Decidability of reachability problems for classes of two counters automata

A. Finkel, G. Sutre

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

Abstract. We present a global and comprehensive view of the properties of subclasses of two counters automata for which counters are only accessed through the following operations: increment $(+1)$, decrement (-1), reset $(c := 0)$, transfer (the whole content of counter c is transfered into counter c'), and testing for zero. We first extend Hopcroft-Pansiot's result (an algorithm for computing a finite description of the semilinear set $post^*$) to two counters automata with only one test for zero (and one reset and one transfer operations). Then, we prove the semilinearity and the computability of pre^* for the subclass of 2 counters automata with one test for zero on c_1, two reset operations and one transfer from c_1 to c_2. By proving simulations between subclasses, we show that this subclass is the maximal class for which pre^* is semilinear and effectively computable. All the (effective) semilinearity results are obtained with the help of a new symbolic reachability tree algorithm for counter automata using an `Acceleration` function. When `Acceleration` has the so-called stability property, the constructed tree computes exactly the reachability set.

1 Introduction

Context. The highly successful model-checking approach for finite systems is mainly a consequence that finite state automata enjoy a lot of good properties like strong relations with logics (first order and monadic second order, linear temporal logic (LTL)), decidability and complexity results, efficient algorithmics, closure properties, etc. Thus, finite automata are a nice framework for the model-checking of programs which operate on bounded and finite variables. But the knowledge of the domains of variables is sometimes impossible to compute and this is still more true when the domain is infinite.

Before we survey the decidability results about extended automata, we give a precise view of the properties we are interested in. One of the most important class of properties is called safety properties. Model-checking safety properties often reduces to the effective computation of the set of predecessors (pre^*) or/and the set of successors ($post^*$). We thus want to find an infinite-state model for which the pre^* or/and $post^*$ images always belong to a "good" class. In our framework, these two sets are two (infinite) subsets of $Q \times \mathbb{N}^n$ (where Q is the finite set of control states and n is the number of counters on which the automaton operates). Semilinear sets of $Q \times \mathbb{N}^n$ form a "good" class since (1) they are closed under $\cap$ and under projection, (2) inclusion is decidable, and (3) they are also the sets expressible in Presburger arithmetics over integers and also the set of regular subsets in the commutative monoid $Q \times \mathbb{N}^n$. Hence, our objective is to find an extension of finite automata for which pre^* and/or $post^*$ are semilinear and, if possible, effectively computable.

Related work. Automata with counters, but without any zero-testing primitive, are known as Vector Addition Systems with States (VASS) and they are equivalent to Petri nets. The reachability problem is thus decidable but the comparison between two reachability sets is undecidable. Moreover, in dimension n, $n > 2$, the reachability set is not always semilinear [HP79].

Stack automata are another well-known extension of finite automata. They have regular reachability sets (in $Q \times \Sigma^*$); this property has been recently rediscovered [BS86, Cau92, FWW97, BEM97] and used for model-checking of stack automata. Of course, stack automata contain 1-counter automata (with the capability of zero-testing). At last, it has been proved that lossy FIFO automata have regular non-computable reachability sets [CFP96] while half-duplex and quasi-stable fifo automata have effectively computable regular reachability sets [CF97]. For flat counters automata (flat means that there is no nested loop in the control transition graph), the reachability relation can be expressed in the Presburger logics [CJ98].

A result of Hopcroft and Pansiot [HP79] said that the reachability set of a 2-dim Vector Addition Systems with States (2-dim VASS) is semilinear and that it is effectively computable. Hopcroft and Pansiot gave an algorithm which computes a description of the semilinear reachability set. The complexity of Hopcroft-Pansiot's algorithm has been studied later by Rosier and Yen [HRHY86] who showed that the deterministic complexity is in 2^{2^n} where n is the size of the automaton with its initial state.

More recently, some papers appeared dealing with verification of reset/transfer Petri nets (or counters automata for which counters are only accessed through the following operations: increment (+1), decrement (-1), reset ($c := 0$), transfer (the whole content of counter c is transfered into counter c'). For these weak counter automata (that do not have a full-fledged test for zero), reachability of a control state is decidable for reset/transfer Petri nets, thus also for counters automata with the following operations {+1, -1, reset, transfer} [DFS98]. Reachability and boundedness are undecidable in dimension 3 (since there are three counters) [DFS98] ; the limit between decidability/undecidability for the boundedness problem is precised as follows: boundedness is decidable for {+1, -1, reset}-counters automata for which only two counters can be reset [DJS99]. Reachability is decidable for counters automata such that only a unique counter can be tested for zero [Rei95]. These extensions except [Rei95] do not contain the complete test for zero.

These results from the litterature do not give a global and comprehensive view of the properties of classes of automata with two {+1, -1, reset, transfer, zero-test}-counters. Can we extend the result of [HP79] to more powerful models of two counters automata ? Are pre^* and/or $post^*$ also semilinear for extended automata ?

