Affine Extensions of Integer Vector Addition Systems with States

Michael Blondin[1]
Technische Universität München, Germany
blondin@in.tum.de
https://orcid.org/0000-0003-2914-2734

Christoph Haase
University of Oxford, United Kingdom
christoph.haase@cs.ox.ac.uk
https://orcid.org/0000-0002-5452-936X

Filip Mazowiecki[2]
LaBRI, Université de Bordeaux, France
filip.mazowiecki@u-bordeaux.fr

Abstract

We study the reachability problem for affine $\mathbb{Z}$-VASS, which are integer vector addition systems with states in which transitions perform affine transformations on the counters. This problem is easily seen to be undecidable in general, and we therefore restrict ourselves to affine $\mathbb{Z}$-VASS with the finite-monoid property (afmp-$\mathbb{Z}$-VASS). The latter have the property that the monoid generated by the matrices appearing in their affine transformations is finite. The class of afmp-$\mathbb{Z}$-VASS encompasses classical operations of counter machines such as resets, permutations, transfers and copies. We show that reachability in an afmp-$\mathbb{Z}$-VASS reduces to reachability in a $\mathbb{Z}$-VASS whose control-states grow polynomially in the size of the matrix monoid. Our construction shows that reachability relations of afmp-$\mathbb{Z}$-VASS are semilinear, and in particular enables us to show that reachability in $\mathbb{Z}$-VASS with transfers and $\mathbb{Z}$-VASS with copies is PSPACE-complete.

2012 ACM Subject Classification Theory of computation → Logic and verification, Theory of computation → Automata over infinite objects, Theory of computation → Complexity classes

Keywords and phrases Vector addition systems, affine transformations, reachability, semilinear sets, computational complexity

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.14

Acknowledgements We are thankful to James Worrell for insightful discussions on transfer VASS.

1 Introduction

Vector addition systems with states (VASS) are a fundamental model of computation comprising a finite-state controller with a finite number of counters ranging over the natural numbers. When a transition is taken, a counter can be incremented or decremented provided that the resulting counter value is greater than or equal to zero. Since the counters of a

[1] Supported by the Fonds de recherche du Québec – Nature et technologies (FRQNT).
[2] This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Programme IdEx Bordeaux (ANR-10-IDEX-03-02).

VASS are unbounded, a VASS gives rise to an infinite transition system. One of the biggest advantages of VASS is that most of the standard decision problems such as configuration reachability and coverability are decidable [26, 32, 27, 29]. Those properties make VASS and their extensions a prime choice for reasoning about and modelling concurrent, distributed and parametrised systems, see *e.g.* the recent surveys by Abdulla and Delzanno [2, 15].

In order to increase their modelling power, numerous extensions of plain VASS have been proposed and studied in the literature over the last 25 years. Due to the infinite-state nature of VASS, even minor extensions often cross the undecidability frontier. For example, while in the extension of VASS with hierarchical zero-tests on counters both reachability and coverability remain decidable [37, 10], all important decision problems for VASS with two counters which can arbitrarily be tested for zero become undecidable [33]. Another example is the extension of VASS with resets and transfers. In a *reset VASS*, transitions may set a counter to zero, whereas *transfer VASS* generalise reset VASS and allow transitions to move the contents of a counter onto another. While it was initially widely believed that any extension of VASS either renders both reachability and coverability undecidable, reset and transfer VASS have provided an example of an extension which leads to an undecidable reachability [5] yet decidable coverability problem [16]. Nevertheless, the computational costs for those extensions are high: while coverability is EXPSPACE-complete for VASS [30, 35], it becomes Ackermann-complete in the presence of resets and transfers [38, 19]. For practical purposes, the extension of VASS with transfers is particularly useful since transfer VASS allow for reasoning about broadcast protocols and multi-threaded non-recursive C programs [17, 25]. It was already observed in [17] that transfer VASS can be viewed as an instance of so-called *affine VASS*. An affine VASS is an extended VASS with transitions labelled by pairs $(\mathbf{A}, \boldsymbol{b})$, where $\mathbf{A}$ is a $d \times d$ matrix over the integers and $\boldsymbol{b} \in \mathbb{Z}^d$ is an integer vector. A transition switches the control-state while updating the configuration of the counters $\boldsymbol{v} \in \mathbb{N}^d$ to $\mathbf{A} \cdot \boldsymbol{v} + \boldsymbol{b}$, provided that $\mathbf{A} \cdot \boldsymbol{v} + \boldsymbol{b} \geq \mathbf{0}$; otherwise, the transition is blocked. Transfer VASS can be viewed as affine VASS in which the columns of all matrices are d-dimensional unit vectors [17].

Due to the symbolic state-explosion problem and Ackermann-hardness of coverability, standard decision procedures for transfer VASS such as the backward algorithm [1] do not *per se* scale to real-world instances. In recent years, numerous authors have proposed the use of over-approximations in order to attenuate the symbolic state-explosion problem for VASS and some of their extensions (see, *e.g.*, [18, 6, 8]). Most commonly, the basic idea is to relax the integrality or non-negativity condition on the counters and to allow them to take values from the integers or non-negative rational numbers. It is easily seen that if a configuration is not reachable under the relaxed semantics, then the configuration is also not reachable under the standard semantics. Hence, those over-approximations can, for instance, be used in order to prune the sets of minimal basis elements in every iteration of the backward algorithm. In this paper, we investigate reachability in *integer over-approximations* of affine VASS, *i.e.*, affine VASS in which a configuration of the counters is a point in $\mathbb{Z}^d$, and in which all transitions are non-blocking. Subsequently, we refer to such VASS as *affine $\mathbb{Z}$-VASS*.

Main contributions

We focus on affine $\mathbb{Z}$-VASS with the *finite-monoid property* (afmp-$\mathbb{Z}$-VASS), *i.e.* where the matrix monoid generated by all matrices occurring along transitions in the affine $\mathbb{Z}$-VASS is finite. By a reduction to reachability in $\mathbb{Z}$-VASS, we obtain decidability of reachability for the whole class of afmp-$\mathbb{Z}$-VASS and semilinearity of their reachability relations.

More precisely, we show that reachability in an afmp-$\mathbb{Z}$-VASS can be reduced to reachability in a $\mathbb{Z}$-VASS whose size is polynomial in the size of the original afmp-$\mathbb{Z}$-VASS and in the size of the finite monoid $\mathcal{M}$ generated by the matrices occurring along transitions,

denoted by $\|\mathcal{M}\|$. For all classes of affine transformations considered in the literature, $\|\mathcal{M}\|$ is bounded exponentially in the dimension of the matrices. This enables us to deduce a general PSPACE upper bound for extensions of $\mathbb{Z}$-VASS such as transfer $\mathbb{Z}$-VASS and copy $\mathbb{Z}$-VASS. By a slightly more elaborated analysis of this construction, we are also able to provide a short proof of the already known NP upper bound for reset $\mathbb{Z}$-VASS [21].

We also show that a PSPACE lower bound of the reachability problem already holds for the extension of reset $\mathbb{Z}$-VASS with permutations. This gives PSPACE-completeness of some interesting classes such as transfer $\mathbb{Z}$-VASS and copy $\mathbb{Z}$-VASS. Finally, we show that an affine $\mathbb{Z}$-VASS that allows for both transfers and copies may not have the finite-monoid property, and the reachability problem for this class becomes undecidable. All complexity results obtained in this paper are summarized in Figure 1.

Related work

Our work is primarily related to the work of Finkel and Leroux [20], Iosif and Sangnier [24], Haase and Halfon [21], and Cadilhac, Finkel and McKenzie [12, 13]. In [20], Finkel and Leroux consider a model more general than affine $\mathbb{Z}$-VASS in which transitions are additionally equipped with guards which are Presburger formulas defining admissible sets of vectors in which a transition does not block. Given a sequence of transitions σ, Finkel and Leroux show that the reachability set obtained from repeatedly iterating σ, *i.e.*, the *acceleration* of σ, is definable in Presburger arithmetic. Note that the model of Finkel and Leroux does not allow for control-states and the usual tricks of encoding each control-state by a counter or all control-states into three counters [22] do not work over $\mathbb{Z}$ since transitions are non blocking. Iosif and Sangnier [24] investigated the complexity of model checking problems for a variant of the model of Finkel and Leroux with guards defined by convex polyhedra and with control-states over a flat structure. Haase and Halfon [21] studied the complexity of the reachability, coverability and inclusion problems for $\mathbb{Z}$-VASS and reset $\mathbb{Z}$-VASS, two submodels of the affine $\mathbb{Z}$-VASS that we study in this paper. In [12, 13], Cadilhac, Finkel and McKenzie consider an extension of Parikh automata to affine Parikh automata with the finite-monoid restriction like in our paper. These are automata recognizing boolean languages, but the finite-monoid restriction was exploited in a similar way to obtain some decidability results in that context. We finally remark that our models capture variants of cost register automata that have only one + operation [4, 3].

Structure of the paper

We introduce general notations and affine $\mathbb{Z}$-VASS in Section 2. In Section 3, we give the reduction from afmp-$\mathbb{Z}$-VASS to $\mathbb{Z}$-VASS. Subsequently, in Section 4 we show that afmp-$\mathbb{Z}$-VASS have semilinear reachability relations and discuss semilinearity of affine $\mathbb{Z}$-VASS in general. In Section 5, we show the PSPACE and NP upper bounds of the reachability problem for some classes of afmp-$\mathbb{Z}$-VASS; and in Section 6 we show PSPACE-hardness and undecidability results for some classes of affine $\mathbb{Z}$-VASS. Some concluding remarks will be made in Section 7.

