A. Finkel and J. Leroux

How to compose Presburger–Accelerations: Applications to Broadcast Protocols

Research Report LSV–02–14, Sep. 2002

Laboratoire Spécification et Vérification

Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

http://www.lsv.ens-cachan.fr/Publis/
Research Report LSV-02-14, Lab. Spécification et Vérification, CNRS & ENS de Cachan, France, Sep. 2002
This is the long version of the paper with same title in Proc. FST&TCS 2002.

How To Compose Presburger-Accelerations : Applications to Broadcast Protocols

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. *Finite linear systems* are finite sets of linear functions whose guards are defined by Presburger formulas, and whose the squares matrice associated generate a finite multiplicative monoid. We prove that for finite linear systems, the accelerations of sequences of transitions always produce an effective Presburger-definable relation. We then show how to choose the good sequences of length n whose number is *polynomial* in n although the total number of cycles of length n is exponential in n. We implement these theoretical results in the tool `FAST` [FAS] (Fast Acceleration of Symbolic Transition systems). `FAST` computes in few seconds the minimal deterministic finite automata that represent the reachability sets of 8 well-known broadcast protocols.

Key words : Presburger model checking, verification, infinite-state systems, reachability set, acceleration.

1 Introduction

Reachability is a central problem in verification. Let us recall that verification of safety properties can be reduced to reachability of a given state from a set of initial states. For some particular models like lossy channel systems [AJ96], [ABJ98] automata with stacks [FWW97], [BEM97], reactive FIFO automata [SFRC99], flat counters automata ([CJ98] and reset/transfer 2-counters automata [FS00b], [FS00a], it is possible to test reachability from a regular set of initial states because the reachability set (or the set of all predecessors) is regular and is effectively computable. But, these "decidable" models are often too restricted because we may wish to use counters with no constraints on guards and actions, as for example, broadcast protocols with parameterized initial set of states [EN98], [EFM99], [Del00a]. Hence, our desired model must be able to *easily* simulate counters automata and as an unavoidable consequence, most of reachability problems will become undecidable.

We make the observation that almost all systems with integer variables can be seen as a *finite set of linear functions* $f(x) = M.x + v : D \subseteq \mathbb{Z}^m \to \mathbb{Z}^m$

having some *definition domains* D. For being able to effectively handle these functions f, we will suppose that their definition domains D (also called guards) are Presburger-definable; such systems are now called *linear systems*. It is clear that the reachability problem is undecidable for linear systems even of dimension 3 (by using a Post problem reduction [FB95]).

We call *Presburger Model Checking* the Regular Model Checking [BJNT00], [FO97b], [BF00], [BGP97] in which Presburger formulas (instead of regular languages) are used as a symbolic representation of infinite set of states. Presburger formulas enjoy good properties because, as regular sets, they are closed under union, intersection and negation; moreover, the satisfiability and the validity are both decidable [GS66].

We say that we *accelerate a sequence of transitions labeled by* σ from a set S of states when we symbolically compute the infinite union of all $\sigma^n(S)$; this set is also called the σ^*-acceleration set. We will compute and use accelerations of different sequences, to speed up the reachability set construction and help its termination. Accelerations are also called meta-transitions in [BW94], [ABJN99], [Boi98] or exact widening in the field of abstract interpretation.

Boigelot's model and results

B. Boigelot and P. Wolper in [BW94], [WB98] and [Boi98] consider a *unique* linear function f whose guard is of the form $Ax \leq b$. They gave two decidable technical conditions [B1] and [B2] over a linear function f (called here a Boigelot's function) which insures the reachability set $f^*(S)$ (i.e., the acceleration of f) to be effectively Presburger-definable when S is Presburger-definable. Conditions [B1] and [B2] become necessary and sufficient when we consider totally-defined linear functions (it means guards are equal to $\mathbb{Z}^m$).

Our contributions

- **Finite linear systems enable composition of accelerations**
  The starting point of our research has been to find out conditions to be able to make acceleration of any composition of functions of a linear system. We remarked that the two technical conditions [B1] and [B2] exactly correspond to the finiteness of the multiplicative monoid generated by the square matrix M associated to f. Hence, it is natural to consider the class of linear systems whose the monoid generated by all the square matrices M, is finite (called *finite linear systems*). This condition is decidable [Jac78],[MS77]. We prove that acceleration of f_σ is possible for any cycle σ.
- **How to find out the good accelerations ?**
  The second step to really use accelerations is to compute how many different accelerations are necessary for calculating the reachability relation or the reachability set. We prove that from the exponential number in k of different accelerations of words in $\Sigma^{\leq k}$, we can extract a polynomial number of "good accelerations". Let f and g be two functions, under conditions on f and g,

we replace acceleration of f and g by acceleration of the unique function $f \vee g$ whose the definiton domain is equal to the definition domains of f and g. To do so, we use the property that the union of two Presburger-guards is still a Presburger-guard (remark that the guards used by Boigelot are not stable by union because they are equal to convex sets of the form $A.x \leq b$). Moreover, the Theorem 4 shows us that for a subclass of totally-defined linear systems, we have $k \leq |\mathcal{M}_{\boldsymbol{L}}|$ ($\mathcal{M}_{\boldsymbol{L}}$ is the finite multiplicative monoid associated to a linear system $\boldsymbol{L}$).

– **The tool `FAST`**
A tool named `FAST` (Fast Acceleration of Symbolic Transition systems) implements the theoretical results of this paper. The current version of `FAST` computes, in few seconds, all the complete reachability sets (represented by finite automata having less than 12 states) of the 8 well-known broadcast protocols (and some other systems known to be difficult to analyze). To the best of our knowledge, it is the first time that a tool computes automatically the complete reachability sets of all the 8 broadcast protocols (and not only the set of all predecessors of an upward closed set of bad states).

Related works on acceleration

H. Comon and Y. Jurski analyze a relational model based on multiple counters automata in [CJ98] but their guards cannot express set of integers satisfying $x+y=z$. On the other hand, actions are relations (they are not restricted to be functions). Hence, their model is not comparable with finite linear systems. A. Finkel and G. Sutre [Sut00], [FS00b], [FS00a] study 2-counters automata with reset/transfer/+1/-1 actions and guards of the form $x_i \geq c$ and $x_i = 0$ (zero test). They prove that cycle-acceleration leads to the effective computation of the Presburger-definable reachability set.
[FS00b] considers acceleration from a theoretical point of view with applications to 2-counters automata and to FIFO channel systems. [AAB00] guesses the result of the infinite iteration of a cycle, in a hybrid system, and verify whether the guess is correct or not. [Rev90], [FO97b], [FO97a] and [BF99] automatically accelerate a given non-elementary cycle. [PS00] attacks the acceleration techniques using formulas in a variant of $WS1S$ for the symbolic representation of sets of states: the authors experiment their tool on examples but no decidability result is given about the convergence in general. Among a lot of papers on some what we called the Presburger Model Checking, we can cite [BF00] and [BGP97] which both also use the tool `MONA` [MON]. For FIFO channels systems, acceleration of loops have been recently studied in [FPS00] with SLRE, in [BGWW97] with QDD and in [BH99] with CQDD; for lossy FIFO channels systems, acceleration of loops and SRE have been studied in [ABJ98] and [AAB99].

