An algorithm constructing the semilinear post* for 2-dim Reset/Transfer VASS (Extended abstract)

A. Finkel, G. Sutre

LSV, ENS Cachan & CNRS UMR 8643, France.
{finkel, sutre}@lsv.ens-cachan.fr

Abstract. The main result of this paper is a proof that the reachability set (post*) of any 2-dim Reset/Transfer Vector Addition System with States is an effectively computable semilinear set. Our result implies that numerous boundedness and reachability properties are decidable for this class. Since the traditional Karp and Miller's algorithm does not terminate when applied to 2-dim Reset/Transfer VASS, we introduce, as a tool for the proof, a new technique to construct a finite coverability tree for this class.

1 Introduction

Context. Model checking consists in verifying that a model — usually a transition system — satisfies a property (usually a temporal logic formula). It has become very popular because it is fully automatic when the transition system has a finite state space. However, many programs cannot be modelled by a finite-state transition system, for instance communication protocols for which the size of channels is not known in advance, distributed systems parametrized by the number of processes. The success of model checking for finite-state transition systems has suggested that an efficient verification technology could be developped for infinite-state transition systems as well.

Model checking infinite-state transition systems often reduces to effective computation of two potentially infinite sets: the set of predecessors pre* and the set of successors post*. In order to effectively compute post* (resp. pre*), one generally needs to find a class $\mathcal{C}$ of (finitely describable) infinite sets which has the following good properties: (1) closure under union, (2) closure under post (resp. pre) and (3) membership and inclusion are decidable with an elementary complexity.

We focus in this paper on programs with integer variables, or more precisely on several kinds of counter automata. Basically, counters hold nonnegative integer values and they can be modified through the following operations: increment (+1), decrement (-1), zero-test, reset and transfer (from one counter to another). Hence the infinite state space is $Q \times \mathbb{N}^k$, where Q is the set of control states. It is well known that 2 counters automata with increment, decrement and zero-test are sufficient to simulate any Turing machine. Thus, post* and pre* may be nonrecursive for this class.

Upward closed sets enjoy the three good required properties for the class of lossy counter automata and hence they are particularly well suited to the computation of post* and pre* for this class [BM99]. However, upward closed sets do not allow to exactly represent pre* and post* for general (nonlossy) counter automata. The class of semilinear sets (which contains upward closed sets) enjoy the three good required properties for general counter automata. However, even for 3 counters automata with only increment and decrement, post* and pre* may be nonsemilinear. Hence, our objective is to find classes of 2-counters automata for which post* and pre* are effectively computable semilinear sets.

Related work. Vector Addition Systems with States (VASS) (or equivalently Petri nets) correspond to the class of counter automata with increment and decrement. For the class of VASS, post* and pre* are recursive [May84, Kos82]. Hopcroft and Pansiot proved that for 2-dim VASS (i.e. VASS with 2 counters), post* — and hence pre* — are semilinear and constructible[1]. However, for 3-dim VASS, post* and pre* may be nonsemilinear [HP79].

Some papers recently appeared dealing with verification of n-dim Reset/Transfer VASS (VASS extended with resets and transfers), 2-dim VASS extended with zero-tests on one counter and Reset VASS with lossy counters:

counters	**operations**	**post***	**pre***
3	+1, -1 and resets	nonrecursive [DFS98]	nonrecursive [DFS98]
3 lossy	+1, -1 and resets	recursive, semilinear, non constructible [BM99]	semilinear, constructible [BM99]
2	+1, -1 and zero-tests on one counter	semilinear, constructible [FS00a]	semilinear, constructible [FS00a]
2	+1, -1, resets and transfers	recursive, semilinear [FS00a], **constructible** [this paper]	semilinear, constructible [FS00a]

We already know from [FS00a] that pre* is semilinear and constructible for 2-dim Reset/Transfer VASS, and hence post* is recursive ; we also know that post* is semilinear for this class. However, this does not imply that post* is constructible since for 3-dim Lossy Reset VASS, pre* is a constructible semilinear set whereas post* is a *non-constructible* semilinear set.

Our contribution. Our main result is a technically nontrivial proof that post* is semilinear and constructible for 2-dim Reset/Transfer VASS. The proof is achieved with the following four main steps.