Our contribution. We present three types of results:

(1) A *new general symbolic reachability tree (semi-)algorithm* for automata with {+1, -1, reset, transfer, zero-test}-counters.

This algorithm is *generic* and *modular*, and it uses an `Acceleration` function. It exactly computes the reachability set ($post^*$) when the `Acceleration` function satisfies a so-called *stability* property. Our algorithm, given for any n-dim Extended VASS, both generalizes Hopcroft-Pansiot's algorithm (given for 2-dim VASS) and refines Karp-Miller's algorithm (given for n-dim VASS) [KM69]. It gives an exact description of $post^*$ (as Hopcroft-Pansiot's algorithm and not only a coverability set like Karp-Miller's algorithm). We prove termination and correctness of this algorithm for some subclasses of automata with two {+1, -1, reset, transfer, zero-test}-counters.

(2) A *hierarchy result.*

We systematically study all the subclasses of two {+1, -1, reset, transfer, zero-test}-counters automata and we give a complete hierarchy between classes using new simulations. A "maximal decidable model" appears to be 2-counters automata ($\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$) with the following extended operations: test for zero on the first counter, reset on the two counters, and, transfer from the first to the second counter (see Figure 1 on page 11).

(3) *Three main technical results.*

We prove the three following theorems: pre^* is an effective semilinear set for the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ (Theorem 4) ; $post^*$ is an effective semilinear set for the class $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$ (Theorem 3) ; and, $post^*$ is a semilinear set for the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ (Theorem 5).

Plan. Section 2 introduces Extended Vector Addition Systems with States (E-VASS), and we present in Section 3 semilinear sets and projections. In Section 4, we give a new symbolic reachability tree algorithm for E-VASS which uses as a parameter a function `Acceleration` and we define the stability property. Section 5 extends Hopcroft-Pansiot's result (effective semilinearity of $post^*$) to 2-dim Extended VASS with one test for zero (and with one reset and one transfer operations). Section 6 proves the effective semilinearity of pre^* for the maximal class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$. Section 7 shows that $post^*$ is still semilinear for the maximal class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$.

The proofs are technically nontrivial and are omitted here due to space constraints. They can be found in the full version available form the authors and at the URL `http://www.lsv.ens-cachan.fr/Publis/`.

2 Extended Vector Addition Systems with States

Let $\mathbb{Z}$ (resp. $\mathbb{N}$, $\mathbb{N}^*$, $\mathbb{Q}^+$) denote the set of integers (resp. nonnegative integers, positive integers, nonnegative rational numbers). If $i, j \in \mathbb{N}$, we write $[i, j]$ for the set $\{k \in \mathbb{N} \ / \ i \leq k \leq j\}$. Let $\mathbb{N}^n$ (resp. $\mathbb{Z}^n$) denote the set of n-tuples of elements of $\mathbb{N}$ (resp. $\mathbb{Z}$). If x is an n-tuple and $i \in [1, n]$, $x(i)$ is the i^{th} *component* of x. The i^{th} *unit vector* is the n-tuple e_i defined by $e_i(j) = 0$ if $j \neq i$ and $e_i(i) = 1$. Operations on n-tuples are componentwise extensions of the usual operations and when 0 is used as an n-tuple, it denotes the all zero n-tuple. These operations are classically extended on sets of n-tuples (e.g. for $P, P' \subseteq \mathbb{N}^n$, $P + P' = \{p + p' \ / \ p \in P \text{ and } p' \in P'\}$). Moreover, in an operation involving sets of n-tuples, we shortly write v for the singleton $\{v\}$ (e.g. for $P \subseteq \mathbb{N}^n$ and $x \in \mathbb{N}^n$, we write $x + P$ for $\{x\} + P$).

For every set X, we write $\wp(X)$ for the set of subsets of X and $|X|$ for the cardinal of X. For any infinite sequence $(x_i)_{i\in\mathbb{N}}$ in X, an infinite *subsequence* (of (x_i)) is any infinite sequence $(x_{i_k})_{i_k\in\mathbb{N}}$ with $i_0 < i_1 < i_2 \cdots i_k < i_{k+1} \cdots$.

An *ordering* is any reflexive, transitive and antisymmetric relation $\preceq$ over some set X. A *well ordering* is any ordering $\preceq$ such that, for any infinite sequence $(x_i)_{i\in\mathbb{N}}$ in X, there exists an infinite increasing subsequence $x_{i_0} \preceq x_{i_1} \preceq x_{i_2} \cdots x_{i_k} \preceq x_{i_{k+1}} \cdots$. If $\preceq$ is an ordering on X, then for every subset $Y \subseteq X$, the set $\text{Min}(Y)$ of *minimal elements* of Y is defined by $\text{Min}(Y) = \{y \in Y \ / \ \forall\, y' \in Y \setminus \{y\}, y' \not\preceq y\}$ and it is finite when $\preceq$ is a well ordering.