Plan of the paper

Section 2 and 3 intuitively introduce, on a simple example, acceleration of cycles. Then, after recalling the basic notions (labeled transition systems, matrices and

functions, Presburger formulas), Section 4 presents the so-called finite linear systems model. The main theoretical result of this section is: the cycle-acceleration relation of a finite linear system is Presburger-definable (Theorem 2). An "if and only if" condition is given for finite totally-defined linear systems. The finiteness of the monoid is decidable and it is EXPTIME for positive finite linear systems.

Sections 5 deals with the problem of finding a little number of useful cycles for acceleration. The finiteness of the monoid is used for proving that only a polynomial number of cycles have to be considered for acceleration.

Section 6 reports the use of the tool `FAST` (Fast Acceleration of Symbolic Transition systems) specially on broadcast protocols. The size of automata corresponding to Presburger formulas is short in practice.

2 Preliminaries

A *labeled transition system* LTS is a tuple $LTS = (S, \Sigma, \rightarrow)$ where S is the set of *states*, Σ is a finite alphabet of *actions*, and $\xrightarrow{a} \subseteq S \times S$ is the *transition relation* for the action a. For some letters a_1, ..., a_n, the relation $\xrightarrow{a_1 \cdots a_n}$ is naturally defined by $s \xrightarrow{a_1 \cdots a_n} s'$ if and only if $s \xrightarrow{a_1} \cdots \xrightarrow{a_n} s'$; and for every s, we have $s \xrightarrow{\epsilon} s$ where ϵ is the empty word. A *labeled transition system with initial states* is a couple (LTS, S_0) where LTS is a labeled transition system and $S_0 \subseteq S$ is a set of *initial states*. The *reachability set* of a labeled transition system is $Reach(LTS, S_0) = \{s' \in S; \exists \sigma \in \Sigma^* \; \exists s_0 \in S_0 \; s_0 \xrightarrow{\sigma} s'\}$.

Throughout this paper, $\mathbb{Z}$ is the set of *integers* and $\mathbb{N}$ is the set of *non negative integers*. The set of *square matrices of size* $m \times m$ over the integers $\mathbb{Z}$ is denoted by $\mathcal{M}_m(\mathbb{Z})$ and the set of *column matrices* of size $m \times 1$, also called *vectors*, is written $\mathbb{Z}^m$. Given an integer $1 \leq i \leq m$, v_i is the i-th component of a vector $v \in \mathbb{Z}^m$. The vector x such that $x_i = 1$ and $x_j = 0$ for every $j \neq i$ is written $\mathbf{x}_i$.

Consider a finite set Y of *variables*. The set of *arithmetical terms* over Y, is defined by the grammar $t ::= 0|1|y|t - t|t + t$. The set of *Presburger formulas* over Y is defined by the grammar:

$$\phi ::= t < t|\neg\phi|\phi \vee \phi|\exists x'.\phi|true$$

The subset of variables $x' \in Y$ that appear in a quantification is written X'. The set $X = Y - X'$ is called the set of free variables. A Presburger formula over the set of free variables X is written $\phi(X)$. An *interpretation* v of $\phi(X)$ is a vector in $\mathbb{Z}^X$. We say that v *satisfies* $\phi(X)$ which is written $v \models \phi(X)$ if by replacing any occurrence of $x \in \phi(X)$ by the integer $v(x)$ we obtain a true formula over the integers. The set of vectors satisfying a Presburger formula $\phi(X)$ is written $S(\phi)$. A set $S \subseteq \mathbb{Z}^X$ is said to be *Presburger-definable* if there exists a Presburger formula $\phi(X)$ such that $S = S(\phi)$.

Consider some functions $(f_a)_{a \in \Sigma}$ where Σ is a finite alphabet, the function $f_{a_n} \circ \cdots \circ f_{a_1}$ is written $f_{a_1 \cdots a_n}$ to improve the readability. For the same reason, the square matrices product $M_{a_n} \cdots M_{a_1}$ is written $M_{a_1 \cdots a_n}$. The empty word ϵ

has a length 0 and by convention the matrix M_ϵ is equal to the *identity matrix* I and the function f_ϵ is equal to the *identity function.*

3 What is acceleration ?

The reachability set of a labeled transition systems is not finitely computable in general. In fact, assume that the transition system has an infinite number of reachable states, then an algorithm that computes one reachable state at one step never terminates. To handle such infinite-state labeled transition systems, we must find out a method that computes, in a finite number of steps, an infinite number of reachable states. To do so, we accelerate sequences of transitions (we say also that we compte a meta-transition [BW94] or in the field of abstract interpretation, we say that we use exact widening)

Definition 1 (σ^*-acceleration). *For a labeled transition system LTS and a sequence σ, the* σ^*-acceleration relation *is the relation R such that sRs' if and only if there exists an integer $k \in \mathbb{N}$ such that $s \xrightarrow{\sigma^k} s'$. For a set of states S, the* σ^*-acceleration set *from S is the set the set of states s' such that there exists $s \in S$ and $k \in \mathbb{N}$ such that $s \xrightarrow{\sigma^K} s'$. Without ambiguity, the σ^*-acceleration relation and the σ^*-acceleration set will be called by the same name: σ^*-acceleration or just acceleration.*

We present this method here on a simple example: the Producer-Consumer algorithm [BGP97] with a buffer of size a represented in Fig. 1. It contains 3 events corresponding to a write-action e_W and two read-actions e_{R_1} and e_{R_2}.

