Place-Boundedness for Vector Addition Systems with one zero-test*

Rémi Bonnet[1], Alain Finkel[1], Jérôme Leroux[2], and Marc Zeitoun[1,2]

1 LSV, ENS Cachan, CNRS & INRIA, France.
firstname.lastname@lsv.ens-cachan.fr

2 LaBRI, Univ. Bordeaux & CNRS, France.
firstname.lastname@labri.fr

Abstract

Reachability and boundedness problems have been shown decidable for Vector Addition Systems with one zero-test. Surprisingly, place-boundedness remained open. We provide here a variation of the Karp-Miller algorithm to compute a basis of the downward closure of the reachability set which allows to decide place-boundedness. This forward algorithm is able to pass the zero-tests thanks to a finer cover, hybrid between the reachability and cover sets, reclaiming accuracy on one component. We show that this filtered cover is still recursive, but that equality of two such filtered covers, even for usual Vector Addition Systems (with no zero-test), is undecidable.

Keywords and phrases Vector addition systems; Inhibitor arcs; Karp-Miller algorithms; Reachability sets; Cover sets; Well-quasi orders.

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2010.192

1 Introduction

Context. Petri Nets, Vector Addition Systems (VAS), and Vector Addition Systems with control states (VASS) are equivalent well-known classes of counter systems for which the reachability problem is decidable [19, 17, 18], even if its complexity is still an open problem. On the other hand, testing equality of the reachability sets of two such systems is undecidable [12]. For that reason, one cannot compute a canonical finite representation of the reachability set that would make it possible to test for equality. However, there is such an effective finite representation for the *cover*, a useful over-approximation of the reachability set which is connected to various verification problems.

If we add to VAS the ability to test at least two counters to zero, one obtains a model equivalent to Minsky machines, for which all nontrivial properties are undecidable. The study of VAS with *a single* zero-test transition began recently, and very few results are known for this model. Reinhardt [21] has shown that the reachability problem is decidable for VASS with one zero-test transition (as well as for hierarchical zero-tests). Abdulla and Mayr have shown that the coverability problem is decidable in [2], by using the backward procedure of Well Structured Transition Systems [1]. See [10] for a survey. The boundedness problem (whether the reachability set is finite), the termination and the reversal-boundedness problem (whether the counters can alternate infinitely often between the increasing and the decreasing modes) are all decidable by using a forward procedure, a finite but *non-complete* Karp and Miller tree [9]. The place-boundedness problem, and more generally the possibility to

* Supported by the Agence Nationale de la Recherche, AVERISS (grant ANR-06-SETIN-001) and AVERILES (grant ANR-05-RNTL-002).

IARCS Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010).
Editors: Kamal Lodaya, Meena Mahajan; pp. 192–203
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

compute a finite representation of the cover were still open problems. Only in the particular case of dimension 2 with control states, the reachability set is semilinear and its basis and periods are computable [11] and then the place-boundedness is decidable; but this result cannot be extended in dimension 3, even without zero-test [14].

Our contribution. We give an algorithm for computing a finite representation of the cover for a VAS with one zero-test. This result makes it possible to decide the place-boundedness, which is in general undecidable for VAS extensions (such as VAS with resets [5] or Lossy Minsky machines, *i.e.* Lossy VAS with zero-test transitions [3, 20]). Our proof techniques introduce a filtered cover, an hybrid between the reachability and cover sets, which unlike the cover reclaims *accuracy* on one component. We show that this set is recursive, but that one cannot decide the equality of such filtered covers of two VAS (even without zero-test). Thus, our work is a contribution to understanding the limits of decidability, taking into account two parameters: the models (VAS and VAS with zero-test) and the problems (reachability, cover and filtered cover).

The difficulty. The central problem is to compute the cover of a VAS *with one zero-test*. Let us explain the reasons why the usual Karp and Miller is not sufficient for that purpose. A natural idea appearing in [9] is to adapt the classical Karp-Miller construction [15], first building the Karp-Miller tree, but *without* firing the zero test. To continue the construction after this first stage, we need to fire the zero test from the leaves of the Karp-Miller tree carrying a 0 value on the component tested to 0. The problem is that accelerations performed while building the Karp-Miller tree may have produced, on this component in the label of such a leaf, an ω value which represents infinity, and abstracts actual values. For that reason, one may not be able to determine if the zero test succeeds or not. We therefore want a more accurate information for the labeling of the leaves, for the component tested to 0. This is what the filtered cover actually captures.

The schema of our proof.

1. We start in Section 3 with usual VAS: we extend the decidability of the reachability problem for VAS, in proving that the set $\mathsf{Lim\,Reach}$ of *limits* of increasing sequences of reachable states is also recursive ($\mathsf{Lim\,Reach}$ contains the reachability set). The set $\mathsf{Lim\,Reach}$ is a more sophisticated set than both the cover and the reachability set. It allows us to know whether an element in $(\mathbb{N} \cup \{\omega\})^d$ is a reachable state or is the limit of a sequence of reachable states. This information is not given by the reachability set neither by the cover. The proof carries on by using Higman's Lemma, using a nontrivial ordering.
2. In Section 4, we refine the definition of the cover in which the first component has now to be exactly known (and not only bounded by a maximum). We prove that, for VAS, a finite basis of this filtered cover is *still computable* by using the recursivity of $\mathsf{Lim\,Reach}$.
3. We finally compute in Section 5 the finite basis of the cover of a VAS with one zero-test by using a variation of the Karp and Miller algorithm that uses the previously defined filtered covers in order to convey enough information to go through the zero-test.

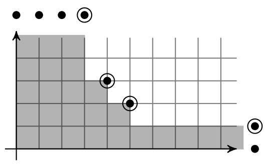
Due to lack of space, some proofs are omitted.

2 Vector Addition Systems

Orderings and vectors. An *ordering* $\preccurlyeq$ on a set X is a reflexive, transitive and antisymmetric binary relation on X. Given $x, y \in X$, we write $x \prec y$ for $x \preccurlyeq y$ and $x \neq y$. For $d \geqslant 1$, we write any $\boldsymbol{x} \in X^d$ as $\boldsymbol{x} = (\boldsymbol{x}(1), \ldots, \boldsymbol{x}(d))$, with $\boldsymbol{x}(i) \in X$. The *pointwise ordering* on X^d, still denoted $\preccurlyeq$, is defined by $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$ if $\boldsymbol{x}(i) \preccurlyeq \boldsymbol{y}(i)$ for all i. For $\boldsymbol{x_1} \in X^{d_1}$ and $\boldsymbol{x_2} \in X^{d_2}$, we let $(\boldsymbol{x_1}, \boldsymbol{x_2}) \in X^{d_1+d_2}$ be the vector obtained by gluing $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. For $X = \mathbb{N}$, let $\mathbf{0}$ be the vector whose components are all 0, and for $i \in \{1, \ldots, d\}$, let $\boldsymbol{e_i}$ be the vector such that $\boldsymbol{e_i}(i) = 1$ and $\boldsymbol{e_i}(k) = 0$ if $k \neq i$. Finally, given $Y \subseteq X$, let $\downarrow_{\preccurlyeq} Y = \{x \in X \mid \exists y \in Y,\ x \preccurlyeq y\}$ denote the downward closure of Y with respect to $\preccurlyeq$. The set Y is said *downward closed* if $Y = \downarrow_{\preccurlyeq} Y$. When working in $\mathbb{N}^d$ or $\mathbb{N}^d_\omega$ (see below) we shorten the downward closure operator $\downarrow_{\leqslant}$ as $\downarrow$.