The relation $\leq$ between n-tuples in $\mathbb{Z}^n$ is defined by $x \leq y$ if for all $i \in [1, n]$ we have $x(i) \leq y(i)$. This relation is an ordering on $\mathbb{Z}^n$ and it is a well ordering on $\mathbb{N}^n$ (it is not a well ordering on $\mathbb{Z}^n$).

A *labelled transition system* is a structure $\mathcal{TS} = (S, A, \rightarrow)$ 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, $\mathcal{TS}$ is a *finite labelled transition system*. We note $\xrightarrow{*}$ the reflexive and transitive closure of $\rightarrow$. For every subset X of S, we write $post(X)$ for the set $\{s \in S \ / \ \exists\ r \in X, r \rightarrow s\}$ of *immediate successors* of X, $post^*(X)$ for the set $\{s \in S \ / \ \exists\ r \in X, r \xrightarrow{*} s\}$ of *successors* of X, and $pre^*(X)$ for the set $\{s \in S \ / \ \exists\ r \in X, s \xrightarrow{*} r\}$ of *predecessors* of X.

We present Extended Vector Addition Systems with States (E-VASS) in two steps : we first describe the finite control structure of an E-VASS and we then define an E-VASS as an operational semantics associated with such a control structure, which leads to an infinite labelled transition system.

An *n-dim E-VASS Control* is any finite labelled transition system $\mathsf{A} = (Q, Op, \rightarrow_{\mathsf{A}})$ such that $Op \subseteq \{add(v) \ / \ v \in \mathbb{Z}^n\} \cup \{test(i), weaktest(i), reset(i), transfer(i \rightarrow j) \ / \ i, j \in [1, n], i \neq j\}$.

Definition 1. An *n-dim E-VASS* $\mathcal{A}$ is a labelled transition system $\mathcal{A} = (S, Op, \rightarrow_{\mathcal{A}})$ based on an *n-dim E-VASS Control* $\mathsf{A} = (Q, Op, \rightarrow_{\mathsf{A}})$, where:

- $S = Q \times \mathbb{N}^n$ is the set of states, and,
- 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:

1. if $q \xrightarrow{add(v)}_{\mathsf{A}} q'$ then for all $x \in \mathbb{N}^n$ such that $x+v \geq 0$, $(q,x) \xrightarrow{add(v)}_{\mathcal{A}} (q', x+v)$, and
2. if $q \xrightarrow{test(i)}_{\mathsf{A}} q'$ then for all $x \in \mathbb{N}^n$ such that $x(i) = 0$, $(q,x) \xrightarrow{test(i)}_{\mathcal{A}} (q', x)$, and
3. if $q \xrightarrow{weaktest(i)}_{\mathsf{A}} q'$ then for all $x \in \mathbb{N}^n$ such that $x(i) = 0$ and for all $\alpha \in \mathbb{N}$, $(q,x) \xrightarrow{weaktest(i)}_{\mathcal{A}} (q', x + \alpha e_i)$, and
4. if $q \xrightarrow{reset(i)}_{\mathsf{A}} q'$ then for all $x \in \mathbb{N}^n$, $(q,x) \xrightarrow{reset(i)}_{\mathcal{A}} (q', x')$ where $x'(i) = 0$ and for all $j \neq i, x'(j) = x(j)$, and
5. if $q \xrightarrow{transfer(i \to j)}_{\mathsf{A}} q'$ then for all $x \in \mathbb{N}^n$, $(q,x) \xrightarrow{transfer(i \to j)}_{\mathcal{A}} (q', x')$ where $x'(i) = 0$, $x'(j) = x(i) + x(j)$ and for all $k \notin \{i,j\}, x'(k) = x(k)$.

A $weaktest(i)$ transition is the inverse of a $reset(i)$ in the sense that we have $(q,x) \xrightarrow{weaktest(i)} (q',x')$ iff $(q',x') \xrightarrow{reset(i)} (q,x)$. We will use $weaktest(i)$ transitions in section 6 to compute pre^* for E-VASS with $reset(i)$ transitions. Notice that any ε-transition may be simulated by an $add(0)$ transition. An *n-dim* (non extended) *VASS* is any n-dim E-VASS $\mathcal{A} = (S, Op, \to_{\mathcal{A}})$ such that $Op \subseteq \{add(v) \ / \ v \in \mathbb{Z}^n\}$.