2 Preliminaries

General notation

For every $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, 2, \ldots, n\}$. For every $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathbb{Z}^d$ and every $i \in [d]$, we write $\boldsymbol{x}(i) \stackrel{\text{def}}{=} x_i$. We denote the identity matrix and the zero-vector by $\mathbf{I}$ and $\mathbf{0}$ in every dimension, as there will be no ambiguity. For every $\boldsymbol{x} \in \mathbb{Z}^d$ and $\mathbf{A} \in \mathbb{Z}^{d \times d}$, we define

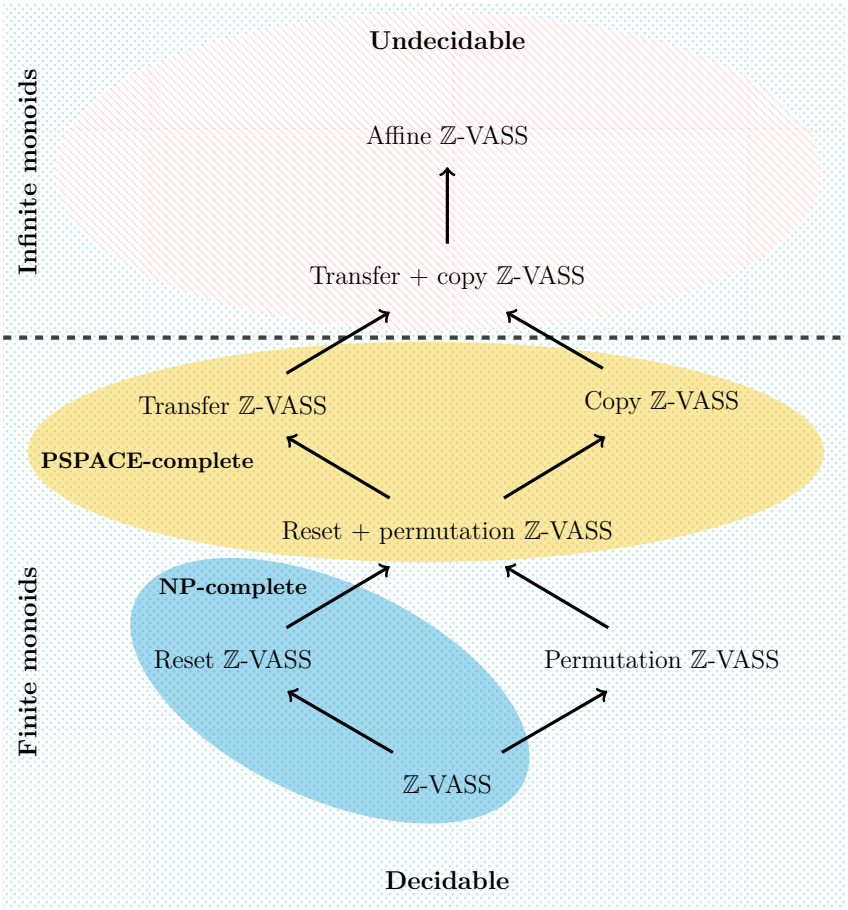

Figure 1 Classification of the complexity of reachability in affine ℤ-VASS in terms of classes of matrices. The rectangular regions below and above the horizontal dashed line correspond to classes of matrices with finite and infinite monoids respectively. The green rectangular dotted region and the red elliptical striped region correspond to the classes where reachability is decidable and undecidable, respectively. The blue elliptical region and the orange elliptical region correspond to the classes where reachability is NP-complete and PSPACE-complete respectively. Reachability in permutation ℤ-VASS is NP-hard and belongs to PSPACE.

the *max-norm* of $\boldsymbol{x}$ and $\mathbf{A}$ as $\|\boldsymbol{x}\| \stackrel{\text{def}}{=} \max\{|\boldsymbol{x}(i)| : i \in [d]\}$ and $\|\mathbf{A}\| \stackrel{\text{def}}{=} \max\{\|\mathbf{A}_i\| : i \in [d]\}$ where $\mathbf{A}_i$ denotes the i^{th} column of $\mathbf{A}$. We assume that numbers are represented in binary, hence the entries of vectors and matrices can be exponential in the size of their encodings.

Affine Integer VASS

An *affine integer vector addition system with states (affine ℤ-VASS)* is a tuple $\mathcal{V} = (d, Q, T)$ where $d \in \mathbb{N}$, Q is a finite set and $T \subseteq Q \times \mathbb{Z}^{d\times d} \times \mathbb{Z}^d \times Q$. Let us fix such a $\mathcal{V}$. We call d the *dimension* of $\mathcal{V}$ and the elements of Q and T respectively *control-states* and *transitions*. For every transition $t = (p, \mathbf{A}, \boldsymbol{b}, q)$, let $\mathrm{src}(t) \stackrel{\text{def}}{=} p$, $\mathrm{tgt}(t) \stackrel{\text{def}}{=} q$, $M(t) \stackrel{\text{def}}{=} \mathbf{A}$ and $\Delta(t) \stackrel{\text{def}}{=} \boldsymbol{b}$, and let $f_t \colon \mathbb{Z}^d \to \mathbb{Z}^d$ be the affine transformation defined by $f_t(\boldsymbol{x}) = \mathbf{A} \cdot \boldsymbol{x} + \boldsymbol{b}$. The *size* of $\mathcal{V}$, denoted $|\mathcal{V}|$, is defined as $|\mathcal{V}| \stackrel{\text{def}}{=} d + |Q| + \|T\|$ where $\|T\| \stackrel{\text{def}}{=} \sum_{t\in T} d^2 \cdot \lceil \log(\|M(t)\| + \|\Delta(t)\| + 1) \rceil$.

A *configuration* of $\mathcal{V}$ is a pair $(q, \boldsymbol{v}) \in Q \times \mathbb{Z}^d$ which we write as $q(\boldsymbol{v})$. For every $t \in T$ and $p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{Z}^d$, we write $p(\boldsymbol{u}) \xrightarrow{t} q(\boldsymbol{v})$ if $p = \mathrm{src}(t)$, $q = \mathrm{tgt}(t)$ and $\boldsymbol{v} = f_t(\boldsymbol{u})$. We naturally extend $\to$ to sequences of transitions as follows. For every $w \in T^*$ and

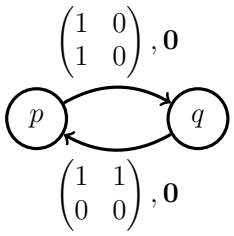

Figure 2 Example of a transfer + copy $\mathbb{Z}$-VASS $\mathcal{V}$ which does not have the finite-monoid property.

$p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{Z}^d$, we write $p(\boldsymbol{u}) \xrightarrow{w} q(\boldsymbol{v})$ if either $|w| = 0$ and $p(\boldsymbol{u}) = q(\boldsymbol{v})$, or $|w| = k > 0$ and there exist $p_0(\boldsymbol{u}_0), p_1(\boldsymbol{u}_1), \ldots, p_k(\boldsymbol{u}_k) \in Q \times \mathbb{Z}^d$ such that

$$p(\boldsymbol{u}) = p_0(\boldsymbol{u}_0) \xrightarrow{w_1} p_1(\boldsymbol{u}_1) \xrightarrow{w_2} \cdots \xrightarrow{w_k} p_k(\boldsymbol{u}_k) = q(\boldsymbol{v}).$$

We write $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ if there exists some $w \in T^*$ such that $p(\boldsymbol{u}) \xrightarrow{w} q(\boldsymbol{v})$. The relation $\xrightarrow{*}$ is called the *reachability relation* of $\mathcal{V}$. If $p(\boldsymbol{u}) \xrightarrow{w} q(\boldsymbol{v})$, then we say that w is a *run from* $p(\boldsymbol{u})$ *to* $q(\boldsymbol{v})$, or simply a *run* if the source and target configurations are irrelevant. We also say that w is a *path* from p to q, and if $p = q$ then we say that w is a *cycle*.

Let $M(\mathcal{V}) \stackrel{\text{def}}{=} \{M(t) : t \in T\}$ and $\Delta(\mathcal{V}) \stackrel{\text{def}}{=} \{\Delta(t) : t \in T\}$. If $\mathcal{V}$ is clear from the context, we sometimes simply write M and Δ. The *monoid of* $\mathcal{V}$, denoted $\mathcal{M}_\mathcal{V}$ or sometimes simply $\mathcal{M}$, is the monoid generated by $M(\mathcal{V})$, *i.e.* it is the smallest set that contains $M(\mathcal{V})$, is closed under matrix multiplication, and contains the identity matrix. We say that a matrix $\mathbf{A} \in \mathbb{Z}^{d\times d}$ is respectively a (i) *reset*, (ii) *permutation*, (iii) *transfer*, (iv) *copyless*, or (v) *copy* matrix if $\mathbf{A} \in \{0,1\}^{d\times d}$ and

- **(i)** $\mathbf{A}$ does not contain any 1 outside of its diagonal;
- **(ii)** $\mathbf{A}$ has exactly one 1 in each row and each column;
- **(iii)** $\mathbf{A}$ has exactly one 1 in each column;
- **(iv)** $\mathbf{A}$ has at most one 1 in each column;
- **(v)** $\mathbf{A}$ has exactly one 1 in each row.