Downward closed sets of $\mathbb{N}^d$. Given an ordered set, one may under suitable hypotheses construct a topological completion of this set to recover a *finite description* of downward closed sets [7, 8]. The completion of $\mathbb{N}^d$ is $\mathbb{N}^d_\omega$, with $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$, where we extend $\leqslant$ by $n \leqslant \omega$ for all $n \in \mathbb{N}_\omega$. The results of [7, 8] in this case yield that, if $D \subseteq \mathbb{N}^d$ is downward closed, then $D = \mathbb{N}^d \cap \downarrow B$ for some finite set $B \subseteq \mathbb{N}^d_\omega$, which we call a (finite) *basis* of D. One can show that the maximal elements of any basis B of D still form a basis which does not depend of B. It is minimal for inclusion among all basis, and is called the *minimal basis*.

An example. Let us consider in $\mathbb{N}^2$ the downward closed set $\{(x, y) \in \mathbb{N}^2 \mid x \leqslant 3 \vee y \leqslant 1\} \cup \{(4,2), (4,3), (5,2)\}$. A (non-minimal) basis is $(\{0,1,2,3\} \times \{\omega\}) \cup \{(4,3), (5,2)\} \cup \{\omega\} \times \{0,1\}$. It is shown with dots $\bullet$ in the figure, where elements involving ω fall beyond the grid. The elements of the minimal basis are circled.



▶ **Definition 1. (VAS$_0$).** A *Vector Addition System of dimension d with one zero-test (VAS$_0$)* is a tuple $\langle A, a_Z, \delta, \boldsymbol{x}_{\text{in}} \rangle$, where A is a finite alphabet of *actions*, $a_Z \notin A$ is called the *zero-test*, $\delta : A \cup \{a_Z\} \to \mathbb{Z}^d$ is a mapping, and $\boldsymbol{x}_{\text{in}} \in \mathbb{N}^d$ is the *initial state*.

Intuitively, a VAS$_0$ works on d counters, one for each component, whose initial values are given by $\boldsymbol{x}_{\text{in}}$. Executing $a \in A \cup \{a_Z\}$ translates the counters according to $\delta(a) \in \mathbb{Z}^d$. The mapping δ extends to a monoid morphism $\delta : (A \cup \{a_Z\})^* \to \mathbb{Z}^d$, so that $\delta(\varepsilon) = \mathbf{0}$ and $\delta(uv) = \delta(u) + \delta(v)$ for $u, v \in (A \cup \{a_Z\})^*$. A word $u \in (A \cup \{a_Z\})^*$ is *fireable from* $\boldsymbol{x} \in \mathbb{N}^d$ if

(a) for every prefix v of u, we have $\boldsymbol{x} + \delta(v) \geqslant \mathbf{0}$, and
(b) for every prefix wa_Z of u, we have $[\boldsymbol{x} + \delta(w)](1) = 0$.

The first condition means that all counters must remain nonnegative while firing actions. The second one says that the zero-test a_Z is possible only when the first counter is zero. We write $\boldsymbol{x} \xrightarrow{u} \boldsymbol{y}$ if u is fireable from $\boldsymbol{x}$ and $\boldsymbol{y} = \boldsymbol{x} + \delta(u)$. This implies in particular that $\boldsymbol{x}, \boldsymbol{y} \geqslant \mathbf{0}$.

▶ **Definition 2. (VAS).** A *Vector Addition System (VAS) of dimension* d is a tuple $\langle A, \delta, \boldsymbol{x}_{\text{in}} \rangle$ where A is a finite alphabet, $\delta : A \to \mathbb{Z}^d$ is a mapping, and $\boldsymbol{x}_{\text{in}} \in \mathbb{N}^d$ is the *initial state*.

A VAS is a particular VAS$_0$: choosing $\delta(a_Z) = -\boldsymbol{e}_1$ makes the zero-test a_Z never fireable. Given the VAS $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{\text{in}} \rangle$, we say that $u \in A^*$ is *fireable* if condition (a) above is satisfied.

For a VAS$_0$ or a VAS $\mathcal{S}$, the *reachability set* $\mathsf{Reach}(\mathcal{S})$ and the *cover* $\mathsf{Cover}(\mathcal{S})$ of $\mathcal{S}$ are:

$$\mathsf{Reach}(\mathcal{S}) = \{\boldsymbol{x}_{\text{in}} + \delta(u) \mid u \text{ is fireable in } \mathcal{S}\},$$
$$\mathsf{Cover}(\mathcal{S}) = \downarrow \mathsf{Reach}(\mathcal{S}).$$

We call elements of $\mathsf{Reach}(\mathcal{S})$ reachable *states* (also called reachable markings in related work). The reachability (resp. coverability) problem consists in deciding membership in the set $\mathsf{Reach}(\mathcal{S})$ (resp. in $\mathsf{Cover}(\mathcal{S})$). Reachability is decidable for VAS [19, 17, 18] and VAS_0 [21].

▶ **Theorem 3.** *Given a VAS $\mathcal{S}$, the reachability problem is decidable.*

Testing membership in the cover set is easier. One even gets a more precise result [15, 10, 8].

▶ **Theorem 4.** *Given a VAS $\mathcal{S}$, one can effectively compute a finite basis of* $\mathsf{Cover}(\mathcal{S})$.

Observe that from a finite basis B of a downward closed set D, one can effectively test membership in D. Therefore, one can effectively test membership in $\mathsf{Cover}(\mathcal{S})$. Computing a finite basis of the cover makes it possible to decide place-boundedness, that is, whether the projection of $\mathsf{Reach}(\mathcal{S})$ on some given component is bounded. In the next sections, we will show that one can also effectively compute a finite basis for the cover of a VAS_0.

3 Limits of reachable states of a VAS

Limits in $\mathbb{N}_\omega^d$. A sequence $(\ell_n)_{n\geqslant 0}$ (also written $(\ell_n)_n$) of elements of $\mathbb{N}_\omega$ *has limit* $\ell \in \mathbb{N}_\omega$, noted $\lim_n \ell_n = \ell$, if either it is ultimately constant with value ℓ, or its subsequence of integer values is infinite, it tends to infinity, and $\ell = \omega$. A sequence $(\boldsymbol{x_n})_n$ of vectors of $\mathbb{N}_\omega^d$ has limit $\boldsymbol{x} \in \mathbb{N}_\omega^d$, noted $\lim_n \boldsymbol{x_n} = \boldsymbol{x}$, if $\lim_n \boldsymbol{x_n}(i) = \boldsymbol{x}(i)$ for all $i \in \{1, \dots, d\}$.

For $M \subseteq \mathbb{N}_\omega^d$, we denote by $\mathsf{Lim}\, M$ the set of limits of sequences of elements of M. Note that $M \subseteq \mathsf{Lim}\, M$. Topologically speaking, $\mathsf{Lim}\, M$ is the least closed set (for the topology associated with the ordering) containing M and is usually called the (topological) closure of M. Also note that for $M \subseteq \mathbb{N}^d$, if $\mathsf{Lim}\, M$ is recursive, then so is $M = \mathbb{N}^d \cap \mathsf{Lim}\, M$. However, in general, M may be recursive while $\mathsf{Lim}\, M$ is not.