1. We first prove that it is possible to compute the finite minimal coverability set of a 2-dim Reset VASS by using an extended Karp and Miller's algorithm and a recent technical result of Jančar [DJS99]. Notice that the traditional Karp and Miller's algorithm does not terminate for 2-dim Reset VASS [DFS98].

[1] We say that a semilinear set L is *constructible* when a Presburger formula for L is computable.

2. Then we obtain, by simulation, the minimal coverability set of a 2-dim Reset/Transfer VASS.
3. We show that any 2-dim Reset/Transfer VASS may be simulated by a 2-dim Extended VASS in the class $\mathbf{T}_1\mathbf{R}_2$ (i.e. VASS with the capacity of zero-testing on the first counter and reset on the second counter) preserving post* (and also the same minimal coverability set).
4. We deduce from [FS00a] that we can build an algorithm for constructing the semilinear post* of a 2-dim Reset/Transfer VASS.

We can deduce from this result that, for instance, reachability, reachability equivalence, boundedness and counter-boundedness are all decidable for the class of 2-dim Reset/Transfer VASS.

The proofs are technically non trivial and are omitted for space reasons. A complete presentation of these results can be found in the full paper [FS00b].

2 Reset/Transfer Vector Addition Systems with States

We write $[i\,..\,j]$ for the set $\{k \in \mathbb{N} \ / \ i \leq k \leq j\}$. If $x \in X^2$ and $\gamma \in \{1,2\}$, we denote by $x(\gamma)$ the γ^{th} *component* of x. If $\gamma \in \{1,2\}$, we write $\overline{\gamma}$ for the unique element of $\{1,2\} \setminus \{\gamma\}$.

Let $\mathbb{N}_\omega$ denote the classical completion $\mathbb{N} \cup \{\omega\}$ of the set $\mathbb{N}$, where $n < \omega$ for all $n \in \mathbb{N}$. Operations are extended on $\mathbb{N}_\omega$ as follows: for every $z \in \mathbb{Z}$, we define $z + \omega = \omega + z = \omega$; and for every $x \in \mathbb{N}_\omega \setminus \{0\}$, we define $x \cdot \omega = \omega \cdot x = \omega$ and $0 \cdot \omega = \omega \cdot 0 = 0$. For every infinite nondecreasing sequence $(a_i)_{i \in \mathbb{N}}$ in $\mathbb{N}$, we define

$$\mathrm{lub}(a_i) = \begin{cases} \omega & \text{if } \{a_i \ / \ i \in \mathbb{N}\} \text{ is infinite} \\ \mathrm{Max}(\{a_i \ / \ i \in \mathbb{N}\}) & \text{otherwise} \end{cases}$$

Operations $+$, $\cdot$ and lub on $\mathbb{N}^2_\omega$ are componentwise extensions of the respective operations on $\mathbb{N}_\omega$ and we also extend $\leq$ on $\mathbb{N}^2_\omega$ (and $\mathbb{Z}^2$) as expected, with $v \leq v'$ if for all $\gamma \in \{1,2\}$ we have $v(\gamma) \leq v'(\gamma)$. These two relations (written $\leq$) on $\mathbb{N}^2_\omega$ and $\mathbb{Z}^2$ are orderings. We write $z < z'$ when $z \leq z'$ and $z \neq z'$.

A *labelled transition system* is a structure $\mathcal{LTS} = (S, A, \rightarrow)$ where S is a set of *states*, A is a finite set of *actions* and $\rightarrow \subseteq S \times A \times S$ is a set of *transitions*. When S is finite, we say that $\mathcal{LTS}$ is a *finite* labelled transition system. A (finite) *path* π in a labelled transition system is any sequence of transitions $\pi = (s_0 \xrightarrow{\alpha_0} s'_0), (s_1 \xrightarrow{\alpha_1} s'_1), \ldots, (s_k \xrightarrow{\alpha_k} s'_k)$ such that $s'_i = s_{i+1}$ for every $i \in [0\,..\,k-1]$ and π is shortly written $\pi = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \cdots s_k \xrightarrow{\alpha_k} s'_k$; moreover, we say that π is a *path from* s_0 *to* s'_k. We write $s \xrightarrow{*} s'$ when there exists a path from s to s'. For every subset $R \subseteq S$, we write $\mathrm{post}^*(\mathcal{LTS}, R)$ (shortly for $\mathrm{post}^*(R)$) for the set $\{s \in S \ / \ \exists\, r \in R, r \xrightarrow{*} s\}$ of *successors* of R.