Data Variables: i, b, o_1, o_2: integer
Constants: a: integer
Initial Condition: $i = a \wedge (b = o_1 = o_2 = 0)$
Events:
e_W **enabled:** $i > 0$
action: $i' = i - 1 \wedge b' = b + 1$
e_{R_1} **enabled:** $b > 0$
action: $o_1' = o_1 + 1 \wedge b' = b - 1$
e_{R_2} **enabled:** $b > 0$
action: $o_2' = o_2 + 1 \wedge b' = b - 1$

Fig. 1. A Producer-Consumer Algorithm (see [BGP97])

The value of the variables and the constant of the algorithm form a vector $(a, i, b, o_1, o_2) \in \mathbb{Z}^5$. The corresponding labeled transition system LTS_{PCA} has thus $S = \mathbb{Z}^5$ as set of states and $\Sigma = \{e_W, e_{R_1}, e_{R_2}\}$ as alphabet. The relations

$\xrightarrow{e_W}$, $\xrightarrow{e_{R_1}}$, $\xrightarrow{e_{R_2}}$ are defined in a natural way: for instance $s \xrightarrow{e_W} s'$ if and only if $s_2 > 0$ and $s' = s + (0, -1, 1, 0, 0)$.

Intuitively, one may see that the action e_W can be accelerated from $S_0 = \{(a, i, b, o_1, o_2) \in \mathbb{N}^5; a = i \wedge b = o_1 = o_2 = 0\}$, it means that one may iterate indefinitely e_W^* and then one reaches the set of states $S_1 = \{(a, i, b, o_1, o_2) \in \mathbb{N}^5; a = i + b \wedge o_1 = o_2 = 0\}$. Then one may accelerate $(e_{R_1} e_W)$...

Consider the labeled transition system LTS_{PCA} associated with the Producer-Consumer Algorithm (Fig. 1) where the set of initial states is $S_0 = \{(a, i, b, o_1, o_2) \in \mathbb{N}^5; a = i \wedge b = o_1 = o_2 = 0\}$. By combining different accelerations, we may prove that

$$\begin{aligned} S_1 &:= \mathit{Reach}(LTS_{PCA}, S_0, e_W^*) && = \{a = i + b \wedge o_1 = o_2 = 0\} \\ S_2 &:= \mathit{Reach}(LTS_{PCA}, S_1, (e_{R_1} e_W)^*) && = \{a = i + b + o_1 \wedge o_2 = 0\} \\ S_3 &:= \mathit{Reach}(LTS_{PCA}, S_2, (e_{R_2} e_W)^*) && = \{a = i + b + o_1 + o_2\} \end{aligned}$$

From this computation, we see that $\mathit{Reach}(LTS_{PCA}, S_0) = S_3 = \{(a, i, b, o_1, o_2) \in \mathbb{N}^5; a = i + b + o_1 + o_2\}$. Hence we can easily deduce the invariant $i = b = 0 \rightarrow a = o_1 + o_2$ which is established in [BGP97].

Such a method for computing the set of reachable states implies that we have found a class of labeled transition systems for which we have a representation of infinite subsets of integers such that

- we can compute a representation of the σ^*-accelerations, and
- we can find out the "good" σ to accelerate which are here e_W, $e_{R_1} e_W$ and $e_{R_2} e_W$.

A subset of $\mathbb{Z}^m$ will be represented by a Presburger formula, the class of finite linear systems will be proved to be a good class of labeled transition systems in the two next sections.

4 How to compute an acceleration ?

We are interested by the computation of accelerations of transitions for labeled transition systems that use tuples of integers to represent their states and *Presburger-linear functions* to represent their actions. A Presburger-linear function f is a function which can be represented by a tuple (M, v, ϕ) where $M \in \mathcal{M}_m(\mathbb{Z})$, $v \in \mathbb{Z}^m$ and $\phi(x_1, \ldots, x_m)$ is Presburger formula, such that $f(s) = M.s + v$ for every $s \models \phi$. Without ambiguity and to improve the readability, such a tuple (M, v, ϕ) and the Presburger-linear function associated with it will be noted by the same letter f. A Presburger-linear function is said to be *totally defined* if it is defined over the whole set $\mathbb{Z}^m$. The totally defined Presburger-linear function associated with a Presburger-linear function $f = (M, v, \phi)$ is written $\bar{f} = (M, v, true)$.

Definition 2 (Linear systems). *A* linear system $\boldsymbol{L}$ *is a triple* $\boldsymbol{L} = (\Sigma, m, f_\Sigma)$ *where* Σ *is a finite alphabet,* $m > 0$ *is a positive integer and* $f_\Sigma = \{f_a = (M_a, v_a, \phi_a); a \in \Sigma\}$ *is a set of Presburger-linear functions into* $\mathbb{Z}^m$.

A labeled transition system can easily be associated with a linear system $\boldsymbol{L} = (\Sigma, m, f_\Sigma)$ by considering $LTS = (\mathbb{Z}^m, \Sigma, \rightarrow)$ where $s \xrightarrow{a} s$ if and only if $s' = f_a(s)$. The labeled transition system associated with a linear system will be written without ambiguity by the same name.

Definition 3 (The monoid $\mathcal{M}_{\boldsymbol{L}}$). *The multiplicative monoid generated by the matrices M_a (for $a \in \Sigma$) of a linear system $\boldsymbol{L}$ is written $\mathcal{M}_{\boldsymbol{L}}$. When $\boldsymbol{L}$ is composed of a unique function $f(s) = M.s + v$, then the monoid $\mathcal{M}_{\boldsymbol{L}}$ is simply written $< M >$.*

The structure of such a monoid has already been studied by Boigelot.

Theorem 1. *[Boi98] For every linear function $f(s) = M.s + v$ with a guard defined by $A.x \leq b$, where $A \in \mathbb{Z}^{k \times m}$, $b \in \mathbb{Z}^k$, the set $f^*(S_0)$ is Presburger-definable for every Presburger-definable set S_0 if there exists an integer $p > 0$ such that [B1] and [B2] hold, with:*

- *[B1] the eigenvalues of the matrix M^p belong to $\{0, 1\}$, and,*
- *[B2] the matrix M^p is diagonalizable.*

Moreover, if the guard of f is equal to $\mathbb{Z}^k$, the conditions [B1] and [B2] become necessary and sufficient.

It is easy to see that the two conditions (*[B1]* and *[B2]*) proposed by Boigelot in the previous theorem are equivalent to the fact that the monoid $< M >$ is finite (a proof of this result can be found in the Appendix in lemma 1). What we want to do, is to be able to compute acceleration of *any* word $\sigma \in \Sigma^*$. Therefore, the following definition seems to be a good one.

Definition 4 (Finite linear systems). *A linear system $\boldsymbol{L}$ is finite if $\mathcal{M}_{\boldsymbol{L}}$ is finite.*

Deciding the finiteness of a linear system is equivalent to deciding the finiteness of a multiplicative monoid generated by a finite number of matrices in $\mathcal{M}_m(\mathbb{Z})$. This last problem was proved to be decidable [Jac78],[MS77]. To the best of our knowledge, the complexity of this problem is not known. However, for *positive linear systems* (i.e., all matrices are positive ($M_a \in \mathcal{M}_m(\mathbb{N})$)) we can easily deduce from [MS77] an algorithm to decide the finiteness of a linear system in EXPTIME.