We prove in this section the following statement.

▶ **Theorem 5.** $\mathsf{Lim}\,\mathsf{Reach}(\mathcal{S})$ *is recursive.*

We do so by proving that $\mathsf{Lim}\,\mathsf{Reach}(\mathcal{S})$ and its complement in $\mathbb{N}_\omega^d$ are both recursively enumerable. We start by proving that $\mathsf{Lim}\,\mathsf{Reach}(\mathcal{S})$ is recursively enumerable, by introducing *productive sequences*, a notion inspired by Hauschildt [13].

▶ **Definition 6.** Let $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{\mathrm{in}} \rangle$ be a VAS. A sequence $\pi = (u_i)_{0\leqslant i\leqslant k}$ of words $u_i \in A^*$ is *productive* in $\mathcal{S}$ for a word $v = a_1 \cdots a_k$ ($a_i \in A$) if

(1) the partial sums $\delta(u_0) + \cdots + \delta(u_i)$ are nonnegative for every $i \in \{0, \dots, k\}$, and
(2) the word $u_0 a_1 u_1 \cdots a_k u_k$ is fireable from $\boldsymbol{x}_{\mathrm{in}}$.

The total sum $\sum_{i=0}^{k} \delta(u_i)$ is called the *production* of π and is simply denoted $\delta(\pi)$.

The following lemma provides a characterization of the productive sequences.

▶ **Lemma 7.** *A sequence $\pi = (u_i)_{0\leqslant i\leqslant k}$ is productive for $v = a_1 \cdots a_k$ if and only if the words $u_0^n a_1 u_1^n \cdots a_k u_k^n$ are fireable from $\boldsymbol{x}_{in}$ for all $n \geqslant 1$. In particular, every marking $\boldsymbol{x}_{in} + \delta(v) + n\delta(\pi)$ where $n \geqslant 1$ is reachable from $\boldsymbol{x}_{in}$.*

Proposition 9 below shows that limits of reachable states can be witnessed by productive sequences. Its essential argument is Higman's Lemma. Recall that an ordering $\preccurlyeq$ is *well* if every infinite sequence $(\ell_n)_{n\in\mathbb{N}}$ admits an infinite increasing subsequence $(\ell_{n_k})_{k\in\mathbb{N}}$, *i.e.*, such that $\ell_{n_0} \preccurlyeq \ell_{n_1} \preccurlyeq \ell_{n_2} \preccurlyeq \cdots$. The pointwise ordering on $\mathbb{N}^d$ or on $\mathbb{N}_\omega^d$ is well (Dickson's Lemma).

Higman's Lemma. For a (possibly infinite) set Σ, we denote by Σ^* the set of finite words over Σ. Given an ordering $\preccurlyeq$ on Σ, let $\preccurlyeq^*$ be the ordering on Σ^* defined as follows: for $u, v \in \Sigma^*$, we have $u \preccurlyeq^* v$ if $u = a_1 \cdots a_n$ with $a_i \in \Sigma$, $v = v_0 b_1 v_1 \cdots v_{n-1} b_n v_n$, with $v_i \in \Sigma^*$, $b_j \in \Sigma$, and for all $i = 1, \ldots, n$, we have $a_i \preccurlyeq b_i$. In other words, u is obtained from v by removing some letters, and then replacing some of the remaining letters by smaller ones. Higman's Lemma is the following result, see [4] for instance for a proof.

▶ **Lemma 8.** ***(Higman)*** *If $\preccurlyeq$ is a well ordering on A, then $\preccurlyeq^*$ is a well ordering on A^*.*

We extend the multiplication on $\mathbb{N}_\omega$ by $\omega \cdot 0 = 0 = 0 \cdot \omega$ and $\omega \cdot k = \omega = k \cdot \omega$ if $k \neq 0$. This multiplication then extends componentwise to the scalar multiplication of $\mathbb{N}_\omega^d$ by $\mathbb{N}_\omega$.

▶ **Proposition 9.** *Let $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{in} \rangle$ be a VAS. Then*

$$\mathsf{Lim\,Reach}(\mathcal{S}) = \{\boldsymbol{x}_{in} + \delta(v) + \omega\delta(\pi) \mid \pi \textit{ is productive for } v\}.$$

Proof. For the inclusion from right to left, if π is a productive sequence for a word v, then $\boldsymbol{x}_{\text{in}} + \delta(v) + \omega\delta(\pi)$ is the limit of the sequence $(\boldsymbol{x_n})_{n\in\mathbb{N}}$ with $\boldsymbol{x_n} = \boldsymbol{x}_{\text{in}} + \delta(v) + n\delta(\pi)$, and $\boldsymbol{x_n}$ is a reachable state by Lemma 7. We prove the reverse inclusion thanks to Higman's lemma.

We first introduce a well ordering $\sqsubseteq$ over $\mathsf{Reach}(\mathcal{S})$, using a temporary ordering $\preccurlyeq$. Consider the infinite set $\Sigma = A \times \mathbb{N}_\omega^d$. This set is well ordered by $\preccurlyeq$, defined by

$$(a, \boldsymbol{y}) \preccurlyeq (b, \boldsymbol{z}) \text{ if and only if } a = b \text{ and } \boldsymbol{y} \leqslant \boldsymbol{z}.$$

Since $\preccurlyeq$ is a well ordering, Higman's lemma shows that $\preccurlyeq^*$ is a well-ordering over Σ^*. Let us now associate to every reachable state $\boldsymbol{y} \in \mathsf{Reach}(\mathcal{S})$ a word $\alpha_{\boldsymbol{y}}$ in Σ^* as follows: since $\boldsymbol{y}$ is reachable, we can choose a word $v = a_1 \cdots a_k$, with $a_i \in A$, such that $\boldsymbol{x}_{\text{in}} \xrightarrow{v} \boldsymbol{y}$. We introduce the sequence $(\boldsymbol{y_i})_{0\leqslant i\leqslant k}$ of states defined by $\boldsymbol{y_i} = \boldsymbol{x}_{\text{in}} + \delta(a_1 \cdots a_i)$, and we let:

$$\alpha_{\boldsymbol{y}} = (a_1, \boldsymbol{y_1}) \cdots (a_k, \boldsymbol{y_k}).$$

The ordering $\sqsubseteq$ over $\mathsf{Reach}(\mathcal{S})$ is defined by $\boldsymbol{y} \sqsubseteq \boldsymbol{z}$ if $\alpha_{\boldsymbol{y}} \preccurlyeq^* \alpha_{\boldsymbol{z}}$ and $\boldsymbol{y} \leqslant \boldsymbol{z}$. Since the orderings $\preccurlyeq^*$ over Σ^* and $\leqslant$ over $\mathbb{N}^d$ are well, we deduce that $\sqsubseteq$ is a well ordering over $\mathsf{Reach}(\mathcal{S})$.