We present 2-dim Reset/Transfer Vector Addition Systems with States in two steps : we first describe the *finite control structure* (noted A) of a 2-dim Reset/Transfer VASS and we then define a 2-dim Reset/Transfer VASS (noted

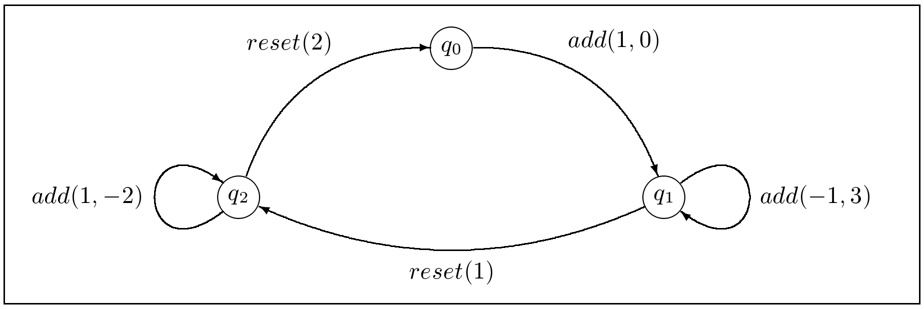

Fig. 1. A 2-dim Reset VASS

$\mathcal{A}$) as an *operational semantics* associated with such a control structure, which leads to a (potentially) infinite labelled transition system.

A 2-*dim Reset/Transfer VASS Control* is a finite labelled transition system $\mathsf{A} = (Q, Op, \rightarrow_{\mathsf{A}})$ such that $Op \subseteq \{add(z) \ / \ z \in \mathbb{Z}^2\} \cup \{reset(1), reset(2)\} \cup \{transfer(1 \rightarrow 2), transfer(2 \rightarrow 1)\}$.

Definition 2.1. *A* 2-dim Reset/Transfer VASS *is a labelled transition system $\mathcal{A} = (S, Op, \rightarrow_{\mathcal{A}})$ based on a 2-dim Reset/Transfer VASS Control $\mathsf{A} = (Q, Op, \rightarrow_{\mathsf{A}})$, where $S = Q \times \mathbb{N}^2$ is the set of states, Op is the set of actions, and the set of transitions $\rightarrow_{\mathcal{A}}$ is the smallest subset of $S \times Op \times S$ verifying the three following conditions:*

1. *if $q \xrightarrow{add(z)}_{\mathsf{A}} q'$ then for all $a \in \mathbb{N}^2$ such that $a + z \geq 0$, $(q, a) \xrightarrow{add(z)}_{\mathcal{A}} (q', a + z)$*
2. *if $q \xrightarrow{reset(\gamma)}_{\mathsf{A}} q'$ then for all $a \in \mathbb{N}^2$, $(q, a) \xrightarrow{reset(\gamma)}_{\mathcal{A}} (q', a')$ where $a'(\gamma) = 0$ and $a'(\overline{\gamma}) = a(\overline{\gamma})$*
3. *if $q \xrightarrow{transfer(\gamma \rightarrow \overline{\gamma})}_{\mathsf{A}} q'$ then for all $a \in \mathbb{N}^2$, $(q, a) \xrightarrow{transfer(\gamma \rightarrow \overline{\gamma})}_{\mathcal{A}} (q', a')$ where $a'(\gamma) = 0$ and $a'(\overline{\gamma}) = a(\gamma) + a(\overline{\gamma})$.*

Let $\mathcal{A}$ be a 2-dim Reset/Transfer VASS. States of A are called *control states* of $\mathcal{A}$ and transitions of A are called *control transitions* of $\mathcal{A}$. Every path $\pi = (q_0, a_0) \xrightarrow{\alpha_0} (q_1, a_1) \xrightarrow{\alpha_1} (q_2, a_2) \cdots (q_k, a_k) \xrightarrow{\alpha_k} (q_{k+1}, a_{k+1})$ in $\mathcal{A}$ is also shortly written $\pi = (q_0, a_0) \xrightarrow{\sigma} (q_{k+1}, a_{k+1})$ where $\sigma = q_0 \xrightarrow{\alpha_0} q_1 \xrightarrow{\alpha_1} q_2 \cdots q_k \xrightarrow{\alpha_k} q_{k+1}$ (notice that σ is a path in A).