Following [DFS98] and [FS00b], we call Transfer/Reset/Inhibitor Petri Nets extended Petri nets which are able to (1) transfer the content of a place into another, (2) to reset (or empty) a place and (3) to zero-test a place. In fact, all those protocols can be proved to be finite linear system. Moreover as all broadcast protocols are parameterized Transfer/Reset/Inhibitors Petri Nets, we can prove that Transfer/Reset/Inhibitors Petri Nets and all broadcast protocols [EN98], [EFM99], [Del00a], [Del00b] are finite linear systems.

Proposition 1. *Transfer/Reset/Inhibitor Petri Nets [DFS98] and all broadcast protocols [EN98], [EFM99], [Del00a], [Del00b] are finite linear systems.*

Proof. All these classes are linear systems. Moreover, the square matrices used by the linear sunctions are *Reset/Transfer/Inhibitor matrix* (*RTI-matrix*): a RTI-matrix is a matrix such that all its entries are equal to "0" expect at most one entry per column that can be equal to "1". We see that the product of two RTI-matrices is an RTI-matrix. As there exists at most a finite number of RTI-matrices with a given size, we have proved that every linear system such that all matrices M_a are RTI-matrices is finite. □

In practice a linear system is finite every time we have "no duplication of the variables"(the variables x_1 is said duplicated by f if $f(x_1, x_2) = (x_1, x_2) + v$ for every $(x_1, x_2) \in \mathbb{N}^m$).

For finite linear systems it seems natural that we can represent the σ^*-acceleration of a Presburger-definable set by a Presburger formula. However, this is not an obvious consequence because the guards used by Boigelot are only of the form $A.x \leq b$ and do not permit to represent every Presburger definable set. In fact, remark that this class of guards are not closed by union. Also remark that in the next section, this closure by union is proved to be an important property that enable us to find the "good" words σ to accelerate. In fact, we can do it and we can even do better by computing the σ^*-acceleration relation.

Proposition 2. *The relation $s' \in f^*(s)$ is effectively equivalent to a Presburger formula if the function f is Presburger-linear and if the monoid $< M >$ is finite.*

Proof. Consider such a Presburger-linear function $f = (M, v, \phi)$. Recall that the totally defined function associated with f is written $\bar{f} = (M, v, true)$. As the monoid $< M >$ is finite, there exist two integers a and b such that $M^{a+b} = M^a$. One can easily check that we then have $\bar{f}^a(\bar{f}^b(s)) = \bar{f}^a(s) + M^a.\bar{f}^b(0)$. Therefore for every integer $q \geq 0$, we have $\bar{f}^{a+q.b}(s) = \bar{f}^a(s) + q.M^a.\bar{f}^b(0)$. Then remark that the formula $s' = \bar{f}^i(s) \wedge i \geq 0$ with the three free variables s', i, s is equivalent to a Presburger formula:

$$\begin{bmatrix} s' = \bar{f}^i(s) \\ \wedge\, i \geq 0 \end{bmatrix} \Longleftrightarrow \left(\bigvee_{0 \leq j < a} \begin{bmatrix} s' = \bar{f}^i(s) \\ \wedge i = j \end{bmatrix} \right) \vee \left(\bigvee_{0 \leq r < b} \exists q \geq 0 \begin{bmatrix} s' = \bar{f}^i(s) \\ \wedge i - a = r + q.b \end{bmatrix} \right)$$

$$\Longleftrightarrow \left(\bigvee_{0 \leq j < a} \begin{bmatrix} s' = \bar{f}^j(s) \\ \wedge i = j \end{bmatrix} \right) \vee \left(\bigvee_{0 \leq r < b} \exists q \geq 0 \begin{bmatrix} s' = \bar{f}^r(\bar{f}^a(s) + q.M^a.\bar{f}^b(0))) \\ \wedge i - a = r + q.b \end{bmatrix} \right)$$

Finally just remark that $s' \in f^*(s)$ is equivalent to the Presburger formula:

$$s' \in f^*(s) \Longleftrightarrow \exists k \geq 0 \ \left[s' = \bar{f}^k(s) \wedge (\forall i \ 0 \leq i < k \Rightarrow \phi(\bar{f}^i(s))) \right]$$

□

We can deduce from the previous Proposition 2 the main result: for any finite linear system, all the σ^*-accelerations are computable.

Theorem 2. *For a finite linear system $\boldsymbol{L}$ and for every word $\sigma \in \Sigma^*$, the σ^*-acceleration relation is effectively Presburger definable.*

Let us remark that the previous theorem is a powerful one because finite linear systems form a highly undecidable model. In fact, the reachability problem (is a state s in $Reach(\boldsymbol{L}, S_0)$?) is undecidable for finite linear systems of dimension $m = 3$ with $S_0 = \{s_0\}$. This can be proved by reducing the Post problem to the reachability problem with the same method than in [FB95].

Finally, remark that we can construct counter examples to the converse of the Theorem 2 just by choosing non finite linear system such that the guards of its linear functionss are not satisfiable ($\phi_a = false$). However, for the class of totally defined linear systems (see Theorem 4 for an other property satisfying by this class), the converse of the Theorem 2 is true.

Definition 5 (Totally defined linear systems). *A totally defined linear system is a linear system such that all its Presburger-linear functions are totally defined.*

Proposition 3. *A totally defined linear system such that for every word σ, the relation $s' \in f_\sigma^*(s)$ is Presburger definable is finite.*

Proof. Consider such a totally defined linear system $\boldsymbol{L}$. Consider a matrix $M \in \mathcal{M}_{\boldsymbol{L}}$ and remark that there exists a word σ such that $M = M_\sigma$. As $f_\sigma^*(S_0) = Reach(\boldsymbol{L}, S_0, \sigma)$ is Presburger-definable for every Presburger-definable set S_0, then by using [Boi98], the monoid $< M >$ is finite. The Burnside theorem [MZ75], [Jac78], [MS77] implies that $\mathcal{M}_{\boldsymbol{L}}$ is finite. □

5 How to find out the good accelerations ?

For a finite linear system $\boldsymbol{L} = (\Sigma, m, f_\Sigma)$, the number of words σ with a length less than or equal to an integer $k \geq 0$ is equal to $\sum_{i=1}^{k} |\Sigma|^k$. Therefore, it seems that a brute force algorithm that computes the accelerations by all these words will take too much time. However, for finite linear systems this exponential number of cycles can be put together such that we have only to consider a polynomial number of cycles ! Let us show why.