We define the following classes of 2-dim E-VASS. For every $I \subseteq \{1,2\}$, $J \subseteq \{1,2\}$ and $K \subseteq \{(1,2),(2,1)\}$, we write $\mathbf{T}_I\mathbf{R}_J\mathbf{Tr}_K$ for the class of 2-dim E-VASS $\mathcal{A}$ such the set of actions Op of $\mathcal{A}$ satisfies:

$$\begin{aligned} Op \subseteq \quad & \{add(v) \ / \ v \in \mathbb{Z}^n\} \\ & \cup \{test(i), reset(j), transfer(k \to k') \ / \ i \in I, j \in J \text{ and } (k,k') \in K\} \end{aligned}$$

Notice that $\mathbf{T}_\emptyset$ (resp. $\mathbf{R}_\emptyset$, $\mathbf{Tr}_\emptyset$) means that there is no zero-test transition (resp. no reset transition, no transfer transition) and we will omit to write $\mathbf{T}_\emptyset$ (resp. $\mathbf{R}_\emptyset$, $\mathbf{Tr}_\emptyset$). For instance, $\mathbf{T}_1\mathbf{R}_\emptyset\mathbf{Tr}_{12,21}$ (shortly written $\mathbf{T}_1\mathbf{Tr}_{12,21}$) is the class of 2-dim E-VASS $\mathcal{A}$ where the allowed extended transitions are labelled by $test(1)$, $transfer(1 \to 2)$ or $transfer(2 \to 1)$.

Simulations

Let $\mathbf{C}$ and $\mathbf{D}$ be two classes of E-VASS. We say $\mathbf{C}$ is *effectively simulable* by $\mathbf{D}$ if for every E-VASS $\mathcal{A}$ in $\mathbf{C}$ with control states Q_{A}, there exists an E-VASS $\mathcal{B}$ in $\mathbf{D}$ with control states Q_{B} computable from $\mathcal{A}$ such that:

1. we have $Q_{\mathsf{B}} \supseteq \overline{Q_{\mathsf{A}}}$, where $\overline{Q_{\mathsf{A}}}$ is a copy of Q_{A}, and,
2. for any states (q,x) and (q',x') in $\mathcal{A}$, we have $(q,x) \xrightarrow{*}_{\mathsf{A}} (q',x')$ iff $(\overline{q},x) \xrightarrow{*}_{\mathsf{B}} (\overline{q'},x')$.

We will use the notion of effective simulation (which is different from the classical notion of simulation) to have simpler proofs for the effective computation of $post^*$ and pre^*. Assume that $\mathbf{C}$ is effectively simulable by $\mathbf{D}$. If there exists an algorithm which computes for every E-VASS in $\mathbf{D}$ a finite description of $post^*$ (resp. pre^*) then we get that there exists an algorithm which computes for every E-VASS in $\mathbf{C}$ a finite description of $post^*$ (resp. pre^*).

Theorem 1. *We have the following simulations between classes of 2-dim E-VASS:*

1. *the class* $\mathbf{T}_{1,2}$ *is effectively simulable by the class* $\mathbf{T}_1\mathbf{Tr}_{21}$*,*
2. *the class* $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ *is effectively simulable by the class* $\mathbf{T}_{1,2}$*,*
3. *the class* $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ *is effectively simulable by the class* $\mathbf{T}_1\mathbf{R}_2$*,*
4. *the class* $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$ *is effectively simulable by the class* $\mathbf{T}_1$*.*
5. *the class* $\mathbf{R}_{1,2}\mathbf{Tr}_{12,21}$ *is effectively simulable by the class* $\mathbf{T}_1\mathbf{R}_2$*,*

3 Semilinear sets and projections

Let $P = \{p_1, p_2, \cdots, p_k\}$ be a finite subset of $\mathbb{N}^n$. We write P^* for the set $P^* = \{\Sigma_{i=1}^k \alpha_i p_i \;/\; \forall\, i \in [1,k], \alpha_i \in \mathbb{N}\}$. For every $x \in \mathbb{N}^n$, $(x + P^*)$ is called a *linear* set. A *semilinear* set is a finite union of linear sets. A *description* of a semilinear set L is any finite family $\{(x_i, P_i) \;/\; i \in I\}$ such that $L = \bigcup_{i \in I}(x_i + {P_i}^*)$. We will say that a semilinear set L is *effectively computable* if one can give *explicitly* an algorithm which computes a description of L.

Let us remark that for every $x \in \mathbb{N}^n$ and $v \in \mathbb{Z}^n$, the set $L = ((x+P^*)+v) \cap \mathbb{N}^n$ is semilinear as it can be written as $L = (B + P^*)$ where $B = \mathrm{Min}(((x + P^*) + v) \cap \mathbb{N}^n)$ is finite. Moreover, it can be shown that B is computable [HP79].

For every $i \in [1, n]$, we write H_i for the linear set $H_i = \{x \in \mathbb{N}^n \;/\; x(i) = 0\}$. We define two kinds of projection on H_i, extended on subsets of $\mathbb{N}^n$ in the usual way:

- for every $i \in [1, n]$, $proj_i : \mathbb{N}^n \to \mathbb{N}^n$ is the *orthogonal projection* on H_i defined by $proj_i(x) = x'$ where $x'(i) = 0$, and $\forall\, k \neq i, x'(k) = x(k)$.
- for every $i, j \in [1, n]$ such that $i \neq j$, $proj_{ij} : \mathbb{N}^n \to \mathbb{N}^n$ is the *diagonal projection* on H_i defined by $proj_{ij}(x) = x'$ where $x'(i) = 0, x'(j) = x(i)+x(j)$ and $\forall\, k \notin \{i, j\}, x'(k) = x(k)$.

