Mixing coverability and reachability to analyze VASS with one zero-test

Alain Finkel[1,⋆] and Arnaud Sangnier[2,⋆⋆]

[1]LSV, ENS Cachan & CNRS, France
finkel@lsv.ens-cachan.fr
[2]Dipartimento di Informatica, Università di Torino, Italy
sangnier@di.unito.it

Abstract. We study Vector Addition Systems with States (VASS) extended in such a way that one of the manipulated integer variables can be tested to zero. For this class of system, it has been proved that the reachability problem is decidable. We prove here that boundedness, termination and reversal-boundedness are decidable for VASS with one zero-test. To decide reversal-boundedness, we provide an original method which mixes both the construction of the coverability graph for VASS and the computation of the reachability set of reversal-bounded counter machines. The same construction can be slightly adapted to decide boundedness and hence termination.

1 Introduction

Vector Addition Systems with States (VASS), which are equivalent to Petri nets, are a model which has received a lot of attention and thousands of papers exist on this subject [17]. Whereas many problems are decidable for VASS [4], it is well-known that VASS with the ability for testing to zero (or with inhibitor arcs) have the power of Turing machines. Hence all the non-trivial problems are undecidable for this class of models.
Recently in [18], Reinhardt proved that the reachability problem for VASS with an unique integer variable (or counter) tested to zero is decidable in reducing this problem to the reachability problem for Petri nets, which is decidable (see the papers of Kosaraju [12] and Mayr [14] and Leroux [13] for a conceptual decidability proof of reachability). For VASS, many problems like zero reachability, coverability (is it possible to reach a configuration larger than a given configuration?), boundedness (whether the reachability set is finite?) and termination (is there an infinite execution?) can be reduced to reachability and this is still true for extended (well-structured) Petri nets [3,9] (polynomial reductions of reachability for well-structured Petri nets extensions are given in [2]). For VASS with one zero-test, coverability reduces (as usual) to reachability but it is less clear for the other properties like boundedness whether the known reductions can be adapted. Note that in [1], Abdulla and Mayr proposes a method to decide coverability for VASS with one zero-test without using the Reinhardt's result.

⋆ Partly supported by project AVERISS (ANR-06-SETIN-001)
⋆⋆ Supported by a post-doctoral scolarship from DGA/ENS Cachan

In many verification problems, it is convenient not only to have an algorithm for the reachability problem, but also to be able to compute effectively the reachability set. In [10], the class of reversal-bounded counter machines is introduced as follows: each counter can only perform a bounded number of alternations between increasing and decreasing mode. Ibarra shows that reversal-bounded counter machines enjoy the following nice property: their reachability set is a semi-linear set which can be effectively computed. In a recent work [7], we have proved that reversal-boundedness is decidable for VASS, whereas for VASS extended with two counters which can be tested to zero this property is undecidable.
Our contribution. We investigate here the three following problems: given a VASS with one zero-test, can we decide whether it is bounded, whether it is reversal-bounded and whether it terminates. We first consider the most difficult problem, which is the reversal-boundedness problem and from the algorithm for solving it, we deduce another algorithm for solving boundedness. The decidability of termination is then obtained by a classical reduction into boundedness. The algorithm we propose mix the classical construction of the coverability graph for VASS [11] and the computing of the reachability set of reversal-bounded counter machines.

Due to lack of space, some details are omitted and can be found in [8].

2 VASS with one zero-test and reversal-bounded property

2.1 Useful notions

Let $\mathbb{N}$ (resp. $\mathbb{Z}$) denotes the set of nonnegative integers (resp. integers). The usual total order over $\mathbb{Z}$ is written $\leq$. By $\mathbb{N}_\omega$, we denote the set $\mathbb{N} \cup \{\omega\}$ where ω is a new symbol such that $\omega \notin \mathbb{N}$ and for all $k \in \mathbb{N}_\omega$, $k \leq \omega$. We extend the binary operation $+$ and $-$ to $\mathbb{N}_\omega$ as follows : for all $k \in \mathbb{N}$, $k + \omega = \omega$ and $\omega - k = \omega$. For $k, l \in \mathbb{N}_\omega$ with $k \leq l$, we write $[k..l]$ for the interval of integers $\{i \in \mathbb{N} \mid k \leq i \leq l\}$.
Given a set X and $n \in \mathbb{N}$, X^n is the set of n-dim vectors with values in X. For any index $i \in [1..n]$, we denote by $\mathbf{v}(i)$ the i^{th} component of a n-dim vector $\mathbf{v}$. We write $\mathbf{0}$ the vector such that $\mathbf{0}(i) = 0$ for all $i \in [1..n]$. The classical order on $\mathbb{Z}^n$ is also denoted $\leq$ and is defined by $\mathbf{v} \leq \mathbf{w}$ if and only if for all $i \in [1..n]$, we have $\mathbf{v}(i) \leq \mathbf{w}(i)$. We also define the operation $+$ over n-dim vectors of integers in the classical way (ie for $\mathbf{v}$, $\mathbf{v}' \in \mathbb{Z}^n$, $\mathbf{v} + \mathbf{v}'$ is defined by $(\mathbf{v} + \mathbf{v}')(i) = \mathbf{v}(i) + \mathbf{v}'(i)$ for all $i \in [1..n]$).
Let $n \in \mathbb{N}$. A subset $S \subseteq \mathbb{N}^n$ is *linear* if there exist $k + 1$ vectors $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_k$ in $\mathbb{N}^n$ such that $S = \{\mathbf{v} \mid \mathbf{v} = \mathbf{v}_0 + \lambda_1.\mathbf{v}_1 + \ldots + \lambda_k.\mathbf{v}_k$ with $\lambda_i \in \mathbb{N}$ for all $i \in [1..k]\}$. A *semi-linear set* is any finite union of linear sets. We extend the notion of semi-linearity to subsets of $Q \times \mathbb{N}^n$ where Q is a finite (non-empty) set. This can be easily done assuming Q is for instance a finite subset of $\mathbb{N}$. For an alphabet Σ, we denote by Σ^* the set of finite words over Σ and ϵ represents the empty word.