To show the inclusion from left to right, pick $\boldsymbol{x} \in \mathsf{Lim\,Reach}(\mathcal{S})$: $\boldsymbol{x}$ is the limit of a sequence $(\boldsymbol{x_k})_{k\in\mathbb{N}}$ of reachable states. By extracting a subsequence we can assume that $(\boldsymbol{x_k}(i))_{k\in\mathbb{N}}$ is strictly increasing if $\boldsymbol{x}(i) = \omega$, and $\boldsymbol{x_k}(i) = \boldsymbol{x}(i)$ if $\boldsymbol{x}(i) < \omega$. Denote by α_k the word $\alpha_{\boldsymbol{x_k}}$ associated to the reachable state $\boldsymbol{x_k}$. Since $\sqsubseteq$ is a well ordering, there exist $m < n$ such that $\boldsymbol{x_m} \sqsubseteq \boldsymbol{x_n}$. By construction of α_m there exists a word $v = a_1 \cdots a_k$ with $a_j \in A$ such that the sequence $(\boldsymbol{y_j})_{0\leqslant j\leqslant k}$ defined by $\boldsymbol{y_j} = \boldsymbol{x}_{\text{in}} + \delta(a_1 \cdots a_j)$ for every $j \in \{1, \ldots, k\}$ satisfies:

$$\alpha_m = (a_1, \boldsymbol{y_1}) \cdots (a_k, \boldsymbol{y_k}).$$

Since $\boldsymbol{x_m} \preccurlyeq^* \boldsymbol{x_n}$ and by definition of $\preccurlyeq^*$, there exist a sequence $(\boldsymbol{z_j})_{1\leqslant j\leqslant k}$ of states with $\boldsymbol{y_j} \leqslant \boldsymbol{z_j}$, and a sequence $(\beta_j)_{0\leqslant j\leqslant k}$ of words in Σ^* such that the following equality holds:

$$\alpha_n = \beta_0(a_1, \boldsymbol{z_1})\beta_1 \cdots (a_k, \boldsymbol{z_k})\beta_k$$

We call *label* of a word $(b_1, \boldsymbol{t_1}) \cdots (b_\ell, \boldsymbol{t_\ell})$ over Σ the word $b_1 \cdots b_\ell$ over A. Consider the sequence $\pi = (u_j)_{0\leqslant j\leqslant k}$ where u_j is the label of β_j. By definition of α_n, we have

$$\boldsymbol{x}_{\text{in}} \xrightarrow{u_0 a_1} \boldsymbol{z_1} \cdots \xrightarrow{u_{k-1} a_k} \boldsymbol{z_k} \xrightarrow{u_k} \boldsymbol{x_n}$$

In particular, $\boldsymbol{z_j} = \boldsymbol{y_j} + \delta(u_0) + \cdots + \delta(u_{j-1})$ for every $j \in \{1, \ldots, k\}$ and $\boldsymbol{x_n} = \boldsymbol{z_k} + \delta(u_k) = \boldsymbol{y_k} + \delta(\pi) = \boldsymbol{x_m} + \delta(\pi)$. As $\boldsymbol{y_j} \leqslant \boldsymbol{z_j}$ for every $j \in \{1, \ldots, k\}$ and $\boldsymbol{x_m} \leqslant \boldsymbol{x_n}$, we deduce that π is productive for v.

Finally, let us prove that $\boldsymbol{x} = \boldsymbol{y}$ where $\boldsymbol{y} = \boldsymbol{x}_{\text{in}} + \delta(v) + \omega\delta(\pi)$. We have $\boldsymbol{x_n} = \boldsymbol{x_m} + \delta(\pi)$. Let us consider $i \in \{1, \ldots, d\}$. If $\boldsymbol{x}(i) < \omega$ then $\boldsymbol{x_m}(i) = \boldsymbol{x}(i) = \boldsymbol{x_n}(i)$. Thus $\delta(\pi)(i) = 0$ and we deduce that $\boldsymbol{x}(i) = \boldsymbol{y}(i)$. If $\boldsymbol{x}(i) = \omega$ then $\boldsymbol{x_m}(i) < \boldsymbol{x_n}(i)$ and we deduce that $\delta(\pi)(i) > 0$ and in particular $\boldsymbol{x}(i) = \omega = \boldsymbol{y}(i)$. Thus $\boldsymbol{x} = \boldsymbol{y}$. We have proved that there exists a productive sequence π for a word v such that $\boldsymbol{x} = \boldsymbol{x}_{\text{in}} + \delta(v) + \omega\delta(\pi)$. ◀

It is easier to prove that the *complement* of $\mathsf{Lim\,Reach}(\mathcal{S})$ recursively enumerable. We just give the construction. Let $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{\text{in}} \rangle$ and $\boldsymbol{y} \in \mathbb{N}_\omega^d$. We introduce d distinct additional elements $b_1, \ldots, b_d \notin A$. Let $B = \{b_1, \ldots, b_d\}$. We introduce the VAS $\mathcal{S}_{\boldsymbol{y}} = \langle A \uplus B, \delta_{\boldsymbol{y}}, \boldsymbol{x}_{\text{in}} \rangle$, where $\delta_{\boldsymbol{y}}$ extends δ by:

$$\delta_{\boldsymbol{y}}(b_i) = \begin{cases} \boldsymbol{0} & \text{if } \boldsymbol{y}(i) < \omega, \\ -\boldsymbol{e_i} & \text{if } \boldsymbol{y}(i) = \omega. \end{cases}$$

Finally, we define from $\boldsymbol{y}$ a sequence $(\boldsymbol{y_\ell})_\ell$ converging to $\boldsymbol{y}$, by $\boldsymbol{y_\ell}(i) = \begin{cases} \ell & \text{if } \boldsymbol{y}(i) = \omega, \\ \boldsymbol{y}(i) & \text{if } \boldsymbol{y}(i) < \omega. \end{cases}$

▶ **Lemma 10.** *Let $\mathcal{S}_{\boldsymbol{y}}$ and $(\boldsymbol{y_\ell})_\ell$ constructed from $\boldsymbol{y}$ as above. Then,*

$$\boldsymbol{y} \notin \mathsf{Lim\,Reach}(\mathcal{S}) \iff \exists \ell \in \mathbb{N},\ \boldsymbol{y_\ell} \notin \mathsf{Reach}(\mathcal{S}_{\boldsymbol{y}}). \tag{1}$$

In particular, the complement of $\mathsf{Lim\,Reach}(\mathcal{S})$ is recursively enumerable.

Theorem 5 now follows from Proposition 9 and Lemma 10.

4 Between the cover and the reachability set: the filtered covers

In this section, we introduce a set hybrid between the reachability and cover sets, which to our knowledge has not yet been considered. Instead of the downward closure $\mathsf{Cover}(\mathcal{S})$ of $\mathsf{Reach}(\mathcal{S})$ wrt. the pointwise ordering $\leqslant$, we consider $\mathsf{Cover}_{\leqslant_P}(\mathcal{S}) = \downarrow_{\leqslant_P} \mathsf{Reach}(\mathcal{S})$, that is, we replace $\leqslant$ with an ordering $\leqslant_P$ parametrized by a set of "positions" $P \subseteq \{1, \ldots, d\}$:

$$\boldsymbol{x} \leqslant_P \boldsymbol{y} \quad \text{if} \quad \begin{cases} \boldsymbol{x}(i) = \boldsymbol{y}(i) & \text{for } i \in P, \\ \boldsymbol{x}(i) \leqslant \boldsymbol{y}(i) & \text{for } i \notin P. \end{cases}$$

The set P contains the components for which we insist on keeping equality. Thus, $\leqslant_\emptyset$ is the usual pointwise ordering $\leqslant$, while $\leqslant_{\{1,\ldots,d\}}$ boils down to equality. Note that $\leqslant_P$ is *not* a well ordering, except if $P = \emptyset$ (*e.g.*, $\mathbb{N}$ ordered by $\leqslant_{\{1\}}$ consists only of incomparable elements).

