Integer Vector Addition Systems with States

Christoph Haase* and Simon Halfon

Laboratoire Spécification et Vérification (LSV), CNRS
École Normale Supérieure (ENS) de Cachan, France

Abstract. This paper studies reachability, coverability and inclusion problems for Integer Vector Addition Systems with States ($\mathbb{Z}$-VASS) and extensions and restrictions thereof. A $\mathbb{Z}$-VASS comprises a finite-state controller with a finite number of counters ranging over the integers. Although it is folklore that reachability in $\mathbb{Z}$-VASS is NP-complete, it turns out that despite their naturalness, from a complexity point of view this class has received little attention in the literature. We fill this gap by providing an in-depth analysis of the computational complexity of the aforementioned decision problems. Most interestingly, it turns out that while the addition of reset operations to ordinary VASS leads to undecidability and Ackermann-hardness of reachability and coverability, respectively, they can be added to $\mathbb{Z}$-VASS while retaining NP-completeness of both coverability and reachability.

1 Introduction

Vector Addition Systems with States (VASS) are a prominent class of infinite-state systems. They comprise a finite-state controller with a finite number of counters ranging over the natural numbers. When taking a transition, an integer can be added to a counter, provided that the resulting counter value is non-negative. A configuration of a VASS is a tuple $q(\boldsymbol{v})$ consisting of a control state q and a vector $\boldsymbol{v} \in \mathbb{N}^d$, where $d > 0$ is the number of counters or, equivalently, the dimension of the VASS. The central decision problems for VASS are reachability, coverability and inclusion. Given configurations $q(\boldsymbol{v})$, $q'(\boldsymbol{v}')$ of a VASS $\mathcal{A}$, reachability is to decide whether there is a path connecting the two configurations in the transition system induced by $\mathcal{A}$. Coverability on the other hand asks whether there is a path from $q(\boldsymbol{v})$ to a configuration that is "above" $q'(\boldsymbol{v}')$, *i.e.*, a path to some $q'(\boldsymbol{w})$ such that $\boldsymbol{w} \geq \boldsymbol{v}'$, where $\geq$ is interpreted component-wise. Finally, given VASS $\mathcal{A}$ and $\mathcal{B}$, inclusion asks whether the set of counter values reachable in the transition system induced by $\mathcal{A}$ is contained in those reachable by $\mathcal{B}$. All of the aforementioned problems have extensively been studied over the course of the last forty years. One of the earliest results was obtained by Lipton, who showed that reachability and coverability are EXPSPACE-hard [20]. Later, Rackoff established a matching upper bound for coverability [23], and Mayr showed that reachability is decidable [21]. For inclusion, it is known that

* Supported by the French ANR, ReacHard (grant ANR-11-BS02-001).

this problem is in general undecidable [14] and Ackermann ($\mathbf{F}_\omega$)-complete [18] when restricting to VASS with a finite reachability set. Moreover, various extensions of VASS with, for instance, resets or polynomial updates on counter values have been studied in the literature. Resets allow for setting a counter to zero along a transition, and polynomial updates allow for updating a counter with an arbitrary polynomial. In general, reachability in the presence of any such extension becomes undecidable [4, 6], while the complexity of coverability increases significantly to $\mathbf{F}_\omega$-completeness in the presence of resets [25].

What makes VASS hard to deal with, both in the computational and in the mathematical sense, is the restriction of the counters to non-negative integers. This restriction allows for enforcing an order in which transitions can be taken, which is at the heart of many hardness proofs. In this paper, we relax this restriction and study $\mathbb{Z}$-VASS which are VASS whose counters can take values from the integers, and extensions thereof. Thus, the effect of transitions can commute along a run of a $\mathbb{Z}$-VASS, which makes deciding reachability substantially easier, and it is in fact folklore that reachability in $\mathbb{Z}$-VASS is NP-complete. It appears, however, that many aspects of the computational complexity of standard decision problems for $\mathbb{Z}$-VASS and extensions and restrictions thereof have not received much attention in the literature.

Our contribution. The main focus of this paper[1] is to study the computational complexity of reachability, coverability and inclusion for $\mathbb{Z}$-VASS equipped with resets ($\mathbb{Z}$-VASS$_\mathrm{R}$). Unlike in the case of VASS, we can show that reachability and coverability are naturally logarithmic-space inter-reducible. By generalizing a technique introduced by Seidl *et al.* [26] for defining Parikh images of finite-state automata in existential Presburger arithmetic, we can show that a given instance of reachability (and *a fortiori* coverability) in $\mathbb{Z}$-VASS$_\mathrm{R}$ can be reduced in logarithmic-space to an equivalent sentence in existential Presburger arithmetic, and henceforth both problems are NP-complete. Moreover, by exploiting a recent result on the complexity of Presburger arithmetic with a fixed number of quantifier alternations [12], this reduction immediately yields coNEXP-membership of the inclusion problem for $\mathbb{Z}$-VASS$_\mathrm{R}$. We also show that a matching lower bound can be established via a reduction from validity in Π_2-Presburger arithmetic. This lower bound does not require resets and thus already holds for $\mathbb{Z}$-VASS. Along the way, wherever possible we sharpen known lower bounds and propose some further open problems.

Related Work. The results obtained in this paper are closely related to decision problems for commutative grammars, *i.e.* Parikh images of, for instance, finite-state automata or context-free grammars. A generic tool that is quite powerful in this setting is to define Parikh images as the set of solutions to certain systems of linear Diophantine equations. This approach has, for instance, been taken

[1] A full version containing all proofs omitted due to space constraints can be obtained from `http://arxiv.org/abs/1406.2590`.

in [5, 22, 26, 13, 15]. As stated above, we generalize the technique of Seidl *et al.*, which has also been the starting point in [15] in order to show decidability and complexity results for pushdown systems equipped with reversal-bounded counters.

Furthermore, results related to ours have also been established by Kopczyński & To. In [19], they consider inclusion problems for regular and context-free commutative grammars, and show that for a fixed alphabet those problems are coNP- and Π_2^{P}-complete, respectively. As a matter of fact, the proof of the Π_2^{P}-upper bound is established for context-free commutative grammars in which, informally speaking, letters can be erased, which can be seen as a generalization of $\mathbb{Z}$-VASS. In general, inclusion for context-free commutative grammars is in coNEXP [16], but it is not known whether this bound is tight. Also related is the work by Reichert [24], who studies the computational complexity of reachability games on various classes of $\mathbb{Z}$-VASS. Finally, $\mathbb{Z}$-VASS are an instance of valence automata, which have recently, for instance, been studied by Buckheister & Zetzsche [3]. However, their work is more concerned with language-theoretic properties of valence automata rather than aspects of computational complexity. Language-theoretic aspects of $\mathbb{Z}$-VASS have also been studied by Greibach [11].

As discussed above, $\mathbb{Z}$-VASS achieve a lower complexity for standard decision problems in comparison to VASS by relaxing counters to range over the integers. Another approach going into a similar direction is to allow counters to range over the positive reals. It has been shown in recent work by Fraca & Haddad [7] that the decision problems we consider in this paper become substantially easier for such continuous VASS, with reachability even being decidable in P.

2 Preliminaries

In this section, we provide most of the definitions that we rely on in this paper. We first introduce some general notation and subsequently an abstract model of register machines from which we derive $\mathbb{Z}$-VASS as a special subclass. We then recall and tighten some known complexity bounds for $\mathbb{Z}$-VASS and conclude this section with a brief account on Presburger arithmetic.