Consider two Presburger-linear functions $f_1 = (M_1, v_1, \phi_1)$ and $f_2 = (M_2, v_2, \phi_2)$ such that $\bar{f_1} = \bar{f_2}$ (recall that it is equivalent to $M_1 = M_2$ and $v_1 = v_2$) and let us define the Presburger-linear function $(M, v, \phi_1 \vee \phi_2)$ where $M = M_1 = M_2$ and $v = v_1 = v_2$ written $f_1 \vee f_2$. We then have $s' \in (f_1 \vee f_2)^*(s)$ if and only if $s' \in (f_1, f_2)^*(s)$ where $s' \in (f_1, f_2)^*(s)$ is defined by $s' \in f_1^* \circ f_2^* \circ \cdots \circ f_1^* \circ f_2^*(s)$.

Therefore, the good measure for counting the number of σ^*-accelerations with a length less than or equal to an integer $k \geq 0$ we have to consider to be exhaustive is not the number of words in $\Sigma^{\leq k}$, but the cardinal of the set $\{\bar{f}_\sigma; \sigma \in \Sigma^{\leq k}\}$ called *the number of σ^*-accelerations of length less than or equal to k.*

Theorem 3. *The number of σ^*-accelerations of length less than or equal to an integer k for a finite linear system is polynomial in the variable k.*

Proof. Consider the finite set $V_{\boldsymbol{L}} = \{M.v_a; M \in \mathcal{M}_{\boldsymbol{L}}; a \in \Sigma\}$ and a word $\sigma = a_1 \dots a_k$. We have $\bar{f}_\sigma(s) = M_\sigma.s + \sum_{i=1}^{k} M_{a_1} \cdots M_{a_{i-1}}.v_{a_i}$. Therefore $f_\sigma(s)$ is equal to $M_\sigma.s$ plus the sum of k elements in $V_{\boldsymbol{L}}$ (not necessary different). So, there exists a finite sequence of non negative integers $(\alpha_v)_{v \in V_{\boldsymbol{L}}}$ such that $\bar{f}_\sigma(s) = M_\sigma.s + \sum_{v \in V_{\boldsymbol{L}}} \alpha_v.v$ and $\sum_{v \in V_{\boldsymbol{L}}} \alpha_v = k$. As the number of sequences of non negative integers $(\alpha_v)_{v \in V}$ such that $\sum_{v \in V_{\boldsymbol{L}}} \alpha_v \leq k$ is bounded by $k^{|V_{\boldsymbol{L}}|}$, the number of σ^*-accelerations of length less than or equal to an integer k for a linear system $\boldsymbol{L} = (\Sigma, m, f_\Sigma)$ is bounded by $|\mathcal{M}_{\boldsymbol{L}}|.k^{|V_{\boldsymbol{L}}|}$. □

An algorithm that computes all the σ^*-accelerations of length less than or equal to k can be easily deduced from the previous theorem. We have implemented this simple algorithm in the tool `FAST` and very promising results have been obtained (see section 6).

Moreover remark that there exist Presburger-linear functions f_1 and f_2 such that $\bar{f}_1 = \bar{f}_2$ and the relation $s' \in (f_1, f_2)^*(s)$ cannot be obtained by a finite composition of the relations f_1^* and f_2^* (for instance consider the function $f_1(s)$ defined over the odd integers and the function $f_2(s)$ defined over the even integers by $\bar{f}_1(s) = \bar{f}_2(s) = s + 1$). That shows the powerful of this tips of acceleration.

Recall once again that this tips cannot be implemented directly with LASH [LAS] because the guards used by Boigelot are of the form $A.x \leq b$ and therefore they are not stable by union.

So, we have proved that for finite linear systems, we can increase the length of the σ^*-accelerations we consider without exponential grow up. Now the question is "Can we bound the length of the σ^*-accelerations we have to consider ?". The exact problem is the following one: being given a linear system such that its reachability set can be computed by σ^*-accelerations, can we bound the length of the used words σ ? We have only succeeded in answering this question for the class of totally defined finite linear systems.

Theorem 4. *Consider a totally defined finite linear system $\boldsymbol{L}$, the reachability relation is Presburger-definable. Moreover, we can compute t words σ_1, ..., σ_t with length less than or equal to $|\mathcal{M}_{\boldsymbol{L}}|$ such that the reachability relation is equal to $f_{\sigma_1}^* \circ \cdots \circ f_{\sigma_t}^*$.*

Proof. We can associate to a finite linear system an automaton whose the set of states is equal to the finite monoid $\mathcal{M}_{\boldsymbol{L}}$. In fact, let us consider the automaton $A_{\boldsymbol{L}} = (Q, \Sigma, \delta_{\boldsymbol{L}})$ defined by a set of states $Q = \mathcal{M}_{\boldsymbol{L}}$ and a relation $\delta_{\boldsymbol{L}}$ defined by $\delta_{\boldsymbol{L}}(a, M) = M.M_a$ for every $a \in \Sigma$ and every $M \in \mathcal{M}_{\boldsymbol{L}}$. A path in this automaton is as usual, a sequence $M_0, a_1, M_1, \dots, a_n, M_n$ also written $M_0 \xrightarrow{a_1 \dots a_n} M_n$. The length of a path is written $|P| = n$. The notion of cycles, elementary cycles, elementary paths are defined as usual. The set of elementary cycles is written $EC(\boldsymbol{L})$. If the concatenation of two paths P and P' is defined, the resulting path is written $P.P'$.

A sequence of paths P_0, $C_1^{n_1}$, P_1, ..., $C_t^{n_t}$, P_t is said to be *correct* if the concatenation of these paths is defined, if P_0 is a path from the state I and if all the paths C_i are elementary cycles. We can associate the couple $(|P_0| + \cdots + |P_t|, t)$

to such a sequence. We say that a couple (l,t) is associated with a word σ if this couple is associated with a correct sequence of paths labeled by a word σ' satisfying $F_\sigma = F_{\sigma'}$ where $F_{a_1 \dots a_n} = f_{a_n \dots a_1}$. The set of couples associated with a word σ is written $Ass_{\boldsymbol{L}}(\sigma)$.

Let us consider the lexicographic order $\leq_{lex}$ over $\mathbb{N} \times \mathbb{N}$ defined by:

$$(n_1, n_2) \leq_{lex} (m_1, m_2) \Longleftrightarrow (n_1 < m_1) \text{ or } (n_1 = m_1 \text{ and } n_2 \leq m_2)$$

Remark that every subset of $\mathbb{N} \times \mathbb{N}$ has a unique minimal element for $\leq_{lex}$. Therefore, the set $Ass_{\boldsymbol{L}}(\sigma)$ has a unique minimal element written (l,t). We are going to prove that this couple is bounded.

