal SDA that represents X. □

Remark 3. Find a theorem equivalent to the previous one for the class of NDD seems difficult.

4 Polynomial time computation of $\mathrm{Pre}_S(X')$

For counters systems S, the computation of the set of immediate predecessors $\mathrm{Pre}_S(X')$ for the SDA representation, is proved to be polynomial in time.

Definition 5. *A saturated digit automaton $\mathcal{A}$ represents a function $f : \mathbb{N}^m \rightarrow \mathbb{N}^m$ if it represents the following set of $\mathbb{N}^{2m}$:*

$$\{(x_1, x'_1, \ldots, x_m, x'_m);\ (x'_1, \ldots, x'_m) = f((x_1, \ldots, x_m))\}$$

Naturally, a function f is said *SDA-definable* if there exists a saturated digit automaton that represents f.

Remark 4. The previous definition can be extended to binary relation.

Definition 6. *A counters system S is a tuple $S = (\mathbb{N}^m, \Sigma, (f_a)_{a\in\Sigma})$ where Σ is a finite set of actions and $f_a : \mathbb{N}^m \rightarrow \mathbb{N}^m$ is a SDA-definable function.*

Remark 5. In practice, the function f_a is given by Presburger formula. However, remark 2 shows that any Presburger definable set is SDA-definable.

The set of immediate predecessors of $X' \subseteq \mathbb{N}^m$ is naturally defined by $\mathrm{Pre}_S(X') = \bigcup_{a\in\Sigma} f_a^{-1}(X')$.

Remark 6. Any counter automaton can be "simulated" by a counter system just by added another counter bounded by the number of control states.

The size $|S|$ of an effective counters system S represented by a sequence of SDA $(\mathcal{A}_a)_{a\in\Sigma}$ is $|S| = \sum_{a\in\Sigma} |\mathcal{A}_a|$.

Theorem 2. *Let $g : \mathbb{N}^m \rightarrow \mathbb{N}^m$ and $X' \subseteq \mathbb{N}^m$ be represented respectively by the SDA $\mathcal{A}^f$ and by the SDA $\mathcal{A}'$. The set $g^{-1}(X')$ can be effectively represented by a SDA in time $O(r.(|\mathcal{A}'| + 1)^{|\mathcal{A}^g|})$.*

Proof. Let us denote by $Q^g_\perp$ the set of states $q^g \in Q^g$ such that there does not exist a path from q^g to a final state. Symmetrically, we define $Q'_\perp$. Let $K = (Q^g_\perp \times Q') \cup (Q^g \times Q'_\perp)$. We are going to prove that the following automaton $\mathcal{A} = (Q, \Sigma_r, \delta, \{q_0\}, F)$ is a SDA that represents $g^{-1}(X)$:

$$\begin{cases} Q = \mathcal{P}(Q' \times Q^g) \\ \delta(q, b) = \{(\delta'(q', b'), \delta^g(q^g, bb'));\ (q', q^g) \in q;\ b' \in \Sigma_r\} \backslash K \\ q_0 = \{(q'_0, q^g_0)\} \backslash K \\ F = \{q_f \in Q;\ \exists q \in Q;\ q \cap (F' \times F^g) \neq \emptyset;\ q_f \xrightarrow{0^*} q_f\} \end{cases}$$

Let us prove that the number of reachable states of $\mathcal{A}$ is bounded by $(|Q'|+1)^{|Q^g|}$. Let q be a reachable state of $\mathcal{A}$ and let us prove that for any (q'_1, q^g) and (q'_2, q^g) in q, we have $q'_1 = q'_2$. There exists a sequence $b_1, \ldots, b_n$ in Σ_r such that $q =$

$\delta(q_0, b_1 \dots b_n)$. By definition of $\mathcal{A}$, there exists two sequences $b'_{1,1}, \dots b'_{n,1}$ and $b'_{1,2}, \dots, b'_{n,2}$ in Σ_r such that

$$\begin{cases} q^g = \delta^g(q_0^g, b_1 b'_{1,1} \dots b_n b'_{n,1}) \\ q^g = \delta^g(q_0^g, b_1 b'_{1,2} \dots b_n b'_{n,2}) \\ q'_1 = \delta'(q'_0, b'_{1,1} \dots b'_{n,1}) \\ q'_2 = \delta'(q'_0, b'_{1,2} \dots b'_{n,2}) \end{cases}$$