General Notation. In the following, $\mathbb{Z}$ and $\mathbb{N}$ are the sets of integers and natural numbers, respectively, and $\mathbb{N}^d$ and $\mathbb{Z}^d$ are the set of dimension d vectors in $\mathbb{N}$ and $\mathbb{Z}$, respectively. We denote by $[d]$ the set of positive integers up to d, *i.e.* $[d] = \{1, \ldots, d\}$. By $\mathbb{N}^{d\times d}$ and $\mathbb{Z}^{d\times d}$ we denote the set of $d \times d$ square matrices over $\mathbb{N}$ and $\mathbb{Z}$, respectively. The identity matrix in dimension d is denoted by I_d and $\boldsymbol{e}_i$ denotes the i-th unit vector in any dimension d provided $i \in [d]$. For any d and $i, j \in [d]$, E_{ij} denotes the $d \times d$-matrix whose i-th row and j-th column intersection is equal to one and all of its other components are zero, and we use E_i to abbreviate E_{ii}. For $\boldsymbol{v} \in \mathbb{Z}^d$ we write $\boldsymbol{v}(i)$ for the i-th component of $\boldsymbol{v}$ for $i \in [d]$. Given two vectors $\boldsymbol{v_1}, \boldsymbol{v_2} \in \mathbb{Z}^d$, we write $\boldsymbol{v_1} \geq \boldsymbol{v_2}$ iff for all $i \in [d]$, $\boldsymbol{v_1}(i) \geq \boldsymbol{v_2}(i)$. Given a vector $\boldsymbol{v} \in \mathbb{Z}^d$ and a set $S \subseteq [d]$, by $\boldsymbol{v}_{|S}$ we denote the vector $\boldsymbol{w}$ derived from $\boldsymbol{v}$ with components from S reset, *i.e*, for all

$j \in [d]$, $\boldsymbol{w}(j) = \boldsymbol{v}(j)$ when $j \notin S$, and $\boldsymbol{w}(j) = 0$ otherwise. Given $i \in [d]$, $\boldsymbol{v}_{|i}$ abbreviates $\boldsymbol{v}_{|\{i\}}$. If not stated otherwise, all numbers in this paper are assumed to be encoded in binary.

Presburger Arithmetic. Recall that *Presburger arithmetic (PA)* is the first-order theory of the structure $\langle \mathbb{N}, 0, 1, +, \geq \rangle$, *i.e.*, quantified linear arithmetic over natural numbers. The size $|\Phi|$ of a PA formula is the number of symbols required to write it down, where we assume unary encoding of numbers[2]. For technical convenience, we may assume with no loss of generality that terms of PA formulas are of the form $\boldsymbol{z} \cdot \boldsymbol{x} \geq b$, where $\boldsymbol{x}$ is an n-tuple of first-order variables, $\boldsymbol{z} \in \mathbb{Z}^n$ and $b \in \mathbb{Z}$. It is well-known that the existential (Σ_1-)fragment of PA is NP-complete, see *e.g.* [2]. Moreover, validity for the Π_2-fragment of PA, *i.e.* its restriction to a $\forall^*\exists^*$-quantifier prefix, is coNEXP-complete [10, 12].

Given a PA formula $\Phi(x_1, \ldots, x_d)$ in d free variables, we define

$$[\![\Phi(x_1, \ldots, x_d)]\!] = \{(n_1, \ldots, n_d) \in \mathbb{N}^d : \Phi(n_1/x_1, \ldots, n_d/x_d) \text{ is valid}\}.$$

Moreover, a set $M \subseteq \mathbb{N}^d$ is *PA-definable* if there exists a PA formula $\Phi(x_1, \ldots, x_d)$ such that $M = [\![\Phi(x_1, \ldots, x_d)]\!]$. Recall that a result due to Ginsburg & Spanier states that PA-definable sets coincide with the so-called *semi-linear sets* [9].

Integer Vector Addition Systems. The main objects studied in this paper can be derived from a general class of integer register machines which we define below.

Definition 1. *Let $\mathfrak{A} \subseteq \mathbb{Z}^{d \times d}$, a* dimension d-integer register machine over $\mathfrak{A}$ ($\mathbb{Z}$-RM($\mathfrak{A}$)) *is a tuple $\mathcal{A} = (Q, \Sigma, d, \Delta, \tau)$ where*

- *Q is a finite set of* control states*,*
- *Σ is a finite* alphabet*,*
- *$d > 0$ is the* dimension *or the number of* counters*,*
- *$\Delta \subseteq Q \times \Sigma \times Q$ is a finite set of* transitions*,*
- *$\tau : \Sigma \to (\mathbb{Z}^d \to \mathbb{Z}^d)$ maps each $a \in \Sigma$ to an* affine transformation *such that $\tau(a) = \boldsymbol{v} \mapsto A\boldsymbol{v} + \boldsymbol{b}$ for some $A \in \mathfrak{A}$ and $\boldsymbol{b} \in \mathbb{Z}^d$.*

We will often consider τ as a morphism from Σ^* to the set of affine transformations such that $\tau(\epsilon) = I_d$ and for any $w \in \Sigma^*$ and $a \in \Sigma$, $\tau(wa)(\boldsymbol{v}) = \tau(a)(\tau(w)(\boldsymbol{v}))$. The set $C(\mathcal{A}) = Q \times \mathbb{Z}^d$ is called the *set of configurations of* $\mathcal{A}$. For readability, we write configurations as $q(\boldsymbol{v})$ instead of $(q, \boldsymbol{v})$. Given configurations $q(\boldsymbol{v}), q'(\boldsymbol{v}') \in C$, we write $q(\boldsymbol{v}) \xrightarrow{a}_{\mathcal{A}} q(\boldsymbol{v}')$ if there is a transition $(q, a, q') \in \Delta$ such that $\boldsymbol{v}' = \tau(a)(\boldsymbol{v})$, and $q(\boldsymbol{v}) \to_{\mathcal{A}} q'(\boldsymbol{v}')$ if $q(\boldsymbol{v}) \xrightarrow{a}_{\mathcal{A}} q(\boldsymbol{v}')$ for some $a \in \Sigma$. A *run on a word* $\gamma = a_1 \cdots a_n \in \Sigma^*$ is a finite sequence of configurations $\varrho : c_0 c_1 \cdots c_n$ such that $c_i \xrightarrow{a_{i+1}}_{\mathcal{A}} c_{i+1}$ for all $0 \leq i < n$, and we write

[2] This is with no loss of generality since binary encoding can be simulated at the cost of a logarithmic blowup of the formula size. Note that in particular all complexity lower bounds given in this paper still hold assuming unary encoding of numbers.

$c_0 \xrightarrow{\gamma}_{\mathcal{A}} c_n$ in this case. Moreover, we write $c \to^*_{\mathcal{A}} c'$ if there is a run ϱ on some word γ such that $c = c_0$ and $c' = c_n$. Given $q(\boldsymbol{v}) \in C(\mathcal{A})$, the *reachability set starting from* $q(\boldsymbol{v})$ is defined as

$$\mathit{reach}(\mathcal{A}, q(\boldsymbol{v})) = \{\boldsymbol{v}' \in \mathbb{Z}^d : q(\boldsymbol{v}) \to^*_{\mathcal{A}} q'(\boldsymbol{v}') \text{ for some } q' \in Q\}.$$

In this paper, we study the complexity of deciding reachability, coverability and inclusion.

$\mathbb{Z}$-RM($\mathfrak{A}$) Reachability/Coverability/Inclusion

INPUT: $\mathbb{Z}$-RM($\mathfrak{A}$) $\mathcal{A}, \mathcal{B}$, configurations $q(\boldsymbol{v}), q'(\boldsymbol{v}') \in C(\mathcal{A}), p(\boldsymbol{w}) \in C(\mathcal{B})$.
QUESTION: *Reachability:* Is there a run $q(\boldsymbol{v}) \to^*_{\mathcal{A}} q'(\boldsymbol{v}')$?
Coverability: Is there a $\boldsymbol{z} \in \mathbb{Z}^d$ s.t. $q(\boldsymbol{v}) \to^*_{\mathcal{A}} q'(\boldsymbol{z})$ and $\boldsymbol{z} \geq \boldsymbol{v}'$?
Inclusion: Does $\mathit{reach}(\mathcal{A}, q(\boldsymbol{v})) \subseteq \mathit{reach}(\mathcal{B}, p(\boldsymbol{w}))$ hold?