2.2 Counter machines

We call a *n-dim guarded translation* (shortly a translation) any function $t : \mathbb{N}^n \rightarrow \mathbb{N}^n$ characterized by $\# \in \{=, \leq, \geq\}^n$, $\mu \in \mathbb{N}^n$ and $\delta \in \mathbb{Z}^n$ such that $dom(t) = \{\mathbf{v} \in$

$\mathbb{N}^n \mid \mathbf{v}\#\mu$ and $\mathbf{v}+\delta \in \mathbb{N}^n\}$ and for all $\mathbf{v} \in dom(t)$, $t(\mathbf{v}) = \mathbf{v}+\delta$. We will sometimes use the encoding $(\#, \mu, \delta)$ to represent a translation. With this notation $\#$ and μ encode a test and δ an update. In the following, T_n will denote the set of the n-dim guarded translations.

Definition 1. *A n-dim* counter machine *(shortly* counter machine*) is a tuple $S = \langle Q, E\rangle$ where Q is a finite set of control states and E is a finite relation $E \subseteq Q \times T_n \times Q$.*

The semantics of a counter machine $S = \langle Q, E\rangle$ is given by its associated transition system $TS(S) = \langle Q \times \mathbb{N}^n, \rightarrow\rangle$ where $\rightarrow \subseteq (Q \times \mathbb{N}^n) \times T_n \times (Q \times \mathbb{N}^n)$ is a relation defined as follows :$(q, \mathbf{v}) \xrightarrow{t} (q', \mathbf{v}')$ iff $\exists\ (q, t, q') \in E$ such that $\mathbf{v} \in dom(t)$ and $\mathbf{v}' = t(\mathbf{v})$. We write $(q, \mathbf{v}) \rightarrow (q', \mathbf{v}')$ if there exists $t \in T_n$ such that $(q, \mathbf{v}) \xrightarrow{t} (q', \mathbf{v}')$. The relation $\rightarrow^*$ represents the reflexive and transitive closure of $\rightarrow$. Given a configuration $(q, \mathbf{v})$ of $TS(S)$, $\texttt{Reach}(S, (q, \mathbf{v})) = \{(q', \mathbf{v}') \mid (q, \mathbf{v}) \rightarrow^* (q', \mathbf{v}')\}$. Given a counter machine $S = \langle Q, E\rangle$ and an initial configuration $c_0 \in Q \times \mathbb{N}^n$, the pair (S, c_0) is an intialized counter machine. Since, the notations are explicit, in the following we shall write counter machine for both (S, c_0) and S.

Definition 2. *A counter machine (S, c_0) is bounded if there exists $k \in \mathbb{N}$ such that for all $(q, \boldsymbol{v}) \in \texttt{Reach}(S, c_0)$ and for all $i \in [1..n]$, we have $\boldsymbol{v}(i) \leq k$.*

Note that a counter machine has a finite number of reachable configurations if and only if it is bounded. It is well-known [15] that many verification problems, such as the reachability of a control state or the boundedness, are undecidable for 2-dim counter machines. We present in the sequel some restricted classes of counter machines for which these problems become decidable.

2.3 VASS with one zero-test

Definition 3. *A n-dim counter machine$\langle Q, E\rangle$ is a* Vector Addition System with States *(shortly VASS) if for all transitions $(q, t, q') \in E$, t is a guarded translation $(\#, \mu, \delta)$ such that $\# = (\geq, \ldots, \geq)$,*

Hence in VASS, it is not possible to test if a counter value is equal to a constant but only if it is greater than a constant. In opposite to general counter machines, many problems are decidable for VASS, for instance the problem of the reachability of a configuration or the boundedness [11,12,14]. We finally present here another class of counter machine, which extends the VASS.

Definition 4. *A n-dim counter machine $S = \langle Q, E\rangle$ is a* VASS with one zero-test *if $Q = Q' \uplus \{q_{?0}, q_{=0}\}$, $E = E_{\geq} \uplus \{(q_{?0}, g_{=0?}, q_{=0})\}$ where $g_{=0?}$ is the guarded translation $((=, \geq, \ldots, \geq), \boldsymbol{0}, \boldsymbol{0})$ and $S_{\geq} = \langle Q' \uplus \{q_{?0}\}, E_{\geq}\rangle$ is a VASS.*

Without loss of generality we will impose that the transition $(q_{?0}, g_{=0?}, q_{=0})$ is the only transition leading to $q_{=0}$ in S. We see that in a VASS with one zero-test, the transition $(q_{?0}, g_{=0?}, q_{=0})$ has the ability to test if the value of the first counter is 0. Even if this class of counter machines has been less studied than VASS, it has been proved in [18] that the problem of reachability of a configuration is decidable for it. Note that we restrict this class of machines to use only one transition with an equality test over the first counter, but this is only to improve the readability of our work. In fact, our results still hold for any VASS with more than one zero-test on the first counter.

