Monus Semantics in Vector Addition Systems with States

Pascal Baumann
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Khushraj Madnani
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Filip Mazowiecki
University of Warsaw, Poland

Georg Zetzsche
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Abstract

Vector addition systems with states (VASS) are a popular model for concurrent systems. However, many decision problems have prohibitively high complexity. Therefore, it is sometimes useful to consider overapproximating semantics in which these problems can be decided more efficiently.

We study an overapproximation, called monus semantics, that slightly relaxes the semantics of decrements: A key property of a vector addition systems is that in order to decrement a counter, this counter must have a positive value. In contrast, our semantics allows decrements of zero-valued counters: If such a transition is executed, the counter just remains zero.

It turns out that if only a subset of transitions is used with monus semantics (and the others with classical semantics), then reachability is undecidable. However, we show that if monus semantics is used throughout, reachability remains decidable. In particular, we show that reachability for VASS with monus semantics is as hard as that of classical VASS (i.e. Ackermann-hard), while the zero-reachability and coverability are easier (i.e. EXPSPACE-complete and NP-complete, respectively). We provide a comprehensive account of the complexity of the general reachability problem, reachability of zero configurations, and coverability under monus semantics. We study these problems in general VASS, two-dimensional VASS, and one-dimensional VASS, with unary and binary counter updates.

2012 ACM Subject Classification Theory of computation → Concurrency

Keywords and phrases Vector addition systems, Overapproximation, Reachability, Coverability

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2023.10

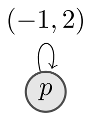
Related Version *Full Version*: https://arxiv.org/abs/2308.14926

Funding *Filip Mazowiecki*: Supported by the ERC grant INFSYS, agreement no. 950398.

Acknowledgements The authors are grateful to Wojciech Czerwiński and Sylvain Schmitz for discussions, and to Sylvain Schmitz for suggesting the term 'monus'.

1 Introduction

Vector addition systems with states (VASS) are an established model used in formal verification with a wide range of applications, *e.g.* in concurrent systems [22], business processes [39] and others (see the survey [37]). They are finite automata with transitions labeled by vectors over integers in some fixed dimension d. A configuration of a VASS consists of a pair $(p, \mathbf{v})$, denoted $p(\mathbf{v})$, where p is a state and $\mathbf{v}$ is a vector in $\mathbb{N}^d$. As a result of applying a transition labeled by some $\mathbf{z} \in \mathbb{Z}^d$, the vector in the resulting configuration is $\mathbf{v} + \mathbf{z}$. Thus in particular $\mathbf{v} + \mathbf{z} \geq \mathbf{0}$ must hold for the transition to be applicable. The latter requirement is often called the VASS semantics. To avoid ambiguity we will refer to it as the *classical VASS semantics*.



classical	$p(2,0) \rightarrow p(1,2) \rightarrow p(0,4) \not\rightarrow$
integer	$p(2,0) \xrightarrow[\mathbb{Z}]{} p(1,2) \xrightarrow[\mathbb{Z}]{} p(0,4) \xrightarrow[\mathbb{Z}]{} p(-1,6) \xrightarrow[\mathbb{Z}]{*} p(-n,4+2n)$
monus	$p(2,0) \Rightarrow p(1,2) \Rightarrow p(0,4) \Rightarrow p(0,6) \overset{*}{\Rightarrow} p(0,4+2n)$

Figure 1 A VASS in dimension 2 with one state p and one transition t. It has only one transition labeled with $(-1,2)$. We consider possible runs assuming that the initial configuration is $p(2,0)$. We use different notation for steps in each semantics: $\rightarrow$, $\xrightarrow[\mathbb{Z}]{}$, $\Rightarrow$. For the classical semantics ($\rightarrow$) after reaching the configuration $p(0,4)$ the transition can no longer be applied. For the integer semantics ($\xrightarrow[\mathbb{Z}]{*}$) the transition can be applied even in $p(0,4)$, reaching all configurations of the form $p(-n,4+2n)$. Similarly for the monus semantics ($\overset{*}{\Rightarrow}$), but there the configurations reachable from $p(0,4)$ are of the form $p(0,4+2n)$.