If we allow an arbitrary number of control states, whenever it is convenient we may assume $\boldsymbol{v}, \boldsymbol{v}'$ and $\boldsymbol{w}$ in the definition above to be equal to $\mathbf{0}$. Of course, $\mathbb{Z}$-RM are very general, and all of the aforementioned decision problems are already known to be undecidable, we will further elaborate on this topic below. We therefore consider subclasses of $\mathbb{Z}$-RM($\mathfrak{A}$) in this paper which restrict the transformation mappings or the number of control states: $\mathcal{A}$ is called

- *integer vector addition system with states and resets ($\mathbb{Z}$-VASS$_R$)* if $\mathfrak{A} = \{\lambda_1 E_1 + \cdots + \lambda_d E_d : \lambda_i \in \{0, 1\}, i \in [d]\}$;
- *integer vector addition system with states ($\mathbb{Z}$-VASS)* if $\mathfrak{A} = I_d$;
- *integer vector addition system ($\mathbb{Z}$-VAS)* if $\mathcal{A}$ is a $\mathbb{Z}$-VASS and $|Q| = 1$.

Classical vector addition systems with states (VASS) can be recovered from the definition of $\mathbb{Z}$-VASS by defining the set of configurations as $Q \times \mathbb{N}^d$ and adjusting the definition of $\to_{\mathcal{A}}$ appropriately. It is folklore that coverability in VASS is logarithmic-space reducible to reachability in VASS. Our first observation is that unlike in the case of VASS, reachability can be reduced to coverability in $\mathbb{Z}$-VASS, this even holds for $\mathbb{Z}$-VASS$_{\mathrm{R}}$. Thanks to this observation, all lower and upper bounds for reachability carry over to coverability, and *vice versa*.

Lemma 2. *Reachability and coverability are logarithmic-space inter-reducible in each of the classes $\mathbb{Z}$-VASS$_R$, $\mathbb{Z}$-VASS and $\mathbb{Z}$-VAS. The reduction doubles the dimension.*

Proof. The standard folklore construction to reduce coverability in VASS to reachability in VASS also works for all classes of $\mathbb{Z}$-VASS$_{\mathrm{R}}$. For brevity, we therefore only give the reduction in the converse direction.

Let $\mathcal{A}$ be from any class of $\mathbb{Z}$-VASS in dimension d and let $q(\boldsymbol{v}), q'(\boldsymbol{v}') \in C(\mathcal{A})$. We construct a $\mathbb{Z}$-VASS $\mathcal{B}$ in dimension $2d$ with the property $q(\boldsymbol{v}) \to^*_{\mathcal{A}} q'(\boldsymbol{v}')$ iff $q(\boldsymbol{v}, -\boldsymbol{v}) \to^*_{\mathcal{B}} q'(\boldsymbol{v}', -\boldsymbol{v}')$ as follows: any affine transformation $\boldsymbol{v} \mapsto A\boldsymbol{v} + \boldsymbol{b}$ is replaced by $\boldsymbol{v} \mapsto A'\boldsymbol{v} + \boldsymbol{b}'$, where

$$A' = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & A \end{bmatrix} \qquad \boldsymbol{b}' = \begin{vmatrix} \boldsymbol{b} \\ -\boldsymbol{b} \end{vmatrix}.$$

Any run $\varrho : q_0(\boldsymbol{v}_0) \cdots q_n(\boldsymbol{v}_n)$ in $\mathcal{B}$ such that $q_0(\boldsymbol{v}_0) = q(\boldsymbol{v}, -\boldsymbol{v})$ and $q_n(\boldsymbol{v}_n) = q'(\boldsymbol{v}', -\boldsymbol{v}')$ corresponds in the first d components to a run in $\mathcal{A}$. Moreover, ϱ has the property that for any $0 \leq i \leq n$ and $q_i(\boldsymbol{v}_i)$, $\boldsymbol{v}_i(j) = -\boldsymbol{v}_i(j+d)$ for all $j \in [d]$. Therefore, $q(\boldsymbol{v}, -\boldsymbol{v}) \to_{\mathcal{B}}^* q'(\boldsymbol{w}, -\boldsymbol{w})$ for some $q'(\boldsymbol{w}, -\boldsymbol{w})$ that covers $q'(\boldsymbol{v}', -\boldsymbol{v}')$ if, and only if, $\boldsymbol{w} \geq \boldsymbol{v}'$ and $-\boldsymbol{w} \geq -\boldsymbol{v}'$, *i.e.*, $\boldsymbol{w} = \boldsymbol{v}'$ and thus in particular whenever $\mathcal{A}$ reaches $q'(\boldsymbol{v}')$ from $q(\boldsymbol{v})$. □

Known Complexity Results for $\mathbb{Z}$-VASS. It is folklore that reachability in $\mathbb{Z}$-VASS is NP-hard. Most commonly, this is shown via a reduction from Subset Sum, so this hardness result in particular relies on binary encoding of numbers and the presence of control states. Here, we wish to remark the following observation.

Lemma 3. *Reachability in $\mathbb{Z}$-VAS is* NP*-hard even when numbers are encoded in unary.*

The proof is given in the appendix of the full version of this paper and follows straight-forwardly via a reduction from feasibility of a system of linear Diophantine equations $A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$, which is known to be NP-complete even when unary encoding of numbers is assumed [8]. Apart from that, it is folklore that reachability in $\mathbb{Z}$-VASS is in NP. To the best of the authors' knowledge, no upper bounds for reachability, coverability or inclusion for $\mathbb{Z}$-VASS$_{\mathrm{R}}$ have been established so far.

Next, we recall that slightly more general transformation matrices lead to undecidability of reachability: when allowing for arbitrary diagonal matrices, *i.e.* affine transformations along transitions, reachability becomes undecidable already in dimension two [6]. Consequently, by a straight forward adaption of Lemma 2 we obtain the following.

Lemma 4. *Let $\mathfrak{D}_d$ be the set of all diagonal matrices in dimension d. Coverability in $\mathbb{Z}$-RM$(\mathfrak{D}_d)$ is undecidable already for $d = 4$.*

Of course, undecidability results for reachability in matrix semi-groups obtained in [1] can be applied in order to obtain undecidability results for more general classes of matrices, and those undecidability results do not even require the presence of control states.

3 Reachability in $\mathbb{Z}$-VASS$_{\mathbf{R}}$ is in NP

One main idea for showing that reachability for $\mathbb{Z}$-VASS$_{\mathrm{R}}$ is in NP is that since there are no constraints on the counter values along a run, a reset on a particular counter allows to forget any information about the value of this counter up to this point, *i.e.*, a reset cuts the run. Hence, in order to determine the value of a particular counter at the end of a run, we only need to sum up the effect of the operations on this counter since the last occurrence of a reset on this counter. This in turn requires us to guess and remember the last occurrence of a reset on each counter.

Subsequently, we introduce monitored alphabets and generalized Parikh images in order to formalize our intuition behind resets. A *monitored alphabet* is an alphabet $\Sigma \uplus R$ with $R = \{r_1, \dots, r_k\}$ being the monitored letters. Given $S \subseteq [k]$, we denote by $\Sigma_S = \Sigma \cup \{r_i : i \in S\}$ the alphabet containing only monitored letters indexed from S. Any word $\gamma \in (\Sigma \cup R)^*$ over a monitored alphabet admits a unique decomposition into *partial words*

$$\gamma = \gamma_0 r_{i_1} \gamma_1 r_{i_2} \cdots r_{i_\ell} \gamma_\ell$$

for some $\ell \leq k$ such that all i_j are pairwise distinct and for all $j \in [\ell]$, $\gamma_j \in \Sigma^*_{\{r_{i_{j+1}}, \dots, r_{i_\ell}\}}$. Such a decomposition simply keeps track of the last occurrence of each monitored letter. For instance for $k = 4$ and $\Sigma = \{a, b\}$, the word $\gamma = aabr_1br_3abr_3ar_1$ can uniquely be decomposed as $(aabr_1br_3ab)r_3(a)r_1$.

In this paper, the *Parikh image* $\pi_\Sigma(w)$ of a word $w \in (\Sigma \uplus R)^*$ restricted to the alphabet $\Sigma = \{a_1, \dots, a_n\}$ is the vector $\pi_\Sigma(w) \in \mathbb{N}^n$ such that $\pi(w)(i) = |w|_{a_i}$ is the number of occurrences of a_i in w. Moreover, $\mathfrak{S}_k$ denotes the permutation group on k symbols.