Similarly, we say that $\mathcal{V}$ is respectively a *reset*, *permutation*, *transfer*, *copyless*, or *copy* $\mathbb{Z}$-VASS if all matrices of $M(\mathcal{V})$ are reset, permutation, transfer, copyless, or copy matrices. The monoids of such affine $\mathbb{Z}$-VASS are finite and respectively of size at most 2^d, $d!$, d^d, $(d+1)^d$ and d^d. Copyless $\mathbb{Z}$-VASS correspond to a model of copyless cost-register automata studied in [3] (see the remark below). If $M(\mathcal{V})$ only contains the identity matrix, then $\mathcal{V}$ is simply called a $\mathbb{Z}$-VASS. We define $\|\mathcal{M}_\mathcal{V}\| \stackrel{\text{def}}{=} |\mathcal{M}_\mathcal{V}| \cdot d^2 \cdot \max\{\log(\|\mathbf{A}\| + 1) : \mathbf{A} \in \mathcal{M}_\mathcal{V}\}$. Note that $\|\mathcal{M}_\mathcal{V}\| = |\mathcal{M}_\mathcal{V}| \cdot d^2$ for any monoid obtained from one of the above matrices types.

A *class of matrices* $\mathcal{C}$ is a union $\bigcup_{d\geq 1} \mathcal{C}_d$ where $\mathcal{C}_d$ is a finitely generated, but possibly infinite, submonoid of $\mathbb{Z}^{d\times d}$ for every $d \geq 1$. We say that $\mathcal{V}$ belongs to a class $\mathcal{C}$ of $\mathbb{Z}$-VASS if $\mathcal{M}_\mathcal{V} \subseteq \mathcal{C}$. If each $\mathcal{C}_d$ is finite, then we say that this class of affine $\mathbb{Z}$-VASS has the *finite-monoid property* (afmp-$\mathbb{Z}$-VASS). For two classes $\mathcal{C}$ and $\mathcal{C}'$ we write $\mathcal{C} + \mathcal{C}'$ to denote the smallest set $\mathcal{D} = \bigcup_{d\geq 1} \mathcal{D}_d$ such that $\mathcal{D}_d$ is a monoid that contains both $\mathcal{C}_d$ and $\mathcal{C}'_d$ for every $d \geq 1$. Notice that this operation does not preserve finiteness and for example the class of transfer + copy matrices is infinite (see Figure 2 and Section 6).

We discuss the $\mathbb{Z}$-VASS $\mathcal{V}$ in Figure 2 to give some intuition behind the names transfer and copy $\mathbb{Z}$-VASS. The transition from p to q is a copy transition and the transition from q to p is a transfer transition. Notice that for every vector $(x, y) \in \mathbb{Z}^2$, we have $p(x, y) \to q(x, x)$, *i.e.* the value of the first counter is copied to the second counter. Similarly, for the other

transition we have $q(x, y) \to p(x + y, 0)$, that is the value of the second counter is transferred to the first counter (resetting its own value to 0). Let $\mathbf{A}$ and $\mathbf{B}$ be the two matrices used in $\mathcal{V}$. Note that $(\mathbf{A} \cdot \mathbf{B})^n$ is the matrix with all entries equal to 2^{n-1}, and hence $\mathcal{M}_\mathcal{V}$ is infinite.

▶ Remark. The variants of affine $\mathbb{Z}$-VASS that we consider are related to cost register automata (CRA) with only the $+$ operation [4, 3] and without an output function. These are deterministic models with states and registers that upon reading an input, update their registers in the form $x \leftarrow y + c$, where x, y are registers and c is an integer. An affine $\mathbb{Z}$-VASS does not read any input, but is nondeterministic. Thus, one can identify an affine $\mathbb{Z}$-VASS with a CRA that reads sequences of transitions as words. In particular, the restrictions imposed on the studied CRAs correspond to copy $\mathbb{Z}$-VASS [4] and copyless $\mathbb{Z}$-VASS [3].

Decision problems

We consider the *reachability* and the *coverability* problems parameterized by classes of matrices $\mathcal{C}$:

$\mathsf{Reach}_\mathcal{C}$ (reachability problem)

GIVEN:	an affine $\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$ and configurations $p(\boldsymbol{u}), q(\boldsymbol{v})$ such that $\mathcal{M}_\mathcal{V} \subseteq \mathcal{C}$.
DECIDE:	whether $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$?

$\mathsf{Cover}_\mathcal{C}$ (coverability problem)

GIVEN:	an affine $\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$ and configurations $p(\boldsymbol{u}), q(\boldsymbol{v})$ such that $\mathcal{M}_\mathcal{V} \subseteq \mathcal{C}$.
DECIDE:	whether there exists $\boldsymbol{v}' \in \mathbb{Z}^d$ such that $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v}')$ and $\boldsymbol{v}' \geq \boldsymbol{v}$?

For standard VASS (where configurations cannot hold negative values), the coverability problem is considered much simpler than the reachability problem. However, for affine $\mathbb{Z}$-VASS, these two problems coincide as observed in [21, Lemma 2]: the two problems are inter-reducible in logarithmic space at the cost of doubling the number of counters. Therefore we will only study the reachability problem in this paper.

3 From affine $\mathbb{Z}$-VASS with the finite-monoid property to $\mathbb{Z}$-VASS

The main result of this section is that every affine $\mathbb{Z}$-VASS $\mathcal{V}$ with the finite monoid can be simulated by a $\mathbb{Z}$-VASS with twice the number of counters whose size is polynomial in $\|\mathcal{M}\|$ and $|\mathcal{V}|$. More formally, we show the following:

▶ **Theorem 1.** *For every afmp-$\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$ there exist a $\mathbb{Z}$-VASS $\mathcal{V}' = (d', Q', T')$ and $p', q' \in Q'$ such that*

- $d' = 2 \cdot d$,
- $|Q'| \leq 4 \cdot \|\mathcal{M}\|^2 \cdot |Q|$,
- $\|T'\| \leq 8d \cdot \|\mathcal{M}\|^2 \cdot |Q| + \|\mathcal{M}\|^4 \cdot \|T\|$,
- $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ *in* $\mathcal{V}$ *if and only if* $p'(\boldsymbol{u}, \mathbf{0}) \xrightarrow{*} q'(\mathbf{0}, \boldsymbol{v})$ *in* $\mathcal{V}'$.

Moreover, $\mathcal{V}'$, p' and q' are effectively computable from $\mathcal{V}$.

▶ **Corollary 2.** *The reachability problem for afmp-$\mathbb{Z}$-VASS is decidable.*

Proof. By Theorem 1, it suffices to construct, for a given afmp-$\mathbb{Z}$-VASS $\mathcal{V}$, the $\mathbb{Z}$-VASS $\mathcal{V}'$ and to test for reachability in $\mathcal{V}'$. It is known that reachability for $\mathbb{Z}$-VASS is in NP [21]. To effectively compute $\mathcal{V}'$ it suffices to provide a bound for $\|\mathcal{M}_\mathcal{V}\|$. It is known that if $|\mathcal{M}_\mathcal{V}|$ is finite then it is bounded by a computable function, which is an exponential tower (see [31]), and hence $\|\mathcal{M}_\mathcal{V}\|$ is also computable. ◀

For the remainder of this section, let us fix some affine $\mathbb{Z}$-VASS $\mathcal{V}$ such that $\mathcal{M}_\mathcal{V}$ is finite. We proceed as follows to prove Theorem 1. First, we introduce some notations and intermediary lemmas characterizing reachability in affine $\mathbb{Z}$-VASS. Next, we give a construction that essentially proves the special case of Theorem 1 where the initial configuration is of the form $p(\mathbf{0})$. Finally, we prove Theorem 1 by extending this construction to the general case.

It is worth noting that proving the general case is not necessary if one is only interested in deciding reachability. Indeed, an initial configuration $p(\boldsymbol{v})$ can be turned into one of the form $p'(\mathbf{0})$ by adding a transition that adds $\boldsymbol{v}$. The reason for proving the general case is that it establishes a stronger relation that allows us to prove semilinearity of afmp-$\mathbb{Z}$-VASS reachability relations in Section 4.

3.1 A characterization of reachability

For every $\sigma \in T^*$, $t \in T$ and $\boldsymbol{u} \in \mathbb{Z}^d$, let

$$M(\varepsilon) \stackrel{\text{def}}{=} \mathbf{I}, \qquad \varepsilon(\boldsymbol{u}) \stackrel{\text{def}}{=} \boldsymbol{u},$$
$$M(\sigma t) \stackrel{\text{def}}{=} M(t) \cdot M(\sigma), \qquad \sigma t(\boldsymbol{u}) \stackrel{\text{def}}{=} M(t) \cdot \sigma(\boldsymbol{u}) + \Delta(t).$$

Intuitively, for any sequence $w \in T^*$, $w(\boldsymbol{u})$ is the effect of w on $\boldsymbol{u}$, regardless of whether w is an actual path of the underlying graph. A simple induction yields the following characterization:

▶ **Lemma 3.** *For every $w \in T^*$ and $p(\boldsymbol{u}), q(\boldsymbol{v}) \in Q \times \mathbb{Z}^d$, it is the case that $p(\boldsymbol{u}) \xrightarrow{w} q(\boldsymbol{v})$ if and only if*

(a) *w is a path from p to q in the underlying graph of $\mathcal{V}$, and*
(b) *$\boldsymbol{v} = w(\boldsymbol{u})$.*

Testing for reachability with Lemma 3 requires evaluating $w(\boldsymbol{u})$. This value can be evaluated conveniently as follows:

▶ **Lemma 4.** *For every $w \in T^k$ and $\boldsymbol{u} \in \mathbb{Z}^d$, the following holds:*

$$w(\boldsymbol{u}) = M(w) \cdot \boldsymbol{u} + \sum_{i=1}^{k} M(w_{i+1}w_{i+2} \cdots w_k) \cdot \Delta(w_i). \tag{1}$$

Moreover, $w(\boldsymbol{u}) = M(w) \cdot \boldsymbol{u} + w(\mathbf{0})$.