As (q^g, q'_1) is not in K, we have $q_g \notin Q^g_\perp$. So, there exists a word $\sigma \in \Sigma_r^*$ and a final state $q^g_f \in F^g$ such that $q^g \xrightarrow{\sigma} q^g_f$ is an accepting path in $\mathcal{A}^g$. As $\mathcal{A}^g$ is a SDA, by replacing σ by $\sigma.0$, we can assume that $|\sigma|$ is even. We have $\sigma = b_{n+1} b'_{n+1} \dots b_k b'_k$ where $b_i, b'_i \in \Sigma_r$. Let x'_1 x'_2 and x be the vectors in $\mathbb{N}^m$ defined by:

$$\begin{cases} x'_1 = \rho_m(b'_{1,1} \dots b'_{n,1} b'_{n+1} \dots b'_k) \\ x'_2 = \rho_m(b'_{1,2} \dots b'_{n,2} b'_{n+1} \dots b'_k) \\ x = \rho_m(b_1 \dots b_k) \end{cases}$$

As $b_1 b'_{1,1} \dots b_n b'_{n,1} b_{n+1} b'_{n+1} \dots b_k b'_k$ and $b_1 b'_{1,2} \dots b_n b'_{n,2} b_{n+1} b'_{n+1} \dots b_k b'_k$ are two accepted words in $\mathcal{L}(\mathcal{A}^g)$, we have $x'_1 = g(x) = x'_2$. As $b'_{1,1} \dots b'_{n,1} b'_{n+1} \dots b'_k$ and $b'_{1,2} \dots b'_{n,2} b'_{n+1} \dots b'_k$ are two words with the same length that represent the same vector $x'_1 = x'_2$, we have $b'_{1,1} \dots b'_{n,1} b'_{n+1} \dots b'_k = b'_{1,2} \dots b'_{n,2} b'_{n+1} \dots b'_k$. In particular, we have proved that $q'_1 = \delta'(q'_0, b'_{1,1} \dots b'_{n,1}) = \delta'(q'_0, b'_{1,2} \dots b'_{n,2}) = q'_2$. Therefore, the number of reachable states of $\mathcal{A}$ is bounded by $(|Q'|+1)^{|Q^g|}$.

Now, let us prove that $\mathcal{L}(\mathcal{A}) \subseteq \rho_m^{-1}(g^{-1}(X))$. Consider an accepting path $q \xrightarrow{w} q_f$ in $\mathcal{A}$. There exists a path $q_f \xrightarrow{0^i} q$ such that $q \cap (F' \times F^g) \neq \emptyset$. Let us decompose the word $w.0^i$ as a sequence of digits $w.0^i = b_1 \dots .b_n$. There exists a word $b'_1 \dots b'_n$ such that $b_1 b'_1 \dots b_n b'_n \in \mathcal{L}(\mathcal{A}^g)$ and such that $b'_1 \dots b'_n \in \mathcal{L}(\mathcal{A}')$. Let $x = \rho_m(b_1 \dots b_m)$ and $x' = \rho_m(b'_1 \dots b'_m)$. Remark that $\rho_{2.m}(b_1 b'_1 \dots b_n b'_n) = (x_1, x'_1, \dots, x_m, x'_m)$. Therefore $x' = g(x)$. Moreover, from $b'_1 \dots b'_n \in \mathcal{L}(\mathcal{A}')$, we deduce $x' \in X'$. So $x \in g^{-1}(X)$. As $\rho_m(w) = \rho_m(w.0^i) = \rho_m(b_1 \dots b_n)$, we have proved $w \in \rho_m^{-1}(g^{-1}(X))$.

Finally, let us prove that $\rho_m^{-1}(g^{-1}(X)) \subseteq \mathcal{L}(\mathcal{A})$. Consider $w \in \rho^{-1}(g^{-1}(X'))$ and let $x = \rho_m(w)$ and $x' = g(x)$. There exists a word $\sigma \in \mathcal{L}(\mathcal{A}^g)$ such that $\rho_{2.m}(\sigma) = (x_1, x'_1, \dots, x_m, x'_m)$. As $\mathcal{A}^g$ is a SDA, we can replace σ by $\sigma.0^j$ for any $j \geq 0$. In particular, $|\sigma|$ can be assumed even and greater than $2.|w|$. Let us write $\sigma = b_1 b'_1 \dots b_n b'_n$ such that $b_i, b'_i \in \Sigma_r$ and remark that $x = \rho_m(b_1 \dots b_n)$ and $x' = \rho_m(b'_1 \dots b'_n)$. Hence $q_0 \xrightarrow{b_1 \dots b_n} q$ is a path in $\mathcal{A}$ such that $q \cap (F' \times F^g) \neq \emptyset$. As $\rho_m(b_1 \dots b_n) = \rho_m(w)$ and $|w| \leq n$, there exists $i \geq 0$ such that $b_1 \dots b_n = w.0^i$. By definition of F, we have $w \in \mathcal{L}(\mathcal{A})$. □