Definition 2.2. *A* 2-dim Reset VASS *(resp.* 2-dim VASS*) is any 2-dim Reset/ Transfer VASS $\mathcal{A}$ whose control A contains no transfer transition (resp. no transfer transition and no reset transition).*

3 Computation of the Minimal Coverability Set

We show in this section that the minimal coverability set of a 2-dim Reset/Transfer VASS is computable.

Let $\mathcal{A}$ be a 2-dim Reset/Transfer VASS with a set Q of control states and let (q_0, a_0) be an initial state of $\mathcal{A}$. We extend $\leq$ on $Q \times \mathbb{N}_\omega^2$ (and it is still an ordering) by $(q, v) \leq (q', v')$ if $q = q'$ and $v \leq v'$.

Definition 3.1. *A* coverability set *of $(\mathcal{A}, (q_0, a_0))$ is any set $CS \subseteq Q \times \mathbb{N}_\omega^2$ satisfying the two following conditions:*

1. *for every $(q, a) \in \mathrm{post}^*(q_0, a_0)$, there exists $(q', v') \in CS$ such that $(q, a) \leq (q', v')$*
2. *for every $(q, v) \in CS$, there exists an infinite nondecreasing sequence $(q, b_i)_{i \in \mathbb{N}}$ in $\mathrm{post}^*(q_0, a_0)$ such that $v = \mathrm{lub}(b_i)$*

A coverability set CS is minimal *if no proper subset of CS is a coverability set.*

Notice that a minimal coverability set does not contain two comparable elements. The following theorem is an obvious consequence of the fact that 2-dim Reset/Transfer VASS are Well Structured Transition Systems [Fin90].

Theorem 3.2 ([Fin90]). *Any 2-dim Reset/Transfer VASS $\mathcal{A}$ with an initial state (q_0, a_0) has a unique minimal coverability set, noted* $\mathrm{MCS}(\mathcal{A}, (q_0, a_0))$ *(shortly* $\mathrm{MCS}(q_0, a_0)$*), which is finite and satisfies* $\mathrm{MCS}(q_0, a_0) = \mathrm{Max}(CS)$ *for any finite coverability set CS.*

3.1 Parametrized States and Path Schemes

We now focus on 2-dim Reset VASS and we present in this section parametrized states and path schemes. This whole section is higly inspired from [DJS99]. We have slightly adapted several definitions to our context.

For every $n \in \mathbb{N}$ and for every states $(q, a), (q', a')$ of a 2-dim Reset VASS $\mathcal{A}$, we write $(q, a) \xrightarrow{*}_n (q', a')$ when there exists a path from (q, a) to (q', a') containing at most n reset transitions. We also write $(q, a) \xrightarrow{*\, reset(\gamma)}_n (q', a')$ when there exists a path ending with a $reset(\gamma)$ from (q, a) to (q', a') and containing at most n reset transitions.

A *parametrized state* $s = \langle q, \gamma \rangle$ is a control state q equipped with a *position* $\gamma \in \{1, 2\}$. For every $x \in \mathbb{N}$, we write $s(x)$ for the state (q, γ_x) of $\mathcal{A}$, where $\gamma_x = (x, 0)$ if $\gamma = 1$ and $\gamma_x = (0, x)$ if $\gamma = 2$.

Definition 3.3. *A* path scheme of order n *(shortly a* path scheme*) related to a parametrized state $s = \langle q, \gamma \rangle$ is a 3-tuple (σ, f, x_0), where $\sigma : \mathbb{N} \to (\to_{\mathsf{A}})^*$, $f : \mathbb{N} \to \mathbb{N}$ and $x_0 \in \mathbb{N}$, satisfying for every $x \geq x_0$ the three following conditions:*

1. *we have $s(x) \xrightarrow{\sigma(x)} s(f(x))$, and,*
2. *$\sigma(x)$ contains at most n reset control transitions, and,*
3. *$\sigma(x)$ ends with a reset control transition $p \xrightarrow{reset(\gamma)}_{\mathsf{A}} q$.*

A path scheme (σ, f, x_0) is maximal *if for every $x \geq x_0$ there is no $y > f(x)$ such that $s(x) \xrightarrow{*\, reset(\gamma)}_n s(y)$.*

Observe that the *definition domains* of f and σ, $\mathrm{Dom}(f)$ and $\mathrm{Dom}(\sigma)$ respectively, necessarily contain the set $\{x \in \mathbb{N} \ / \ x \geq x_0\}$.