The ordering $\leqslant_{\{1\}}$ will be abbreviated as $\leqslant_1$. It is a natural order to study for a VAS_0 (recall that the zero-test occurs on the first component). Indeed, the transition relation of a VAS_0 is monotonic regarding this order: if $\boldsymbol{x} \xrightarrow{u} \boldsymbol{x'}$ and $x \leqslant_1 y$, then there exists $\boldsymbol{y'}$ with $\boldsymbol{y} \xrightarrow{u} \boldsymbol{y'}$ and $\boldsymbol{x'} \leqslant_1 \boldsymbol{y'}$. More precisely, testing if $\mathsf{Cover}_{\leqslant_1}(\mathcal{S})$ contains a vector whose first component is 0 is what we need to design our algorithm computing the cover of a VAS with one zero test. Unfortunately, this set has infinitely many maximal elements for $\leqslant_1$, and thus cannot be represented by a finite basis. The following theorem shows that we cannot find a sensible way to compute a representation of this set, as any representation would not allow to test for equality.

▶ **Theorem 11.** *Given two VAS $\mathcal{S}_1$ and $\mathcal{S}_2$ of the same dimension d, the equality problem $\mathsf{Cover}_{\leqslant_1}(\mathcal{S}_1) = \mathsf{Cover}_{\leqslant_1}(\mathcal{S}_2)$ is undecidable.*

Proof. We reduce this problem to the equality problem $\mathsf{Reach}(\mathcal{S}_1) = \mathsf{Reach}(\mathcal{S}_2)$. This problem is known to be undecidable [12]. Let us first consider a VAS $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{\text{in}} \rangle$ of dimension d. We introduce a VAS $\mathcal{S}' = \langle A, \delta', \boldsymbol{x}'_{\text{in}} \rangle$ of dimension $d+1$ that counts in the first component the sum of the other components. Formally, $\boldsymbol{x}'_{\text{in}} = (\sum_{i=1}^{d} \boldsymbol{x}_{\text{in}}(i), \boldsymbol{x}_{\text{in}})$ and $\delta'(a) = (\sum_{i=1}^{d} \delta(a)(i), \delta(a))$ for every $a \in A$. Observe that the following equivalence holds:

$$(n, \boldsymbol{x}) \in \mathsf{Reach}(\mathcal{S}') \iff \boldsymbol{x} \in \mathsf{Reach}(\mathcal{S}) \text{ and } n = \sum_{i=1}^{d} \boldsymbol{x}(i)$$

Finally let us consider two VAS $\mathcal{S}_1$ and $\mathcal{S}_2$ and just observe that $\mathsf{Reach}(\mathcal{S}_1) = \mathsf{Reach}(\mathcal{S}_2)$ if and only if $\mathsf{Cover}_{\leqslant_1}(\mathcal{S}'_1) = \mathsf{Cover}_{\leqslant_1}(\mathcal{S}'_2)$. ◀

So, we cannot hope for a useful representation of the sets $\mathsf{Cover}_{\leqslant_P}(\mathcal{S})$ themselves. However, one can capture the needed information differently, by replacing the downward closure $\downarrow_{\leqslant_P}$ in $\mathsf{Cover}_{\leqslant_P}(\mathcal{S}) = \downarrow_{\leqslant_P}\mathsf{Reach}(\mathcal{S})$ with an operator $\Downarrow_{\boldsymbol{f}}$ parametrized by a vector $\boldsymbol{f}$ of $\mathbb{N}_\omega^d$. Informally, $\Downarrow_{\boldsymbol{f}} M$ takes into account only elements of M that agree with $\boldsymbol{f}$ on its finite components. Formally, for $\boldsymbol{f} \in \mathbb{N}_\omega^d$ and $M \subseteq \mathbb{N}^d$, let

$$\mathsf{Filter}(M, \boldsymbol{f}) = \Big\{ \boldsymbol{x} \in M \mid \bigwedge_{i=1}^{d} \big[\, \boldsymbol{f}(i) < \omega \Longrightarrow \boldsymbol{x}(i) = \boldsymbol{f}(i) \big] \Big\},$$

$$\Downarrow_{\boldsymbol{f}} M = \big\downarrow \mathsf{Filter}(M, \boldsymbol{f}).$$

Note that $\Downarrow_{\boldsymbol{f}} M = \downarrow M$ for $\boldsymbol{f} = (\omega, \omega, \ldots, \omega)$. On the other hand, if $\boldsymbol{f} \in \mathbb{N}^d$, then $\Downarrow_{\boldsymbol{f}} M = \downarrow \boldsymbol{f}$ if $\boldsymbol{f} \in M$, and $\Downarrow_{\boldsymbol{f}} M = \emptyset$ otherwise. Observe also that $\Downarrow_{\boldsymbol{f}} M$ is downward closed and that the maximal elements of any basis of $\Downarrow_{\boldsymbol{f}} M$ agree with $\boldsymbol{f}$ on every component i where $\boldsymbol{f}(i)$ is finite. The next lemma provides a relationship between the sets $\Downarrow_{\boldsymbol{f}} M$ and $\downarrow_{\leqslant_P} M$.

▶ **Lemma 12.** *Let $M \subseteq \mathbb{N}^d$. Then, the following conditions are equivalent:*

(a) *For all $\boldsymbol{f} \in \mathbb{N}_\omega^d$, one can effectively compute the basis of $\Downarrow_{\boldsymbol{f}} M$.*
(b) *For all $P \subseteq \{1, \ldots, d\}$, the set $\mathsf{Lim} \downarrow_{\leqslant_P} M$ is recursive.*

The main result of this section states that both conditions of Lemma 12 actually hold when M is the reachability set of a VAS. This is obtained by first proving that $\mathsf{Cover}_{\leqslant_P}(\mathcal{S}) = \mathsf{Reach}(\mathcal{S}_P)$ where $\mathcal{S}_P$ is a VAS constructed from $\mathcal{S}$ and P. From this equality, we deduce that $\mathsf{Lim}\,\mathsf{Cover}_{\leqslant_P}(\mathcal{S}) = \mathsf{Lim}\,\mathsf{Reach}(\mathcal{S}_P)$. Applying Theorem 5, it follows that this set is recursive, which proves condition (b) for $M = \mathsf{Reach}(\mathcal{S})$. Then by Lemma 12, condition (a) also holds.