Corollary 1. *Let S be a counters system. The minimal SDA $\mathcal{A}(\mathrm{Pre}_S(X'))$ is computable in polynomial time in function of $\mathcal{A}(X')$.*

Proof. Just remark that $\mathrm{Pre}_S(X') = \bigcup_{a \in \Sigma} f_a^{-1}(X')$. By using an Hopcroft algorithm [Hop71], we can compute the minimal SDA $\mathcal{A}(\mathrm{Pre}_S(X'))$ from a SDA $\mathcal{A}$ that represents $\mathrm{Pre}_S(X')$ in time $O(|\mathcal{A}|.\ln(|\mathcal{A}|))$. □

The previous corollary shows that $\mathcal{A}(\mathrm{Pre}_S(X'))$ can be computed in polynomial time in the size of $\mathcal{A}(X')$ for counters system S. Remark that the complexity is also exponential in $|S|$. However, in the computation of $\mathrm{Pre}_S^{\leq k}(X')$, the size $|S|$ does not depend on k. Moreover, in practice, the size of S is small compared to the size of $\mathcal{A}(X')$.

Remark 7. In the case of the computation of the immediate successors $\mathrm{Post}_S(X) = \bigcup_{a\in\Sigma} f_a(X)$, the number of states of $\mathcal{A}(\mathrm{Post}_S(X))$ can be exponential in the number of states of $\mathcal{A}(X)$. This exponential blow up provides from the fact that $f(X)$ correspond to a "projection" for the function $f : \mathbb{N}^m \to \mathbb{N}^m$ defined by $f(x_1, \ldots, x_m) = (0, x_2, \ldots, x_m)$ ([Ler03b]).

5 Asymptotic size of $\mathrm{Pre}_S^{\leq k}(X')$

The polynomial time computation of $\mathrm{Pre}_S(X')$ is a first step to be able to efficiently compute the set of predecessors in k steps. If each step multiplies the size of the SDA by 2, after k steps, the size of the SDA that represents the set of predecessors is greater than 2^k. In this section, we give sufficient conditions such that this exponential blow up cannot appear.

Definition 7. *A counters system S is affine if for any $a \in \Sigma$, there exists an affine function $f_a : D_a \to \mathbb{N}^m$, $D_a \subseteq \mathbb{N}^m$, such that $x\mathcal{R}_a x'$ iff $x' = f_a(x)$.*

Precisely, we show that if D_a and X' are definable in the interval logic (almost all the counters systems studied in practice, satisfy this condition [Str98], [Del00], [BB02], [FS01], [FL02]), the asymptotic size in k of $\mathcal{A}(\mathrm{Pre}_S^{\leq k}(X'))$ is polynomial in k.

The size of $\mathcal{A}(X)$ is first bounded in the *granularity* of the set X defined as bellow.

Definition 8. *The granularity of an interval-definable set X is the least integer* $\mathrm{gran}(X) \geq 0$*, such that X is the set of vectors accepted by a formula in the interval logic with $c < \mathrm{gran}(X)$:*

$$\phi := v_i = c|\phi \vee \phi|\phi \wedge \phi|\neg\phi|true|false$$

Proposition 2. *For any interval-definable set X, we have:*

$$|\mathcal{A}(X)| \leq (r.\mathrm{gran}(X))^m + 2^{3^m}$$

Proof. Recall that the size of the SDA $\mathcal{A}(X)$ is equal to the number of elements in $\{\gamma_\sigma^{-1}(X);\ \sigma \in \Sigma_r^*\}$. We first prove that for any word $\sigma \in (\Sigma_r^m)^*$ and for any interval-definable set X such that $r^{|\sigma|/m} \geq \mathrm{gran}(X)$, the granularity of $\gamma_\sigma^{-1}(X)$ is bounded by 1. Next, we show that there exists at most 2^{3^m} interval-definable sets whose the granularity is bounded by 1. Finally, from these two results, we prove the proposition.