Definition 5. *Let $\Sigma \uplus R$ be a monitored alphabet such that $|\Sigma| = n$ and $|R| = k$. A tuple $(\boldsymbol{\alpha}, \sigma) = (\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_k, \sigma) \in (\mathbb{N}^n)^{k+1} \times \mathfrak{S}_k$ is* a generalized Parikh image of $\gamma \in (\Sigma \uplus R)^*$ *if there exist $0 \leq p \leq k$ and a decomposition $\gamma = \gamma_p r_{\sigma(p+1)} \gamma_{p+1} r_{\sigma(p+2)} \cdots r_{\sigma(k)} \gamma_k$ such that:*

(a) for all $p \leq i \leq k$, $\gamma_i \in \Sigma^*_{R_i}$, where $R_i = \{r_{\sigma(i+1)}, \dots, r_{\sigma(k)}\}$; and
(b) for all $0 \leq i < p$, $\boldsymbol{\alpha}_i = \mathbf{0}$ and for all $p \leq i \leq k$, $\boldsymbol{\alpha}_i = \pi_\Sigma(\gamma_i)$, the Parikh image of γ_i restricted to Σ, i.e. *monitored alphabet symbols are ignored.*

The generalized Parikh image of a language $L \subseteq (\Sigma \uplus R)^*$ is the set $\Pi(L) \subseteq (\mathbb{N}^n)^{k+1} \times \mathfrak{S}_k$ of all generalized Parikh images of all words $\gamma \in L$.

This definition formalizes the intuition given by the decomposition described above with some additional padding of dummy vectors for monitored letters not occurring in γ in order to obtain canonical objects of *uniform size*. Even though generalized Parikh images are not unique, two generalized Parikh images of the same word only differ in the order of dummy monitored letters. For instance for $k = 4$, the word $\gamma = aabr_1br_3abr_3ar_1$ has two generalized Parikh images: they coincide on $\boldsymbol{\alpha}_0 = \boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2 = (0, 0)$, $\boldsymbol{\alpha}_3 = (3, 3)$, $\boldsymbol{\alpha}_4 = (1, 0)$ and $\sigma(3) = 3$, $\sigma(4) = 1$, and only differ on $\sigma(1)$ and $\sigma(2)$ that can be 2 and 4, or 4 and 2, respectively.

Generalized Parikh images can now be applied to reachability in $\mathbb{Z}$-VASS_R as follows. Without loss of generality, we may assume that a $\mathbb{Z}$-VASS_R in dimension d is given as $\mathcal{A} = (Q, \Sigma \uplus R, d, \Delta, \tau)$ for some alphabet $\Sigma = \{a_1, \dots, a_n\}$ and $R = \{r_1, \dots, r_d\}$ such that $\tau(r_i) = \boldsymbol{v} \mapsto v_{|i}$ for any $i \in [d]$ and for any $a_i \in \Sigma$, $\tau(a_i) = \boldsymbol{v} \mapsto \boldsymbol{v} + \boldsymbol{b}_i$ for some $\boldsymbol{b}_i \in \mathbb{Z}^d$. This assumption allows for isolating transitions performing a reset and enables us to apply monitored alphabets by monitoring when a reset occurs in each dimension the last time. Consequently, the counter value realized by some $\gamma \in (\Sigma \uplus R)^*$ starting from $\mathbf{0}$ is fully determined by a generalized Parikh image of γ.

Lemma 6. *Let $\mathcal{A}$ be a $\mathbb{Z}$-VASS$_R$, $\gamma \in (\Sigma \uplus R)^*$, $(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_d, \sigma) \in \Pi(\gamma)$ and $B \in \mathbb{Z}^{d \times n}$ the matrix whose columns are the vectors $\boldsymbol{b}_i$. Then the following holds:*

$$\tau(\gamma)(\mathbf{0}) = \sum\nolimits_{1 \leq i \leq d} (B\boldsymbol{\alpha}_{i-1})_{|\{\sigma(i),\ldots,\sigma(d)\}} + B\boldsymbol{\alpha}_d.$$

It thus remains to find a suitable way to define the generalized Parikh image of the language of the non-deterministic finite state automaton (NFA) underlying a $\mathbb{Z}$-VASS$_{\mathrm{R}}$. In [26], it is shown how to construct in linear time an existential Presburger formula representing the Parikh image of the language of an NFA. We generalize this construction to generalized Parikh images of NFA over a monitored alphabet, the original result being recovered in the absence of monitored alphabet symbols, *i.e.* when $k = 0$. To this end, we introduce below some definitions and two lemmas from the construction provided in [26] which we employ for our generalization. First, a *flow* in an NFA $\mathcal{B} = (Q, \Sigma, \Delta, q_0, F)$ is a triple (f, s, t) where $s, t \in Q$ are states, and $f : \Delta \to \mathbb{N}$ maps transitions $(p, a, q) \in \Delta$ to natural numbers. Let us introduce the following abbreviations:

$$\mathrm{in}_f(q) = \sum_{(p,a,q) \in \Delta} f(p, a, q) \quad \text{and} \quad \mathrm{out}_f(p) = \sum_{(p,a,q) \in \Delta} f(p, a, q).$$

A flow (f, s, t) is called *consistent* if for each $p \in Q$, $\mathrm{in}_f(p) = \mathrm{out}_f(p) + h(p)$, where $h(s) = -1$, $h(t) = 1$, and $h(p) = 0$ otherwise. A flow is *connected* if the undirected graph obtained from the graph underlying the automaton when removing edges with zero flow is connected. A consistent and connected flow simply enforces Eulerian path conditions on the directed graph underlying $\mathcal{B}$ so that any path starting in s and ending in t yields a unique such flow.

Lemma 7 ([26]). *A vector $\boldsymbol{\alpha} \in \mathbb{N}^n$ is in the Parikh image of $\mathcal{L}(\mathcal{B})$ if, and only if, there is a consistent and connected flow (f, s, t) such that*

- *$s = q_0$, $t \in F$, and*
- *for each $a_i \in \Sigma$, $\boldsymbol{\alpha}(i) = \sum_{(p,a_i,q) \in \Delta} f(p, a_i, q)$*

Subsequently, in order to conveniently deal with states and alphabet symbols in Presburger arithmetic, we write $Q = \{\tilde{1}, \ldots, \tilde{m}\}$, $\Sigma = \{\dot{1}, \ldots, \dot{n}\}$ and $R = \{(n \dot{+} 1), \ldots, (n \dot{+} k)\}$. This enables us to write within the logic terms like $p = q$ for $\tilde{p}, \tilde{q} \in Q$. Moreover, it is easy to construct a formula $\varphi_\Delta(p, a, q)$ such that $\varphi_\Delta(p, a, q)$ holds if, and only if, $(\tilde{p}, \dot{a}, \tilde{q}) \in \Delta$. In particular, φ_Δ can be constructed in linear time, independently of the encoding of the NFA and its graph structure. With this encoding, it is not difficult to see how the conditions from Lemma 7 can be checked by an existential Presburger formula.

Lemma 8 ([26]). *There exists a linear-time computable existential Presburger formula $\varphi_{\mathcal{B}}(\boldsymbol{f}, s, t)$ with the following properties:*

- *$\boldsymbol{f}$ represents a flow,* i.e.*, is a tuple of variables $x_{(p,a,q)}$ for each $(p, a, q) \in \Delta$;*
- *s and t are free variables constrained to represent states of Q; and*

– $(m_{\delta_1}, \ldots, m_{\delta_g}, m_s, m_t) \in [\![\varphi_{\mathcal{B}}(\boldsymbol{f}, s, t)]\!]$ *if, and only if, the flow* $(f_m, \tilde{m_s}, \tilde{m_t})$ *defined by* $f_m(\delta_i) = m_{\delta_i}$ *is connected and consistent in* $\mathcal{B}$.

We can now show how to generalize the construction from [26] to monitored alphabets and generalized Parikh images. Subsequently, recall that k is the number of monitored letters.

Theorem 9. *Given an NFA* $\mathcal{B} = (Q, \Sigma \uplus R, \Delta, \tilde{q_0}, F)$ *over a monitored alphabet* $\Sigma \uplus R$*, an existential Presburger formula* $\Psi_{\mathcal{B}}(\boldsymbol{\alpha}, \boldsymbol{\sigma})$ *defining the generalized Parikh image of the language* $\mathcal{L}(B)$ *of* $\mathcal{B}$ *can be constructed in time* $O(k^2|\mathcal{B}|)$.