Let $\mathcal{S} = \langle A, \delta, \boldsymbol{x}_{\text{in}} \rangle$ be a VAS and $P \subseteq \{1, \ldots, d\}$. Let us define a VAS $\mathcal{S}_P$ such that $\mathsf{Reach}(\mathcal{S}_P) = \mathsf{Cover}_{\leqslant_P}(\mathcal{S})$. We consider d distinct additional elements $b_1, \ldots, b_d \notin A$. Let $B = \{b_1, \ldots, b_d\}$. We consider the VAS $\mathcal{S}_P = \langle A \uplus B, \delta_P, \boldsymbol{x}_{\text{in}} \rangle$, where δ_P extends δ by:

$$\delta_P(b_i) = \begin{cases} \boldsymbol{0} & \text{if } i \in P \\ -\boldsymbol{e_i} & \text{if } i \notin P. \end{cases}$$

▶ **Lemma 13.** *Let $\mathcal{S}_P$ constructed from $\mathcal{S}$ and P as above. Then $\mathsf{Cover}_{\leqslant_P}(\mathcal{S}) = \mathsf{Reach}(\mathcal{S}_P)$.*

Proof. Consider a state $\boldsymbol{x} \in \mathsf{Cover}_{\leqslant_P}(\mathcal{S})$. By definition, there exists $\boldsymbol{y} \in \mathsf{Reach}(\mathcal{S})$ such that $\boldsymbol{x} \leqslant_P \boldsymbol{y}$. Observe that $\boldsymbol{x}_{\text{in}} \xrightarrow{*} \boldsymbol{y} \xrightarrow{u} \boldsymbol{x}$ in $\mathcal{S}_P$ with $u = \prod_{i=1}^{d} b_i^{\boldsymbol{y}(i) - \boldsymbol{x}(i)}$. Hence $\boldsymbol{x} \in \mathsf{Reach}(\mathcal{S}_P)$. Conversely let $\boldsymbol{x} \in \mathsf{Reach}(\mathcal{S}_P)$ and let $u \in (A \cup B)^*$ such that $\boldsymbol{x}_{\text{in}} \xrightarrow{u} \boldsymbol{x}$ in $\mathcal{S}_P$. Consider the word v obtained from u by erasing all letters of B. Since $\delta_P(b) \leqslant \boldsymbol{0}$ for $b \in B$, the word v is still fireable from $\boldsymbol{x}_{\text{in}}$. Thus $\boldsymbol{y} = \boldsymbol{x}_{\text{in}} + \delta(v) \in \mathsf{Reach}(\mathcal{S})$. Moreover, by definition of $\mathcal{S}_P$ we have $\boldsymbol{x} \leqslant_P \boldsymbol{y}$. Therefore $\boldsymbol{x} \in \mathsf{Cover}_{\leqslant_P}(\mathcal{S})$. ◀

Combining Lemma 13, Theorem 5 and Lemma 12 as explained above yields:

▶ **Theorem 14.** *Given $\boldsymbol{f} \in \mathbb{N}_\omega^d$ and a VAS $\mathcal{S}$, one can effectively compute a basis of $\Downarrow_{\boldsymbol{f}} \mathsf{Reach}(\mathcal{S})$.*

5 Computing the cover of a VAS with one zero-test

We provide an algorithm computing the basis of $\mathsf{Cover}(\mathcal{S})$ of any VAS_0 $\mathcal{S} = \langle A, a_Z, \delta, \boldsymbol{x}_{\text{in}} \rangle$. Intuitively the algorithm, inspired by the Karp and Miller algorithm for VAS [15], builds a tree with nodes labeled by vectors in $\{0\} \times \mathbb{N}_\omega^{d-1}$ such that the finite set R of node labels satisfies the following equality when the algorithm terminates:

$$\Downarrow_{\boldsymbol{f}} \mathsf{Reach}(\mathcal{S}) = (\downarrow R) \cap \mathbb{N}^d, \quad \text{where } \boldsymbol{f} = (0, \omega, \dots, \omega).$$

In order to simplify the presentation, we assume without loss of generality that $\boldsymbol{x}_{\text{in}} \in \{0\} \times \mathbb{N}^{d-1}$ and $\delta(a_Z) \in \{0\} \times \mathbb{Z}^{d-1}$. In the sequel we denote by $\mathcal{S}_{\text{VAS}}$ the VAS $\mathcal{S}_{\text{VAS}} = (A, \delta, \boldsymbol{x}_{\text{in}})$ obtained from $\mathcal{S}$ by removing the zero-test a_Z. Moreover, given $\boldsymbol{s} \in \{0\} \times \mathbb{N}^{d-1}$ we denote by $\mathcal{S}(\boldsymbol{s})$ and $\mathcal{S}_{\text{VAS}}(\boldsymbol{s})$ the VASs obtained respectively from $\mathcal{S}$ and $\mathcal{S}_{\text{VAS}}$ by replacing the initial state $\boldsymbol{x}_{\text{in}}$ by $\boldsymbol{s}$.

At any step of the execution, in the tree built in the algorithm, every ancestor node n of a node n' satisfies the invariant $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x'}$ where $\boldsymbol{x}, \boldsymbol{x'}$ are the labels of n, n' and where $\overset{*}{\Rightarrow}$ is the binary relation defined over the vectors in $\{0\} \times \mathbb{N}_\omega^{d-1}$ by:

$$\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x'} \;\text{ if }\; (\downarrow\boldsymbol{x'}) \cap \mathbb{N}^d \subseteq \bigcup_{\boldsymbol{s} \in (\downarrow \boldsymbol{x}) \cap \mathbb{N}^d} \Downarrow_{\boldsymbol{f}} \mathsf{Reach}(\mathcal{S}(\boldsymbol{s})).$$

By the next lemma, it is sufficient to maintain this invariant along each parent-child edge.

▶ **Lemma 15.** *The binary relation $\overset{*}{\Rightarrow}$ is reflexive and transitive.*

Proof. The reflexivity is immediate. For the transitivity, we first introduce the binary relation $\xrightarrow{*}$ over $\mathbb{N}^d$ defined by $\boldsymbol{x} \xrightarrow{*} \boldsymbol{x'}$ if there exists $u \in (A \cup \{a_Z\})^*$ such that $\boldsymbol{x} \xrightarrow{u} \boldsymbol{x'}$. We observe that $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x'}$ if and only if the following relation holds:

$$\forall \boldsymbol{s'} \in (\downarrow\boldsymbol{x'}) \cap \mathbb{N}^d \; \exists \boldsymbol{s} \in (\downarrow\boldsymbol{x}) \cap \mathbb{N}^d \; \exists \boldsymbol{z} \in \{0\} \times \mathbb{N}_\omega^{d-1} \quad \boldsymbol{s} \xrightarrow{*} \boldsymbol{s'} + \boldsymbol{z}.$$

Assume that $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x'}$ and $\boldsymbol{x'} \overset{*}{\Rightarrow} \boldsymbol{x''}$. Let $\boldsymbol{s''} \in (\downarrow\boldsymbol{x''}) \cap \mathbb{N}^d$. From $\boldsymbol{x'} \overset{*}{\Rightarrow} \boldsymbol{x''}$, we deduce that there exist $\boldsymbol{z'} \in \{0\} \times \mathbb{N}^{d-1}$ and $\boldsymbol{s'} \in (\downarrow\boldsymbol{x'}) \cap \mathbb{N}^d$ such that $\boldsymbol{s'} \xrightarrow{*} \boldsymbol{s''} + \boldsymbol{z'}$. From $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x'}$, we deduce that there exist $\boldsymbol{z} \in \{0\} \times \mathbb{N}^{d-1}$ and $\boldsymbol{s} \in (\downarrow\boldsymbol{x}) \cap \mathbb{N}^d$ such that $\boldsymbol{s} \xrightarrow{*} \boldsymbol{s'} + \boldsymbol{z}$. In particular we deduce that $\boldsymbol{s} \xrightarrow{*} \boldsymbol{s''} + \boldsymbol{z} + \boldsymbol{z'}$. We have proved that $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{x''}$. ◀

Assume now that $\boldsymbol{x} \in \{0\} \times \mathbb{N}_\omega^{d-1}$ labels a leaf. We create a child of this leaf if the vector $\boldsymbol{y} = \boldsymbol{x} + \delta(a_Z)$ is nonnegative. Note that in this case $\boldsymbol{y} \in \{0\} \times \mathbb{N}_\omega^{d-1}$, since $\delta(a_Z)(1) = 0$. We do not violate the invariant when creating the child labeled $\boldsymbol{y}$ since $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{y}$. We may also add new children labeled by elements of the minimal basis $B(\boldsymbol{x})$ of the following downward-closed set:

$$\bigcup_{\boldsymbol{s} \in (\downarrow\boldsymbol{x}) \cap \mathbb{N}^d} \Downarrow_{\boldsymbol{f}} \mathsf{Reach}(\mathcal{S}_{\text{VAS}}(\boldsymbol{s}))$$

We observe that $\boldsymbol{x} \overset{*}{\Rightarrow} \boldsymbol{b}$ for every $\boldsymbol{b} \in B(\boldsymbol{x})$, so that the invariant will still be fulfilled after adding elements of $B(\boldsymbol{x})$.