So, let us first consider $\sigma \in (\Sigma_r^m)^*$ and an interval-definable set X such that $r^{|\sigma|/m} \geq \text{gran}(X)$ and let us prove that $\text{gran}(\gamma_\sigma^{-1}(X)) \leq 1$. Remark that if $\text{gran}(X) = 0$ then $X = \mathbb{N}^m$ or $X = \emptyset$. As in these two cases, we have $|\mathcal{A}(X)| = 1$, we can assume that $\text{gran}(X) \geq 1$. From $\gamma_\sigma^{-1}(X \cap Y) = \gamma_\sigma^{-1}(X) \cap \gamma_\sigma^{-1}(Y)$, $\gamma_\sigma^{-1}(\mathbb{N}^m \backslash X) = \mathbb{N}^m \backslash \gamma_\sigma^{-1}(X)$, we can assume that there exists $i \in \{1, \dots, m\}$ such that $X = \{x \in \mathbb{N}^m;\ x_i = \text{gran}(X) - 1\}$. We have $\gamma_\sigma^{-1}(X) = \{x \in \mathbb{N}^m;\ (\gamma_\sigma(x))_i = \text{gran}(X) - 1\} = \{x \in \mathbb{N}^m;\ x_i = c\}$ where $c = \frac{\text{gran}(X) - 1 - \rho_m(\sigma)}{r^{|\sigma|/m}} < 1$. Remark that if $c \notin \mathbb{N}$ then $\gamma_\sigma^{-1}(X) = \emptyset$ and if $c \in \mathbb{N}$ then $c = 0$. In these two cases, we have proved that $\text{gran}(\gamma_\sigma^{-1}(X)) \leq 1$.

Next, let us prove that there exists at most 2^{3^m} interval-definable sets X such that $\text{gran}(X) \leq 1$. Remark that such a set is a finite union of sets defined by a formula of the form $\bigwedge_{i \in I}(x_i = 0) \bigvee_{i' \in I'}(x_{i'} \neq 0)$ where $I, I' \subseteq \{1, \dots, m\}$ and $I \cap I' = \emptyset$. So, there exists at most 2^{3^m} interval-definable sets whose the granularity is bounded by 1.

Finally, let X be an interval-definable set such that $\text{gran}(X) \geq 1$ and consider $k \geq 0$ such that $r^k \geq \text{gran}(X) \geq r^{k-1}$. The number of states of $\mathcal{A}(X)$ is bounded by $\sum_{i=0}^{m.k-1} r^i + 2^{3^m} \leq (r.\text{gran}(X))^m + 2^{3^m}$. □

Next, we characterize the affine function f such that the inverse image of an interval-definable set remains interval-definable.

Definition 9. *Let X be a subset of $\mathbb{N}^m$, $n \geq 0$ and $I \subseteq \{1, \dots, m\}$, the set $X_{I,n}$ is defined by:*

$$X_{I,n} = \{x \in X;\ \forall i \in I, x_i = n;\ \forall i \notin I, x_i < n\}$$

Proposition 3. *For any interval-definable set X and for any $n \geq \text{gran}(X)$, we have:*

$$X = \bigcup_{I \subseteq \{1,\dots,m\}} X_{I,n} + \sum_{i \in I} \mathbb{N}.\boldsymbol{e}_i$$

Proof. Let us consider a formula ϕ in the logic $\phi := v_i = c|\phi \vee \phi|\phi \wedge \phi|\neg\phi|true|false$ such that $c < \text{gran}(X)$ and such that the set of vectors satisfying ϕ is equal to X. By developing ϕ, we can assume that ϕ is a finite disjunction of formula of the form $\bigwedge_{j \in J_{\neq}}(v_j \neq c_j) \bigwedge_{j \in J_=}(v_j = c_j)$ where $J_{\neq} \cap J_= = \emptyset$ and $c_j < \text{gran}(X)$. Remark that we can assume that $\phi = \bigwedge_{j \in J_{\neq}}(v_j \neq c_j) \bigwedge_{j \in J_=}(v_j = c_j)$ to prove the proposition.

Let $x \in X$ and let us prove that $x \in \bigcup_{I \subseteq \{1,\dots,m\}} X_{I,n} + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$. Let us consider the set $I = \{i \in \{1, \dots, m\};\ x_i \geq n\}$. As for any $j \in J_=$, we have $x_j = c_j < \text{gran}(X)$, we deduce $I \subseteq \{1, \dots, m\} \backslash J_=$. Let us consider the vector $y \in \mathbb{N}^m$ defined by $y_i = n$ if $i \in I$ and $y_i = x_i$ otherwise. As x satisfies ϕ, the vector y also satisfies ϕ. Therefore $y \in X_{I,n}$. From $x \in y + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$, we deduce the inclusion $X \subseteq \bigcup_{I \subseteq \{1,\dots,m\}} X_{I,n} + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$. Let us prove the converse inclusion. Let $x \in \bigcup_{I \subseteq \{1,\dots,m\}} X_{I,n} + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$. There exists $I \subseteq \{1, \dots, m\}$ such that $x \in X_{I,n} + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$. Let $y \in X_{I,n}$ such that $x \in y + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$. As $y \in X_{I,n} \subseteq X$, y satisfies ϕ. As for any $i \in I$, we have $y_i = n$, we have