As $(l,t) \in Ass_{\boldsymbol{L}}(\sigma)$, there exists a correct sequence of paths P_0, $C_1^{n_1}$, P_1, ..., $C_t^{n_t}$, P_t associated with σ such that $l = |P_0| + \dots + |P_t|$.

Let us assume that there exists an index i such that $|P_i| > |\mathcal{M}_{\boldsymbol{L}}|$. Therefore the path P_i cannot be elementary and so the path P_i is equal to $P'.C.P''$ where C is an elementary cycle. In this case we can build a new correct sequence of paths associated with the couple $(l - |C|, t+1) <_{lex} (l,t)$ labeled by the same word which is in contradiction with the minimum property of (l,t). So, for every i, we have $|P_i| \leq |\mathcal{M}_{\boldsymbol{L}}|$.

Consider a path $I = M_0 \xrightarrow{w_1} M_1 \cdots \xrightarrow{w_t} M_t$ and let us define $F_{w_i}(x) = m_i.x + v_i$. We then have $M_{i-1}.m_i = M_i$. Remark that

$$\begin{aligned} F_{w_1 \dots w_t}(x) = F_{w_1} \circ \dots \circ F_{w_t}(x) &= m_1 \dots m_t.x + \sum_{i=1}^{t} m_1 \dots m_{i-1}.v_i \\ &= M_t.x + \sum_{i=1}^{t} M_{i-1}.F_{w_i}(0) \end{aligned}$$

Consider a correct sequence of paths P_0, C^n, P_1, C^m, P_2 labeled by a word w. The sequence of paths $P_0, C^{n+m}, P_1.P_2$ labeled by a word w' is also correct and moreover, the previous remark shows us that $F_w = F_{w'}$. Now, let us assume that $t > |EC(\boldsymbol{L})|$. In this case, there exist $i \neq j$ such that $C_i = C_j$. By reordering the cycles as previously seen, we obtain a new correct sequence associated with the couple $(l, t-1) <_{lex} (l,t)$ which is once again in contradiction with the minimal property of (l,t). We therefore obtain a contradiction. So we have proved that $t \leq |EC(\boldsymbol{L})|$.

Therefore, for every word σ, there exists $t \leq |EC(\boldsymbol{L})|$ words c_1, ...,c_t in $EC(\boldsymbol{L})$ (the labels of the paths C_i) and $t+1$ words w_0, ..., w_t in $\Sigma^{\leq |\mathcal{M}_L|}$ (the labels of the paths P_i) such that $f_\sigma \in f_{w_0} \circ f_{c_1}^* \circ f_{w_1} \circ \dots \circ f_{c_t}^* \circ f_{w_t}$ (the notation $R \in R'$ means for two binary relations R and R' that if sRs' then $sR's'$). Let us ordering the elements of $\Sigma^{\leq m} = \{\sigma_1, \dots \sigma_{|\Sigma^{\leq m}|}\}$ and let us consider $f_{\Sigma^{\leq m}} = f_{\sigma_1}^* \circ \dots \circ f_{\sigma_{|\Sigma^{\leq m}|}}^*$. Remark that f_{w_i} and $f_{c_i}^*$ are in $f_{\Sigma^{\leq m}}$. Therefore, for every $\sigma \in \Sigma^*$, we have $f_\sigma \in (f_{\Sigma^{\leq m}})^{|EC(\boldsymbol{L})|}$. □

This class of totally defined finite linear systems is therefore a really important one because it permits us to obtain for a finite linear system an over approximation of its reachability relation by considering the reachability relation of its naturally associated totally defined finite linear system.

6 Fast Acceleration of Symbolic Transition systems: application to broadcast protocols

FAST is a tool that takes as input a finite linear system $\boldsymbol{L}$ and a set of initial states S_0 and computes recursively the set $f^*_{\sigma_n} \circ \cdots \circ f^*_{\sigma_1}(S_0)$.

Even if the order of the word σ_i to compute the reachability set seems important, let us prove that it is not the case in a certain extent. An infinite sequence of words $(\sigma_n)_{n\geq 0}$ is called a *strategy* for computing a linear system $\boldsymbol{L}$ if every word $\sigma \in \{\sigma_n; n \in \mathbb{N}\}$ appears infinitely often in the sequence $(\sigma_n)_{n\geq 0}$: the set $\{n; \sigma_n = \sigma\}$ is infinite. Such a strategy is said to be *good* for $(\boldsymbol{L}, S_0)$ if there exists an integer $n \geq 0$ such that $f^*_{\sigma_n} \circ \cdots \circ f^*_{\sigma_1}(S_0)$ is equal to the reachability set of $(\boldsymbol{L}, S_0)$. Now remark that if $(\sigma_n)_{n\geq 0}$ is a good strategy for $(\boldsymbol{L}, S_0)$ and if (σ'_n) is an other strategy that uses the same words: $\{\sigma_n; n \geq 0\} = \{\sigma'_n; n \geq 0\}$ then $(\sigma'_n)_{n\geq 0}$ is also a good strategy. Therefore, the order of the words σ_i chosen by FAST is not so important.

However for optimizing the computation, FAST starts with the set of words $F = \Sigma^{\leq k}$ with $k = 1$ an attempt a strategy that use only words in F. If the computation grows up and becomes unmanageable, the integer k is increased and the computation restarts from the set of initial states (let us recall that the set of σ^*-accelerations with a length less than or equal to k is polynomial in k). This simple idea has given us good results in practice. In fact, the set of reachability sets of 8 cache coherence protocols have been computed with FAST in less than 10 seconds on an Intel Pentium 800 and the value of k is remained equal to $k = 1$. We have also used FAST on the SWIMMING POOL protocols [FO97b] and FAST stops after 3 minutes and the integer k has taken the value $k = 4$. Moreover, the automata which represent those reachability sets have a very small number of states. The following table gives precisely the obtained results. The specification of all those protocols can be found on our home page http://www.lsv.ens-cachan.fr/~leroux/fast/.

Technically, FAST is a tool programmed in $C++$ and freely available for download from http://www.lsv.ens-cachan.fr/~leroux/fast/ that uses the DFA library of MONA 1.4. This library has enable us to implement our ideas for finite linear systems such that the reachable states remain in $\mathbb{N}^m$. In fact, the DFA library of MONA lacks some functions over the manipulated automata to have an optimized and usable representation of subset of $\mathbb{Z}^m$. Even if this problem does not seem important because in practice protocols use states in $\mathbb{N}^m$, in fact it is not the case when we want to study the totally defined linear system associated with a finite linear system. A solution is just to replace the MONA library by the LASH library. This modification will be implemented as soon as possible.