The VASS model is also studied with other semantics. One of the most natural variants of VASS semantics is the *integer semantics* (or simply *$\mathbb{Z}$-semantics*), where configurations are of the form $p(\mathbf{v})$, where $\mathbf{v} \in \mathbb{Z}^d$ [25]. There, a transition can always be applied, *i.e.* the resulting configuration is $\mathbf{v}+\mathbf{z}$ and we do not require $\mathbf{v}+\mathbf{z} \geq \mathbf{0}$. In this paper we consider VASS with the *monus semantics*, whose behavior partly resembles both classical and integer semantics. There, a transition can always be applied (as in $\mathbb{Z}$-semantics), however, if as a result the vector in the new configuration would have negative entries, then these are replaced with 0. Thus, vectors in configurations are over the naturals (as in classical semantics). The name monus semantics comes from the monus binary operator, which is a variant of the minus operator.[1] Note that every instance of a VASS can be considered with all three semantics. See Figure 1 for an example.

We study classical decision problems for VASS: reachability and coverability. The input for these problems is a VASS $\mathcal{V}$, an initial configuration $p(\mathbf{v})$, and a final configuration $q(\mathbf{w})$. The *reachability* problem asks whether there is a run from $p(\mathbf{v})$ to $q(\mathbf{w})$. A variant of this problem, called *zero reachability*, requires additionally that in the input the final vector is fixed to $\mathbf{w} = \mathbf{0}$. The *coverability* problem asks whether there is a run from $p(\mathbf{v})$ to $q(\mathbf{w}')$, where $\mathbf{w}' \geq \mathbf{w}$. Note that all three problems can be considered with respect to any of the three VASS semantics. As an example consider the VASS in Figure 1. Then for all three semantics $p(1,2)$ is both reachable and coverable from $p(2,0)$; and $p(0,2)$ is not reachable from $p(2,0)$ (but it is coverable as $(1,2) \geq (0,2)$).

Contribution I: Arbitrary dimension. Our first contribution is settling the complexities of reachability and coverability for VASS with the monus semantics (see Table 1). We prove that reachability is Ackermann-complete by showing that it is inter-reducible with classical VASS reachability, which is known to be Ackermann-complete [30, 9, 29]. This comes as a surprise, since in monus semantics, every transition can always be applied, just like in $\mathbb{Z}$-semantics, where reachability is merely NP-complete [25]. Thus, the monus operation encodes enough information in the resulting configuration that reachability remains extremely hard.

The Ackermann-hardness relies crucially on the fact that the final configuration is non-zero: We also show that the zero reachability problem is EXPSPACE-complete in monus semantics. This uses inter-reducibility with classical VASS coverability, which is EXPSPACE-complete

[1] One can also think that monus semantics is integer semantics, where after every step we apply the ReLU function.

due to seminal results of Lipton and Rackoff [33, 35]. The fact that zero-reachability is significantly easier than general reachability is in contrast to classical semantics, where zero reachability is interreducible with the reachability problem (intuitively, one can modify the input VASS by adding an extra edge that decrements by $\mathbf{w}$).

In another unexpected result, the complexity of coverability drops even more: We prove that it is NP-complete in monus semantics. We complete these results by showing that mixing classical and monus semantics (*i.e.* each transition is designated to either work in classical or monus semantics) makes reachability undecidable.

Contribution II: Fixed dimension. Understanding the complexity of reachability problems in VASS of fixed dimension has received a lot of attention in recent years and is now well understood. This motivates our second contribution: An almost complete complexity analysis of reachability, zero reachability and coverability for VASS with the monus semantics in dimensions 1 and 2. Here, the complexity depends on whether the counter updates are encoded in unary or binary (see Table 1).

We restrict our attention to dimensions 1 and 2 as most research in fixed dimension for the classical semantics. For the classical semantics not much is known about reachability in dimension $d \geq 3$. Essentially, the only known results consist of an upper bound of $\mathbf{F}_7$ that follows from the Ackermann upper bound in the general case [30], and a PSPACE-lower bound that holds already for $d = 2$ [5]. An intuition as to why the jump from 2 to 3 is so difficult is provided already by Hopcroft and Pansiot [27] who prove that the reachability set is always semilinear in dimension 2, and show an example that this is not the case in dimension 3. In contrast, coverability is well understood, and already Rackoff's construction [35] shows that for fixed dimension $d \geq 2$ coverability is in NL and in PSPACE, for unary and binary encoding, respectively (with matching lower bounds [5]).