$I \subseteq \{1, \ldots, m\} \backslash J_=$. From $x \in y + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$, we deduce that x satisfies ϕ. Therefore $x \in X$. $\square$

Proposition 4. *Let $f : D \to \mathbb{N}^m$ with $D \subseteq \mathbb{N}^m$ be an affine function. The two following assertions are equivalent:*

- *D is interval-definable.*
- *For any interval-definable set X', $f^{-1}(X')$ is interval-definable.*

Moreover, in this case, we have $\mathrm{gran}(f^{-1}(X')) \leq \mathrm{gran}(X') + \mathrm{gran}(D)$.

Proof. Remark that if $f^{-1}(X')$ is interval-definable for any interval-definable set X', then in particular, as $\mathbb{N}^m$ is interval-definable, the definition domain $D = f^{-1}(\mathbb{N}^m)$ is also interval-definable. So let us consider an affine function $f : D \to \mathbb{N}^m$ such that $D \subseteq \mathbb{N}^m$ is interval-definable and let X' be an interval-definable set. We first prove that we can assume that $\mathrm{gran}(X') \geq 1$. In fact, if $\mathrm{gran}(X') = 0$, then $X' = \emptyset$ or $X' = \mathbb{N}^m$. In the first case, we have $f^{-1}(X') = \emptyset$ and the set $f^{-1}(X')$ is an interval-definable set such that $\mathrm{gran}(f^{-1}(X')) = 0 \leq \mathrm{gran}(X') + \mathrm{gran}(D)$ and in the second case, we have $f^{-1}(X') = D$ and the set $f^{-1}(X')$ is interval-definable and verify $\mathrm{gran}(f^{-1}(X')) = \mathrm{gran}(D) \leq \mathrm{gran}(X') + \mathrm{gran}(D)$. So, we can assume that $\mathrm{gran}(X') \geq 1$.

As f is an affine function, there exists a square matrix $M \in \mathcal{M}_m(\mathbb{Q})$ and a vector $v \in \mathbb{Q}^m$ such that $f(x) = M.x + v$ for any $x \in D$. Proposition 3 shows that the sets X' and D can be decomposed as follow where $D_I = D_{I,\mathrm{gran}(D)}$ and $X'_I = X'_{I,\mathrm{gran}(X')}$:

$$D = \bigcup_{J \subseteq \{1,\ldots,m\}} D_J + \sum_{j \in J} \mathbb{N}.\mathbf{e}_j$$

$$X' = \bigcup_{I \subseteq \{1,\ldots,m\}} X'_I + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i$$

We have:

$$\begin{aligned}
&f^{-1}(X') \\
&= \bigcup_{J \subseteq \{1,\ldots,m\}} \{x \in D_J + \sum_{j \in J} \mathbb{N}.\mathbf{e}_j;\ f(x) \in X'\} \\
&= \bigcup_{J \subseteq \{1,\ldots,m\}} \bigcup_{d \in D_J} d + \{x \in \sum_{j \in J} \mathbb{N}.\mathbf{e}_j;\ f(d) + M.x \in X'\} \\
&= \bigcup_{\substack{J \subseteq \{1,\ldots,m\} \\ I \subseteq \{1,\ldots,m\}}} \bigcup_{\substack{d \in D_J \\ x' \in X'_I}} d + \{x \in \sum_{j \in J} \mathbb{N}.\mathbf{e}_j;\ f(d) + M.x \in x' + \sum_{i \in I} \mathbb{N}.\mathbf{e}_i\}
\end{aligned}$$

Let us consider a subset J such that D_J is not empty. In this case let us consider $d \in D_J$ and remark that for every $j \in J$, we have $d + \mathbb{N}.\mathbf{e}_j \subseteq D$. Therefore $f(d) + \mathbb{N}.M.\mathbf{e}_j \subseteq \mathbb{N}^m$. So, for every $j \in J$ and for every $i \in \{1, \ldots, m\}$ we have $M_{ij} \geq 0$.