Proof. The formula we construct has free variables $\alpha_0^1, \ldots, \alpha_0^n, \alpha_1^1, \ldots, \alpha_k^n$ representing the $k+1$ vectors $\boldsymbol{\alpha_0}, \ldots, \boldsymbol{\alpha_k}$ and free variables $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_k)$ to represent the permutation σ. First, we construct a formula φ_{perm} expressing that $\boldsymbol{\sigma}$ is a permutation from $[k]$ to $[k]$:

$$\varphi_{\text{perm}}(\boldsymbol{\sigma}) = \bigwedge_{i \in [k]} \left(1 \leq \sigma_i \leq k \ \wedge \bigwedge\nolimits_{j \in [k]} i \neq j \rightarrow \sigma_i \neq \sigma_j \right).$$

This formula has already size $O(k^2)$. Now we have to compute the flow for each of the $k+1$ parts of the runs corresponding to the $k+1$ partial words, but first we have to "guess" the starting and ending states of each of these partial runs, in order to use the formula from Lemma 8. Let $\boldsymbol{s} = (s_0, \ldots, s_k)$ and $\boldsymbol{t} = (t_0, \ldots, t_k)$, we define

$$\varphi_{\text{states}}(\boldsymbol{\sigma}, p, \boldsymbol{s}, \boldsymbol{t}) = s_0 = q_0 \wedge \bigvee\nolimits_{\tilde{q} \in F} t_k = q \wedge \\ \bigwedge_{i \in [k]} [i \leq p \rightarrow s_{i-1} = t_{i-1} \wedge t_{i-1} = s_i] \wedge [p < i \rightarrow \varphi_{\Delta}(t_{i-1}, n + \sigma_i, s_i)].$$

Here, p is used as in Definition 5. We can now express the $k+1$ flows: we need one variable per transition for each partial run.

$$\varphi_{\text{flows}}(\boldsymbol{\sigma}, p, \boldsymbol{f}, \boldsymbol{s}, \boldsymbol{t}) = \bigwedge_{0 \leq i \leq k} i < p \rightarrow \sum_{(p,a,q) \in \Delta} x^i_{(p,a,q)} = 0 \wedge$$

$$\wedge \bigwedge_{0 \leq i \leq k} p \leq i \rightarrow \left(\varphi_{\mathcal{B}}(\boldsymbol{f}_i, s_i, t_i,) \wedge \bigwedge_{1 \leq j < i} \bigwedge_{(p,\dot{a},q) \in \Delta} a = n + \sigma_j \rightarrow x^i_{(p,\dot{a},q)} = 0 \right),$$

where $\boldsymbol{f} = (\boldsymbol{f}_0, \ldots, \boldsymbol{f}_k)$ and $\boldsymbol{f}_i$ is the tuple of free variables of the form $x^i_{(p,a,q)}$ for all $(p, a, q) \in \Delta$. This formula essentially enforces the constraints from Definition 5. The first line enforces that the "dummy flows" $\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{p-1}$ have zero flow. The second line ensures that the flows $\boldsymbol{f}_p, \ldots, \boldsymbol{f}_k$ actually correspond to partial words γ_i in the decomposition described in Definition 5, and that monitored letters that, informally speaking, have expired receive zero flow. Now

putting everything together yields:

$$\Psi_{\mathcal{B}}(\boldsymbol{\alpha},\boldsymbol{\sigma}) = \exists p, \boldsymbol{f}_0, \dots \boldsymbol{f}_k, \boldsymbol{s}, \boldsymbol{t}.\, 0 \leq p \leq k \wedge \varphi_{\mathrm{perm}}(\boldsymbol{\sigma}) \wedge \\ \wedge \varphi_{\mathrm{states}}(\boldsymbol{\sigma}, p, \boldsymbol{s}, \boldsymbol{t}) \wedge \varphi_{\mathrm{flows}}(\boldsymbol{\sigma}, p, \boldsymbol{f}, \boldsymbol{s}, \boldsymbol{t}) \wedge \bigwedge_{0 \leq i \leq k} \bigwedge_{a \in [n]} \alpha_i^a = \sum_{(p,\dot{a},q) \in \Delta} x^i_{(p,\dot{a},q)}.$$

The size of $\Psi_{\mathcal{B}}(\boldsymbol{\alpha},\boldsymbol{\sigma})$ is dominated by the size of $\varphi_{\mathrm{flows}}(\boldsymbol{\sigma}, p, \boldsymbol{f}, \boldsymbol{s}, \boldsymbol{t})$ which is $O(k^2|\mathcal{B}|)$. □

Note that it is easy to modify $\Psi_{\mathcal{B}}$ in order to have q_0 as a free variable. By combining $\Psi_{\mathcal{B}}$ with Lemma 6, we obtain the following corollary.

Corollary 10. *Let $\mathcal{A}$ be a $\mathbb{Z}$-VASS$_R$ and $p, q \in Q$. There exists a logarithmic-space computable existential Presburger formula*[3] *$\Phi_{\mathcal{A}}(p, q, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\sigma})$ such that $(p, q, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\sigma}) \in [\![\Phi_{\mathcal{A}}]\!]$ if, and only if, there is $\gamma \in (\Sigma \uplus R)^*$ such that $\tilde{p}(\boldsymbol{v}) \xrightarrow{\gamma}_{\mathcal{A}} \tilde{q}(\boldsymbol{w})$ and $(\boldsymbol{\alpha}, \sigma) \in \Pi(\gamma)$, where $\sigma(i) = \boldsymbol{\sigma}(i)$.*

In particular, this implies that the reachability set of $\mathbb{Z}$-VASS$_{\mathrm{R}}$ is semi-linear, and that reachability in $\mathbb{Z}$-VASS$_{\mathrm{R}}$ is NP-complete.

4 Inclusion for $\mathbb{Z}$-VASS

In this section, we show the following theorem.

Theorem 11. *Inclusion for $\mathbb{Z}$-VAS is NP-hard and in Π_2^{P}, and coNEXP-complete for $\mathbb{Z}$-VASS and $\mathbb{Z}$-VASS$_R$.*

The upper bounds follow immediately from the literature. For $\mathbb{Z}$-VAS we observe that we are asking for inclusion between linear sets. Huynh [17] shows that inclusion for semi-linear sets is Π_2^{P}-complete, which yields the desired upper bound. Regarding inclusion for $\mathbb{Z}$-VASS$_{\mathrm{R}}$, from Corollary 10 we have that the reachability set of a $\mathbb{Z}$-VASS$_{\mathrm{R}}$ is Σ_1-PA definable. Let $\mathcal{A}, \mathcal{B}$ be $\mathbb{Z}$-VASS$_{\mathrm{R}}$ in dimension d, $q(\boldsymbol{v}) \in C(\mathcal{A})$, $p(\boldsymbol{w}) \in C(\mathcal{B})$, and let $\phi_{\mathcal{A},q(\boldsymbol{v})}(\boldsymbol{x})$ and $\phi_{\mathcal{B},p(\boldsymbol{w})}(\boldsymbol{x})$ be appropriate Σ_1-PA formulas from Corollary 10 with $\boldsymbol{x} = (x_1, \dots, x_d)$. We have

$$\mathit{reach}(\mathcal{A}, q(\boldsymbol{v})) \subseteq \mathit{reach}(\mathcal{B}, p(\boldsymbol{w})) \Leftrightarrow \neg(\exists \boldsymbol{x}.\phi_{\mathcal{A},q(\boldsymbol{v})}(\boldsymbol{x}) \wedge \neg(\phi_{\mathcal{B},p(\boldsymbol{w})}(\boldsymbol{x}))) \text{ is valid.}$$

Bringing the above formula into prenex normal form yields a Π_2-PA sentence for which validity can be decided in coNEXP [12]. For that reason we focus on the lower bounds in the remainder of this section.

For $\mathbb{Z}$-VAS, an NP-lower bound follows straight-forwardly via a reduction from the feasibility problem of a system of linear Diophantine equations $A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$. Despite some serious efforts, we could not establish a stronger lower bound. Even though it is known that inclusion for semi-linear sets is Π_2^{P}-hard [16], this lower bound does not seem to carry over to inclusion for $\mathbb{Z}$-VAS.