Proof of Lemma 4. We prove (1) by induction on k. The base case follows from $\varepsilon(\boldsymbol{u}) = \boldsymbol{u} = \mathbf{I} \cdot \boldsymbol{u} + \mathbf{0} = M(\varepsilon) \cdot \boldsymbol{u} + \mathbf{0}$. Assume that $k > 0$ and that the claim holds for sequences of length $k - 1$. For simplicity we denote $\sigma \stackrel{\text{def}}{=} w_1 \dots w_{k-1}$. We have:

$$\begin{aligned}
w(\boldsymbol{u}) &= \sigma w_k(\boldsymbol{u}) \\
&= M(w_k) \cdot \sigma(\boldsymbol{u}) + \Delta(w_k) && (2)\\
&= M(w_k) \cdot \left(M(\sigma) \cdot \boldsymbol{u} + \sum_{i=1}^{k-1} M(w_{i+1}w_{i+2} \cdots w_{k-1}) \cdot \Delta(w_i) \right) + \Delta(w_k) && (3)\\
&= M(w_k) \cdot M(\sigma) \cdot \boldsymbol{u} + \sum_{i=1}^{k-1} M(w_k) \cdot M(w_{i+1}w_{i+2} \cdots w_{k-1}) \cdot \Delta(w_i) + \Delta(w_k) \\
&= M(\sigma w_k) \cdot \boldsymbol{u} + \sum_{i=1}^{k-1} M(w_{i+1}w_{i+2} \cdots w_k) \cdot \Delta(w_i) + \Delta(w_k) && (4)\\
&= M(w) \cdot \boldsymbol{u} + \sum_{i=1}^{k} M(w_{i+1}w_{i+2} \cdots w_k) \cdot \Delta(w_i)
\end{aligned}$$

where (2), (3) and (4) follow respectively by definition of $\sigma w_k(\boldsymbol{u})$, by induction hypothesis and by definition of $M(\sigma w_k)$.

The last part of the lemma follows from applying (1) to $w(\mathbf{0})$ and $w(\boldsymbol{u})$, and observing that subtracting them results in $w(\boldsymbol{u}) - w(\mathbf{0}) = M(w) \cdot \boldsymbol{u}$. ◀

Observe that Lemma 4 is trivial for the particular case of $\mathbb{Z}$-VASS. Indeed, we obtain $w(\boldsymbol{u}) = \boldsymbol{u} + \sum_{i=1}^{k} \Delta(w_i)$, which is the sum of transition vectors as expected for a $\mathbb{Z}$-VASS.

3.2 Reachability from the origin

We make use of Lemmas 3 and 4 to construct a $\mathbb{Z}$-VASS $\mathcal{V}' = (d, Q', T')$ for the special case of Theorem 1 where the initial configuration is of the form $p(\mathbf{0})$. The states and transitions of $\mathcal{V}'$ are defined as:

$$Q' \stackrel{\text{def}}{=} Q \times \mathcal{M},$$

$$T' \stackrel{\text{def}}{=} \{((\mathrm{src}(t), \mathbf{A}), \mathbf{I}, \mathbf{B} \cdot \Delta(t), (\mathrm{tgt}(t), \mathbf{B})) : \mathbf{A}, \mathbf{B} \in \mathcal{M}, t \in T \text{ and } \mathbf{B} \cdot M(t) = \mathbf{A}\}.$$

The idea behind $\mathcal{V}'$ is to simulate a path w of $\mathcal{V}$ forward while evaluating $w(\mathbf{0})$ backwards. The latter can be evaluated as the sum identified in Lemma 4 provided that $\mathcal{V}'$ initially "knows" $M(w)$. More formally, $\mathcal{V}'$ and $\mathcal{V}$ are related as follows:

▶ **Proposition 5.**

(a) *For every $w \in T^*$ if $p(\mathbf{0}) \xrightarrow{w} q(\boldsymbol{v})$ in $\mathcal{V}$, then $p'(\mathbf{0}) \xrightarrow{*} q'(\boldsymbol{v})$ in $\mathcal{V}'$, where $p' = (p, M(w))$ and $q' = (q, \mathbf{I})$.*

(b) *If $p'(\mathbf{0}) \xrightarrow{*} q'(\boldsymbol{v})$ in $\mathcal{V}'$, where $p' = (p, \mathbf{A})$ and $q' = (q, \mathbf{I})$, then there exists $w \in T^*$ such that $M(w) = \mathbf{A}$ and $p(\mathbf{0}) \xrightarrow{w} q(\boldsymbol{v})$ in $\mathcal{V}$.*

Proof. **(a)** By Lemma 3, $\mathcal{V}$ has a path $w \in T^*$ such that $w(\mathbf{0}) = \boldsymbol{v}$. Let $k \stackrel{\text{def}}{=} |w|$. For every $i \in [k+1]$, let

$$\mathbf{A}_i \stackrel{\text{def}}{=} M(w_i w_{i+1} \cdots w_k)$$

with the convention that $A_{k+1} = \mathbf{I}$. For every $i \in [k]$, let

$$\boldsymbol{b}_i \stackrel{\text{def}}{=} \mathbf{A}_{i+1} \cdot \Delta(w_i),$$

$$w'_i \stackrel{\text{def}}{=} ((\mathrm{src}(w_i), \mathbf{A}_i), \mathbf{I}, \boldsymbol{b}_i, (\mathrm{tgt}(w_i), \mathbf{A}_{i+1})).$$

We claim that $w' \stackrel{\text{def}}{=} w'_1 w'_2 \cdots w'_k$ is such that $(p, \mathbf{A}_1) \xrightarrow{w'} (q, \mathbf{A}_{k+1})$ in $\mathcal{V}'$. Note that the validity of the claim completes the proof since $\mathbf{A}_1 = M(w)$ and $\mathbf{A}_{k+1} = \mathbf{I}$.

It follows immediately from the definition of T' that $w'_i \in T'$ for every $i \in [k]$ and hence that w' is a path from $(p, \mathbf{A}_1)$ to $(q, \mathbf{A}_k)$. By Lemma 3, it remains to show that $w'(\mathbf{0}) = \boldsymbol{v}$:

$$\begin{aligned}
w'(\mathbf{0}) &= \sum_{i=1}^{k} M(w'_{i+1} w'_{i+2} \cdots w'_k) \cdot \Delta(w'_i) && \text{(by Lemma 4 applied to } w'(\mathbf{0})) \\
&= \sum_{i=1}^{k} \Delta(w'_i) && \text{(by } M(w'_i) = \mathbf{I} \text{ for every } i \in [k]) \\
&= \sum_{i=1}^{k} \mathbf{A}_{i+1} \cdot \Delta(w_i) && \text{(by definition of } \Delta(w'_i)) \\
&= \sum_{i=1}^{k} M(w_{i+1} w_{i+2} \cdots w_k) \cdot \Delta(w_i) && \text{(by definition of } \mathbf{A}_{i+1}) \\
&= w(\mathbf{0}) && \text{(by Lemma 4 applied to } w(\mathbf{0})).
\end{aligned}$$

(b) Similarly, by Lemma 3, there exists a path w' of $\mathcal{V}'$ such that $w'(\mathbf{0}) = \boldsymbol{v}$, and it suffices to exhibit a path $w \in T^*$ from p to q in $\mathcal{V}$ such that $w(\mathbf{0}) = \boldsymbol{v}$ and $M(w) = \mathbf{A}$. Let $k \stackrel{\text{def}}{=} |w'|$. For every $i \in [k]$, let $w'_i = ((p_i, \mathbf{A}_i), \mathbf{I}, \boldsymbol{b}_i, (q_i, \mathbf{B}_i))$. By definition of T', for every $i \in [k]$, there exists a (possibly non unique) transition $t_i \in T$ such that $\mathrm{src}(t) = p_i$, $\mathrm{tgt}(t) = q_i$, $\boldsymbol{b}_i = \mathbf{B}_i \cdot \Delta(t_i)$ and $\mathbf{B}_i \cdot M(t_i) = \mathbf{A}_i$. We set $w \stackrel{\text{def}}{=} t_1 t_2 \cdots t_k$. It is readily seen that w is a path from p to q. To prove $w(\mathbf{0}) = \boldsymbol{v}$ and $M(w) = \mathbf{A}$, Lemma 4 can be applied as in the previous implication. ◀

3.3 Reachability from an arbitrary configuration

We now construct the $\mathbb{Z}$-VASS $\mathcal{V}'' = (2d, Q'', T'')$ of Theorem 1 which is obtained mostly from $\mathcal{V}'$. The states of $\mathcal{V}''$ are defined as

$$Q'' \stackrel{\text{def}}{=} Q_i \cup (Q \times \mathcal{M} \times \mathcal{M}) \cup (Q \times \mathcal{M}) \cup Q_f$$

where $Q_i = \{q_i : q \in Q\}$ and $Q_f = \{q_f : q \in Q\}$. To simplify the notation, given two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}^d$ we write $(\boldsymbol{u}, \boldsymbol{v})$ for the vector of $\mathbb{Z}^{2d}$ equal to $\boldsymbol{u}$ on the first d components and equal to $\boldsymbol{v}$ on the last d components. The set T'' consists of five disjoint subsets of transitions $T_{\text{init}} \cup T_{\text{simul}} \cup T_{\text{end}} \cup T_{\text{mult}} \cup T_{\text{final}}$ working in five sequential stages. Intuitively, these transitions allow $\mathcal{V}''$ to guess a matrix $\mathbf{A}_{\text{guess}}$, to simulate a path w of $\mathcal{V}$ such that $\mathbf{A}_{\text{guess}} = M(w)$, to compute $w(\mathbf{0})$ and finally to compute $w(\mathbf{0}) + \mathbf{A}_{\text{guess}} \cdot \boldsymbol{u}$.