Now, let us consider a subset I such that X'_I is not empty and consider $x' \in X'_I$. Remark that we have just to prove that the following set is interval-definable and has a granularity bounded by $\text{gran}(X')$:

$$\{x \in \sum_{j\in J} \mathbb{N}.\mathbf{e}_j;\ f(d) + M.x \in x' + \sum_{i\in I} \mathbb{N}.\mathbf{e}_i\}$$
$$= \begin{array}{l} \bigcap_{i\notin I} \{x \in \sum_{j\in J} \mathbb{N}.\mathbf{e}_j;\ f(d)_i + \sum_{j\in J} M_{ij}.x_j = x'_i\} \\ \bigcap_{i\in I} \bigcap_{c_i\in\{0,\dots,x'_i-1\}} \{x \in \sum_{j\in J} \mathbb{N}.\mathbf{e}_j;\ f(d)_i + \sum_{j\in J} M_{ij}.x_j \neq c_i\} \end{array}$$

Remark that for every $i \notin I$, we have $x'_i - f(d)_i < \text{gran}(X')$ and for any $i \in I$ and for any $c_i \in \{0, \dots, x'_i - 1\}$, we have $c_i - f(d)_i < x'_i = \text{gran}(X')$. Let us consider the sequence $(\alpha_j)_{j\in\{1,\dots,m\}}$ in $\mathbb{N}$ defined by $\alpha_j = M_{ij}$ if $j \in J$ and $\alpha_j = 0$ otherwise. We have just to prove that for every $k < \text{gran}(X')$ the following set is interval-definable and has a granularity bounded by $\text{gran}(X')$:

$$\{x \in \sum_{j\in J} \mathbb{N}.\mathbf{e}_j;\ \sum_{j\in J} \alpha_j.x_j = k\} = \left(\sum_{j\in J} \mathbb{N}.\mathbf{e}_j\right) \cap \{x \in \mathbb{N}^m;\ \sum_{j=1}^{m} \alpha_j.x_j = k\}$$

The granularity of the set $\sum_{j\in J} \mathbb{N}.\mathbf{e}_j$ is bounded by 1 and as $\text{gran}(X') \geq 1$, we have just to prove that for any sequence $(\alpha_j)_{j\in\{1,\dots,m\}}$ in $\mathbb{N}$ and for any $k < \text{gran}(X')$, the following set has a granularity bounded by $\text{gran}(X')$:

$$\{x \in \mathbb{N}^m;\ \sum_{j=1}^{m} \alpha_j.x_j = k\}$$

Let us consider the set $J' = \{j \in \{1, \dots, m\};\ \alpha_j \geq 1\}$ and let $Y = \{y \in \mathbb{N}^m;\ \forall j \notin J y_j = 0;\ \sum_{j=1}^{m} \alpha_j.y_j = k\}$. Remark that for any $y \in Y$ and for any $i \in \{1, \dots, m\}$, we have $y_j \leq k$. Therefore Y is finite. Moreover, from $\{x \in \mathbb{N}^m;\ \sum_{j=1}^{m} \alpha_j.x_j = k\} = Y + \sum_{j\notin J'} \mathbb{N}.\mathbf{e}_j$, we are done. $\square$

We can now bound the asymptotic size of $\mathcal{A}(\text{Pre}_S^{\leq k}(X'))$ in function of k.

Theorem 3. *Let S be an affine counters system with interval-definable definition domains and let X' be an interval-definable set. The asymptotic size in k of $\text{Pre}_S^{\leq k}(X')$ is in $O(k^m)$.*

Proof. Let $c_S = \max_{a\in\Sigma} \text{gran}(D_a)$. From proposition 4, we deduce that for an interval-definable set X', the set $\text{Pre}_S(X')$ is interval-definable and $\text{gran}(\text{Pre}_S(X')) \leq \text{gran}(X') + c_S$. Therefore $\text{gran}(\text{Pre}_S^{\leq k}(X')) \leq c' + k.c_S$ where $c' = \text{gran}(X')$. From the proposition 2, we deduce $|\mathcal{A}(\text{Pre}_S^{\leq k}(X'))| \leq (r.(c' + c_S.k))^m + 2^{3^m}$. $\square$

Corollary 2. *Let S be an affine counters system with interval-definable definition domains and X' be an interval-definable set. We can compute in polynomial time in k the minimal SDA $\mathcal{A}(\text{Pre}_S^{\leq k}(X'))$.*

Remark 8. When the sets D_a and X' are *upward closed* (an upward closed set X is a subset of $\mathbb{N}^m$ such that $X + \mathbb{N}^m = X$), the sequence $\mathrm{Pre}_S^{\leq k}(X')$ converges as any increasing sequence of upward closed sets.

Remark 9. The bound $O(k^m)$ follows directly from proposition 2. In the proof of these proposition, we have assumed that no sharing appears in the SDA representing an interval-definable set. However, in practice, SDA are like BDD and the asymptotic size of $\mathcal{A}(\mathrm{Pre}_S^{\leq k}(X'))$ is in $O(m.\ln(k))$ rather than $O(k^m)$.