[3] Here, we allow $\boldsymbol{v}$ and $\boldsymbol{w}$ to be interpreted over $\mathbb{Z}$, which can easily be achieved by representing an integer as the difference of two natural numbers.

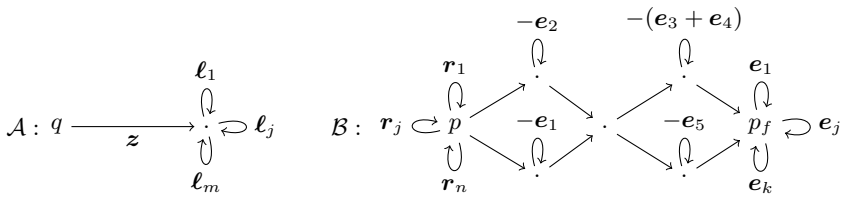

Fig. 1. Illustration of the approach to reduce validity of a Π_2-PA formula $\Phi = \forall\boldsymbol{x}.\exists\boldsymbol{y}.(t_1 \vee t_2) \wedge ((t_3 \wedge t_4) \vee v_5)$ to inclusion for $\mathbb{Z}$-VASS.

Lemma 12. *Inclusion for $\mathbb{Z}$-VASS is* coNEXP*-hard even when numbers are encoded in unary.*

Proof. We reduce from validity in Π_2-PA, which is coNEXP-hard already when numbers are encoded in unary [10, 12]. To this end, let $\Phi = \forall\boldsymbol{x}.\exists\boldsymbol{y}.\varphi(\boldsymbol{x},\boldsymbol{y})$ be a formula in this fragment such that $\boldsymbol{x}$ and $\boldsymbol{y}$ are m- and n-tuples of first-order variables, respectively. As discussed in the introduction, with no loss of generality we may assume that $\varphi(\boldsymbol{x},\boldsymbol{y})$ is a positive Boolean combination of k terms $t_1,\dots,t_k$ of the form $t_i = \boldsymbol{a}_i \cdot \boldsymbol{x} + z_i \geq \boldsymbol{b}_i \cdot \boldsymbol{y}$ with $\boldsymbol{a}_i \in \mathbb{Z}^m, \boldsymbol{b}_i \in \mathbb{Z}^n$ and $z_i \in \mathbb{Z}$. In our reduction, we show how to construct in logarithmic space $\mathbb{Z}$-VASS $\mathcal{A},\mathcal{B}$ with designated control states q,p such that Φ is valid iff $\mathit{reach}(\mathcal{A},q(\mathbf{0})) \subseteq \mathit{reach}(\mathcal{B},p(\mathbf{0}))$. Figure 1 illustrates the structure of the $\mathbb{Z}$-VASS $\mathcal{A}$ and $\mathcal{B}$. A key point behind our reduction is that the counters of $\mathcal{A}$ and $\mathcal{B}$ are used to represent evaluations of left-hand and right-hand-sides of the *terms* of $\varphi(\boldsymbol{x},\boldsymbol{y})$.

In Figure 1, we have that $\boldsymbol{z} \in \mathbb{Z}^k$ is such that $\boldsymbol{z}(i) = z_i$. For $j \in [m]$, $\boldsymbol{\ell}_j \in \mathbb{Z}^k$ is such that $\boldsymbol{\ell}_j(i) = a_i(j)$. Likewise, for $j \in [n]$, $\boldsymbol{r}_j \in \mathbb{Z}^k$ is such that $\boldsymbol{r}_j(i) = b_i(j)$. When moving away from state q, $\mathcal{A}$ adds the absolute term of each t_i to the respective counters. It can then choose any valuation of the $\boldsymbol{x}$ and thus stores the corresponding values of the left-hand sides of each t_i in the counters. Now $\mathcal{B}$ has to match the choice of $\mathcal{A}$. To this end, it can first loop in state p in order to guess a valuation of the $\boldsymbol{y}$ and update the values of the counters accordingly, which now correspond to the right-hand sides of the t_i. Along a path from p to p_f, $\mathcal{B}$ may, if necessary, simulate the Boolean structure of φ: conjunction is simulated by sequential composition and disjunction by branching. For every conjunct of φ, $\mathcal{B}$ can non-deterministically decrement all but one term of every disjunct. Finally, once $\mathcal{B}$ reaches p_f, it may non-deterministically increase the value corresponding to the right-hand sides of every term in order to precisely match any value reached by $\mathcal{A}$. From this example, it is now clear how to construct $\mathcal{A}$ and $\mathcal{B}$ from Φ in general in logarithmic space such that Φ is valid if, and only if, $\mathcal{B}$ has a run beginning in $p(\mathbf{0})$ that matches the counter values reached by any run of $\mathcal{A}$ beginning in $q(\mathbf{0})$. Obviously, the the converse direction holds as well. □

5 Concluding Remarks

We studied reachability, coverability and inclusion problems for various classes of $\mathbb{Z}$-VASS, *i.e.*, VASS whose counter values range over $\mathbb{Z}$. Unsurprisingly, the complexity of those decision problems is lower for $\mathbb{Z}$-VASS when compared to VASS. However, the extend to which the complexity drops reveals an element of surprise: coverability and reachability for VASS in the presence of resets are $\mathbf{F}_\omega$-complete and undecidable, respectively, but both problems are only NP-complete for $\mathbb{Z}$-VASS_R. For the upper bound, we provided a generalization of Parikh images which we believe is a technical construction of independent interest.

Throughout this paper, the dimension of the $\mathbb{Z}$-VASS has been part of the input. A natural line of future research could be to investigate the complexity of the problems we considered in fixed dimensions.

Acknowledgments. We would like to thank the anonymous referees, Sylvain Schmitz and Philippe Schnoebelen for their helpful comments and suggestions on an earlier version of this paper.

References

1. P. Bell and I. Potapov. On undecidability bounds for matrix decision problems. *Theor. Comput. Sci.*, 391(1–2):3–13, 2008.
2. I. Borosh and L.B. Treybing. Bounds on positive integral solutions of linear Diophantine equations. *Proc. AMS*, 55:299–304, 1976.
3. P. Buckheister and G. Zetzsche. Semilinearity and context-freeness of languages accepted by valence automata. In *Proc. MFCS*, volume 8087 of *LNCS*, pages 231–242, 2013.
4. C. Dufourd, A. Finkel, and Ph. Schnoebelen. Reset nets between decidability and undecidability. In *Proc. ICALP*, volume 1443 of *LNCS*, pages 103–115, 1998.
5. J. Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundam. Inform.*, 31(1):13–25, 1997.
6. A. Finkel, S. Göller, and C. Haase. Reachability in register machines with polynomial updates. In *Proc. MFCS*, volume 8087 of *LNCS*, pages 409–420, 2013.
7. E. Fraca and S. Haddad. Complexity analysis of continuous Petri nets. In *Proc. ATPN*, volume 7927 of *LNCS*, pages 170–189, 2013.
8. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
9. S. Ginsburg and E.H. Spanier. Semigroups, Presburger formulas and languages. *Pac. J. Math.*, 16(2):285–296, 1966.
10. E. Grädel. Dominoes and the complexity of subclasses of logical theories. *Ann. Pure Appl. Logic*, 43(1):1–30, 1989.
11. S.A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theor. Comput. Sci.*, 7(3):311 – 324, 1978.
12. C. Haase. Subclasses of Presburger arithmetic and the weak EXP hierarchy. In *Proc. CSL-LICS*, 2014. To appear.
13. C. Haase, S. Kreutzer, J. Ouaknine, and J. Worrell. Reachability in succinct and parametric one-counter automata. In *Proc. CONCUR*, volume 5710 of *LNCS*, pages 369–383, 2009.