If we define the $\leq_1$ in $\mathbb{N}^n \times \mathbb{N}^n$ as follows, $\mathbf{v} \leq_1 \mathbf{v}'$ if and only if $\mathbf{v} \leq \mathbf{v}'$ and $\mathbf{v}(1) = \mathbf{v}'(1)$, then VASS with one zero-test enjoy the following property:

Lemma 5. *Let $S = \langle Q, E \rangle$ be a n-dim VASS with one zero-test and $TS(S) = \langle Q \times \mathbb{N}^n, \rightarrow \rangle$ its associated transition system. We consider $(q, \boldsymbol{v}), (q, \boldsymbol{w})$ in $Q \times \mathbb{N}^n$, if $\boldsymbol{v} \leq_1 \boldsymbol{w}$ and if there exists $(q', \boldsymbol{v}') \in Q \times \mathbb{N}^n$ such that $(q, \boldsymbol{v}) \rightarrow (q', \boldsymbol{v}')$ then there exists $(q', \boldsymbol{w}') \in Q \times \mathbb{N}^n$ such that $(q, \boldsymbol{w}) \rightarrow (q', \boldsymbol{w}')$ and $\boldsymbol{v}' \leq_1 \boldsymbol{w}'$.*

In the following, we will propose a new method to analyze VASS with one zero-test.

2.4 Reversal-bounded counter machines

In [10], the class of reversal-bounded counter machines has been introduced as follows: each counter can only perform a bounded number of alternations between increasing and decreasing mode. This class of counter machines is interesting because it has been shown that these machines have a semi-linear reachability set which can be effectively computed. We recall here their formal definition.
Let $S = \langle Q, E \rangle$ be a n-dim counter machine and $TS(S) = \langle Q \times \mathbb{N}^n, \rightarrow \rangle$ its associated transition system . From it, we define another transition system $TS_{rb}(S) = \langle Q \times \mathbb{N}^n \times \{\downarrow, \uparrow\}^n \times \mathbb{N}^n, \rightarrow_{rb} \rangle$. For a configuration $(q, \mathbf{v}, \mathbf{m}, \mathbf{r}) \in Q \times \mathbb{N}^n \times \{\downarrow, \uparrow\}^n \times \mathbb{N}^n$, the vector $\mathbf{v}$ contains the values of each counter, the vector $\mathbf{m}$ is used to store the current mode of each counter -increasing ($\uparrow$) or decreasing ($\downarrow$)- and the vector $\mathbf{r}$ the numbers of alternations performed by each counter. A formal definition of $TS_{rb}(S)$ is given in [8]. We denote by $\rightarrow_{rb}^*$ the reflexive and transitive closure of $\rightarrow_{rb}$. Given a configuration $(q, \mathbf{v}, \mathbf{r}, \mathbf{m})$ of $TS_{rb}(S)$, $\mathtt{Reach}_{rb}(S, (q, \mathbf{v}, \mathbf{m}, \mathbf{r})) = \{(q', \mathbf{v}', \mathbf{m}', \mathbf{r}') \mid (q, \mathbf{v}, \mathbf{m}, \mathbf{r}) \rightarrow_{rb}^* (q', \mathbf{v}', \mathbf{m}', \mathbf{r}')\}$. We extend this last notation to the configurations of $TS(S)$, saying that if $(q, \mathbf{v}) \in Q \times \mathbb{N}^n$ is a configuration of $TS(S)$, then $\mathtt{Reach}_{rb}(S, (q, \mathbf{v}))$ is equal to the set $\mathtt{Reach}_{rb}(S, (q, \mathbf{v}, \uparrow, \mathbf{0}))$ where $\uparrow$ denotes here the vector with all components equal to $\uparrow$.

Definition 6. *A counter machine (S, c_0) is* reversal-bounded *if and only if there exists $k \in \mathbb{N}$ such that for all $(q, \boldsymbol{v}, \boldsymbol{m}, \boldsymbol{r}) \in \mathtt{Reach}_{rb}(S, c_0)$ and for all $i \in [1..n]$, we have $\boldsymbol{r}(i) \leq k$.*

Using a translation into a finite automaton and the fact that the Parikh map of a regular language is a semi-linear set [16], Ibarra proved in [10] the following result:

Theorem 7. *[10] The reachability set of a reversal-bounded counter machine is an effectively computable semi-linear set.*

3 Computing coverability

3.1 Coverability graph of a VASS

In [11], Karp and Miller provide an algorithm to build from a VASS a labeled tree, the *Karp and Miller tree*. We recall here the construction of this tree. We first define a function $\texttt{Acceleration} : \mathbb{N}_\omega^n \times \mathbb{N}_\omega^n \rightarrow \mathbb{N}_\omega^n$ as follows, for $\mathbf{w}, \mathbf{w}' \in \mathbb{N}_\omega^n$ such that $\mathbf{w} \leq \mathbf{w}'$, we have $\mathbf{w}'' = \texttt{Acceleration}(\mathbf{w}, \mathbf{w}')$ if and only if for all $i \in [1..n]$:

- if $\mathbf{w}(i) = \mathbf{w}'(i)$ then $\mathbf{w}''(i) = \mathbf{w}(i)$,
- if $\mathbf{w}(i) < \mathbf{w}'(i)$ then $\mathbf{w}''(i) = \omega$.

The Karp and Miller tree is a labeled tree (P, δ, r, l) where P is a finite set of nodes, $\delta \subseteq P \times T_n \times P$ is the transition relation, $r \in P$ is the root of the tree, and $l : P \rightarrow Q \times \mathbb{N}_\omega^n$ is a labeling function. To represent a node p with the label $l(p) = (q, \mathbf{w})$, we will sometimes directly write $p[q, \mathbf{w}]$. The Algorithm 1 shows how the Karp and Miller tree is obtained from an initialized VASS.