▶ **Lemma 16.** *The basis $B(\boldsymbol{x})$ is effectively computable.*

Proof. We introduce the set I of components $i \in \{2, \dots, d\}$ such that $\boldsymbol{x}(i) = \omega$. We consider the VAS $\mathcal{S}'_{\text{VAS}} = (A, \delta', \boldsymbol{x}')$ obtained from $\mathcal{S}_{\text{VAS}}(\boldsymbol{x})$ by preventing any modification of components in I. More formaly δ' and $\boldsymbol{x}'$ are defined by $\delta'(a)(i) = 0$ and $\boldsymbol{x}'(i) = 0$ if $i \in I$ and $\delta'(a)(i) = \delta(a)(i)$ and $\boldsymbol{x}'(i) = \boldsymbol{x}(i)$ if $i \notin I$. Theorem 14 shows that we can effectively compute the basis B' of $\Downarrow_{\boldsymbol{f}}\mathsf{Reach}(\mathcal{S}'_{\text{VAS}})$. Now $B(\boldsymbol{x}) = \{y + z \mid y \in B'\}$, where $\boldsymbol{z}$ is the vector defined by $\boldsymbol{z}(i) = \omega$ if $i \in I$ and $\boldsymbol{z}(i) = 0$ if $i \notin I$. ◀

The algorithm termination is obtained by introducing an *acceleration operator* ∇. We define the vector $\boldsymbol{x} \nabla \boldsymbol{y}$ for every $\boldsymbol{x}, \boldsymbol{y} \in \{0\} \times \mathbb{N}_\omega^{d-1}$ such that $\boldsymbol{x} \leqslant \boldsymbol{y}$ by

$$(\boldsymbol{x} \nabla \boldsymbol{y})(i) = \begin{cases} \omega & \text{if } \boldsymbol{x}(i) < \boldsymbol{y}(i) \\ \boldsymbol{x}(i) & \text{if } \boldsymbol{x}(i) = \boldsymbol{y}(i). \end{cases}$$

▶ **Lemma 17.** *If $\boldsymbol{x} \xRightarrow{*} \boldsymbol{y}$ with $\boldsymbol{x} \leqslant \boldsymbol{y}$ then $\boldsymbol{x} \xRightarrow{*} \boldsymbol{x} \nabla \boldsymbol{y}$.*

Let us now describe informally the algorithm. It inductively computes a tree with nodes labeled by vectors in $\{0\} \times \mathbb{N}_\omega^{d-1}$. The tree is rooted at a node labeled by $\boldsymbol{x}_{\text{in}}$ (recall that $\boldsymbol{x}_{\text{in}} \in \{0\} \times \mathbb{N}^{d-1}$). The tree is modified in such a way that for every node n and for every child n' of n, the labels $\boldsymbol{x}, \boldsymbol{x'}$ of n, n' satisfy $\boldsymbol{x} \xRightarrow{*} \boldsymbol{x'}$. While there exists a leaf n' labeled by a vector $\boldsymbol{x'}$ that admits an ancestor n labeled by a vector $\boldsymbol{x}$ such that $\boldsymbol{x} \leqslant \boldsymbol{x'} < \boldsymbol{x} \nabla \boldsymbol{x'}$, we replace the label $\boldsymbol{x'}$ of node n' by $\boldsymbol{x} \nabla \boldsymbol{x'}$. From Lemma 17, we deduce that the invariant still holds. Since this loop just replaces some components by ω, it terminates. Then, the algorithm checks if for every leaf n labeled by $\boldsymbol{x}$, there exists a strict ancestor (*i.e.*, different from n) labeled by the same vector $\boldsymbol{x}$. In this case, the algorithm terminates and it returns the set of node labels. Otherwise the algorithm considers a leaf n not fulfilling this condition, and it creates a new child of n labeled by $\boldsymbol{b}$ for each $\boldsymbol{b} \in B(\boldsymbol{x})$. It also creates a new child labeled by $\boldsymbol{x} + \delta(a_Z)$ if this vector is nonnegative. The modification of the tree is then restarted.

The termination of this algorithm follows from König's lemma. If the algorithm does not terminate, then it would generate an infinite tree. Because this tree has a finite branching degree, by König's lemma, there is an infinite branch. Since $\leqslant$ is a well-ordering over $\{0\} \times \mathbb{N}_\omega^{d-1}$, this implies that we can extract from this infinite branch an infinite increasing subsequence. However, since we add children to a leaf only if there does not exist a strict ancestor labeled by the same vector, this sequence cannot contain the same vector twice, and must therefore be *strictly* increasing. But, due to the use of the operator ∇, a component with an integer is replaced by ω at every acceleration step. Because the number of ω's in the vectors labeling a branch cannot decrease, we obtain a contradiction. We deduce the following proposition.

▶ **Proposition 18.** *Algorithm 1 terminates and it returns a finite set R such that*

$$\Downarrow_{\boldsymbol{f}}\mathsf{Reach}(\mathcal{S}) = \downarrow R \cap \mathbb{N}^d.$$

We have proved that we can effectively compute a basis R of $\Downarrow_{\boldsymbol{f}}\mathsf{Reach}(\mathcal{S})$. Now, observe that the following equality holds:

$$\mathsf{Cover}(\mathcal{S}) = \bigcup_{\boldsymbol{b} \in R} \bigcup_{\boldsymbol{s} \in (\downarrow \boldsymbol{b}) \cap \mathbb{N}^d} \mathsf{Cover}(\mathcal{S}_{\text{VAS}}(\boldsymbol{s})).$$

Algorithm 1 An algorithm to compute a basis of $\Downarrow_{\boldsymbol{f}}\mathsf{Reach}(\mathcal{S})$

- Inputs: A VAS_0 $\mathcal{S}$ such that $\boldsymbol{x}_{\text{in}} \in \{0\} \times \mathbb{N}^{d-1}$ and a $\delta(a_Z) \in \{0\} \times \mathbb{Z}^{d-1}$.
- Outputs: R, a finite subset of $\{0\} \times \mathbb{N}_{\omega}^{d-1}$.
- Internal Variables:
  - $\mathcal{T}$, a tree labeled by elements of $\mathbb{N}_{\omega}^{d}$.
  - N, a set of nodes.
- Algorithm:

Initialize $\mathcal{T}$ as a single root n_{in}, labeled by $\boldsymbol{x}_{\text{in}}$
$N \leftarrow \{n_{in}\}$
while $N \neq \emptyset$ **do**
  Take a node n from N
  $\boldsymbol{x} \leftarrow label(n)$
  if the label of every strict ancestor of n is not equal to $\boldsymbol{x}$ **then**
    for all strict ancestor n_0 of n **do**
      $\boldsymbol{x}_0 \leftarrow label(n_0)$
      if $\boldsymbol{x}_0 \leqslant \boldsymbol{x}$ **then**
        $\boldsymbol{x} \leftarrow \boldsymbol{x}_0 \nabla \boldsymbol{x}$
      end if
    end for
    Replace the label of n by $\boldsymbol{x}$
    if $\boldsymbol{x} + \delta(a_Z) \geqslant \mathbf{0}$ **then**
      Create a new node in $\mathcal{T}$ labeled by $\boldsymbol{x} + \delta(a_Z)$, as a child of n
      Add this node to N
    end if
    for all $\boldsymbol{b} \in B(\boldsymbol{x})$ **do**
      Create a new node in $\mathcal{T}$ labeled by $\boldsymbol{b}$, as a child of n
      Add this node to N
    end for
  end if
end while
$R \leftarrow \{label(n) \mid n \in nodes(\mathcal{T})\}$
return R

A reduction similar to the one provided in the proof of Lemma 16 shows that the basis of $\bigcup_{\boldsymbol{s}\in(\downarrow\boldsymbol{b})\cap\mathbb{N}^d} \mathsf{Cover}(\mathcal{S}_{\mathrm{VAS}}(\boldsymbol{s}))$ can be obtained from a basis of $\mathsf{Cover}(\mathcal{S}'_{\mathrm{VAS}})$, where $\mathcal{S}'_{\mathrm{VAS}}$ is a VAS obtained from $\mathcal{S}_{\mathrm{VAS}}$ and $\boldsymbol{b}$ by removing the components $i \in \{2,\dots,d\}$ such that $\boldsymbol{b}(i) = \omega$. We deduce the following theorem.

▶ **Theorem 19.** *Given a VAS_0 $\mathcal{S}$, one can effectively compute the finite basis of* $\mathsf{Cover}(\mathcal{S})$.

6 Conclusion and perspectives

Our main result is a forward algorithm, à la Karp&Miller, to compute the downward closure of the reachability set of a nonmonotonic transition system: VAS_0. This implies that place-boundedness is decidable. For our purposes, we introduced new sets, sitting between the cover and the reachability set. Unfortunately, we cannot say anything about the complexity of the computation of the cover for VAS_0, because our proof uses the decidability of the reachability problem for VAS as an oracle, whose complexity is still open.

Since we have solved the place-boundedness problem, a natural question would be an instance of a liveness problem, like the repeated control-state reachability problem (RCSRP). One could think of a reduction from the RCSRP to the place-boundedness problem (or to the computation of the cover), by adding a new counter c_q getting increased each time the control-state q is hit. This does actually not work, because c_q might be unbounded even if on each single run, it is bounded. It seems that these two problems are not close: for solving the RCSRP, we need to decide whether there is an infinite run along which a given counter is unbounded, while the cover gives boundedness information about the *global* reachability set, but not on infinite runs. For VAS with one *weak* zero-test (for instance a lossy zero-test, like a reset), the usual Karp and Miller algorithm can be easily extended, and the RCSRP is decidable; for VAS with two weak zero-test (two resets), the techniques used in [6] allow one to show that this problem is undecidable. Finally, the RCSRP remains open for VAS_0.

We have proved new decidability results for VAS_0. One could think that maybe, VAS_0 can be simulated by VAS. The answer is negative: the language $\{a^n b^n \mid n \geqslant 1\}^*$ can be easily recognized by a VAS_0, but not by a VAS [16]. More generally, one may prove that for every VAS-language L, there is a VAS_0 $\mathcal{S}$ such that $L(\mathcal{S}) = L^*$. One can also separate VAS and VAS_0 wrt. the reachability set. Hence, even if their reachability problem is decidable [21] and their cover is computable (this paper), VAS_0 are strictly more powerful than VAS.

References

1. P. A. Abdulla, K. Cerans, B. Jonsson, and Y.-K. Tsay. General decidability theorems for infinite-state systems. In *LICS'96*, pages 313–321, 1996.
2. P. A. Abdulla and R. Mayr. Minimal cost reachability/coverability in priced timed Petri nets. In L. de Alfaro, editor, *FOSSACS*, volume 5504 of *LNCS*, pages 348–363. Springer, 2009.
3. A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. In Ch. Meinel and S. Tison, editors, *STACS'99*, volume 1563, pages 323–333, 1999.
4. R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, fourth edition, 2010.
5. C. Dufourd. *Réseaux de Petri avec Reset/Transfert : décidabilité et indécidabilité*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, Oct. 1998.
6. C. Dufourd, P. Jančar, and Ph. Schnoebelen. Boundedness of reset P/T nets. In J. Wiedermann, P. van Emde Boas, and M. Nielsen, editors, *ICALP'99*, volume 1644 of *LNCS*, pages 301–310. Springer, July 1999.

7 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, part I: Completions. In S. Albers and J.-Y. Marion, editors, *STACS'09*, pages 433–444, 2009.
8 A. Finkel and J. Goubault-Larrecq. Forward analysis for WSTS, Part II: Complete WSTS. In S. Albers et al., editors, *ICALP'09*, volume 5556 of *LNCS*, pages 188–199. Springer, 2009.
9 A. Finkel and A. Sangnier. Mixing coverability and reachability to analyze VASS with one zero-test. In D. Peleg and A. Muscholl, editors, *SOFSEM'10*, volume 5901 of *LNCS*, pages 394–406. Springer, 2010.
10 A. Finkel and Ph. Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1–2):63–92, 2001.
11 A. Finkel and G. Sutre. Decidability of reachability problems for classes of counters automata. In H. Reichel and S. Tison, editors, *STACS'00*, volume 1770 of *LNCS*, pages 346–357. Springer, 2000.
12 M. Hack. The equality problem for vector addition systems is undecidable. *Theor. Comput. Sci.*, 2(1):77–95, 1976.
13 D. Hauschildt. *Semilinearity of the Reachability Set is Decidable for Petri Nets.* PhD thesis, University of Hamburg, 1990.
14 J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theor. Comput. Sci.*, 8:135–159, 1979.
15 R. M. Karp and R. E. Miller. Parallel program schemata. *J. Comput. System Sci.*, 2:147–195, 1969.
16 S. R. Kosaraju. Limitations of Dijkstra's semaphore primitives and Petri nets. *SIGOPS Oper. Syst. Rev.*, 7(4):122–126, 1973.
17 S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *STOC'82*, pages 267–281. ACM, 1982.
18 J. Leroux. The general vector addition system reachability problem by Presburger inductive invariants. In *LICS'09*, pages 4–13, 2009.
19 E. W. Mayr. An algorithm for the general Petri net reachability problem. In *STOC'81*, pages 238–246. ACM, 1981.
20 R. Mayr. Undecidable problems in unreliable computations. *Theor. Comput. Sci.*, 297(1-3):337–354, 2003.
21 K. Reinhardt. Reachability in Petri Nets with inhibitor arcs. *Electr. Notes Theor. Comput. Sci.*, 223:239–264, 2008.