14. M. Hack. The equality problem for vector addition systems is undecidable. *Theor. Comput. Sci.*, 2(1):77–95, 1976.
15. M. Hague and A.W. Lin. Model checking recursive programs with numeric data types. In G. Gopalakrishnan and S. Qadeer, editors, *Proc. CAV*, volume 6806 of *LNCS*, pages 743–759, 2011.
16. D.T. Huynh. The complexity of equivalence problems for commutative grammars. *Inform. Control*, 66(1–2):103–121, 1985.
17. D.T. Huynh. A simple proof for the Σ_2^p upper bound of the inequivalence problem for semilinear sets. *Elektron. Inform. Kybernet.*, 22(4):147–156, 1986.
18. P. Jančar. Nonprimitive recursive complexity and undecidability for Petri net equivalences. *Theor. Comput. Sci.*, 256(1-2):23–30, 2001.
19. E. Kopczyński and A.W. To. Parikh images of grammars: Complexity and applications. In *Proc. LICS*, pages 80–89, 2010.
20. R. Lipton. The reachability problem is exponential-space-hard. Technical report, Yale University, New Haven, CT, 1976.
21. E. W. Mayr. An algorithm for the general Petri net reachability problem. In *Proc. STOC*, pages 238–246, New York, NY, USA, 1981. ACM.
22. W. Plandowski and W. Rytter. Complexity of language recognition problems for compressed words. In J. Karhumäki, H.A. Maurer, G. Păun, and G. Rozenberg, editors, *Jewels are Forever*, pages 262–272, 1999.
23. C. Rackoff. The covering and boundedness problems for vector addition systems. *Theor. Comput. Sci.*, 6(2):223–231, 1978.
24. J. Reichert. On the complexity of counter reachability games. In *Proc. RP*, volume 8169 of *LNCS*, pages 196–208, 2013.
25. Ph. Schnoebelen. Revisiting Ackermann-hardness for lossy counter machines and reset Petri nets. In *Proc. MFCS*, volume 6281 of *LNCS*, pages 616–628, 2010.
26. H. Seidl, Th. Schwentick, A. Muscholl, and P. Habermehl. Counting in trees for free. In *Proc. ICALP*, volume 3142 of *LNCS*, pages 1136–1149, 2004.

A Missing Proofs from Section 2

A.1 Proof of Lemma 3

Lemma 13. *Reachability in $\mathbb{Z}$-VAS is* NP*-hard already when numbers are encoded in unary.*

Proof. Let $S : \exists \boldsymbol{x}.A\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$ be a system of linear Diophantine equations such that A consists of row vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$. Determining whether S is valid is a well-known NP-hard problem even when numbers are encoded in unary [8]. From S, we can easily construct a $\mathbb{Z}$-VASS $\mathcal{A}$ with one control state q such that $q(\boldsymbol{0}) \to^*_{\mathcal{A}} q(\boldsymbol{b})$ if, and only if, S is valid as follows: for every $\boldsymbol{a}_i$, $\mathcal{A}$ has a self-loop reading the alphabet symbol a_i and adding $\boldsymbol{a}_i$ to the counter. Given a word γ witnessing $q(\boldsymbol{0}) \to^*_{\mathcal{A}} q(\boldsymbol{b})$, counting the numbers of times each a_i occurs along γ yields a valuation of $\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$. Conversely, any valuation of $\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$ gives rise to a run $q(\boldsymbol{0}) \to^*_{\mathcal{A}} q(\boldsymbol{b})$. □

Remark 14. If $\mathcal{A}$ is constructed as above, we have that S is valid if, and only if, $\{\lambda \boldsymbol{b} : \lambda \in \mathbb{N}\} \subseteq \mathit{reach}(\mathcal{A}, q(\boldsymbol{0}))$. This shows that inclusion for $\mathbb{Z}$-VAS is NP-hard.

A.2 Proof of Lemma 4

Lemma 15. *Let $\mathfrak{D}_d$ be the set of all diagonal matrices in dimension d. Coverability in $\mathbb{Z}$-RM($\mathfrak{D}_d$) is undecidable already for $d = 4$.*

Proof. Reachability for $\mathbb{Z}$-RM($\mathfrak{D}_2$) is undecidable as announced in [6]. This result has been obtained by J. Reichert and has not yet appeared in written format. For the sake of completeness, here we first repeat Reichert's argument.

Undecidability is shown via reduction from the undecidable Post Correspondence Problem (PCP). Given $u_1, \ldots, u_n, v_1, \ldots, v_n \in \{0,1\}^*$, PCP asks whether there are some $i_1, \ldots, i_p$ ($p > 0$) such that $u_{i_1} \cdots u_{i_p} = v_{i_1} \cdots v_{i_p}$. Below, we define a $\mathbb{Z}$-RM($\mathfrak{D}_2$) $\mathcal{A} = (\{q_0, q_f\} \cup Q, \{0, 1, \tilde{0}, \tilde{1}, \#\}, 2, \Delta, \tau)$ such that there is a run from $q_0(\boldsymbol{0})$ to $q_f(\boldsymbol{0})$ in $\mathcal{A}$ if, and only if, there is a solution to the above PCP instance:

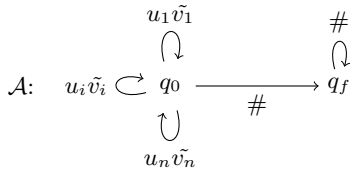

Fig. 2. The $\mathbb{Z}$-RM($\mathfrak{D}_2$) $\mathcal{A}$ used for the reduction from PCP.

$\mathcal{A}$ has n self-loops on q_0, and each of these loops is labeled by a word $w = u_i\tilde{v_i}$. This, of course, actually corresponds to a path with $|w|$ states such that the path

reads w. We now define τ as:

$$\tau(0)(\boldsymbol{v}) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{v} \qquad \tau(\tilde{0})(\boldsymbol{v}) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \boldsymbol{v}$$
$$\tau(1)(\boldsymbol{v}) = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \boldsymbol{v} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \tau(\tilde{1})(\boldsymbol{v}) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \boldsymbol{v} + \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$\tau(\#)(\boldsymbol{v}) = \boldsymbol{v} - \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The idea is that to reach $q_f(\mathbf{0})$ from $q_0(\mathbf{0})$, the two counters must be equal when leaving state q_0. Thinking of counter values encoded in binary, the counters represent the concatenation of the u_i and the v_i, respectively, since in binary, multiplying by 2 corresponds to concatenating 0, and multiplying by 2 and adding 1 corresponds to concatenating 1. Looping non-deterministically on q_0, the machine "guesses" an order to make the two counters, *i.e.*, words, equal.

Note that the only matrices appearing in $\mathcal{A}$ are diagonal, and of dimension 2. By application of Lemma 2, we obtain that coverability is henceforth undecidable for matrices from $\mathfrak{D}_4$. □

Note that reachability for $\mathfrak{D}_1$ is shown decidable in, which implies that coverability is decidable in this setting as well. However, coverability for $\mathfrak{D}_2$ and $\mathfrak{D}_3$ remain an open problems. It is surprising to have such a complexity gap between $\mathbb{Z}$-RM with diagonal matrices and $\mathbb{Z}$-RM with diagonal matrices with only zeros and ones, which respectively make our problems undecidable and NP-complete. Thus, it is natural to wonder whether some decidable class of matrices lies in between.

B Missing Proofs from Section 3

B.1 Proof of Lemma 6

Lemma 16. *Let $\mathcal{A}$ be a $\mathbb{Z}$-VASS$_R$, $\boldsymbol{v} \in \mathbb{Z}^d$, $\gamma \in (\Sigma \uplus R)^*$, $(\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_d, \sigma) \in \Pi(\gamma)$ a generalized Parikh image of γ and $B \in \mathbb{Z}^{d \times n}$ the matrix whose columns are the vectors $\boldsymbol{b}_i$. Then the following holds:*

$$\tau(\gamma)(\mathbf{0}) = \sum_{1 \leq i \leq d} (B\boldsymbol{\alpha}_{i-1})_{|\{\sigma(i),\ldots,\sigma(d)\}} + B\boldsymbol{\alpha}_d.$$

Proof. Let p be the number introduced in Definition 5, we prove the following stronger statement by induction on $j \in [p, d]$:

$$\tau(\gamma_p r_{\sigma(p+1)} \gamma_{p+1} \ldots r_{\sigma(j)} \gamma_j)(\mathbf{0})_{|\{\sigma(j+1),\ldots,\sigma(d)\}} = \sum_{i=0}^{j} (B\boldsymbol{\alpha}_i)_{|\{\sigma(i+1),\ldots,\sigma(d)\}}$$

where $\gamma = \gamma_p r_{\sigma(p+1)} \gamma_{p+1} \ldots r_{\sigma(d)} \gamma_k$ is the decomposition introduced in Definition 5. We then conclude by taking $j = d$.