The first set of transitions is defined as:

$$T_{\text{init}} \stackrel{\text{def}}{=} \{(q_i, \mathbf{I}, (\mathbf{0}, \mathbf{0}), (q, \mathbf{C}, \mathbf{C})) : q \in Q, \mathbf{C} \in \mathcal{M}\}.$$

Its purpose is to move from Q_i to $Q \times \mathcal{M} \times \mathcal{M}$, thereby storing two copies of the guessed matrix $\mathbf{A}_{\text{guess}}$. The second set is defined as:

$$T_{\text{simul}} \stackrel{\text{def}}{=} \{((p, \mathbf{A}, \mathbf{C}), \mathbf{I}, (\mathbf{0}, \boldsymbol{b}), (q, \mathbf{B}, \mathbf{C})) : \mathbf{C} \in \mathcal{M}, ((p, \mathbf{A}), \mathbf{I}, \boldsymbol{b}, (q, \mathbf{B})) \in T'\}.$$

Its purpose is to simulate T' in the two first components of $Q \times \mathcal{M} \times \mathcal{M}$ and to remember $\mathbf{A}_{\text{guess}}$ in the third component. The third set is defined as:

$$T_{\text{end}} \stackrel{\text{def}}{=} \{((q, \mathbf{I}, \mathbf{C}), \mathbf{I}, (\mathbf{0}, \mathbf{0}), (q, \mathbf{C})) : (q, \mathbf{I}, \mathbf{C}) \in Q''\},$$

and its purpose is to move from $Q \times \mathcal{M} \times \mathcal{M}$ to $Q \times \mathcal{M}$, thus guessing the end of a run in $\mathcal{V}'$, i.e. by reaching $\mathbf{I}$. The fourth set is defined as:

$$\begin{aligned} T_{\text{mult}} \stackrel{\text{def}}{=} &\{((q, \mathbf{C}), \mathbf{I}, (-\boldsymbol{e_i}, \mathbf{C} \cdot \boldsymbol{e_i}), (q, \mathbf{C})) : q \in Q, \mathbf{C} \in \mathcal{M}, i \in [d]\} \cup \\ &\{((q, \mathbf{C}), \mathbf{I}, (\boldsymbol{e_i}, -\mathbf{C} \cdot \boldsymbol{e_i}), (q, \mathbf{C})) : q \in Q, \mathbf{C} \in \mathcal{M}, i \in [d]\}, \end{aligned}$$

where $\boldsymbol{e_i}$ is the unit vector such that $\boldsymbol{e_i}(i) = 1$. The purpose of T_{mult} is to compute $\mathbf{A}_{\text{guess}} \cdot \boldsymbol{u}$. Finally, T_{final} is defined as:

$$T_{\text{final}} \stackrel{\text{def}}{=} \{((q, \mathbf{C}), \mathbf{I}, (\mathbf{0}, \mathbf{0}), q_f) : q \in Q, \mathbf{C} \in \mathcal{M}\},$$

and its purpose is to move from $Q \times \mathcal{M}$ to Q_f, guessing the end of the matrix multiplication performed with T_{mult}.

We may now prove Theorem 1:

Proof of Theorem 1. First, note that we obtain

$$\begin{aligned}|Q''| &= (2 + \|\mathcal{M}\| + \|\mathcal{M}\|^2) \cdot |Q| \\ &\leq 4 \cdot \|\mathcal{M}\|^2 \cdot |Q|,\end{aligned}$$

$$\begin{aligned}\|T''\| &= 2 \cdot \|\mathcal{M}\| \cdot |Q| + \|\mathcal{M}\| \cdot \|T'\| + |Q''| + 2d \cdot \|\mathcal{M}\| \cdot |Q| \\ &\leq \|\mathcal{M}\|^4 \cdot \|T\| + 8d \cdot \|\mathcal{M}\|^2 \cdot |Q|,\end{aligned}$$

where we use the fact that $\|T'\| \leq \|\mathcal{M}\|^2 \cdot \|T\| \cdot \max\{\|\mathbf{A}\| : \mathbf{A} \in \mathcal{M}\} \leq \|\mathcal{M}\|^3 \cdot \|T\|$.

It remains to show that $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ in $\mathcal{V}$ if and only if $p_i(\boldsymbol{u}, \mathbf{0}) \xrightarrow{*} q_f(\mathbf{0}, \boldsymbol{v})$ in $\mathcal{V}''$.

$\Rightarrow$) By Lemma 3, there exists a path w of $\mathcal{V}$ such that $w(\boldsymbol{u}) = \boldsymbol{v}$. By definition of T_{init}, T_{simul} and T_{end}, and by Proposition 5, it is the case that $p_i(\boldsymbol{u}, \mathbf{0}) \xrightarrow{*} r(\boldsymbol{u}, w(\mathbf{0}))$ where $r = (q, M(w))$. The transitions of T_{mult} allow to transform $(\boldsymbol{u}, w(\mathbf{0}))$ into $(\mathbf{0}, w(\mathbf{0}) + M(w) \cdot \boldsymbol{u})$. Thus, using T_{final}, we can reach the configuration $q_f(w(\mathbf{0}) + M(w) \cdot \boldsymbol{u})$. This concludes the proof since $w(\boldsymbol{u}) = w(\mathbf{0}) + M(w) \cdot \boldsymbol{u}$ by Lemma 4.

$\Leftarrow$) The converse implication follows the same steps as the previous one. It suffices to observe that the first part of a run of $\mathcal{V}''$ defines the value $w(\mathbf{0})$, while the second part of the run defines $M(w) \cdot \boldsymbol{u}$. ◀

4 Semilinearity of affine $\mathbb{Z}$-VASS

We say that a subset of $\mathbb{Z}^d$ is *semilinear* if it is definable by a Presburger formula [34], *i.e.* by a formula of $\mathsf{FO}(\mathbb{Z}, +, <)$, the first-order logic over $\mathbb{Z}$ with addition and order. Semilinear sets capture precisely finite unions of sets of the form $\boldsymbol{b} + \mathbb{N} \cdot \boldsymbol{p_1} + \mathbb{N} \cdot \boldsymbol{p_2} + \ldots + \mathbb{N} \cdot \boldsymbol{p_k}$, and are closed under basic operations such as finite sums, intersection and complement. Semilinear sets are important in formal verification, in particular because satisfiability of Presburger formulas is decidable [34] and in NP for the existential fragment [11].

The results of Section 3 allow us to show that any affine $\mathbb{Z}$-VASS with the finite-monoid property has a semilinear reachability relation:

▶ **Theorem 6.** *Given an afmp-$\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$ and $p, q \in Q$, it is possible to compute an existential Presburger formula $\varphi_{\mathcal{V},p,q}$ of size at most $O(\text{poly}(|\mathcal{V}|, \|\mathcal{M}_\mathcal{V}\|))$ such that $\varphi_{\mathcal{V},p,q}(\boldsymbol{u}, \boldsymbol{v})$ holds if and only if $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ in $\mathcal{V}$.*

Proof. By Theorem 1, there exist an effectively computable $\mathbb{Z}$-VASS $\mathcal{V}' = (d', Q', T')$ and $p', q' \in Q'$ such that $d' = 2 \cdot d$, $|Q'| \leq 4 \cdot \|\mathcal{M}\|^2 \cdot |Q|$, $\|T'\| \leq 8d \cdot \|\mathcal{M}\|^2 \cdot |Q| + \|\mathcal{M}\|^4 \cdot \|T\|$ and

$$p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v}) \text{ in } \mathcal{V} \text{ if and only if } p'(\boldsymbol{u}, \mathbf{0}) \xrightarrow{*} q'(\mathbf{0}, \boldsymbol{v}) \text{ in } \mathcal{V}'. \tag{5}$$

By [21, Sect. 3], we can compute an existential Presburger formula ψ of linear size in $\mathcal{V}'$ such that $\psi(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{y}, \boldsymbol{y}')$ holds if and only if $p'(\boldsymbol{x}, \boldsymbol{x}') \xrightarrow{*} q'(\boldsymbol{y}, \boldsymbol{y}')$ in $\mathcal{V}'$. By (5), the formula $\varphi_{\mathcal{V},p,q}(\boldsymbol{x}, \boldsymbol{y}) \stackrel{\text{def}}{=} \psi(\boldsymbol{x}, \mathbf{0}, \mathbf{0}, \boldsymbol{y})$ satisfies the theorem. ◀

It was observed in [20, 9] that the reachability relation of a $\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$, such that $|Q| = |M(\mathcal{V})| = 1$, is semilinear if and only if $\mathcal{M}_\mathcal{V}$ is finite. Theorem 6 shows that if we do not bound the number of states and matrices, *i.e.* drop the assumption $|Q| = |M(\mathcal{V})| = 1$, then the left implication remains true. It is natural to ask whether the right implication also remains true.

Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be the affine $\mathbb{Z}$-VASS illustrated in Figure 3 from left to right respectively. Note that $\mathcal{M}_{\mathcal{V}_1}$ and $\mathcal{M}_{\mathcal{V}_2}$ are both infinite due to the matrix made only of 1s. Moreover, the reachability relations of $\mathcal{V}_1$ and $\mathcal{V}_2$ are semilinear since the former can reach any target

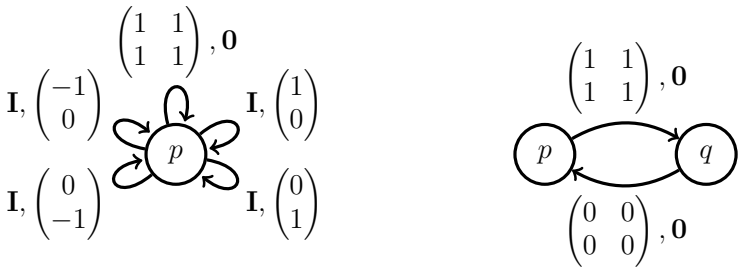

Figure 3 Examples of affine $\mathbb{Z}$-VASS with infinite monoids and semilinear reachability relations.

configuration from any initial configuration, and since the latter can only generate finitely many vectors due to the zero matrix. Since $\mathcal{V}_1$ has a single control-state, $|M(\mathcal{V}_1)| = |M(\mathcal{V}_2)| = 2$ and $\Delta(\mathcal{V}_2) = \{\mathbf{0}\}$, any simple natural extension of the characterization of semilinearity in terms of the number of control-states, matrices and vectors fails.