Algorithm 1 $T = \texttt{KMT}(\langle Q, E\rangle, c_0)$

Input : $(\langle Q, E\rangle, c_0)$ an initialized VASS;
Output : $T = \langle P, \delta, r, l\rangle$ the Karp and Miller tree;
$P = \{r\}, \delta = \emptyset, l(r) = c_0$
$ToBeTreated = \{r\}$
while $ToBeTreated \neq \emptyset$ **do**
  Choose $p[q, \mathbf{w}] \in ToBeTreated$
  if there does not exist a predecessor $p'[q, \mathbf{w}]$ of p in T **then**
    for each $(q, (\#, \mu, \delta), q') \in E$ **do**
      if $\mu \leq \mathbf{w}$ **then**
        let $\mathbf{w}' = \mathbf{w} + \delta$
        if there exists a predecessor $p'[q', \mathbf{w}'']$ of p in T such that $\mathbf{w}'' < \mathbf{w}'$ **then**
          let $\mathbf{w}' = \texttt{Acceleration}(\mathbf{w}'', \mathbf{w}')$
        end if
        Add a new node p' to P such that $l(p') = (q', \mathbf{w}')$
        Add $(p, \langle \#, \mu, \delta\rangle, p')$ to δ
        Add p' to $ToBeTreated$
      end if
    end for
  end if
  Remove p of $ToBeTreated$
end while

The main idea of this tree is to cover in a finite way the reachable configurations using the symbol ω, when a counter is not bounded. It has been proved that the Algorithm 1 always terminates and that the produced tree enjoys some good properties. In particular, this tree can be used to decide the boundedness of a VASS. In [19], Vack and Vidal-Naquet have proposed a further construction based on the Karp and Miller tree in order to test the regularity of the language of the unlabeled traces of a VASS. This last construction is known as the *coverability graph*. To obtain it, the nodes of the Karp and Miller tree with the same label are gathered in an unique node. If (S, c_0) is a n-dim VASS, we denote by $\texttt{KMG}(S, c_0)$ its coverability graph.

For a vector $\mathbf{w} \in \mathbb{N}_\omega^n$, we denote by $\mathtt{Inf}(\mathbf{w})$ the set $\{i \in [1..n] \mid \mathbf{w}(i) = \omega\}$ and $\mathtt{Fin}(\mathbf{w}) = [1..n] \setminus \mathtt{Inf}(\mathbf{w})$. Using these notions, it has been proved that the coverability graph satisfies the following properties.

Theorem 8. *[11,19] Let (S, c_0) be a n-dim VASS with $S = \langle Q, E\rangle$, $TS(S) = \langle Q \times \mathbb{N}^n, \rightarrow\rangle$ its associated transition system and $\mathtt{KMG}(S, c) = \langle P, \delta, r, l\rangle$ its coverability graph.*

1. *If $p[q, \boldsymbol{w}]$ is a node in $\mathtt{KMG}(S, c_0)$, then for all $k \in \mathbb{N}$, there exists $(q, \boldsymbol{v}) \in \mathtt{Reach}(S, c_0)$ such that for all $i \in \mathtt{Inf}(\boldsymbol{w})$, $k \leq \boldsymbol{v}(i)$ and for all $i \in \mathtt{Fin}(\boldsymbol{w})$, $\boldsymbol{w}(i) = \boldsymbol{v}(i)$.*
2. *For $\sigma \in T_n^*$, if $c \xrightarrow{\sigma} (q, \boldsymbol{v})$ then there is a unique path in $\mathtt{KMG}(S, c_0)$ labeled by σ and leading from r to a node $p[q, \boldsymbol{w}]$ and for all $i \in \mathtt{Fin}(\boldsymbol{w})$, $\boldsymbol{v}(i) = \boldsymbol{w}(i)$.*

3.2 Minimal covering set

We present here the notion of minimal covering set of a set, notion that we will use later to build the coverability graph of a reversal-bounded VASS with one zero-test. The minimal covering set of a possibly infinite set of vectors is the smallest set of vectors which cover all the vectors belonging to the considered set. Before giving its definition, we introduce some notations.
If $V \subseteq \mathbb{N}^n$, we denote by $\mathtt{Inc}(V)$, the set of the increasing sequences of elements of V. Each $(v_n)_{n\in\mathbb{N}} \in \mathtt{Inc}(V)$ has a least upper bound in $\mathbb{N}_\omega^n$ denoted $\mathtt{lub}((v_n)_{n\in\mathbb{N}})$. We then define the set $\mathtt{Lub}(V)$ of elements of $\mathbb{N}_\omega^n$ as the set $\{\mathtt{lub}(v_n)_{n\in\mathbb{N}} \mid (v_n)_{n\in\mathbb{N}} \in \mathtt{Inc}(V)\}$. Note that in [6], this last set is defined using the least upper bound of the directed subsets of V, but in the case of vectors of integers, it is equivalent to use the set of increasing sequences. If we consider the maximal elements of $\mathtt{Lub}(V)$ under the classical order over $\mathbb{N}_\omega^n$, we obtain what is called the minimal covering set of V.

Definition 9. *[5,6] Let $n \in \mathbb{N} \setminus \{0\}$ and $V \subseteq \mathbb{N}^n$. The* minimal covering set *of V, denoted by $\mathtt{MinCover}(V)$, is the set $\mathtt{Max}(\mathtt{Lub}(V))$.*

Using the definition of $\mathtt{MinCover}(V)$ and the fact that $(\mathbb{N}_\omega^n, \leq)$ is a well-quasi-order, we have the following proposition.