Protocols	$\lvert\Sigma\rvert$	m	$\lvert A_{\phi_0}\rvert$	$\lvert A_{Reach}\rvert$	k
BERKELEY	3	4	4	7	1
DRAGON	8	5	4	7	1
FIREFLY	8	4	4	7	1
FUTURBUS	10	9	4	12	1
ILLINOIS	6	4	4	7	1
MESI	4	4	4	5	1
MOESI	4	5	4	7	1
SYNAPSE	3	3	4	6	1
SWIMMING POOL	6	9	6	29	4
PRODUCER-CONSUMER	3	5	3	6	1

Conclusion and perspectives

We have proved that we can compute acceleration of any sequence of actions in a finite linear system. Moreover, we have given a method to find out the "good accelerations" by reducing the number of different accelerations we have to consider. These methods are implemented in the tool `FAST`. Even if we can find out some finite linear systems that have a Presburger definable set of reachable states but such that `FAST` does not stop (see lemma 2 in Appendix), this tool enables to compute the set of reachable states of a great number of protocols. Finally, by considering the totally defined linear system associated to a finite linear system, we have proved that we can compute an over approximation of the set of reachable states.

Finally let us make a remark about the commutativity of the functions and the reduction of the number of "good accelerations". We have proved that if we have two Presburger-linear functions f and g such that $\bar{f} \circ \bar{g} = \bar{g} \circ \bar{f}$ then we can replace the set $\{f, g, f \circ g, g \circ f\}$ by $\{f, g, f \circ g \vee g \circ f\}$. However, sometimes we also have $f \circ g = g \circ f$. In this case, we can replace $\{f, g, f \circ g, g \circ f\}$ by $\{f, g\}$. Commutativity enables us to reduce the number of different accelerations. However, it seems that commutativity has not been used as best as possible.

References

[AAB99] P. Abdulla, A. Annichini, and A. Bouajjani. Symbolic verification of lossy channel systems: Application to the bounded retransmission protocol. In *(TACAS'99)*, volume 1579 of *LNCS*, pages 208–222, 1999.

[AAB00] A. Annichini, E. Asarin, and A. Bouajjani. Symbolic techniques for parametric reasoning about counter and clock systems. In *(CAV'00)*, volume 1855 of *LNCS*, pages 419–434, 2000.

[ABJ98] P. A. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. In *(CAV'98)*, volume 1427 of *LNCS*, pages 305–318, 1998.

[ABJN99] P. A. Abdulla, A. Bouajjani, B. Jonsson, and M. Nilsson. Handling global conditions in parameterized system verification. In *(CAV'99)*, volume 1633 of *LNCS*, pages 134–145, 1999.

[AF88] J-M. Arnaudiès and H. Fraysse. *Cours de mathématiques — 1 : Algèbre*. Dunod Université, 1988.

[AJ96] P. A. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Information and Computation*, 127(2):91–101, june 1996.

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model checking. In *(CONCUR'97)*, volume 1243 of *LNCS*, June 1997.

[BF99] B. Bérard and L. Fribourg. Reachability analysis of (timed) Petri nets using real arithmetic. In *(CONCUR'99)*, volume 1664 of *LNCS*, pages 178–193. Springer, 1999.

[BF00] J.-P. Bodeveix and M. Filali. Fmona: a tool for expressing validation techniques over infinite state systems. In *(TACAS'00)*, volume 1785 of *LNCS*, pages 204–219, march 2000.

[BGP97] T. Bultan, R. Gerber, and W. Pugh. Symbolic model checking of infinite state systems using presburger arithmetic. In *(CAV'97)*, volume 1254 of *LNCS*, pages 400–411, 1997.

[BGWW97] B. Boigelot, P. Godefroid, B. Willems, and P. Wolper. The power of QDDs. In *(SAS'97)*, volume 1302 of *LNCS*, pages 172–186, 1997.

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

[BJNT00] A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili. Regular model checking. In *(CAV'00)*, volume 1855 of *LNCS*, July 2000.

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

[BW94] B. Boigelot and P. Wolper. Symbolic verification with periodic sets. In *(CAV'94)*, volume 818 of *LNCS*, pages 55–67, 1994.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and presburger arithmetic. In *(CAV'98)*, volume 1427 of *LNCS*, pages 268–279, 1998.

[Del00a] G. Delzanno. Automatic verification of parameterized cache coherence protocols. In *(CAV'00)*, volume 1855 of *LNCS*, July 2000.

[Del00b] G. Delzanno. Verification of consistency protocols via infinite-state symbolic model checking: A case study. In *(FORTE/PSTV 2000)*, volume 1603 of *LNCS*, October 2000.

[DFS98] C. Dufourd, A. Finkel, and P. Schnoebelen. Reset nets between decidability and undecidability. In *(ICALP'98)*, volume 1443 of *LNCS*, pages 103–115, July 1998.

[EFM99] J. Esparza, A. Finkel, and R. Mayr. On the verification of broadcast protocols. In *(LICS'99)*, volume 1581 of *LNCS*, pages 352–359, July 1999.

[EN98] A. Emerson and K. Namjoshi. On model checking for non-deterministic infinite-state systems. In *(LICS'98)*, 1998.

[FAS] Fast acceleraction of symbolic transition systems: Fast. http://www.lsv.ens-cachan.fr/ leroux/fast/.

[FB95] R. Floyd and R. W Beigel. *Le langage des machines : introduction a la calculabilite et aux langages formels*, chapter 7, page 433. (ITP 1995), 1995.

[FO97a] L. Fribourg and H. Olsén. A decompositional approach for computing least fixed-points of Datalog programs with Z-counters. *Constraints*, 2(3/4):305–335, 1997.

[FO97b] L. Fribourg and H. Olsén. Proving safety properties of infinite state systems by compilation into Presburger arithmetic. In *(CONCUR'97)*, volume 1243 of *LNCS*, pages 213–227, 1997.

[FPS00] A. Finkel, S. Purushothaman Iyer, and G. Sutre. Well-abstracted transition systems. In *(CONCUR'2000)*, volume 1877 of *LNCS*, pages 566–580, 2000.

[FS00a] A. Finkel and G. Sutre. An algorithm constructing the semilinear post* for 2-dim reset/transfer vass. In *(MFCS'2000)*, volume 1893 of *LNCS*, pages 353–362, 2000.

[FS00b] A. Finkel and G. Sutre. Decidability of reachability problems for classes of two counters automata. In *(STACS'2000)*, volume 1770 of *LNCS*, pages 346–357, Feb 2000.