When the sets D_a and X' are just Presburger-definable, the following proposition 5 shows that the asymptotic size in k of $\mathcal{A}(\mathrm{Pre}_S^{\leq k}(X'))$ may be exponential.

Proposition 5. *Let $S = (\mathbb{N}^2, \{a\}, (f_a))$ where $f_a(x_1, x_2) = (r.x_1, x_2)$ over $D_a = \mathbb{N}^2$, and let $X' = \{(x'_1, x'_2) \in \mathbb{N}^2;\ x'_1 = x'_2\}$. For any integer $k \geq 0$, we have:*

$$|\mathcal{A}(\mathrm{Pre}_S^{\leq k}(X'))| \geq r^{k-1}$$

Proof. Let X_i be the subset of $\mathbb{N}^m$ defined for any $i \geq 0$ by $X_i = \{(x, r^i.x);\ x \in \mathbb{N}\}$. We have $\mathrm{Pre}_S^{\leq k}(X') = \bigcup_{i=0}^k X_i$. Assume by contradiction that there exists a SDA $\mathcal{A} = (Q, \Sigma_r, \delta, \{q_0\}, F)$ that represents $\mathrm{Pre}_S^{\leq k}(X')$ and such that $\mathrm{card}(Q) < r^{k-1}$. Let us consider the finite language $\mathcal{L} = (00 + \dots + (r-1)0)^{k-1}10$. For any word $\sigma \in \mathcal{L}$, we have $\delta(q_0, \sigma) \in Q$. As $\mathrm{card}(Q) < r^{k-1} = \mathrm{card}(\mathcal{L})$, there exist two words $\sigma \neq \sigma'$ in $\mathcal{L}$ such that $\delta(q_0, \sigma) = \delta(q_0, \sigma')$. Let $y, y' \in \mathbb{N}$ such that $\rho_2(\sigma) = (y, 0)$ and $\rho_2(\sigma') = (y', 0)$. We have $y, y' \in \{r^{k-1}, \dots, r^k - 1\}$. Let us consider a word $w \in \Sigma_r^*$ such that $\rho_2(w) = (0, y)$. From $\rho_2(\sigma.w) = \rho_2(\sigma) + r^k.\rho_2(w) = (y, r^k.y)$, we deduce that $\rho_2(\sigma.w) \in X_k$. As $\mathcal{A}$ is a SDA that represents $\bigcup_{i=0}^k X_i$, we have proved that $\sigma.w \in \mathcal{L}(\mathcal{A})$. From $\delta(q_0, \sigma) = \delta(q_0, \sigma')$, we deduce that $\sigma'.w \in \mathcal{L}(\mathcal{A})$. Therefore $(y', r^k.y) = \rho_2(\sigma'.w) \in \bigcup_{i-0}^k X_i$. There exists $i \in \{0, \dots, k\}$ such that $(y', r^k.y) \in X_i$. We have $r^k.y = r^i.y'$. From $y \geq r^{k-1}$ and $y' < r^k$, we deduce $i > k - 1$. Hence $i = k$ and we have proved that $y = y'$. As σ and σ' have the same length and as $\rho_2(\sigma) = \rho_2(\sigma')$, we have $\sigma = \sigma'$. We have a contradiction. □

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.

[ABJ98] Parosh Aziz Abdulla, Ahmed Bouajjani, and Bengt Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. In *Proc. 10th Int. Conf. Computer Aided Verification (CAV'98), Vancouver, BC, Canada, June-July 1998*, volume 1427 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 1998.

[ABS01] Aurore Annichini, Ahmed Bouajjani, and Mihaela Sighireanu. TReX: A tool for reachability analysis of complex systems. In *Proc. 13th Int. Conf. Computer Aided Verification (CAV'2001), Paris, France, July 2001*, volume 2102 of *Lecture Notes in Computer Science*, pages 368–372. Springer, 2001.

[Alv] ALV homepage. http://www.cs.ucsb.edu/ bultan/composite/.

[Bab] BABYLON homepage. http://www.ulb.ac.be/di/ssd/lvbegin/CST/-index.html.

[BB02] Constantinos Bartzis and Tevfik Bultan. Efficient symbolic representations for arithmetic constraints in verification. Technical Report ucsb cs:TR-2002-16, University of California, Santa Barbara, Computer Science, 2002.

[BB03] Constantinos Bartziz 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.

[BEF+00] A. Bouajjani, J. Esparza, A. Finkel, O. Maler, P. Rossmanith, B. Willems, and P. Wolper. An efficient automata approach to some problems on context-free grammars. *Information Processing Letters*, 74(5–6):221–227, 2000.