4 Computation of a Symbolic Reachability Tree

Let us consider an n-dim E-VASS $\mathcal{A}$. The algorithm we propose later is based on the construction of a tree labelled by 3-tuples (q, x, P) where q is a control state of $\mathcal{A}$, x is in $\mathbb{N}^n$ and P is a finite subset of $\mathbb{N}^n$. Intuitively, the label $l = (q, x, P)$ of a node t (written $t : l$) represents the set of states $[\![l]\!] = \{q\} \times (x + P^*)$ of $\mathcal{A}$.

Notation. For every control state q of an n-dim E-VASS $\mathcal{A}$, for every $x \in \mathbb{N}^n$ and for every finite subset P of $\mathbb{N}^n$, we write $[\![q, x, P]\!]$ for the set of states $[\![q, x, P]\!] = \{q\} \times (x + P^*)$ of $\mathcal{A}$.

We are now able to define a notion of Symbolic Reachability Tree as follows.

Definition 2. A *Symbolic Reachability Tree* for an n-dim E-VASS $\mathcal{A}$ with a set S_0 of initial states is any rooted directed tree T labelled with 3-tuples (q, x, P) where q is a control state of $\mathcal{A}$, $x \in \mathbb{N}^n$ and P is a finite subset of $\mathbb{N}^n$, such that:

1. we have $\text{post}^*(S_0) = \bigcup_{t:l\ node\ of\ T} [\![l]\!]$, and,
2. for any node $t : l$ in T such that t has a son, then $post([\![l]\!]) \subseteq \bigcup_{i \in I} [\![l_i]\!] \subseteq post^*([\![l]\!])$ where $\{t_i : l_i\}_{i \in I}$ is the set of sons of t.

The `SymbolicTree` algorithm essentially computes a reachability tree. But as the reachability tree of an E-VASS may be infinite in general, an `Acceleration` function is used to accelerate the computation. Similarly to the Karp-Miller strategy [KM69], the `Acceleration` function basically computes the iteration of a repetitive sequence. This `Acceleration` function may depend on the input E-VASS, for instance we will define specific `Acceleration` functions suited to specific classes of E-VASS.

Algorithm 1 `SymbolicTree`$(\mathcal{A}, S_0,$ `Acceleration`$)$

Input: an n-dim E-VASS $\mathcal{A}$, a linear set $S_0 = \{q_0\} \times (x_0 + {P_0}^*)$ of initial states of $\mathcal{A}$ and an `Acceleration` function

create root labelled (q_0, x_0, P_0)
while there are unmarked nodes **do**
  pick an unmarked node $t : l$, where $l = (q, x, P)$
  mark t
  if there is an ancestor $t' : l'$ of t such that $[\![l]\!] \subseteq [\![l']\!]$ **then**
    skip
  else
    $W(t) \leftarrow$ `Acceleration`$(t, \mathcal{A})$
    {We now compute the set $post([\![q, x, W]\!])$ of immediate successors of $[\![q, x, W]\!]$}
    for each transition $q \xrightarrow{add(v)} q'$ in A **do**
      for each $m \in \text{Min}(((x + W^*) + v) \cap \mathbb{N}^n)$ **do**
        construct a son $t' : (q', m, W)$ of t and label the arc $t \xrightarrow{add(v)} t'$
    for each transition $q \xrightarrow{test(i)} q'$ in A **do**
      if $x_i = 0$ **then**
        construct a son $t' : (q', x, W \cap H_i)$ of t and label the arc $t \xrightarrow{test(i)} t'$
    for each transition $q \xrightarrow{weaktest(i)} q'$ in A **do**
      if $x_i = 0$ **then**
        construct a son $t' : (q', x, (W \cap H_i) \cup \{e_i\})$ of t and label the arc $t \xrightarrow{test(i)} t'$
    for each transition $q \xrightarrow{reset(i)} q'$ in A **do**
      construct a son $t' : (q', proj_i(x), proj_i(W) \setminus \{0\})$ of t and label the arc $t \xrightarrow{reset(i)} t'$
    for each transition $q \xrightarrow{transfer(i \to j)} q'$ in A **do**
      construct a son $t' : (q', proj_{ij}(x), proj_{ij}(W))$ of t and label the arc $t \xrightarrow{transfer(i \to j)} t'$

We will often use the following stability property for the different considered `Acceleration` functions.

Definition 3. An `Acceleration` function is *n-stable* if for every n-dim E-VASS $\mathcal{A}$ with a linear set of initial states S_0 and for every node $t : (q, x, P)$ in the tree constructed by $\texttt{SymbolicTree}(\mathcal{A}, S_0, \texttt{Acceleration})$, we have the following property:

$$[\![q, x, P]\!] \subseteq [\![q, x, W(t)]\!] \subseteq post^*([\![q, x, P]\!])$$

where $W(t) = \texttt{Acceleration}(t, \mathcal{A})$.