Proposition 10. *[5] Let $V \subseteq \mathbb{N}^n$. We have then:*

- *$\mathtt{MinCover}(V)$ is finite, and,*
- *for all $\boldsymbol{u} \in \mathtt{MinCover}(V)$, $\forall k \in \mathbb{N}$, there exists $\boldsymbol{v} \in V$ such that $\forall i \in \mathtt{Fin}(\boldsymbol{u})$, $\boldsymbol{v}(i) = \boldsymbol{u}(i)$ and $\forall i \in \mathtt{Inf}(v)$, $k \leq \boldsymbol{v}(i)$ and,*
- *for all $\boldsymbol{v} \in V$, there exists $\boldsymbol{u} \in \mathtt{MinCover}(V)$ such that $\boldsymbol{v} \leq \boldsymbol{u}$.*

Furthermore, for what concerns the minimal covering set of a semi-linear set, we have the following result.

Lemma 11. *Given a semi-linear set L, the set $\mathtt{MinCover}(L)$ can effectively be computed.*

Note that this last result can not be extended to any recursive set V. In fact if we were able to compute the minimal covering set of a recursive set V, we would be able to deduce if it is finite or not, which is known as an undecidable problem (this being a consequence of Rice's theorem).

4 Decidability results for VASS with one zero-test

4.1 Counting the number of alternations in a VASS

In [10], it has been proved that the problem to decide whether a counter machine is reversal-bounded or not is undecidable, but this problem becomes decidable when considering VASS [7]. We recall here how this last result is obtained.
Let $S = \langle Q, E\rangle$ be a n-dim counter machine. We build a $2n$-dim counter machine $\widetilde{S} = \langle Q \times \{\uparrow, \downarrow\}^n, E'\rangle$ in which the n-th last counters count the alternations between increasing and decreasing modes of the n-th first counters. A formal definition of E' is is given in [8]. Note that by construction, since we never test the values of the added counters, if S is a VASS then $\widetilde{S}$ is a VASS too. For an initial configuration $c_0 = (q_0, \mathbf{v}_0)$, if we denote by $\widetilde{c}_0$ the pair $((q_0, \uparrow), (\mathbf{v}_0, \mathbf{0}))$, we have the following proposition:

Proposition 12. *A n-dim counter machine (S, c_0) is reversal-bounded if and only if there exists $k \in \mathbb{N}$ such that for all $((q, \boldsymbol{m}), \boldsymbol{v}) \in \texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$ and for all $i \in [1..n]$, $\boldsymbol{v}(n + i) \leq k$.*

Using the result of Theorem 8, we deduce that a VASS (S, c_0) is reversal-bounded if and only if for all nodes $p[q, \mathbf{w}]$ of the coverability graph of $(\widetilde{S}, \widetilde{c}_0)$ and for all $i \in [1..n]$, $\mathbf{w}(n + i) \neq \omega$. Hence:

Theorem 13. *[7] Reversal-boundedness is decidable for VASS.*

In the sequel, we will see how this method can be adapted to the case of VASS with one zero-test, which will allow us to extend the result of the previous theorem.

4.2 Mixing the coverability graph and reachability analysis

In this section, we will give an algorithm to build a labeled graph which will provide us a necessary and sufficient condition to decide whether a VASS with one zero-test is reversal-bounded. The classical construction of the Karp and Miller Tree cannot be used in the case of VASS with one zero-test, because when we introduce the symbol ω for the counter which might be tested to zero, we do not know for which values this ω stands for, and hence it is not possible to evaluate the test to zero when it occurs.

Let (S, c_0) be a n-dim counter machine with $S = \langle Q, E\rangle$. We define a (S, c_0)-labeled graph G as a tuple $\langle P, \delta, r, l\rangle$ where P is a set of nodes, $\delta \subseteq P \times T_n \times P$ is a set of edges labeled with guarded commands, $r \in P$ is the initial node and $l : P \to Q \times N^n_\omega$ is a labeling function such that $l(r) = c_0$. If $G = \langle P, \delta, r, l\rangle$ is a (S, c_0)-labeled graph, then $\langle P, \delta\rangle$ defines a counter machine we will denote by S_G. Furthermore to S_G we associate the initial configuration $r_0 = (r, \mathbf{v}_0)$ where $\mathbf{v}_0$ is the valuation function associated to c_0.
In the sequel we will consider a n-dim VASS with one zero-test (S, c_0) and its associated $2n$-dim counter machine $(\widetilde{S}, \widetilde{c}_0)$ in which we count the alternations between increasing and decreasing mode. Note that since when we build $\widetilde{S}$ we only introduce counters which never decrease, we have that (S, c_0) is reversal-bounded if and only

Algorithm 2 $G = \texttt{CoverGraph}(S, c_0)$

Input : (S, c_0) VASS with one zero-test
Output : $G = \langle P, \delta, r, l \rangle$ a graph