Key technical ideas. The core insights of our paper are characterizations of the reachability and coverability relations in monus semantics, in terms of reachability and coverability in classical and $\mathbb{Z}$-semantics (Propositions 3.6 and 3.12 and Lemma 3.10). These allow us to apply a range of techniques to reduce reachability problems for one semantics into problems for other semantics, and thereby transfer existing complexity results. There are three cases where we were unable to ascertain the exact complexity: (i) reachability in 2-VASS with unary counter updates, (ii) zero reachability in 1-VASS with binary updates, and (iii) coverability in 1-VASS with binary counter updates. Concerning (i), this is because for 2-VASS with unary updates, it is known that classical reachability is NL-complete [5], but we would need to decide existence of a run that visits intermediate configurations of a certain shape. In the case of 2-VASS with binary updates, the methods from [5] (with a slight extension from [3]) allow this. The other cases, (ii) and (iii), are quite similar to each other. In particular, problem (ii) is logspace-interreducible with classical coverability in 1-VASS with binary updates, for which only an NL lower bound and an NC^2 upper bound are known [2].

Monus semantics as an overapproximation. Recall the example in Figure 1. Notice that every configuration reachable in the classical semantics is also reachable in the integer and monus semantics. It is not hard to see that this is true for every VASS model. Such semantics are called *overapproximations* of the classical VASS semantics. Overapproximations are a standard technique used in implementations of complex problems, in particular for the VASS model (see the survey [4]). They allow to prune the search space of reachable configurations, based on the observation that if a configuration is not reachable by an overapproximation then it cannot be reachable in the classical semantics. This is the core idea behind efficient implementations both of the coverability problem [15, 6] and the reachability problem [12, 7].

The two most popular overapproximations, integer semantics [25] and continuous semantics [20], behave similarly for both reachability and coverability problems, namely both problems are NP-complete. Note that all of the implementations mentioned above rely on such algorithms in NP as they can be efficiently implemented via SMT solvers. Interestingly, the monus semantics is an efficient overapproximation only for the coverability problem. (As far as we know this is the first study of a VASS overapproximation with this property.) Therefore, it seems to be a promising approach to try to speed up backward search algorithms using monus semantics (in the same vein as [6]). Whether this leads to improvements in practice remains to be seen in future work.

Related work. We discuss related work for VASS in classical semantics. A lot of research is dedicated to reachability for the flat VASS model, *i.e.* a model that does not allow for nested cycles in runs. In dimension 2 decision problems for VASS reduce to flat VASS, which is crucial to obtain the exact complexities [5]. It is known that in dimensions $d \geq 3$ such a reduction is not possible, but this raised natural questions of the complexity for flat VASS in higher dimensions [8, 10]. Another research direction is treating the counters in VASS models asymmetrically. For example, it is known that allowing for zero tests in VASS makes reachability and coverability undecidable (they essentially become Minsky machines). However, it was shown that if only one of the 2 counters is allowed to be zero tested then both reachability and coverability remain PSPACE-complete [31]. A different asymmetric question is when one counter is encoded in binary and the other is encoded in unary. Then recently it was shown that coverability is in NP [34] but it is unknown whether there is a matching lower bound. Finally, there are two important extensions of the VASS model: branching VASS (where runs are trees, not paths), and pushdown VASS (with one pushdown stack). For branching VASS, coverability is 2EXPTIME-complete [11]. The complexity of reachability is well understood in dimension 1 [23, 18] but in dimension 2 or higher it is unknown whether it is decidable. For pushdown VASS only coverability in dimension 1 is known to be decidable [32], otherwise decidability of both reachability and coverability remain open problems. Recently some progress was made on restricted pushdown VASS models [14, 21]. The monus semantics is a natural overapproximation that can be studied in all of these variants. Finally, let us mention that VASS with monus semantics fit into the very general framework of G-nets [13], but does not seem to fall into any of the decidable subclasses studied in [13]. However, if we equip VASS with with the usual well-quasi ordering on configurations, it is easy to see that even with monus semantics, they constitute well-structured transition systems (WSTS) [19, 1], which makes available various algorithmic techniques developed for WSTS.

Organization. In Section 2 we formally define the VASS model and the classical, integer and monus semantics. In Section 3 we prove the results in arbitrary dimension. Then in Section 4 and Section 5 we prove the results in dimension 2 and 1, respectively.