Theorem 2. *Let $\mathcal{A}$ be an n-dim E-VASS and $S_0 = \{q_0\} \times (x_0 + {P_0}^*)$ be a linear set of initial states of $\mathcal{A}$. If an* `Acceleration` *function is n-stable then the* `SymbolicTree` *algorithm, applied to $(\mathcal{A}, S_0, \texttt{Acceleration})$, constructs a Symbolic Reachability Tree for $\mathcal{A}$ with initial states S_0.*

Notice that if the `SymbolicTree` algorithm terminates on some n-dim E-VASS $\mathcal{A}$ with a linear set of initial states S_0 and an n-stable `Acceleration` function, then post$^*(S_0)$ is semilinear and effectively computable. Because 3-dim VASS can have non-semilinear reachability set [HP79], we now restrict our study to 2-dim E-VASS. For simplicity, we will shortly write *stable* for 2-stable. In the following of the paper, our aim is to find, for each analyzed class of 2-dim E-VASS, a dedicated stable `Acceleration` function ensuring the termination of the `SymbolicTree` algorithm for this class.

5 Effective semilinearity of *post** for the class $\mathbf{T_1 R_1 Tr_{12}}$

We now may generalize Hopcroft-Pansiot's result in showing that *post** is still semilinear and effectively computable for any E-VASS $\mathcal{A}$ in $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$. We extract from Hopcroft-Pansiot's algorithm an $\texttt{Acceleration}_{HP}$ function [1] which is stable and such that for every 2-dim VASS $\mathcal{A}$ with a linear set of initial states S_0, the `SymbolicTree` algorithm applied to $(\mathcal{A}, S_0, \texttt{Acceleration}_{HP})$ terminates. This $\texttt{Acceleration}_{HP}$ function is then used to define a $\texttt{Acceleration}_{\mathbf{T}_1}$ function which allows us to show that *post** is still semilinear and effectively computable for any E-VASS $\mathcal{A}$ in $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$.

Proposition 1. *The function* $\texttt{Acceleration}_{\mathbf{T}_1}$ *is stable. Moreover, for every 2-dim E-VASS $\mathcal{A}$ in the class $\mathbf{T}_1$ with a linear set of initial states S_0, the* `SymbolicTree` *algorithm applied to $(\mathcal{A}, S_0, \texttt{Acceleration}_{\mathbf{T}_1})$ terminates.*

By using the effective simulation of $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$ by $\mathbf{T}_1$ (Theorem 1), and Proposition 1, we obtain the following theorem.

Theorem 3. *For any 2-dim E-VASS $\mathcal{A}$ in the class $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$ with a semilinear set S_0 of initial states, post$^*(S_0)$ is semilinear and it is effectively computable.*

---

[1] The definition of this function is omitted here, but it can be found in the full version.

Algorithm 2 $\texttt{Acceleration}_{\mathbf{T}_1}(t:(q,x,P),\mathcal{A})$

Input: a node $t:(q,x,P)$ and a 2-dim E-VASS $\mathcal{A}$

```
if the branch from the root r to t may be written as r -*-> u -add(v)-> t for some v then
   return Acceleration_HP(t, A)
else if the branch from the root r to t may be written as r -*-> u -test(1)-> t then
   if there is an ancestor t' : (q, x', P') of t such that (x − x') ∈ {0} × N* and there
   is no extended arc except test(1) arcs between t' and t then
      return P ∪ {x − x'}
else
   return P
```

6 Effective semilinearity of pre^* for the class $\mathbf{T_1R_{1,2}Tr_{12}}$

We present in this section an algorithm for computing pre^* for the largest class (non equivalent to $\mathbf{T}_{1,2}$) of 2-dim E-VASS. We first prove that $post^*$ of a 2-dim E-VASS $\mathcal{A}$ in $\mathbf{T}_1\mathbf{W}_2$ is semilinear and effectively computable (accordingly to previous notations, we note $\mathbf{T}_1\mathbf{W}_2$ the class of 2-dim E-VASS where the allowed extended transitions are labelled by $test(1)$ or $weaktest(2)$.