It is worth noting that an affine $\mathbb{Z}$-VASS with an infinite monoid may have a non semilinear reachability relation. Indeed, Figure 2 depicts a transfer + copy $\mathbb{Z}$-VASS with an infinite monoid and such that $\{\boldsymbol{v} : p(1,1) \xrightarrow{*} q(\boldsymbol{v})\} = \{(2^n, 2^n) : n \in \mathbb{N}\}$, which is known to be non semilinear.

5 Complexity of reachability

In this section, we use the results of Section 3 to show that reachability belongs to PSPACE for a large class of afmp-$\mathbb{Z}$-VASS encompassing all variants of Section 2. Moreover, we give a novel proof to the known NP membership of reachability for reset $\mathbb{Z}$-VASS.

▶ **Theorem 7.** *Let $\mathcal{C} = \bigcup_{d\geq 1} \mathcal{C}_d$ be a class of matrices such that $\mathcal{C}_d$ is finite for every $d \geq 1$. If there exists a polynomial* poly *such $\|\mathcal{C}_d\| \leq 2^{\mathrm{poly}(d)}$ for every $d \geq 1$, then $\mathsf{Reach}_{\mathcal{C}} \in$ PSPACE.*

▶ **Corollary 8.** *The reachability problem of reset, permutation, transfer, copy and copyless $\mathbb{Z}$-VASS is in PSPACE.*

Proof of Theorem 7. Let $\mathcal{V} = (d, Q, T)$ be an affine $\mathbb{Z}$-VASS from class $\mathcal{C}$. Let $\mathcal{V}' = (d, Q', T')$ be the $\mathbb{Z}$-VASS obtained from $\mathcal{V}$ in Theorem 1. Recall that, by Theorem 1, $p(\boldsymbol{u}) \xrightarrow{*} q(\boldsymbol{v})$ in $\mathcal{V}$ if and only if $p'(\boldsymbol{u}, \mathbf{0}) \xrightarrow{*} q'(\mathbf{0}, \boldsymbol{v})$ in $\mathcal{V}'$. Therefore, it suffices to check the latter for determining reachability in $\mathcal{V}$.

We invoke a result of [7] on the flattability of $\mathbb{Z}$-VASS. By [7, Prop. 3], $p'(\boldsymbol{x}) \xrightarrow{*} q'(\boldsymbol{y})$ in $\mathcal{V}'$ if and only if there exist $k \leq |T'|$, $\alpha_0, \beta_1, \alpha_1, \ldots, \beta_k, \alpha_k \in (T')^*$ and $\boldsymbol{e} \in \mathbb{N}^k$ such that

- **(i)** $p'(\boldsymbol{x}) \xrightarrow{\alpha_0 \beta_1^{\boldsymbol{e}(1)} \alpha_1 \cdots \beta_k^{\boldsymbol{e}(k)}} q'(\boldsymbol{y})$ in $\mathcal{V}'$,
- **(ii)** β_i is a cycle for every $i \in [k]$, and
- **(iii)** $\alpha_0\beta_1\alpha_1 \cdots \beta_k\alpha_k$ is a path from p' to q' of length at most $2 \cdot |Q'| \cdot |T'|$.

For every $w \in (T')^*$, let $\Delta(w) \stackrel{\text{def}}{=} \sum_{i=1}^{|w|} \Delta(w_i)$. By Lemma 4 (see the remark below the proof of Lemma 4), we have $w(\boldsymbol{u}) = \boldsymbol{u} + \Delta(w)$ for every $\boldsymbol{u} \in \mathbb{Z}^d$. Thus, by Lemma 3, checking (i), assuming (iii), amounts to testing whether $\boldsymbol{e}$ is a solution of the following system of linear Diophantine equations:

$$\boldsymbol{x} + \sum_{i=0}^{k} \Delta(\alpha_i) + \begin{pmatrix}\Delta(\beta_1) & \Delta(\beta_2) & \cdots & \Delta(\beta_k)\end{pmatrix} \cdot \boldsymbol{e} = \boldsymbol{y}. \tag{6}$$

Let $m \stackrel{\text{def}}{=} 2 \cdot |Q'| \cdot |T'|$. Since $|T'| \leq \|T'\|$ and by Theorem 1, we have $m \leq 128 \cdot d \cdot |\mathcal{M}_\mathcal{V}|^5 \cdot |Q|^2 \cdot \|T\|$, and hence by $M(\mathcal{V}) \subseteq \mathcal{C}_d$ and by assumption on $\mathcal{C}_d$, $m \leq 128 \cdot d \cdot (2^{\text{poly}(d)})^5 \cdot |Q|^2 \cdot \|T\|$.

We describe a polynomial-space non deterministic Turing machine $\mathcal{A}$ for testing whether $p'(\boldsymbol{x}) \stackrel{*}{\to} q'(\boldsymbol{y})$ in $\mathcal{V}'$. The proof follows from NPSPACE = PSPACE. Machine $\mathcal{A}$ guesses $k \leq |T'|$, a path $\pi = \alpha_0\beta_1\alpha_1 \cdots \beta_k\alpha_k$ of length at most m from p' to q', and $\boldsymbol{e} \in \mathbb{N}^k$, and tests whether (6) holds for π. Note that we are not given $\mathcal{V}'$, but $\mathcal{V}$, so we must be careful for the machine to work in polynomial space.

Instead of fully constructing $\mathcal{V}'$ and fully guessing π, we do both on the fly, and also construct $\Delta(\alpha_0), \Delta(\beta_1), \ldots, \Delta(\beta_k), \Delta(\alpha_k)$ on the fly as partial sums as we guess π. Note that to ensure that each β_i is a cycle, we do not need to fully store β_i but only its starting control-state. Moreover, note that $\|\Delta(\alpha_i)\|, \|\Delta(\beta_i)\| \leq m \cdot \max\{\|\Delta(t)\| : t \in T\}$ for every i, and hence each α_i and β_i has a binary representation of polynomial size in $|\mathcal{V}|$.

By [14, Prop. 4], (6) has a solution if and only if it has a solution $\boldsymbol{e} \in \mathbb{N}^k$ such that

$$\|\boldsymbol{e}\| \leq \left((k+1) \cdot \max\{\|\Delta(\beta_i)\| : i \in [k]\} + \|\boldsymbol{x}\| + \|\boldsymbol{y}\| + \sum_{i=0}^{k} \|\Delta(\alpha_i)\| + 1 \right)^{d'}.$$

Since $d' = 2 \cdot d$, this means that we can guess a vector $\boldsymbol{e} \in \mathbb{N}^k$ whose binary representation is of polynomial size, and that we can thus evaluate (6) in polynomial time. ◀

▶ **Theorem 9** ([21]). *The reachability problem for reset $\mathbb{Z}$-VASS belongs to NP.*

Proof. Let $\mathcal{V} = (d, Q, T)$ be a reset $\mathbb{Z}$-VASS. The proof does not follow immediately from Theorem 1 because $\mathcal{M}_\mathcal{V}$ can be of size up to 2^d. We will analyze the construction used in the proof of Theorem 1, where reachability in $\mathcal{V}$ is effectively reduced to reachability in a $\mathbb{Z}$-VASS $\mathcal{V}' = (d', Q', T')$. Recall that $Q' = Q_i \cup (Q \times \mathcal{M}_\mathcal{V}) \cup (Q \times \mathcal{M}_\mathcal{V} \times \mathcal{M}_\mathcal{V}) \cup Q_f$, and thus that the size of $\mathcal{V}'$ depends only on the sizes of Q and $\mathcal{M}_\mathcal{V}$.

It follows from the proof of Theorem 1 and Proposition 5 that for every run $p_i(\boldsymbol{u}, \mathbf{0}) \stackrel{*}{\to} q_f(\mathbf{0}, \boldsymbol{v})$ in $\mathcal{V}'$, there is a corresponding run $p(\boldsymbol{u}) \stackrel{w}{\to} q(\boldsymbol{v})$ in $\mathcal{V}$ for some $w \in T^*$ of length $k \geq 0$. Moreover, the only states of the form $(Q, \mathbf{A}, \mathbf{B})$ or $(Q, \mathbf{A})$ occurring along the run contain matrices $\mathbf{A}, \mathbf{B} \in \mathcal{M}_\mathcal{V}$ of the form $\mathbf{A}_i = M(w_i w_{i+1} \cdots w_k)$ for $i \in [k+1]$. Recall that by definition, for every $i \in [k]$, $\mathbf{A}_i = \mathbf{A}_{i+1} \cdot \mathbf{B}$ for some $\mathbf{B} \in \mathcal{M}_\mathcal{V}$. Since $\mathcal{M}_\mathcal{V}$ consists of reset matrices, it holds that $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ is monotonic, *i.e.* if $\mathbf{A}_i$ has a 1 somewhere on its diagonal, then $\mathbf{A}_{i+1}$ also contains 1 in that position. It follows that $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{k+1}$ is made of at most $d+1$ matrices.