2 Vector addition systems with monus semantics: Main results

Given a vector $\mathbf{v} \in \mathbb{Z}^d$ we write $\mathbf{v}[i]$ for the value in the i-th coordinate, where $i \in \{1, \ldots, d\}$. We also refer to i as the i-th counter and write that it contains $\mathbf{v}[i]$ tokens. Given two vectors $\mathbf{v}$ and $\mathbf{v}'$ we write $\mathbf{v} \geq \mathbf{v}'$ if $\mathbf{v}[i] \geq \mathbf{v}'[i]$ for all $i = 1, \ldots, d$. By $\mathbf{0}^d$ we denote the zero vector in dimension d. We also simply write $\mathbf{0}$ if d is clear from context.

Vector addition systems with states. A vector addition system with states (VASS) is a triple $\mathcal{V} = (d, Q, \Delta)$, where $d \in \mathbb{N}$, Q is a finite set of states and $\Delta \subseteq Q \times \mathbb{Z}^d \times Q$ is a finite set of transitions. Throughout the paper we fix a VASS $\mathcal{V} = (d, Q, \Delta)$.

We start with the formal definitions in the *classical semantics.* A configuration of a VASS is a pair $p(\mathbf{v}) \in Q \times \mathbb{N}^d$, denoted $p(\mathbf{v})$. Any transition $t \in \Delta$ induces a successor (partial) function $\mathsf{Succ}_t : Q \times \mathbb{N}^d \to Q \times \mathbb{N}^d$ such that $\mathsf{Succ}_t(q(\mathbf{v})) = q'(\mathbf{v}')$ iff $t = (q, \mathbf{z}, q')$ and $\mathbf{v}' = \mathbf{v} + \mathbf{z}$. This successor function can be lifted up to Δ to get a step relation $\rightarrow_{\mathcal{V}}$, such that any pair of configuration $C \rightarrow_{\mathcal{V}} C'$ iff there exists $t \in \Delta$ with $\mathsf{Succ}_t(C) = C'$. A *run* is a sequence of configurations

$$q_0(\mathbf{v}_0), q_1(\mathbf{v}_1), q_2(\mathbf{v}_2), \ldots, q_k(\mathbf{v}_k)$$

such that for every $0 < j \leq k$, $q_{j-1}(\mathbf{v}_{j-1}) \rightarrow_{V} q_j(\mathbf{v}_j)$. If there exists such a run we say that $q_k(\mathbf{v}_k)$ is reachable from $q_0(\mathbf{v}_0)$ and denote it $C_0 \xrightarrow{*}_{\mathcal{V}} C_k$. We call $\xrightarrow{*}_{\mathcal{V}}$ the reachability relation in the classical VASS semantics.

In this paper we consider two additional semantics. The first is called the *integer semantics* (or $\mathbb{Z}$*-semantics*). A configuration in this semantics is a pair $p(\mathbf{v}) \in Q \times \mathbb{Z}^d$ (hence, values of vector coordinates can drop below zero). The definitions of successor function, step relation and run are analogous as for the classical semantics. By $\underset{\mathbb{Z}}{\rightarrow}_{\mathcal{V}}$ and $\underset{\mathbb{Z}}{\xrightarrow{*}}_{\mathcal{V}}$, we denote the step and reachability relations in the $\mathbb{Z}$-semantics, respectively.

The second is called *monus semantics.* The configurations are the same as in the classical semantics. The difference is in the successor function. Every transition $t \in \Delta$ induces a successor function $\mathsf{Succ}_t : Q \times \mathbb{N}^d \to Q \times \mathbb{N}^d$ as follows: $\mathsf{Succ}_t(q(\mathbf{v})) = q'(\mathbf{v}')$ iff $t = (q, \mathbf{z}, q')$ and for all $j \in \{1, 2, \ldots d\}$, $\mathbf{v}'[j] = \max(\mathbf{v}[j] + \mathbf{z}[j], 0)$. We write in short $\mathbf{v}' = \max(\mathbf{v} + \mathbf{z}, \mathbf{0})$. Step relation and runs are defined analogously as in the case of classical semantics. By $\Rightarrow_{\mathcal{V}}$ and $\overset{*}{\Rightarrow}_{\mathcal{V}}$, we denote the step and reachability relations in the monus semantics, respectively.

We drop the subscript $\mathcal{V}$ from the above relations when the VASS is clear from context. We write that a run is a *classical run*, a $\mathbb{Z}$ *run* or a *monus run* to emphasize the considered semantics. An example highlighting the differences between the three semantics is in Figure 1.

Decision problems. We study the following decision problems for VASS.

The classical *reachability problem*:

Given A VASS $\mathcal{V} = (d, Q, \Delta)$ and two configurations $p(\mathbf{v})$ and $q(\mathbf{w})$.