Algorithm 3 $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{W}_2}(t:(q,x,P),\mathcal{A})$

Input: a node $t:(q,x,P)$ and a 2-dim E-VASS $\mathcal{A}$

```
if the branch from the root r to t may be written as r -*-> u -weaktest(2)-> t then
   if there is an ancestor t' : (q, x', P') of t such that (x − x') ∈ N* × {0} and there
   is no extended arc except weaktest(2) arcs between t' and t then
      return P ∪ {x − x'}
else
   return Acceleration_T1(t, A)
```

Proposition 2. *The function* $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{W}_2}$ *is stable. Moreover, for every 2-dim E-VASS* $\mathcal{A}$ *in the class* $\mathbf{T}_1\mathbf{W}_2$ *with a linear set of initial states* S_0*, the* `SymbolicTree` *algorithm applied to* $(\mathcal{A},S_0,\texttt{Acceleration}_{\mathbf{T}_1\mathbf{W}_2})$ *terminates.*

Theorem 4. *For any 2-dim E-VASS* $\mathcal{A}$ *in the class* $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ *with a semilinear set* S_0 *of initial states,* $pre^*(S_0)$ *is semilinear and it is effectively computable.*

Corollary 1. *The reachability problem is decidable for the class* $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$.

7 Semilinearity of $post^*$ for the class $\mathbf{T_1R_{1,2}Tr_{12}}$

Let us recall that in section 5, we proved that $post^*$ is semilinear and effectively computable for any $\mathcal{A}$ in $\mathbf{T}_1\mathbf{R}_1\mathbf{Tr}_{12}$. We now extend this semilinearity result to the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$.

Let $\Omega(\mathcal{A}, S_0)$ be the set defined by $\Omega(\mathcal{A}, S_0) = \bigcup_{s \in S_0} \Omega(\mathcal{A}, s)$ where for every state s of $\mathcal{A}$, $\Omega(\mathcal{A}, s)$ is given by:

$$\Omega(\mathcal{A}, s) = \{q \in Q \;/\; |\{x \in \mathbb{N} \;/\; \exists\, \sigma,\ s \xrightarrow{\sigma test(1)} (q, (0, x))\}| = \infty\}$$

where $\mathcal{A}$ is a 2-dim E-VASS with a set Q of control states and S_0 is a set of initial states of $\mathcal{A}$.

The main problem to ensure termination is that we may have on a branch of the constructed tree infinitely many interleavings between $reset(2)$ and $test(1)$ transitions. Hence, the $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{R}_2}$ function uses the set $\Omega(\mathcal{A}, S_0)$ in order to iterate some special loops in this case. Intuitively, if we have just fired a $reset(2)$ transition, then we do not care about the actual value of the second component before the $reset(2)$. Hence if we know that this value may be arbitrarily large, then we can iterate a simple control loop that increases the first component before the firing of the $reset(2)$ transition. Formally, a *simple control loop* l in an E-VASS $\mathcal{A}$ is a path $l = q \xrightarrow{add(v_1)} q_1 \xrightarrow{add(v_2)} q_2 \cdots q_{n-1} \xrightarrow{add(v_n)} q$ in the E-VASS Control A of $\mathcal{A}$ with no repeating state except that the first state and the last state are the same ; moreover we say that l is a control loop *on* q with *displacement* $d(l) = v_1 + v_2 + \cdots + v_n$.

Notice that the set $\Omega(\mathcal{A}, S_0)$ is finite, as it is a subset of Q. Hence, even if we actually do not know whether we can effectively compute this set, we can use this set as an "oracle" in our $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{R}_2}$ function.

Algorithm 4 $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{R}_2}(t : (q, x, P), \mathcal{A})$

Input: a node $t : (q, x, P)$ and a 2-dim E-VASS $\mathcal{A}$

if the branch from the root r to t may be written as $r \xrightarrow{*} u \xrightarrow{reset(2)} t$ **then**
  if there is an ancestor $t' : (q, x', P')$ of t such that $(x - x') \in \mathbb{N}^* \times \{0\}$ and there is no extended arc except $reset(2)$ arcs between t' and t **then**
    return $P \cup \{x - x'\}$
else if the branch from the root r to t may be written as $r \xrightarrow{*} s \xrightarrow{test(1)} t' \xrightarrow{*} t'' \xrightarrow{reset(2)} t$ with no extended arc between t' and t'' **then**
  if the label (q', x', P') of t' is such that $q' \in \Omega(\mathcal{A}, S_0)$ **then**
    if there exists a node $u : (r, y, Q)$ between t' and t'' and a simple control loop on r with displacement $(a, b) \in \mathbb{N}^* \times \mathbb{Z}$ **then**
      return $P \cup \{(a, 0)\}$
else
  return $\texttt{Acceleration}_{\mathbf{T}_1}(t, \mathcal{A})$

Proposition 3. *The function* $\texttt{Acceleration}_{\mathbf{T}_1\mathbf{R}_2}$ *is stable. Moreover, for every 2-dim E-VASS* $\mathcal{A}$ *in the class* $\mathbf{T}_1\mathbf{R}_2$ *with a linear set of initial states* S_0*, the* $\texttt{SymbolicTree}$ *algorithm applied to* $(\mathcal{A}, S_0, \texttt{Acceleration}_{\mathbf{T}_1\mathbf{R}_2})$ *terminates.*

From Theorem 1, the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ is effectively simulated by $\mathbf{T}_1\mathbf{R}_2$. From Proposition 3, we conclude that $post^*$ is semilinear.