Definition 3.4. *A parametrized state* $s = \langle q, \gamma \rangle$ *is* n-sensible *if there exists* $x_0 \in \mathbb{N}$ *such that* $\{y \in \mathbb{N} \ / \ s(x) \xrightarrow{*\, reset(\gamma)}_n s(y)\}$ *is finite for every* $x \geq x_0$.

Notice that a parametrized state s is n-sensible if and only if there exists a maximal path scheme of order n related to s. We will use in the following the notion of *regular* path schemes, first introduced in [DJS99].

Definition 3.5 ([DJS99]). *A function* $f : \mathbb{N} \to \mathbb{N}$ *is* d-regular*, where* $d \in \mathbb{N}$*, if there exists* $\rho \in \mathbb{Q}^+$ *and a function* $h : [0\,..\,d-1] \to \mathbb{Q}$ *such that for every* $x \in \mathrm{Dom}(f)$*,* $f(x) = \rho x + h(x \mod d)$*.*

Definition 3.6 ([DJS99]). *A path scheme* (σ, f, x_0) *is* regular *if* f *is* d*-regular for some* d*, and for each* $i \in [0\,..\,d{-}1]$ *there exist* $m \in \mathbb{N}$*,* $u_1, v_1, u_2, v_2, \ldots, u_m, v_m, u_{m+1} \in (\to_{\mathsf{A}})^*$ *and* m d*-regular functions* $g_1, g_2, \ldots, g_m$ *such that for every* $x \geq x_0$*, if* $x \mod d = i$ *then we have* $\sigma(x) = u_1 v_1^{g_1(x)} u_2 v_2^{g_2(x)} \cdots u_m v_m^{g_m(x)} u_{m+1}$*.*

According to [DJS99], the set $RPS(s)$ of regular path schemes related to a parametrized state s is recursive. Hence, for every parametrized state s, one may enumerate $RPS(s)$. We can easily derive from [DJS99] that:

Lemma 3.7. *Let* s *be a parametrized state and let* $n \in \mathbb{N}$*. There exists a maximal regular path scheme of order* n *related to* s *if and only if* s *is* n*-sensible.*

3.2 Coverability Tree for 2-dim Reset VASS

Definition 3.8. *An* unbounded parametrized state *is a pair* (s, x_0) *where* s *is a parametrized state,* $x_0 \in \mathbb{N}$ *and such that there exists a regular path scheme* (σ, f, x_0) *related to* s *satisfying* $\forall\, x \geq x_0,\ f(x) > x$*.*

Note that for every d-regular function f, given by $f(x) = \rho x + h(x \mod d)$, the property $\forall\, x \geq x_0,\ f(x) > x$ is decidable.

We start by showing that the minimal coverability set of a 2-dim Reset VASS is computable. In order to compute a coverability set for a 2-dim Reset VASS, we run *several* algorithms *in parallel*, as given by the `MinimalCoverabilitySet` algorithm. First, using the `RegularPathSchemes` algorithm, we enumerate regular path schemes in order to find unbounded parametrized states. At the same time, we develop a coverability tree using Karp and Miller's strategy and using also the list of (currently discovered) unbounded parametrized states (see the `CoverabilityTree` algorithm).

The `CoverabilityTree` algorithm constructs a tree in a breadth first manner (see line 3 of the algorithm). Lines 9–12 of the algorithm come from the following observation:

- if (s, x_0) is an unbounded parametrized state then we have $\mathrm{lub}((f^i(x))_{i \in \mathbb{N}}) = \omega$, and
- moreover, for every $x \geq x_0$, if $s(x) \in \mathrm{post}^*(q_0, a_0)$ then for every $i \in \mathbb{N}$, $s(f^i(x)) \in \mathrm{post}^*(q_0, a_0)$.