Question Does $p(\mathbf{v}) \overset{*}{\Rightarrow} q(\mathbf{w})$ hold?

The classical *zero reachability problem*:

Given A VASS $\mathcal{V} = (d, Q, \Delta)$, a configuration $p(\mathbf{v})$ and a state q.

Question Does $(p, \mathbf{v}) \overset{*}{\Rightarrow} q(\mathbf{0}^d)$ hold?

The classical *coverability problem*:

Given A VASS $\mathcal{V} = (d, Q, \Delta)$ and two configurations $p(\mathbf{v})$ and $q(\mathbf{w})$.

Question Does $p(\mathbf{v}) \overset{*}{\Rightarrow} q(\mathbf{w}')$ hold for some $\mathbf{w}' \geq \mathbf{w}$?

Similarly, the above problems in $\mathbb{Z}$ and classical semantics are defined by replacing $\overset{*}{\Rightarrow}$ with $\underset{\mathbb{Z}}{\xrightarrow{*}}$ and $\xrightarrow{*}$, respectively.

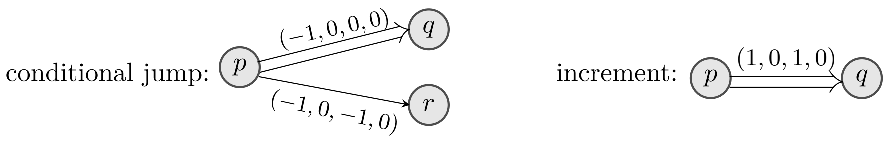


Figure 2 Two gadgets for realizing a zero-testable counter.

Main results. The main complexity results of this work are summarized in Table 1. In Table 2, we recall complexity results for VASS with classical semantics for comparison. We do not split the cases of unary and binary encoding for arbitrary dimensions, since there all lower bounds work for unary, whereas all upper bounds work for binary.

Concerning the *reachability problem*, we note that in all cases where we obtain the exact complexity, it is the same as for the classical VASS semantics. For the other decision problems, there are stark differences: First, while in the classical semantics, *zero reachability* is easily inter-reducible with general reachability, in the monus semantics, its complexity drops in two cases: In 1-VASS with binary counter updates, monus zero reachability is in NC^2 (thus polynomial time), compared to NP in the classical setting. Moreover, in arbitrary dimension, monus zero reachability is EXPSPACE-complete, compared to Ackermann in the classical semantics. For the *coverability problem*, the monus semantics also lowers the complexity in two cases: For binary encoded 2-VASS (NP in monus semantics, PSPACE in classical) and in the general case (NP in monus semantics, EXPSPACE in classical semantics).

Undecidability. To stress the subtle effects of monus semantics, we mention that it leads to undecidability if combined with classical semantics: If one can specify the applied semantics (classical vs. monus) for each transition, then (zero) reachability becomes undecidable.

We sketch the proof using Figure 2. It shows two gadgets, where "$\rightarrow$" transitions use classical semantics and "$\Rightarrow$" transitions use monus semantics. The two gadgets realize a counter with zero test: The left gadget is a conditional jump ("if zero, then go to q, otherwise decrement and go to r"), whereas the right gadget is just an increment. In intended runs (i.e. where the left gadget always takes the intended transition), the counter value is stored both in components 1 and 3. (To realize a full two-counter machine, the same gadgets on components 2 and 4 realize the other testable counter.) Thus, initially, all components are zero. Note that if the left gadget always takes the transitions as intended, then the first and third counter will remain equal. If the gadget takes the upper transition when the counter is not actually zero, then the first counter becomes smaller than the third, and will then always stay smaller. Hence, to reach $(0, 0, 0, 0)$, the left gadget must always behave as intended.

However, coverability remains decidable if we can specify the semantics of each transition. Indeed, suppose we order the configurations of a VASS by the usual well-quasi ordering (i.e. the control states have to agree, and the counter values are ordered component-wise). Then it is easy to see that this results in a well-structured transition system (WSTS) [19, 1]. This also implies, e.g. that termination is decidable in this general setting.

3 Arbitrary dimension

In this section, we prove the complexity results concerning VASS with arbitrary dimension. This will include the characterizations of monus reachability, monus zero reachability, and monus coverability in terms of classical and $\mathbb{Z}$-semantics. We begin with some terminology.

Table 1 Complexity results shown in this work.