Theorem 5. *For any 2-dim E-VASS $\mathcal{A}$ in the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$ with a semilinear set S_0 of initial states, $post^*(S_0)$ is semilinear. Moreover, if $\Omega(\mathcal{A}, S_0)$ is effectively computable, then $post^*(S_0)$ is effectively computable.*

It remains an open problem to know whether it is possible to compute a finite description of the semilinear reachability set for the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$.

8 Conclusion

We have systematically studied all the subclasses of two {+1, -1, reset, transfer, zero-test}-counters automata and given a complete hierarchy between classes using new simulations. It turns out that a "maximal decidable model" consists of the class of 2-counters automata ($\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$) with the following extended operations: test for zero on the first counter, reset on the two counters, and transfer from the first to the second counter. For this model, the pre^* image is effectively semilinear.

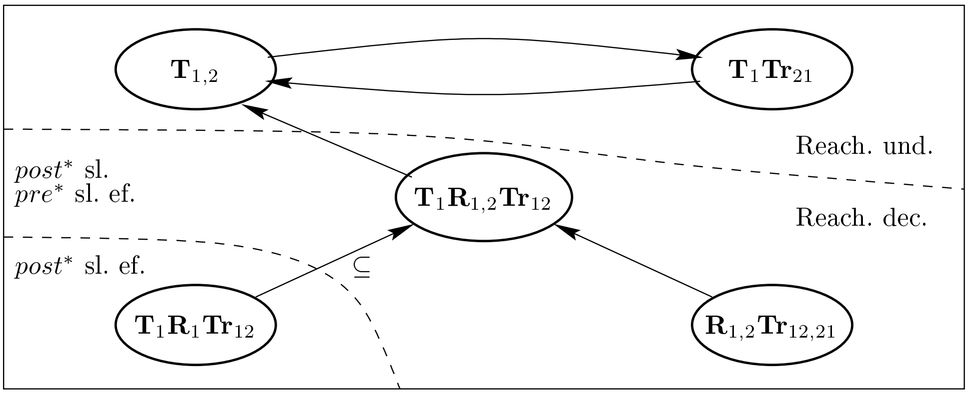

Fig. 1. Graph of inclusions and simulations

Our results cannot be generalized to three {+1, -1}-counters automata because they may have non-semilinear reachability sets; moreover, reachability is undecidable for three {+1, -1, reset}-counters automata such that one of the counters never uses the reset operation [DFS98, Duf98].

Our symbolic algorithm has been defined for any {+1, -1, reset, transfer, zero-test}-counters automata. It can also be used as a semi-algorithm when termination cannot be guaranteed.

A lot of other open problems still arise: for example, we have no answer whether $post^*$ is *effectively* semilinear for the class $\mathbf{T}_1\mathbf{R}_{1,2}\mathbf{Tr}_{12}$. The complexity of the symbolic algorithm is certainly at least in 2^{2^n} [HRHY86] but we don't know if it is primitive recursive or not. Another point which has to be stated is the comparison between classes of 2-counters automata and stack automata.

References

[BEM97] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: Application to model-checking. In *Proc. 8th Int. Conf. Concurrency Theory (CONCUR'97), Warsaw, Poland, Jul. 1997*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997.

[BS86] M. Benois and J. Sakarovitch. On the complexity of some extended word problems defined by cancellation rules. *Information Processing Letters*, 23(6):281–287, 1986.

[Cau92] D. Caucal. On the regular structure of prefix rewriting. *Theoretical Computer Science*, 106(1):61–86, 1992.

[CF97] G. Cécé and A. Finkel. Programs with quasi-stable channels are effectively recognizable. In *Proc. of the* 9^{th} *Conference on Computer-Aided Verification (CAV)*, volume 1254, pages 304–315. LNCS, June 1997.

[CFP96] G. Cécé, A. Finkel, and I. S. Purushothaman. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1996.

[CJ98] H. Comon and Y. Jurski. Multiple counters automata, safety analysis and Presburger arithmetic. 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 268–279. Springer, 1998.

[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.

[Duf98] C. Dufourd. Réseaux de Petri avec reset/transfert: Décidabilité et indécidabilité. Thèse de Docteur en Sciences de l'École Normale Supérieure de Cachan, October 1998.

[FWW97] A. Finkel, B. Willems, and P. Wolper. A direct symbolic approach to model checking pushdown systems. *Electronic Notes in Theoretical Computer Science*, 9, 1997.

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

[HRHY86] R. R. Howell, L. E. Rosier, D. T. Huynh, and H-C. Yen. Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states. *Theoretical Computer Science*, 46:107–140, 1986.

[KM69] R. M. Karp and R. E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.

[Rei95] K. Reinhardt. Reachability in Petri nets with inhibitor arcs, November 1995. Unpublished manuscript. See `www-fs.informatik.uni-tuebingen.de/~reinhard`.