To prove the NP upper bound we proceed as follows. We guess at most $d+1$ matrices of $\mathcal{M}_\mathcal{V}$ that could appear in sequence $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{k+1}$. We construct the $\mathbb{Z}$-VASS $\mathcal{V}'$ as in Theorem 1, but we discard each control-state of Q' containing a matrix not drawn from the guessed matrices. Since the constructed $\mathbb{Z}$-VASS is of polynomial size, reachability can be verified in NP [21]. ◀

▶ Remark. Observe that the proof of Theorem 9 holds for any class of affine $\mathbb{Z}$-VASS with a finite monoid such that every path of its Cayley graph contains at most polynomially many different vertices. For a reset $\mathbb{Z}$-VASS of dimension d, the number of vertices on every path of the Cayley graph is bounded by $d+1$.

6 Hardness results for reachability

It is known that the reachability problem for $\mathbb{Z}$-VASS is already NP-hard [21], which means that reachability is NP-hard for all classes of affine $\mathbb{Z}$-VASS. In this section, we show that PSPACE-hardness holds for some classes, matching the PSPACE upper bound derived in Section 5. Moreover, we observe that reachability is undecidable for transfer + copy $\mathbb{Z}$-VASS.

▶ **Theorem 10.** *The reachability problem for permutation + reset $\mathbb{Z}$-VASS is PSPACE-hard.*

Proof. We give a reduction from the membership problem of linear bounded automata, which is known to be PSPACE-complete (see, *e.g.*, [23, Sect. 9.3 and 13]). Let $\mathcal{A} = (P, \Sigma, \Gamma, \delta, q^{\text{ini}}, q^{\text{acc}}, q^{\text{rej}})$ be a linear bounded automaton, where:

- P is the set of states,
- $\Sigma \subseteq \Gamma$ is the input alphabet,
- Γ is the tape alphabet,
- δ is the transition function, and
- $q^{\text{ini}}, q^{\text{acc}}, q^{\text{rej}}$ are the initial, accepting and rejecting states respectively.

The transition function is a mapping $\delta : P \times \Gamma \to P \times \Gamma \times \{\textsc{Left}, \textsc{Right}\}$. The intended meaning of a transition $\delta(p, a) = (q, b, D)$ is that whenever $\mathcal{A}$ is in state p and holds letter a at the current position of its tape, then $\mathcal{A}$ overwrites a with b and moves to state q and to the next tape position in direction D.

Let us fix the word that we will check for membership $w \in \Sigma^n$ (so $|w| = n$). We construct an affine $\mathbb{Z}$-VASS $\mathcal{V} = (d, Q, T)$ and configurations $r(\boldsymbol{u})$ and $r'(\boldsymbol{v})$ such that $\mathcal{A}$ accepts w if and only if $r(\boldsymbol{u}) \xrightarrow{*} r'(\boldsymbol{v})$.

We set $d \stackrel{\text{def}}{=} n \cdot |\Gamma| + 1$ and associate a counter to each position of w and each letter of the tape alphabet Γ, plus one additional counter. For readability, we denote these counters respectively as $x_{i,a}$ and y, where $i \in [n]$ and $a \in \Gamma$. The idea is to maintain, for every $i \in [n]$, a single "token" among counters $\{x_{i,a} : a \in \Gamma\}$ in order to represent the current letter in the i^{th} tape cell of $\mathcal{A}$. The initial vector is $\boldsymbol{u} \in \{0,1\}^d$ such that $\boldsymbol{u}(y) = 0$ and $\boldsymbol{u}(x_{i,a}) = 1$ if and only if $w_i = a$ for every $i \in [n]$ and $a \in \Gamma$.

The control-states of $\mathcal{V}$ are defined as:

$$Q \stackrel{\text{def}}{=} \{r_{p,i} : p \in P, i \in [n]\} \cup \{r_{p,i,a} : p \in P, i \in [n], a \in \Gamma\} \cup \{r_{\text{acc}}\}.$$

The purpose of states of the form $r_{p,i}$ is to store the current state p and tape cell i of $\mathcal{A}$. States of the form $r_{p,i,a}$ are intermediary control-states and the state r_{acc} will be the target control-state.

We associate transitions to every triple $(p, a, i) \in P \times \Gamma \times [n]$, which denotes a configuration of $\mathcal{A}$: the automaton is in state p in position i, where letter a is stored. Let us fix a transition $\delta(p, a) = (q, b, D)$; and let $j = i + 1$ if $D = \textsc{Right}$, and $j = i - 1$ if $D = \textsc{Left}$. For every $i \in [n]$, if $j \in [n]$ then we add to T the transitions

$$(r_{p,i}, \mathbf{A}, \mathbf{0}, r_{p,i,a}) \text{ and } (r_{p,i,a}, \mathbf{B}, \mathbf{0}, r_{q,j}), \tag{7}$$

where $\mathbf{A}$ is a permutation matrix that swaps the values of $x_{i,a}$ and $x_{i,b}$; and $\mathbf{B}$ resets $x_{i,\sigma}$ for every $\sigma \in \Gamma \setminus \{b\}$. The two transitions are depicted on the left of Figure 4 (for $D = \mathsf{Right}$). The purpose of the first transition is to simulate the transition of $\mathcal{A}$, upon reading a in tape cell i and state p, by moving the i^{th} "token" from $x_{i,a}$ to $x_{i,b}$. Note that this transition may be faulty, *i.e.* it can simulate reading letter a even though tape cell i contains another letter. The purpose of the second transition is to detect such faulty behaviour: if the first

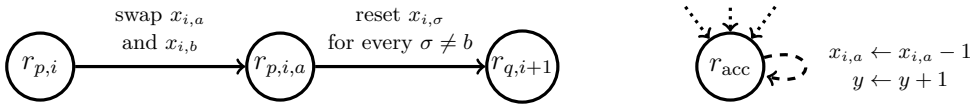

Figure 4 *Left*: transitions of $\mathcal{V}$ simulating transition $\delta(p, a) = (q, b, \mathsf{Right})$ of $\mathcal{A}$. *Right*: transitions to verify whether the accepting state has been reached with no error during the simulation.

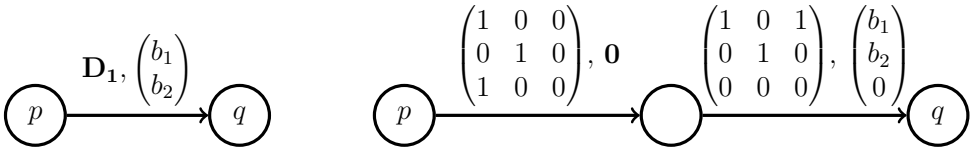

Figure 5 Gadget (on the right) made of copy and transfer transitions simulating the doubling transition on the left.

transition is taken and tape cell i does not contain a, then due to the resets, all counters of $\{x_{i,a} : a \in \Gamma\}$ end up in 0, and the i^{th} "token" is lost.

Recall that in the initial vector $\boldsymbol{u} \in \{0,1\}^d$ there were exactly n counters with 1 and $\sum_{i\in[d]} \boldsymbol{u}(i) = n$. By construction of $\mathcal{V}$, all configurations reachable from $r_{q^{\text{ini}},1}(\boldsymbol{u})$, using transitions defined in (7), have vectors in $\{0,1\}^d$ with at most n counters equal to 1. They have exactly n counters equal to 1 only if all corresponding transitions were valid for the automaton $\mathcal{A}$. We conclude that $\mathcal{A}$ accepts w if and only if there exist $i \in [n]$ and $\boldsymbol{u}' \in \{0,1\}^d$ such that $r_{q^{\text{ini}},1}(\boldsymbol{u}) \xrightarrow{*} r_{q^{\text{acc}},i}(\boldsymbol{u}')$ and $\sum_{i\in[d]} \boldsymbol{u}'(i) = n$.

To test whether such index i and vector $\boldsymbol{v}$ exist, we add some transitions to T as illustrated on the right of Figure 4. For every $i \in [n]$, we add to T the transition $(r_{q^{\text{acc}},i}, \mathbf{I}, \mathbf{0}, r_{\text{acc}})$. For every $i \in [n]$ and $a \in \Gamma$, we add to T the transition $(r_{\text{acc}}, \mathbf{I}, \boldsymbol{b}, r_{\text{acc}})$ where $\boldsymbol{b}$ is the vector whose only non zero components are $\boldsymbol{b}(x_{i,a}) = -1$ and $\boldsymbol{b}(y) = 1$. The purpose of these transitions is to (weakly) transfer the values of all counters to y. Recall that $\boldsymbol{v}$ is the vector whose only non zero component is $\boldsymbol{v}(y) = n$. We conclude that the language of $\mathcal{A}$ accepts w if and only if $r_{q^{\text{ini}},1}(\boldsymbol{u}) \xrightarrow{*} r_{\text{acc}}(\boldsymbol{v})$. ◀

▶ **Corollary 11.** *The reachability problem is PSPACE-complete for permutation + reset $\mathbb{Z}$-VASS, transfer $\mathbb{Z}$-VASS and copy $\mathbb{Z}$-VASS.*

Proof. The hardness for permutation + reset $\mathbb{Z}$-VASS follows from Theorem 10, and the upper bound for transfer $\mathbb{Z}$-VASS and copy $\mathbb{Z}$-VASS follows from Theorem 7. It remains to argue that transfers and copies can both simulate permutations and resets. By definition, permutation matrices are also transfer and copy matrices. Resetting a counter x can be simulated by adding an extra counter y. In the case of transfers, it suffices to transfer x to y and to allow for y to be arbitrarily incremented or decremented. In the case of copies, it suffices to keep $y = 0$ at all times and to copy y onto x. ◀

▶ **Proposition 12** ([36]). *The reachability problem for transfer + copy $\mathbb{Z}$-VASS is undecidable, even when restricted to three counters.*

Proof. Reichert [36] gives a reduction from the Post correspondence problem over the alphabet $\{0,1\}$ to reachability in affine $\mathbb{Z}$-VASS with two counters. The trick of the reduction is to represent two binary sequences as the natural numbers the sequences encode, one in each counter. If we add an artificial 1 at the beginning of the two binary sequences, then these sequences are uniquely determined by their numerical values. We only need to be

able to double the counter values, which corresponds to shifting the sequences. This can be achieved using the following matrices:

$$\mathbf{D_1} \stackrel{\text{def}}{=} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \text{ and } \mathbf{D_2} \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}.$$

The only matrices used in the construction of Reichert are $\mathbf{I}$, $\mathbf{D_1}$ and $\mathbf{D_2}$. The two last matrices can be simulated by a gadget made of copy and transfer matrices and by introducing a third counter. This gadget is depicted in Figure 5 for the case of matrix $\mathbf{D_1}$. The other gadget is symmetric. Note that if a run enters control-state p of the gadget with vector $(x, y, 0)$, then it leaves control-state q in vector $(2x + b_1, y + b_2, 0)$ as required. ◀

▶ Remark. A monoid $\mathcal{M}$ is *positive* if it contains only matrices with non negative entries. The classes of Section 2 and the matrices used in Proposition 12 have this property. The coverability problem for affine VASS with positive (and possibly infinite) monoids is known to be decidable in Ackermann time [19]. Recall that coverability and reachability are inter-reducible for affine $\mathbb{Z}$-VASS. Thus, Proposition 12 gives an example of a decision problem, namely coverability, which is more difficult for affine $\mathbb{Z}$-VASS than for affine VASS.