$HasChanged = True$ *\/*This boolean becomes True when G is changed*\/*
Compute $(\widetilde{S}, \widetilde{c}_0)$ *\/*See the definition on the previous page*\/*
$\langle P, \delta, r, l \rangle = \texttt{KMG}(\widetilde{S}_{\geq}, \widetilde{c}_0)$ *\/*$\widetilde{S}_{\geq}$ is a VASS obtained from $\widetilde{S}$ deleting the zero-tests*\/*
$G = \langle P, \delta, r, l \rangle$
while $HasChanged = True$ and $\texttt{ConditionRB}(G) = True$ **do**
    $HasChanged = False$
    for each $p[(q_{?0}, \mathbf{m}), \mathbf{u}] \in P$ **do**
        $V_p = \{\mathbf{v} \mid (p, \mathbf{v}) \in \texttt{Reach}(S_G, r_0) \wedge \mathbf{v}(1) = 0\}$
        \/(S_G, r_0) is the counter machine obtained from G *\/*
        Compute $\texttt{MinCover}(V_p)$
        for each $\mathbf{u} \in \texttt{MinCover}(V_p)$ **do**
            if there exists a predecessor[1] $p'[(q_{=0}, \mathbf{m}), \mathbf{u}']$of p such that $\mathbf{u}' \leq_1 \mathbf{u}$ **then**
                $\mathbf{u} = \texttt{Acceleration}(\mathbf{u}', \mathbf{u})$
            end if
            if there is no one-step successor[2] $p''[(q_{=0}, \mathbf{m}), \mathbf{u}'']$of p such that $\mathbf{u} \leq \mathbf{u}''$ **then**
                $HasChanged = True$
                Let $t \in T_{2n}$ with $dom(t) = \{\mathbf{v} \mid \forall i \in \texttt{Fin}(\mathbf{u}).\ \mathbf{v}(i) \ \leq \mathbf{u}(i)\}$ and $\forall \mathbf{v}. t(\mathbf{v}) = \mathbf{v}$
                if there exists a predecessor $p'''[(q_{=0}, \mathbf{m}), \mathbf{u}]$of p **then**
                    Add (p, t, p''') to δ
                else
                    Add a new node $newp[(q_{=0}, \mathbf{m}), \mathbf{u}]$ to P
                    Add $(p, t, newp)$ to δ
                    $G' = \texttt{KMG}(\widetilde{S}_{\geq}, ((q_{=0}, \mathbf{m}), \mathbf{u}))$
                    Add G' to G merging the root node of G' and $newp$
                end if
            end if
        end for
    end for
end while

if $(\widetilde{S}, \widetilde{c}_0)$ is reversal-bounded. As for S, we denote by $\widetilde{S}_{\geq}$ the $2n$-dim VASS obtained from $\widetilde{S}$ removing all the transitions of the form $(q, g_{=0?}, q')$.

We propose the Algorithm 2 to build a partial coverability graph of $(\widetilde{S}, \widetilde{c}_0)$. We will then use this graph to decide whether the input VASS with one zero-test is reversal-bounded or not. Our algorithm builds a $(\widetilde{S}, \widetilde{c}_0)$-labeled graph G as follows:

- First, we build the coverability graph of $(\widetilde{S}_{\geq}, \widetilde{c}_0)$ and test if $(\widetilde{S}_{\geq}, \widetilde{c}_0)$ is reversal-bounded. The predicate $\texttt{ConditionRB}(G)$ will ensure that.
- If $(\widetilde{S}_{\geq}, \widetilde{c}_0)$ is not reversal-bounded, we can already deduce that $(\widetilde{S}, \widetilde{c}_0)$ is not reversal-bounded and we stop our construction.

[1] p' is a predecessor of p if there exists a path in G of length greater than or equal to 1 from p' to p.
[2] p'' is a one step successor of p if there exists t such that $(p, t, p'') \in \delta$.

– If $(\widetilde{S}_{\geq}, \widetilde{c}_0)$ is reversal-bounded, so is (S_G, r_0). We then compute the reachability set of (S_G, r_0) to know which test to zero will be accepted and we compute the minimal covering set of the vectors we obtain after realizing one test to zero (Lines 8-9 of Algorithm 2).
– From this covering set, we obtain a new set of labeled nodes from which we build again the coverability graph of $\widetilde{S}_{\geq}$. Doing so we complete the graph G. We then again test if (S_G, r_0) is reversal-bounded and if it is the case we proceed as previous considering again all the nodes from which a zero-test is done.
– Finally, in order to ensure termination (in case $\texttt{ConditionRB}(G)$ is always evaluated to $True$) we insert ω when computing the reachability set for the zero-test we encounter a covering vector bigger than a preceding one (Line 11-12 of Algorithm 2).

An example of the result of the computation of Algorithm 2 is provided in [8].

We will now analyze more formally the Algorithms 2. First, we define the condition $\texttt{ConditionRB}(G)$ for a $(\widetilde{S}, \widetilde{c}_0)$-labeled graph G as follows: $\texttt{ConditionRB}(G) = True$ if and only if for all nodes $p[q, \mathbf{u}]$ of G, for all $i \in [1..n]$, we have $\mathbf{u}(n+i) \neq \omega$.

Note that the first graph we compute being the coverability graph of $(\widetilde{S}_{\geq}, \widetilde{c}_0)$, according to Proposition 12, we have that $(\widetilde{S}_{\geq}, \widetilde{c}_0)$ is reversal-bounded if and only if the predicate $\texttt{ConditionRB}(G)$ is true.

In order to prove that our algorithm is correct, we need to prove the following points:

1. If G is a $(\widetilde{S}, \widetilde{c}_0)$-labeled graph computed during the execution of the Algorithm 2 and if $\texttt{ConditionRB}(G) = True$ then (S_G, r_0) is reversal-bounded,
2. For any VASS with one zero-test (S, c_0), the algorithm $\texttt{CoverGraph}(S, c_0)$ terminates.

The first point is a sufficient condition which allows us to compute effectively the set V_p at Line 8 of the Algorithm 2 and also the set $\texttt{MinCover}(V_p)$. In fact, if (S_G, r_0) is reversal-bounded, according to Theorem 7, the set $\texttt{Reach}(S_G, r_0)$ is an effectively computable semi-linear set and consequently so is the corresponding set $V_p = \{\mathbf{v} \mid (p, \mathbf{v}) \in \texttt{Reach}(S_{G_i}, r_0) \wedge \mathbf{v}(1) = 0\}$, and hence from Lemmas 10 and 11, we also deduce that $\texttt{MinCover}(V_p)$ is finite and can be effectively computed.