Dimension & encoding	Monus Reachability	Monus zero reachability	Monus coverability
1-dim, unary	NL-complete	NL-complete	NL-complete
1-dim, binary	NP-complete	in NC^2	in NC^2
2-dim, unary	in PSPACE	NL-complete	NL-complete
2-dim, binary	PSPACE-complete	PSPACE-complete	NP-complete
arbitrary	Ack-complete	EXPSPACE-complete	NP-complete

Table 2 Known complexities for classical VASS semantics, for comparison.

Dimension & encoding	Reachability	Zero reachability	Coverability
1-dim, unary	NL-complete [38]	NL-complete [38]	NL-complete [38]
1-dim, binary	NP-complete [26]	NP-complete [26]	in NC^2 [2]
2-dim, unary	NL-complete [5]	NL-complete [5]	NL-complete [36]
2-dim, binary	PSPACE-complete [5]	PSPACE-complete [5]	PSPACE-complete [5, 36, 16]
arbitrary	Ack-compl. [30, 29, 9]	Ack-compl. [30, 29, 9]	EXPSPACE-compl. [33, 35]

Paths. A sequence of transitions $(p_1, \mathbf{z}_1, q_1), \ldots, (p_k, \mathbf{z}_k, q_k)$ is valid iff $q_i = p_{i+1}$ for every $1 \le i < k-1$. Furthermore, we say that it is valid from a given configuration $(p, \mathbf{v})$ if $p = p_0$. We call a valid sequence of transitions a *path*.

Given two paths ρ_1 and ρ_2 if the last state of ρ_1 is equal to the first state of ρ_2 then by $\rho = \rho_1\rho_2$ we denote the path defined as the sequence ρ_1 followed by the sequence ρ_2. Similarly, we use this notation with more paths, *e.g.* $\rho = \rho_1\rho_2 \ldots \rho_k$ means that the path ρ is composed from k paths: $\rho_1, \ldots \rho_k$.

Fix a path $\rho = (p_0, \mathbf{z}_0, p_1), \ldots, (p_{k-1}, \mathbf{z}_{k-1}, p_k)$. We say that $\mathbf{z} = \sum_{i=0}^{k-1} \mathbf{z}_i$ is the effect of the path ρ. Notice that while for classical and $\mathbb{Z}$-semantics the effect of a path can be computed by subtracting the vectors in the last and first configurations, this is not necessarily true for monus semantics. In Figure 1 consider the path $\rho = t, t, t$. The effect is $(-3, 6)$. In the $\mathbb{Z}$-semantics $(2, 0) \xrightarrow[\mathbb{Z}]{*} (-1, 6)$ and the difference $(-1, 6) - (2, 0)$ is precisely the effect of ρ. In the monus semantics it is not the case as $(2, 0) \xRightarrow{*} (0, 6)$. This is because a run in monus semantics can lose some decrements, unlike in classical and $\mathbb{Z}$-semantics.

▶ Remark 3.1. Observe that every classical and $\mathbb{Z}$ run defines a unique path from the initial configuration. For monus semantics uniqueness is not guaranteed as it is possible that a run induces more than one path. Indeed, suppose $p(2, 0) \Rightarrow q(1, 0)$. This could be realised by any transition of the form $(p, (-1, z), q)$, where $z \le 0$. Conversely, a path induces a unique run for $\mathbb{Z}$ and monus semantics. Formally, consider a path $(p_0, \mathbf{z}_1, p_1), \ldots, (p_{k-1}, \mathbf{z}_k, p_k)$ from a configuration $s(\mathbf{v})$. Then, in the $\mathbb{Z}$ and monus semantics there exists a unique corresponding run. In the classical semantics a path might be blocked if a counter drops below zero (see *e.g.* Figure 1). We write $p_0(\mathbf{v}_0) \xrightarrow{\rho} p_k(\mathbf{v}_k)$, $p_0(\mathbf{v}_0) \xrightarrow[\mathbb{Z}]{\rho} p_k(\mathbf{v}_k)$ and $p_0(\mathbf{v}_0) \xRightarrow{\rho} p_k(\mathbf{v}_k)$ if $p_0(\mathbf{v}_0), \ldots, p_k(\mathbf{v}_k)$ is a run in classical, integer and monus semantics, respectively. Recall that for classical and $\mathbb{Z}$-semantics $\mathbf{v}_{i+1} - \mathbf{v}_i = \mathbf{z}_i$, and for monus semantics $\mathbf{v}_{i+1} = \max(\mathbf{v}_i + \mathbf{z}_i, \mathbf{0})$.