[Ber77] Leonard Berman. Precise bounds for Presburger arithmetic and the reals with addition: Preliminary report. In *Proc. 18th IEEE Symp. Foundations of Computer Science (FOCS'77), Providence, RI, USA, Oct.-Nov. 1977*, pages 95–99, Providence, Rhode Island, 31 October–2 November 1977. IEEE.

[BF00] J.-P. Bodeveix and M. Filali. FMona: a tool for expressing validation techniques over infinite state systems. 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 204–219. Springer, 2000.

[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.

[BGP97] Tevfik Bultan, Richard Gerber, and William Pugh. Symbolic model-checking of infinite state systems using Presburger arithmetic. In *Proc. 9th Int. Conf. Computer Aided Verification (CAV'97), Haifa, Israel, June 1997*, volume 1254 of *Lecture Notes in Computer Science*, pages 400–411. Springer, 1997.

[BH99] Ahmed Bouajjani and Peter Habermehl. Symbolic reachability analysis of FIFO-channel systems with nonregular sets of configurations. *Theoretical Computer Science*, 221(1–2):211–250, 1999.

[Boi] Bernard Boigelot. On iterating linear transformations over recognizable sets of integers. *Theoretical Computer Science*. To appear.

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

[Bra] BRAIN homepage. http://www.cs.man.ac.uk/ voronkov/BRAIN/-index.html.

[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.

[DFS98] Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Proc. 25th Int. Coll. Automata, Languages, and Programming (ICALP'98), Aalborg, Denmark, July 1998*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998.

[DRV01] Gorgio Delzanno, Jean-Francois Raskin, and Laurent Van Begin. Attacking symbolic state explosion. In *Proc. 13th Int. Conf. Computer Aided Verification (CAV'2001), Paris, France, July 2001*, volume 2102 of *Lecture Notes in Computer Science*, pages 298–310. Springer, 2001.

[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.

[FMP99] Alain Finkel, Pierre McKenzie, and Claudine Picaronny. A well-structured framework for analysing Petri nets extensions. Research Report LSV-99-2, Lab. Specification and Verification, ENS de Cachan, Cachan, France, February 1999.

[FO97] Laurent Fribourg and Hans Olsén. Proving safety properties of infinite state systems by compilation into Presburger arithmetic. In *Proc. 8th Int. Conf. Concurrency Theory (CONCUR'97), Warsaw, Poland, Jul. 1997*, volume 1243 of *Lecture Notes in Computer Science*, pages 213–227. Springer, 1997.

[FPS00] Alain Finkel, S. Purushothaman Iyer, and Grégoire Sutre. Well-abstracted transition systems. In *Proc. 11th Int. Conf. Concurrency Theory (CONCUR'2000), University Park, PA, USA, Aug. 2000*, volume 1877 of *Lecture Notes in Computer Science*, pages 566–580. Springer, 2000.

[FS01] Alain Finkel and Phillipe Schnoebelen. Well structured transition systems everywhere! *Theoretical Computer Science*, 256(1–2):63–92, 2001.

[Hop71] John E. Hopcroft. An $n \log n$ algorithm for minimizing the states in a finite-automaton. In Z. Kohavi, editor, *Theory of Machines and Computations*, pages 189–196. Academic Press, 1971.

[KMS02] Nils Klarlund, A. Møller, and M. I. Schwartzbach. MONA implementation secrets. *Int. J. of Foundations Computer Science*, 13(4):571–586, 2002.

[Las] LASH homepage. http://www.montefiore.ulg.ac.be/ boigelot/research/lash/.

[Ler03a] Jérôme Leroux. The affine hull of a binary automaton is computable in polynomial time. In *5th Int. Workshop on Verification of Infinite-State Systems*, Electronic Notes in Theor. Comp. Sci., 2003. to appear.

[Ler03b] Jérôme Leroux. *Algorithmique de la vérification des systèmes à compteurs. Approximation et accélération. Implémentation de l'outil Fast.* PhD thesis, Ecole Normale Supérieure de Cachan, Laboratoire Spécification et Vérification. CNRS UMR 8643, décembre 2003.

[Mon] MONA homepage. http://www.brics.dk/mona/index.html.

[Sch02] Philippe Schnoebelen. Verifying lossy channel systems has nonprimitive recursive complexity. *Information Processing Letters*, 83(5):251–261, 2002.

[Str98] Karsten Strehl. Using interval diagram techniques for the symbolic verification of timed automata. Technical Report 53, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, July 1998.

[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.