7 Conclusion

We have shown that the reachability problem for afmp-$\mathbb{Z}$-VASS reduces to the reachability problem for $\mathbb{Z}$-VASS, *i.e.* every afmp-$\mathbb{Z}$-VASS $\mathcal{V}$ can be simulated by a $\mathbb{Z}$-VASS of size polynomial in $|\mathcal{V}|$ and $\|\mathcal{M}_\mathcal{V}\|$. In particular, this allowed us to establish that the reachability relation of any afmp-$\mathbb{Z}$-VASS is semilinear.

For all of the variants we studied – reset, permutation, transfer, copy and copyless $\mathbb{Z}$-VASS – the size of $\|\mathcal{M}_\mathcal{V}\|$ is of exponential size, thus yielding a PSPACE upper bound on their reachability problems. We do not know whether an exponential bound on $\|\mathcal{M}_\mathcal{V}\|$ holds for any class of afmp-$\mathbb{Z}$-VASS. We are aware that the work of [31] provides an exponential tower upper bound. Moreover, an exponential upper bound holds when $\mathcal{M}_\mathcal{V}$ is generated by a single matrix [24]; and when $\mathcal{M}_\mathcal{V}$ is a group then we have an exponential bound but only on $|\mathcal{M}_\mathcal{V}|$ (see [28] for an exposition on the group case).

For all the classes of afmp-$\mathbb{Z}$-VASS studied in this paper, we have shown that the reachability problem is either PSPACE-complete or NP-complete, with the exception of permutation $\mathbb{Z}$-VASS reachability which lies between NP and PSPACE, and whose precise complexity remains open.

Another interesting open question is whether reachability is undecidable for every class of infinite matrix monoids, *i.e.* is the top rectangular region of Figure 1 equal to the red ellipse?

References

1 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proc. 11th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 313–321, 1996. doi:10.1109/LICS.1996.561359.

2 Parosh Aziz Abdulla and Giorgio Delzanno. Parameterized verification. *International Journal on Software Tools for Technology Transfer*, 18(5):469–473, 2016. doi:10.1007/s10009-016-0424-3.

3 Rajeev Alur, Adam Freilich, and Mukund Raghothaman. Regular combinators for string transformations. In *Proc. Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 9:1–9:10, 2014. doi:10.1145/2603088.2603151.

4 Rajeev Alur and Mukund Raghothaman. Decision problems for additive regular functions. In *Proc.* 40^{th} *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 37–48, 2013. doi:10.1007/978-3-642-39212-2_7.

5 Toshiro Araki and Tadao Kasami. Some decision problems related to the reachability problem for Petri nets. *Theoretical Computer Science*, 3(1):85–104, 1976. doi:10.1016/0304-3975(76)90067-0.

6 Konstantinos Athanasiou, Peizun Liu, and Thomas Wahl. Unbounded-thread program verification using thread-state equations. In *Proc.* 8^{th} *International Joint Conference on Automated Reasoning (IJCAR)*, pages 516–531, 2016. doi:10.1007/978-3-319-40229-1_35.

7 Michael Blondin, Alain Finkel, Stefan Göller, Christoph Haase, and Pierre McKenzie. Reachability in two-dimensional vector addition systems with states is PSPACE-complete. In *Proc.* 30^{th} *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 32–43, 2015. doi:10.1109/LICS.2015.14.

8 Michael Blondin and Christoph Haase. Logics for continuous reachability in Petri nets and vector addition systems with states. In *Proc.* 32^{nd} *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2017. doi:10.1109/LICS.2017.8005068.

9 Bernard Boigelot. *Symbolic Methods for Exploring Infinite State Spaces*. PhD thesis, Université de Liège, Belgium, 1998.

10 Rémi Bonnet. *Theory of Well-Structured Transition Systems and Extended Vector-Addition Systems*. PhD thesis, École normale supérieure de Cachan, France, 2013.

11 I. Borosh and L. B. Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55(2):299–304, 1976. doi:10.2307/2041711.

12 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Bounded Parikh automata. *International Journal of Foundations of Computer Science*, 23(8):1691–1710, 2012. doi:10.1142/S0129054112400709.

13 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Unambiguous constrained automata. *International Journal of Foundations of Computer Science*, 24(7):1099–1116, 2013. doi:10.1142/S0129054113400339.

14 Dmitry Chistikov and Christoph Haase. The taming of the semi-linear set. In *Proc.* 43^{rd} *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 128:1–128:13, 2016. doi:10.4230/LIPIcs.ICALP.2016.128.

15 Giorgio Delzanno. A unified view of parameterized verification of abstract models of broadcast communication. *International Journal on Software Tools for Technology Transfer*, 18(5):475–493, 2016. doi:10.1007/s10009-016-0412-7.

16 Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In *Proc.* 25^{th} *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 103–115, 1998. doi:10.1007/BFb0055044.

17 E. Allen Emerson and Kedar S. Namjoshi. On model checking for non-deterministic infinite-state systems. In *Proc.* 13^{th} *Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 70–80, 1998. doi:10.1109/LICS.1998.705644.

18 Javier Esparza, Ruslán Ledesma-Garza, Rupak Majumdar, Philipp J. Meyer, and Filip Niksic. An SMT-based approach to coverability analysis. In *Proc.* 26^{th} *International Conference on Computer Aided Verification (CAV)*, pages 603–619, 2014. doi:10.1007/978-3-319-08867-9_40.

19 Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's lemma. In *Proc.* 26^{th} *Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 269–278, 2011. doi:10.1109/LICS.2011.39.

20 Alain Finkel and Jérôme Leroux. How to compose Presburger-accelerations: Applications to broadcast protocols. In *Proc. 22nd Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 145–156, 2002. doi:10.1007/3-540-36206-1_14.

21 Christoph Haase and Simon Halfon. Integer vector addition systems with states. In *Proc. 8th International Workshop on Reachability Problems (RP)*, pages 112–124, 2014. doi:10.1007/978-3-319-11439-2_9.

22 John E. Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979. doi:10.1016/0304-3975(79)90041-0.

23 John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

24 Radu Iosif and Arnaud Sangnier. How hard is it to verify flat affine counter systems with the finite monoid property? In *Proc. 14th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, pages 89–105, 2016. doi:10.1007/978-3-319-46520-3_6.

25 Alexander Kaiser, Daniel Kroening, and Thomas Wahl. A widening approach to multithreaded program verification. *ACM Transactions on Programming Languages and Systems*, 36(4):14:1–14:29, 2014. doi:10.1145/2629608.

26 Richard M. Karp and Raymond E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969. doi:10.1016/S0022-0000(69)80011-5.

27 S. Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proc. 14th Annual ACM Symposium on Theory of Computing (STOC)*, pages 267–281, 1982. doi:10.1145/800070.802201.

28 James Kuzmanovich and Andrey Pavlichenkov. Finite groups of matrices whose entries are integers. *The American Mathematical Monthly*, 109(2):173–186, 2002. doi:10.2307/2695329.

29 Jérôme Leroux. Vector addition systems reachability problem (a simpler solution). In *The Alan Turing Centenary Conference*, pages 214–228, 2012.

30 Richard J. Lipton. The reachability problem requires exponential space. Technical Report 63, Department of Computer Science, Yale University, 1976.

31 Arnaldo Mandel and Imre Simon. On finite semigroups of matrices. *Theoretical Computer Science*, 5(2):101–111, 1977. doi:10.1016/0304-3975(77)90001-9.

32 Ernst W. Mayr. An algorithm for the general Petri net reachability problem. *SIAM Journal on Computing*, 13(3):441–460, 1984. doi:10.1137/0213029.

33 Marvin Lee Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

34 Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *Comptes Rendus du I^{er} Congrès des mathématiciens des pays slaves*, pages 192–201, 1929.

35 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978. doi:10.1016/0304-3975(78)90036-1.

36 Julien Reichert. *Reachability games with counters: decidability and algorithms*. PhD thesis, École normale supérieure de Cachan, France, 2015.

37 Klaus Reinhardt. Reachability in Petri nets with inhibitor arcs. *Electronic Notes in Theoretical Computer Science*, 223:239–264, 2008. doi:10.1016/j.entcs.2008.12.042.

38 Philippe Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *Proc. 35th International Symposium Mathematical Foundations of Computer Science (MFCS)*, pages 616–628, 2010. doi:10.1007/978-3-642-15155-2_54.