Algorithm 1 `RegularPathSchemes`$(\mathcal{A}, s, \texttt{var}\, L)$

Input: a 2-dim Reset VASS $\mathcal{A}$, a parametrized state s and a list L of unbounded parametrized states

1: **for each** regular path scheme (σ, f, x_0) related to s **do**
2:   **if** $\forall\, x \geq x_0,\ f(x) > x$ **then**
3:     add (s, x_0) to L
4:     **return**

Algorithm 2 `CoverabilityTree`$(\mathcal{A}, (q_0, a_0), \texttt{var}\, L)$

Input: a 2-dim Reset VASS $\mathcal{A}$, an initial state (q_0, a_0) and a list L of unbounded parametrized states

1: create root labelled (q_0, a_0)
2: **while** there are unmarked nodes **do**
3:   pick an unmarked node $t : (q, v)$ among those of smallest depth
4:   **if** there is a marked node $t' : (q, v')$ such that $v' \geq v$ **then**
5:     skip {goto 18}
6:   **else**
7:     **if** there is an ancestor $t' : (q, v')$ of t such that $v \geq v'$, and $\omega > v(\gamma) > v'(\gamma)$ for some $\gamma \in \{1, 2\}$, and there is no reset between t' and t **then**
8:       $v \leftarrow v + \omega \cdot (v - v')$
9:     **if** $v \in \mathbb{N} \times \{0\}$ and there exists $x_0 \leq v(1)$ such that $((q, (\cdot\,, 0)), x_0) \in L$ **then**
10:       $v \leftarrow (\omega, 0)$
11:     **if** $v \in \{0\} \times \mathbb{N}$ and there exists $x_0 \leq v(2)$ such that $((q, (0, \cdot\,)), x_0) \in L$ **then**
12:       $v \leftarrow (0, \omega)$
13:     {We now create sons for the immediate successors of (q, v)}
14:     **for each** transition $q \xrightarrow{add(z)} q'$ in A such that $v + z \geq 0$ **do**
15:       construct a son $t' : (q', v + z)$ of t and label the arc $t \xRightarrow{add(z)} t'$
16:     **for each** transition $q \xrightarrow{reset(\gamma)} q'$ in A **do**
17:       construct a son $t' : (q', v')$ of t, where $v'(\gamma) = 0$ and $v'(\overline{\gamma}) = v(\overline{\gamma})$ and label the arc $t \xRightarrow{reset(\gamma)} t'$
18:   mark t

Algorithm 3 `MinimalCoverabilitySet`$(\mathcal{A}, (q_0, a_0))$

Input: a 2-dim Reset VASS $\mathcal{A}$ and an initial state (q_0, a_0)

1: $L \leftarrow \emptyset$
2: **for each** parametrized state s **do**
3:   {We start a new process enumerating regular path schemes related to s}
4:   **start** `RegularPathSchemes`$(\mathcal{A}, s, \texttt{var}\, L)$
5: {We run in parallel the `CoverabilityTree` algorithm}
6: **start** `CoverabilityTree`$(\mathcal{A}, (q_0, a_0), \texttt{var}\, L)$
7: **wait until** `CoverabilityTree` terminates
8: **stop** all the processes `RegularPathSchemes` started at line 4
9: $CS \leftarrow \{(q, v)\ /\ t : (q, v)$ is a node in the tree constructed by `CoverabilityTree`$\}$
10: **return** $\mathrm{Max}(CS)$

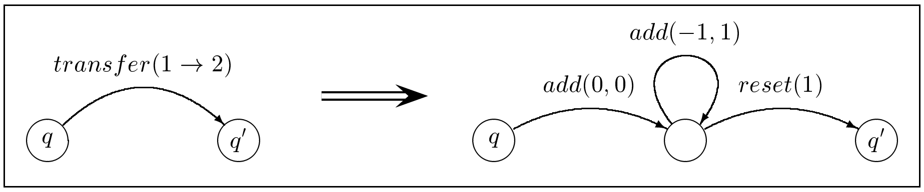

Fig. 2. Weak simulation of $transfer(1 \to 2)$ transitions by $reset(1)$ and add transitions

Intuitively, in order to show that the developped coverability tree is finite, we show that if there was an infinite branch in the tree, then an unbounded parametrized state should have been discovered (at some time) and, hence, the `CoverabilityTree` algorithm should have used this unbounded parametrized state to finish the branch, a contradiction. Thus, we get:

Proposition 3.9. *For any 2-dim Reset VASS $\mathcal{A}$ with an initial state (q_0, a_0), the* `MinimalCoverabilitySet` *algorithm applied to $(\mathcal{A}, (q_0, a_0))$ terminates and we have:*

$$\texttt{MinimalCoverabilitySet}(\mathcal{A}, (q_0, a_0)) = \mathrm{MCS}(q_0, a_0)$$