Consider a run $R = p_0(\mathbf{v}_0), \ldots, p_k(\mathbf{v}_k)$ (in any semantics). We say that the counter $j \in \{1, \cdots, d\}$ hits 0 iff $\mathbf{v}_i[j] = 0$ for some $1 \le i \le k$. Similarly, we say that the counter $j \in \{1, \cdots, d\}$ goes negative in R iff $\mathbf{v}_i[j] < 0$ for some $0 \le i \le k$ (this can happen only in the $\mathbb{Z}$-semantics).

Let $\rho = (p_0, \mathbf{z}_0, p_1) \ldots (p_{k-1}, \mathbf{z}_{k-1}, p_k)$ be a path such that R is the unique run corresponding to ρ from the initial configuration $p_0(\mathbf{v}_0)$. We say that (ρ, R) or $p_0(\mathbf{v}_0) \overset{\rho}{\Rrightarrow} p_k(\mathbf{v}_k)$ is lossy for the counter $j \in \{1, \cdots, d\}$ iff $\mathbf{v}_i[j] - \mathbf{v}_{i-1}[j] \neq \mathbf{z}_{i-1}[j]$ for some $1 \leq i \leq k$ (a lossy run can happen only in the monus semantics).

▶ Remark 3.2. Integer and monus semantics are overapproximations of the classical semantics. That is, $s(\mathbf{v}) \xrightarrow{\rho} t(\mathbf{w})$ implies $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{\rho} t(\mathbf{w})$ and $s(\mathbf{v}) \overset{\rho}{\Rrightarrow} t(\mathbf{w})$. The converse is not always the case (see Figure 1). Moreover, $s(\mathbf{v}) \overset{\rho}{\Rrightarrow} t(\mathbf{w})$ implies $s(\mathbf{v}) \xrightarrow{\rho} t(\mathbf{w})$ if $s(\mathbf{v}) \overset{\rho}{\Rrightarrow} t(\mathbf{w})$ is not lossy. Notice that if in $s(\mathbf{v}) \overset{\rho}{\Rrightarrow} t(\mathbf{w})$, none of the counters $j \in \{1, \ldots, d\}$ hits 0 then it is not a lossy run. Similarly, $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{\rho} t(\mathbf{w})$ implies $s(\mathbf{v}) \xrightarrow{\rho} t(\mathbf{w})$ if, in the former run, none of the counters $j \in \{1, \ldots, d\}$ goes negative.

Characterizing Monus Reachability. Our first goal is to characterize the reachability problem for the monus semantics in terms of the classical semantics. We start with some propositions that relate monus runs to $\mathbb{Z}$ runs and classical runs. Let ρ be a path and $s_0(\mathbf{v}_0)$ a configuration. Let $s_0(\mathbf{v}_0) \ldots s_k(\mathbf{v}_k)$ be the unique $\mathbb{Z}$ run defined by ρ and $s_0(\mathbf{v}_0)$. We define the vector $\mathbf{m} = \min_{\mathbb{Z}}(\rho, s_0, \mathbf{v}_0)$ by $\mathbf{m}[i] = \min(\min_{j=0}^{k} \mathbf{v}_j[i], 0)$. Intuitively, it is the vector of minimal values in the $\mathbb{Z}$ run, but note that $\mathbf{m} \leq \mathbf{0}$.

For the next two propositions we fix a configuration $s_0(\mathbf{v}_0) \in Q \times \mathbb{N}^d$, a path $\rho = (s_0, \mathbf{z}_0, s_1) \ldots (s_{k-1}, \mathbf{z}_{k-1}, s_k)$, and $\mathbf{m} = \min_{\mathbb{Z}}(\rho, s_0, \mathbf{v}_0)$.

▶ **Proposition 3.3.** *Consider the unique runs induced by ρ from $s_0(\mathbf{v}_0)$ in $\mathbb{Z}$-semantics*

$$s_0(\mathbf{v}_0), \ldots, s_{k-1}(\mathbf{v}_{k-1}), s_k(\mathbf{v}_k),$$

and in monus semantics

$$s_0(\mathbf{v}'_0), \ldots, s_{k-1}(\mathbf{v}'_{k-1}), s_k(\mathbf{v}'_k).$$

where $\mathbf{v}'_0 = \mathbf{v}_0$. Then $\mathbf{v}'_k = \mathbf{v}_k - \mathbf{m}$.