– Base case $j = p$:
Let $S = \{\sigma(p+1), \dots, \sigma(d)\}$. Since only resets r_i for $i \in S$ occurs in γ_p by definition of the decomposition, and since addition is commutative and associative, only the number of times each letter appear is important. Therefore:

$$\tau(\gamma_p)(\mathbf{0})_{|S} = \sum_{i=1}^{n} |\gamma_p|_{a_i}.(\boldsymbol{b}_i)_{|S} = (B\boldsymbol{\alpha}_p)_{|S} = \sum_{i=0}^{p} (B\boldsymbol{\alpha}_i)_{|\{\sigma(i+1),\dots,\sigma(d)\}}$$

Remember that for $i < p$, $\boldsymbol{\alpha}_i = \mathbf{0}$, which explains the last equality.
– Induction step: Let $S = \{\sigma(j+1), \dots, \sigma(d)\}$ and $S' = \sigma(j) \cup S$ and $\gamma' = \gamma_p r_{\sigma(p+1)} \cdots \gamma_{j-1}$:

$$\begin{aligned}
&\tau(\gamma_p r_{\sigma(p+1)} \cdots r_{\sigma(j)} \gamma_j)(\mathbf{0})_{|S} \\
&\quad = \tau(\gamma' r_{\sigma(j)} \gamma_j)(\mathbf{0}) \\
&\quad = \tau(r_{\sigma(j)} \gamma_j)(\tau(\gamma')(\mathbf{0}))_{|S} && (1) \\
&\quad = \tau(\gamma_j)([\tau(\gamma')(\mathbf{0})]_{|\sigma(j)})(\mathbf{0}))_{|S} && (2) \\
&\quad = [(\tau(\gamma')(\mathbf{0}))_{|\sigma(j)} + \tau(\gamma_j)(\mathbf{0})]_{|S} && (3) \\
&\quad = [(\tau(\gamma')(\mathbf{0}))_{|\sigma(j)}]_{|S} + \tau(\gamma_j)(\mathbf{0})_{|S} \\
&\quad = \tau(\gamma')(\mathbf{0})_{|S'} + \tau(\gamma_j)(\mathbf{0})_{|S} \\
&\quad = \sum_{i=0}^{j-1} (B\boldsymbol{\alpha}_i)_{|\{\sigma(i+1),\dots,\sigma(d)\}} + (B\boldsymbol{\alpha}_j)_{|S} && (4) \\
&\quad = \sum_{i=0}^{j} (B\boldsymbol{\alpha}_i)_{|\{\sigma(i+1),\dots,\sigma(d)\}},
\end{aligned}$$

where
(1) by definition of τ
(2) by definition of $\tau(r_{\sigma(j)})$
(3) as γ_j has only resets from S
(4) by induction hypothesis.

□

B.2 Proof of Corollary 10

Throughout this section, let $\mathcal{A} = (Q, \Sigma \uplus R, d, \Delta, \tau)$ be a $\mathbb{Z}$-VASS$_{\mathrm{R}}$. Before we give the proof of Corollary 10, we prove the following lemma.

Lemma 17. *There exists a logarithmic-space computable existential Presburger formula $\varphi_{counters}(\boldsymbol{\alpha}, \boldsymbol{\sigma}, p, \boldsymbol{v}, \boldsymbol{v}')$ such that $(\boldsymbol{\alpha}, \boldsymbol{\sigma}, p, \boldsymbol{v}, \boldsymbol{v}') \in [\![\varphi_{counters}]\!]$ if, and only if, there is a word $\gamma \in (\Sigma \uplus R)^*$ such that $\tau(\gamma)(\boldsymbol{v}) = \boldsymbol{v}'$ and $(\boldsymbol{\alpha}, \boldsymbol{\sigma}) \in \Pi(\gamma)$ with p being the number introduced in Definition 5.*

Proof. In Presburger arithmetic, the equality $\tau(\gamma)(\boldsymbol{v}) = \boldsymbol{v}'$ is actually represented by d equalities $\tau(\gamma)(\boldsymbol{v})(i) = \boldsymbol{v}'(i)$ for $i \in [d]$. Lemma 6 states that for any

$i \in [d]$, $\tau(\gamma)(\boldsymbol{v})(i) = \sum_{j=0}^{d} (B\boldsymbol{\alpha}_j)_{|\{\sigma(j+1),\ldots,\sigma(d)}(i) = \sum_{j=0}^{d} \lambda_{ij}(B\boldsymbol{\alpha}_j)(i)$ where

$$\lambda_{ij} = \begin{cases} 0 & \text{iff } i \in \{\sigma(j+1),\ldots,\sigma(d)\} \text{ iff } \sigma^{-1}(i) \geq (j+1) \\ 1 & \text{otherwise.} \end{cases}$$

This last equality is not a syntactically correct Presburger term since it is quadratic instead of linear. We therefore introduce intermediate variables β_j^i to compute the partial sums $\beta_j^i = \sum_{k=0}^{j} \lambda_{ik}(B\boldsymbol{\alpha}_k)(i)$:

$$\varphi_{\text{counters}}(\boldsymbol{\alpha}, \boldsymbol{\sigma}, p, \boldsymbol{v}, \boldsymbol{v}') = \exists \boldsymbol{\beta}. \exists \boldsymbol{\nu}. \bigwedge_{i=1}^{d} \beta_0^i = 0 \wedge \boldsymbol{v}'(i) = \beta_d^i + \boldsymbol{\nu}(i) \wedge$$
$$\wedge \bigwedge_{k=1}^{d} (\sigma(k) = i) \rightarrow \bigwedge_{j=1}^{d} (k > j \rightarrow \beta_j^i = \beta_{j-1}^i) \wedge (k \leq j \rightarrow \beta_j^i = \beta_{j-1}^i + (B\boldsymbol{\alpha}_j)(i) \wedge$$
$$\wedge (k > p \rightarrow \boldsymbol{\nu}(i) = 0) \wedge (k \leq p \rightarrow \boldsymbol{\nu}(i) = \boldsymbol{v}(i)).$$

When reading this formula, one should see k as $\sigma^{-1}(i)$, and therefore recognize the second line to be the condition expressed above. As β_j^i are the partial sums up to j, β_d^i represents the complete sum on the dimension i. The final vector $\boldsymbol{v}'$ is thus equal to the vector made of the β_d^i plus the starting vector $\boldsymbol{v}$ in which the right components have been erased: this is the vector $\boldsymbol{\nu}$. □

Corollary 18 (Corollary 10 in the main text). *Let $\mathcal{A}$ be a $\mathbb{Z}$-VASS$_R$ and $p, q \in Q$. There exists a logarithmic-space computable existential Presburger formula $\Phi_{\mathcal{A}}(q', q, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\sigma})$ such that $(p, q, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\sigma}) \in [\![\Phi_{\mathcal{A}}]\!]$ if, and only if, there is $\gamma \in (\Sigma \uplus R)^*$ such that $\tilde{q'}(\boldsymbol{v}) \xrightarrow{\gamma}_{\mathcal{A}} \tilde{q}(\boldsymbol{w})$ and $(\boldsymbol{\alpha}, \sigma) \in \Pi(\gamma)$, where $\sigma(i) = \boldsymbol{\sigma}(i)$.*

Proof. Note $\Psi'_{\mathcal{B}}$ for the formula $\Psi_{\mathcal{B}}$ (*cf.* Theorem 9) without the quantification on p, and with additionnal free variables q and q' for the initial and final states. By Lemma **??**, we conclude with:

$$\Phi_{\mathcal{A}}(q', q, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\sigma}) = \exists p.\ \Psi'_{\mathcal{B}}(\boldsymbol{\alpha}, \boldsymbol{\sigma}, p, q, q') \wedge \varphi_{\text{counters}}(\boldsymbol{\alpha}, \boldsymbol{\sigma}, p, \boldsymbol{v}, \boldsymbol{v}').$$

□