Now, given a 2-dim Reset/Transfer VASS $\mathcal{A}$, we construct a 2-dim Reset VASS $\mathcal{A}'$ obtained from $\mathcal{A}$ using the "weak simulation" of transfer transitions by reset transitions (and add transitions) given on Figure 2 (the "weak simulation" of $transfer(2 \to 1)$ transitions by $reset(2)$ and add transitions is obtained by symmetry). Observe that this simulation preserves the minimal coverability set (but not the set of successors). Hence, we obtain:

Theorem 3.10. *For any 2-dim Reset/Transfer VASS $\mathcal{A}$ with an initial state (q_0, a_0), the minimal coverability set* $\mathrm{MCS}(q_0, a_0)$ *is computable.*

Example 3.11. The minimal coverability set of the 2-dim Reset VASS of Figure 1 with initial state $(q_0, (0,0))$ is $\{(q_0, (\omega, 0)), (q_1, (\omega, \omega)), (q_2, (\omega, \omega))\}$.

4 Computation of post*

We now focus on the computation of post^* for 2-dim Reset/Transfer VASS. We already proved in [FS00a] that for any 2-dim Reset/Transfer VASS $\mathcal{A}$ with a semilinear set S_0 of initial states, $\mathrm{post}^*(S_0)$ is semilinear. However, the effective computability of post^* for this class was left an open problem in [FS00a] and we show in this section that post^* is constructible.

We first define semilinear sets. For any finite subset $P = \{p_1, p_2, \cdots, p_k\}$ of $\mathbb{N}^2$ and for any $b \in \mathbb{N}^2$ we define $(b + P^*) = \{b + \Sigma_{i=1}^{k} x_i p_i \ / \ \forall\, i \in [1,k], x_i \in \mathbb{N}\}$ and $(b + P^*)$ is called a *linear* set. A *semilinear* set is a finite union of linear sets. We will say that a semilinear set L is *constructible* if there exists an algorithm which computes a finite family $\{(b_i, P_i) \ / \ i \in I\}$ such that $L = \bigcup_{i \in I}(b_i + {P_i}^*)$. If $B \subseteq \mathbb{N}^2$, we shortly write $(B + P^*)$ for $\bigcup_{b \in B}(b + P^*)$.

We will also need the following definitions taken from [FS00a]. The class $\mathbf{T}_1\mathbf{R}_2$ of 2-*dim Extended VASS* is the set of 2-dim VASS extended with $test(1)$ and $reset(2)$ transitions. The operational semantics associated with a $test(1)$ transition, which actually represents a zero-test on counter 1, is as follows:

$$\text{if } q \xrightarrow{test(1)}_{\mathsf{A}} q' \text{ then for all } a \in \mathbb{N}^2, \text{ we have } (q, a) \xrightarrow{test(1)}_{\mathcal{A}} (q', a) \text{ iff } a(1) = 0$$

Let $\mathcal{A}$ be a 2-dim Extended VASS in the class $\mathbf{T}_1\mathbf{R}_2$ with an initial state (q_0, a_0). We define the set $\Omega(\mathcal{A}, (q_0, a_0))$ as the set of control states q such that $\{x \in \mathbb{N} \ / \ \exists\, q', \ (q_0, a_0) \xrightarrow{*} (q', (0, x)) \xrightarrow{test(1)} (q, (0, x))\}$ is infinite. Recall that:

Theorem 4.1 ([FS00a]). *For any* 2*-dim Extended VASS* $\mathcal{A}$ *in the class* $\mathbf{T}_1\mathbf{R}_2$ *with an initial state* (q_0, a_0)*,* $\text{post}^*(q_0, a_0)$ *is semilinear. Moreover, if* $\Omega(\mathcal{A}, (q_0, a_0))$ *is effectively computable, then* $\text{post}^*(q_0, a_0)$ *is constructible.*

Any 2-dim Reset/Transfer VASS $\mathcal{A}$ can be simulated by a 2-dim Extended VASS $\mathcal{A}'$ in the class $\mathbf{T}_1\mathbf{R}_2$, and $\Omega(\mathcal{A}', (q_0, a_0))$ may be computed from $\text{MCS}(\mathcal{A}, (q_0, a_0))$. Thus, we get:

Theorem 4.2. *For any* 2*-dim Reset/Transfer VASS* $\mathcal{A}$ *with a semilinear set* S_0 *of initial states,* $\text{post}^*(S_0)$ *is semilinear and it is constructible.*

Example 4.3. For the 2-dim Reset VASS of Figure 1, the set of successors of $(q_0, (0, 0))$ is:

$$\begin{aligned} \text{post}^*(q_0, (0,0)) = \quad & q_0 \times ((0,0) + \{(1,0)\}^*) \\ & \bigcup q_1 \times (\{(1,0), (0,3)\} + \{(1,0), (0,3)\}^*) \\ & \bigcup q_2 \times (\{(0,0), (1,1), (2,2)\} + \{(3,0), (0,3)\}^*) \end{aligned}$$

5 Conclusion

We have given an algorithm which computes the semilinear set post^* of any 2-dim Reset/Transfer VASS. This result completes those obtained for pre^* in [FS00a]. Recall that for 3-dim Lossy Reset VASS, pre^* is a constructible semilinear set whereas post^* is a *non-constructible* semilinear set (even if it is recursive). In other words, since $post^*$ was already known to be semilinear for 2-dim Reset/Transfer VASS [FS00a], we have proved that the following language is recursive, whereas it is not for 3-dim Lossy Reset VASS:

$$\begin{aligned} \{(\mathsf{A}, L) \ / \ & \text{post}^*(\mathcal{A}, s_0) = L \text{ where } \mathcal{A} \text{ is a 2-dim Reset/Transfer VASS} \\ & \text{with initial state } (q_0, a_0) \text{ and} L \text{ is a semilinear set}\} \end{aligned}$$

Our result is tight since it cannot be generalized to the extended class of 2-dim Reset/Transfer VASS where one adds zero-tests on one counter. We know from [FS00a] that post^* is not even recursive for this extended class.

This result implies that numerous boundedness and reachability properties are decidable for this class. For instance, the computation of post* allows to decide the following problems (which cannot be solved by only using pre*): boundedness, counter-boundedness, containment of a semilinear set in post*, inclusion and equivalence between two post* sets.

Because 2-dim Reset/Transfer VASS are closed under intersection with finite automata, we obtain that LTL on finite sequences is decidable for this class. Since pre* is also constructible [FS00a], we get that the existential fragment $\{\mathrm{EF}, \mathrm{EX}\}$ of CTL is decidable for 2-dim Reset/Transfer VASS.

Let us remark that line 2 in Algorithm 3 enumerates all regular path schemes related to a parameterized state s. This lead to a non-elementary complexity in the `MinimalCoverabilitySet` algorithm. We don't know whether it is possible to compute a minimal coverability set with an elementary complexity.

Our algorithm may also allow us to develop a strategy to over-approximate the reachability set of any two counters automaton $\mathcal{A}$ which is able to increase, decrease, zero-test, reset and transfer the 2 counters (recall the reachability set post$^*(\mathcal{A})$ is not in general recursive). This strategy, which allows to test safety properties, can be generalized to *n-dim Extended VASS* and to the pre* *operator*.

References

[BM99] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In *Proc. 16th Ann. Symp. Theoretical Aspects of Computer Science (STACS'99), Trier, Germany, Mar. 1999*, volume 1563 of *Lecture Notes in Computer Science*, pages 323–333. Springer, 1999.

[DFS98] C. Dufourd, A. Finkel, and Ph. 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.

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

[Fin90] A. Finkel. Reduction and covering of infinite reachability trees. *Information and Computation*, 89(2):144–179, 1990.

[FS00a] A. Finkel and G. Sutre. Decidability of reachability problems for classes of two counters automata. In *Proc. 17th Ann. Symp. Theoretical Aspects of Computer Science (STACS'2000), Lille, France, Feb. 2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 346–357. Springer, 2000.

[FS00b] A. Finkel and G. Sutre. Verification of two counters automata. Research Report LSV-2000-7, Lab. Specification and Verification, ENS de Cachan, Cachan, France, June 2000.

[HP79] J. Hopcroft and J. J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.

[Kos82] S. R. Kosaraju. Decidability of reachability in vector addition systems. In *Proc. 14th ACM Symp. Theory of Computing (STOC'82), San Francisco, CA, May 1982*, pages 267–281, 1982.

[May84] E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.