Let (S, c_0) be a n-dim VASS with one zero-test and let $G = (P, \delta, r, l)$ be a $(\widetilde{S}, \widetilde{c}_0)$-labeled graph obtained at Line 4 after some iterations of the loop of Algorithm 2. We recall that by construction, $l(r) = \widetilde{c}_0$ and that the initial configuration r_0 of the counter machine S_G associated to the graph G is the pair $(r, \mathbf{v}_0)$ where $\mathbf{v}_0$ is the vector associated to the configuration $\widetilde{c}_0$. We have then the following lemma:

Lemma 14. *For all $(p, \boldsymbol{v}) \in \texttt{Reach}(S_G, r_0)$, if $l(p) = (q, \boldsymbol{u})$, then for all $i \in [1..2n]$, we have $\boldsymbol{v}(i) \leq \boldsymbol{u}(i)$.*

Since by construction in the counter machine S_G the n-th last counters count the numbers of alternations of the n-th first counters and since the graph G has a finite number of nodes we deduce that if $\texttt{ConditionRB}(G) = True$ then there exists a constant k which bounds the number of alternations for each counter in (S_G, r_0), consequently:

Proposition 15. *If $\texttt{ConditionRB}(G) = True$ then (S_G, r_0) is reversal-bounded.*

At Line 8 of Algorithm 2, when we compute the reachability set $\texttt{Reach}(S_G, r_0)$, we are hence sure that the counter machine (S_G, r_0) is effectively reversal-bounded and according to Theorem 7 and Lemma 11 we can effectively compute this set and also its minimal covering set. Finally, we have the following proposition:

Proposition 16. *The Algorithm 2 always terminates when its input is a VASS with one zero-test.*

Idea of proof: This is ensured by the fact that if the algorithm does not terminate we can extract an infinite sequence of vectors $(\mathbf{u}_i)_{i\in\mathbb{N}}$ such that for all $i \in \mathbb{N}$, $\mathbf{u}_i(1) = 0$ and $\mathbf{u}_i$ belongs to a node predecessor of the node containing $\mathbf{u}_{i+1}$ and for all $i, j \in \mathbb{N}$, $\mathbf{u}_i \neq \mathbf{u}_j$. Using that $\{\mathbf{u} \in \mathbb{N}^{2n}_\omega \mid \mathbf{u}(1) = 0\}$ together with the order $\leq_1$ is a well-quasi order, we deduce that we can extract from this sequence an infinite strictly increasing sequence of vectors, but this is not possible because if $\mathbf{u}$ preceeds $\mathbf{u}'$ in this sequence then there are strictly more components equal to ω in $\mathbf{u}'$ than in $\mathbf{u}$ (thanks to the function $\texttt{Acceleration}$) and so this sequence cannot be infinite. □

4.3 Reversal-boundedness

In this last section, we will show how to use the $(\widetilde{S}, \widetilde{c}_0)$-labeled graph produced in output of the Algorithm 2 to decide whether the VASS with one zero-test (S, c_0) is reversal-bounded or not. First, we will show that if the graph $G = \texttt{CoverGraph}(S, c_0)$ is such that $\texttt{ConditionRB}(G) = False$, i.e. there exists a node $p[q, \mathbf{u}]$ and $i \in [1..n]$ such that $\mathbf{u}(n + i) = \omega$ then the counter machine $(\widetilde{S}, \widetilde{c}_0)$ is not reversal-bounded. This is due to the following lemma:

Lemma 17. *If $p[q, \boldsymbol{u}]$ is a node of $\texttt{CoverGraph}(S, c_0)$ and if $i \in \texttt{Inf}(\boldsymbol{u})$ then for all $k \in \mathbb{N}$, there exists a configuration $(q, \boldsymbol{v}) \in \texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$ such that $\boldsymbol{v}(i) > k$.*

We then prove that if $G = \texttt{CoverGraph}(S, c_0)$ is such that $\texttt{ConditionRB}(G) = True$ then $(\widetilde{S}, \widetilde{c}_0)$ and hence (S, c_0) are reversal-bounded. To prove this, we have to prove that for each reachable configuration $(q, \mathbf{v})$ of $(\widetilde{S}, \widetilde{c}_0)$ there is a node of G which "covers" this configuration. This point can be proved by induction on the length of any execution of $\texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$ and leads to the following lemma:

Lemma 18. *If $G = \texttt{CoverGraph}(S, c_0)$ is such that $\texttt{ConditionRB}(G) = True$ then for all configurations $(q, \boldsymbol{v}) \in \texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$, there exists a node $p[q', \boldsymbol{u}]$ in G such that $q = q'$ and for all $i \in [1..2n]$, $\boldsymbol{v}(i) \leq \boldsymbol{u}(i)$.*

Using the two previous lemma and the result of Proposition 12, we deduce the following theorem:

Theorem 19. *A n-dim VASS with one zero-test (S, c_0) is reversal-bounded if and only if for all nodes $p[q, \mathbf{u}]$ in $\texttt{CoverGraph}(S, c_0)$, for all $i \in [1..n]$, $\mathbf{u}(n+i) \neq \omega$.*