[FWW97] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems (extended abstract). In *(INFINITY'97)*, volume 9 of *Electronical Notes in Theoretical Computer Science. Elsevier Science*, July 1997.

[GS66] S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

[Jac78] G. Jacob. La finitude des representations lineaires des semi-groupes est decidable. *J. Algebra*, 52:437–459, 1978.

[LAS] Liège automata-based symbolic handler: Lash. http://www.montefiore.ulg.ac.be/ boigelot/research/lash/.

[MON] Monadic second-order logic: Mona. http://www.brics.dk/mona/index.html.

[MS77] A. Mandel and I. Simon. On finite semigroups of matrices. *Theoretical Computer Science*, 5(2):101–111, October 1977.

[MZ75] R. McNaughton and Y. Zalcstein. The burnside theorem for semi-groups. *J. Algebra*, 34:292–299, 1975.

[PS00] A. Pnueli and E. Shahar. Liveness and acceleration in parameterized verication. In *(CAV'00)*, volume 1855 of *LNCS*, pages 328–343, July 2000.

[Rev90] P. Z. Revesz. A closed form for Datalog queries with integer order. In *ICDT'90, Third International Conference on Database Theory*, volume 470 of *Lecture Notes in Computer Science*, pages 187–201, Paris, France, 12–14 December 1990. Springer.

[SFRC99] G. Sutre, A. Finkel, O. Roux, and F. Cassez. Effective recognizability and model checking of reactive fiffo automata. In *(AMAST'98)*, volume 1548 of *LNCS*, pages 106–123, 1999.

[Sut00] G. Sutre. *Abstraction et accélération de systèmes infinis*. PhD thesis, Ecole Normale Supérieure de Cachan, Laboratoire Spécification et Vérification. CNRS UMR 8643, october 2000.

[WB98] P. Wolper and B. Boigelot. Verifying systems with infinite but regular state spaces. In *(CAV'98)*, volume 1427, pages 88–97, June 1998.

7 Appendix

Lemma 1. . *Being given a matrix $M \in \mathcal{M}_m(\mathbb{Z})$, the two Boigelot's conditions are equivalent to the finiteness of the monoid $< M >$.*

Proof. First assume that there exists an integer $p > 0$ such that the eigenvalues of the matrix M^p belong to the set $\{0, 1\}$, and such that the matrix M^p is diagonalizable. We then have $M^{2.p} = M^p$ and the monoid $< M >$ is then finite.

Now, assume that the monoid $< M >$ is finite. There exist an integer $k \geq 0$ and an integer $\Delta > 0$ such that $M^{k+\Delta} = M^k$. Consider an integer $d > 0$ such that $d.\Delta \geq k$. Let $p = d.\Delta$. We have $M^{k+p} = M^k$. By multiplying the previous equality by M^{p-k} we obtain $M^{2p} = M^p$. Hence M^p is a root of the polynomial $P(X) = X(X-1)$. As M^p is a root of a polynomial of the form $(X-\lambda_1)...(X-\lambda_d)$ where the λ_i are distinct, the matrix M^p is therefore diagonalizable and its eigenvalues belong to the roots $\{\lambda_1, ..., \lambda_d\}$ (See [AF88]. Here the roots of $P(X) = X(X-1)$ are $\{0, 1\}$. □

Lemma 2. *There exists finite linear system with Presburger-definable reachability set such that no strategy using only cycle acceleration succeeds to compute the entire reachability set.*

Proof. For instance consider the finite linear system $\boldsymbol{L} = (\Sigma, 3, f_\Sigma)$ (which corresponds to a broadcast protocol with one broadcast sending a and one silent action b) where $\Sigma = \{a, b\}$ and $f_a(s) = M.s + (-1, 1, 0)$ and $f_b(s) = s + (0, -1, 1)$ where $\mathcal{G}_a = \{s; s_1 \geq 1\}$ and $\mathcal{G}_b = \{s; s_2 \geq 0\}$ and where the square matrix M is defined by:

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Consider the word $\sigma_n = ab^n a$ and a vector $s \in \mathbb{Z}^3$. We have $\bar{f_{\sigma_n}}(s) = \{(s_1 + s_2 + s_3 - n, n, 0)$.

We will say that a subset of $S \subseteq \mathbb{Z}^3$ is *weak-bounded* if the second and the third component of S are bounded: the set $\{s_2; \exists s \in S\}$ and $\{s_3; \exists s \in S\}$ are bounded.

Now consider a weak-bounded set S and an action $f(s) = M'.s + v$ defined over a definition domain D obtained by union reduction of the number of σ^*-accelerations. By definition, there exists a finite set F of words such that for every $\sigma \in F$ we have $\bar{f_\sigma} = \bar{f}$ and $D = \bigcup_{\sigma \in F} D_\sigma$.

Assume that $M' \neq I$. In this case, every word $\sigma \in F$ contains at least one letter a because $I \notin \{M^k; k \geq 1\} = \{M, M^2\}$. Consider such a word σ and remark that there exists an integer $n \geq 0$ such that $\sigma_n = a.b^n.a$ is a subword of $\sigma.\sigma$. Therefore the set $\bar{f_\sigma}^*(S)$ is weak-bounded. So the set $f^*(S) \subseteq \bar{f_\sigma}^*(S)$ is also weak-bounded.

Now, assume that $M' = I$. In this case, for the same reason, we have $F \subseteq \{b\}^*$. However $\bar{f_b}^n(s) = s + (0, -n, n)$ so if $\bar{f_b}^n = \bar{f_b}^{n'}$ then $n = n'$. That

means there exists an integer $n \geq 0$ such that $F = \{b^n\}$ and so $f = f_b^n$. As $\mathbf{G}_{b^n} = \{s; s_2 \geq n\}$ and S is weak-bounded, the set $f^*(S)$ remains weak-bounded.

So, every cycle-acceleration strategy only computes weak-bounded set of states from a weak bounded set of initial states.

Now, let us consider the set $S_0 = \{(n, 0, 0)\}$ as set of initial states and remark that the word $\sigma'_n = abab^2ab^3 \dots ab^n$ is such that $f_{\sigma'_n}(n, 0, 0)$ is defined and equal to $(0, 0, n)$. Therefore, the set of reachable states is not weak-bounded whereas the set of initial states S_0 is weak-bounded.

Moreover, as $f_{\sigma'_{n+k-1}.ab^k}(n + k + l, 0, 0)$ is defined and equal to (l, n, k) for any $l, n, k \geq 0$, we have $Reach(\boldsymbol{L}, S_0) = \mathbb{N}^3$ which is Presburger-definable. □