Proof: Assume there exists a node $p[q, \mathbf{u}]$ in $\texttt{CoverGraph}(S, c_0)$ and $i \in [1..n]$, such that $\mathrm{u}(n+i) = \omega$. Then according to Lemma 17 for all $k \in \mathbb{N}$, there exists a configuration $(q, \mathbf{v}) \in \texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$ such that $\mathbf{v}(n+i) > k$, hence using Proposition 12, we deduce that (S, c_0) is not reversal-bounded. If for all nodes $p[q, \mathbf{u}]$ in $\texttt{CoverGraph}(S, c_0)$, for all $i \in [1..n]$, $\mathbf{u}(n+i) \neq \omega$, since the number of nodes in $\texttt{CoverGraph}(S, c_0)$ is finite, we can find a $k \in \mathbb{N}$ such that for all nodes $p[q, \mathbf{u}]$ in $\texttt{CoverGraph}(S, c_0)$, for all $i \in [1..n]$, $\mathbf{u}(n+i) \leq k$. Using lemma 18, we deduce that for all configurations $(q, \mathbf{v}) \in \texttt{Reach}(\widetilde{S}, \widetilde{c}_0)$, for all $i \in [1..n]$, we have $\mathbf{v}(n+i) \leq k$, hence according to Proposition 12 (S, c_0) is reversal-bounded. $\square$

Consequently, we obtain that:

Corollary 20. *Reversal-boundedness is decidable for VASS with one zero-test.*

4.4 Boundedness and termination

We can adapt the reasoning we have performed to decide reversal-boundedness in order to decide boundedness for VASS with one zero-test. In fact, we still use the Algorithm 2 but instead of building the coverability graph of $(\widetilde{S}, \widetilde{c}_0)$, we build directly the one of (S, c_0) and instead of using condition $\texttt{ConditionRB}$, we use the following condition on a (S, c_0)-labeled graph G:

- for all $i \in [1..n]$, for all nodes $p[q, \mathbf{u}]$ of G, we have $\mathbf{u}(i) \neq \omega$.

The idea here is exactly the same as for reversal-boundedness. In fact, at the first step of the Algorithm 2, the coverability graph of $(S_{\geq}, c_0)$ is computed and it can be directly tested if this VASS is bounded or not. If it is not bounded, the algorithm stops, because all the executions in $(S_{\geq}, c_0)$, are also executions in (S, c_0) and hence if $(S_{\geq}, c_0)$ is not bounded, then (S, c_0) is also not bounded. In the other case, if $(S_{\geq}, c_0)$ is bounded, then its coverability graph corresponds exactly to its reachability graph. The algorithm can then proceed its computation exactly as for deciding reversal-boundedness. This consideration allows us to deduce the following result:

Theorem 21. *Boundedness is decidable for VASS with one zero-test.*

Note that this implies also the decidability of the termination problem for VASS with one zero-test. In fact, the termination problem for counter machines, which consists in deciding whether the counter machine has an infinite execution or not, can be reduced easily to the boundedness. This is due to the following consideration: if a counter machine is not bounded, then it has an infinite execution and if it is bounded, then it is possible to build its reachability graph and hence to decide whether there exists an infinite execution or not.

Corollary 22. *Termination is decidable for VASS with one zero-test.*

5 Conclusion

In this paper, we have provided an original method to decide whether a VASS extended with one-zero test is reversal-bounded (resp. bounded) or not. The main idea consists in mixing the construction of the classical coverability graph for VASS and the computing of the reachability set of reversal-bounded VASS. In the future, we would like to continue our investigation on methods to analyze this class of system and our aim would be to find a construction of a complete coverability graph for VASS with one-zero test. This would in particular gives us a way to decide the problem of place-boundedness which consists in deciding whether a set of counters has bounded values or not. In fact, the method we present in this paper does not allow us to solve this problem, because the graph we build is partial and the construction stops whenever it encounters a non reversal-bounded (resp. non bounded) behavior.

References

1. P. A. Abdulla and R. Mayr. Minimal cost reachability/coverability in priced timed Petri nets. In *FoSSaCS'09*, volume 5504 of *LNCS*, pages 348–363. Springer, 2009.
2. C. Dufourd. *Réseaux de Petri avec Reset/Transfert : décidabilité et indécidabilité*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, 1998.
3. C. Dufourd and A. Finkel. Polynomial-time many-one reductions for Petri nets. In *FSTTCS'97*, volume 1346 of *LNCS*, pages 312–326. Springer, 1997.
4. J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundam. Inform.*, 31(1):13–25, 1997.
5. A. Finkel. The minimal coverability graph for Petri nets. In *APN'91*, volume 674 of *LNCS*, pages 210–243. Springer, 1993.
6. A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part II: Complete WSTS. In *ICALP'09*, volume 5556 of *LNCS*. Springer, 2009.
7. A. Finkel and A. Sangnier. Reversal-bounded counter machines revisited. In *MFCS'08*, volume 5162 of *LNCS*, pages 323–334. Springer, 2008.
8. A. Finkel and A. Sangnier. Mixing coverability and reachability to analyze VASS with one zero-test. Research report, Laboratoire Spécification et Vérification, ENS Cachan, 2009.
9. M. Hack. Petri net language. Technical report, Massachusetts Institute of Technology, 1976.
10. O. H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978.
11. R. M. Karp and R. E. Miller. Parallel program schemata: A mathematical model for parallel computation. In *FOCS'67*, pages 55–61. IEEE, 1967.
12. S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *STOC'82*, pages 267–281. ACM, 1982.
13. J. Leroux. The general vector addition system reachability problem by presburger inductive invariants. In *LICS'09*, pages 4–13. IEEE Computer Society Press, 2009.
14. E. W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM J. Comput.*, 13(3):441–460, 1984.
15. M. L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.
16. R. Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.
17. `http://www.informatik.uni-hamburg.de/TGI/PetriNets/`.
18. K. Reinhardt. Reachability in petri nets with inhibitor arcs. *ENTCS*, 223:239–264, 2008.
19. R. Valk and G. Vidal-Naquet. Petri nets and regular languages. *J. Comput. Syst. Sci.*, 23(3):299–325, 1981.