Proof (sketch). We analyse the behavior of every counter j. Recall that the $\mathbb{Z}$ run and the monus run have the same value in the counter j until the first time the value of j becomes negative in the $\mathbb{Z}$ run. We denote this as $\mathbf{v}_i[j] = -u$. Note that $\mathbf{v}'_i[j] = 0$. Hence, $\mathbf{v}_i[j] - \mathbf{v}'_i[j] = -u$. It is not hard to see that every time the value of the counter j reaches a new minimum in the $\mathbb{Z}$-semantics, the difference $\mathbf{v}'_i[j] - \mathbf{v}_i[j]$ will be equal to it. We prove this formally by induction on k. Refer to full version for the formal proof. ◀

▶ Remark 3.4. Let $\mathbf{z} \in \mathbb{Z}^d$. A sequence of configurations $s_0(\mathbf{v}_0) \ldots s_k(\mathbf{v}_k)$ is a run in $\mathbb{Z}$-semantics corresponding to a path ρ iff $s_0(\mathbf{v}_0 - \mathbf{z}) \ldots s_k(\mathbf{v}_k - \mathbf{z})$ is a run in $\mathbb{Z}$-semantics on the same path ρ.

▶ **Proposition 3.5.** *Consider the following unique run corresponding to the path ρ from $s_0(\mathbf{v}_0)$ in the monus semantics*

$$s_0(\mathbf{v}_0), \ldots, s_{k-1}(\mathbf{v}_{k-1}), s_k(\mathbf{v}_k).$$

Then the following run, induced by ρ, exists in the classical semantics

$$s_0(\mathbf{v}'_0), \ldots, s_{k-1}(\mathbf{v}'_{k-1}), s_k(\mathbf{v}'_k).$$

where $\mathbf{v}'_0 = \mathbf{v}_0 - \mathbf{m}$ and $\mathbf{v}'_k = \mathbf{v}_k$.

Proof. This essentially follows from the definition of $\mathbf{m}$ and Remark 3.4. One just needs to observe that the $\mathbb{Z}$ run with configurations shifted by the vector $-\mathbf{m}$ does not go below zero, hence it is a classical run. See full version for the formal proof. ◀

We now characterize monus reachability in terms of classical reachability.

▶ **Proposition 3.6.** *Let $\mathcal{V} = (d, Q, \Delta)$ be a VASS, let $s(\mathbf{v})$ and $t(\mathbf{w})$ be configurations of $\mathcal{V}$, and let ρ be a path of $\mathcal{V}$. Then, $s(\mathbf{v}) \overset{\rho}{\Rightarrow} t(\mathbf{w})$ if and only if there is a subset $Z \subseteq \{1, \ldots, d\}$ and a vector $\mathbf{v}' \geq \mathbf{v}$ such that*
1. *$s(\mathbf{v}') \xrightarrow{\rho} t(\mathbf{w})$,*
2. *For every $z \in Z$, the coordinate z hits 0 in $s(\mathbf{v}') \xrightarrow{\rho} t(\mathbf{w})$,*
3. *For every $j \in \{1, \ldots, d\} \setminus Z$, we have $\mathbf{v}'[j] = \mathbf{v}[j]$.*

Proof. ($\implies$) Let $\mathbf{m} = \min_{\mathbb{Z}}(\rho, s, \mathbf{v})$. This direction is implied by Proposition 3.5 along with the following argument. Every counter $j \in \{1, \ldots, d\}$ hits 0 in $s(\mathbf{v}) \overset{\rho}{\Rightarrow} t(\mathbf{w})$ if and only if it hits 0 in $s(\mathbf{v} - \mathbf{m}) \xrightarrow{\rho} t(\mathbf{w})$. Moreover, if j does not hit 0 in $s(\mathbf{v}) \overset{\rho}{\Rightarrow} t(\mathbf{w})$ then $\mathbf{m}[j] = 0$.

($\impliedby$) Let $\mathbf{v}' \geq \mathbf{v}$ be a vector as in the statement and let $s(\mathbf{v}') \xrightarrow{\rho} t(\mathbf{w})$. We define $Z \subseteq \{1 \ldots d\}$ such that $i \in Z$ if it hits 0. Moreover, let $s(\mathbf{v}) \overset{\rho}{\Rightarrow} t(\mathbf{w}'')$. It suffices to show that $\mathbf{w} = \mathbf{w}''$. We write $s(\mathbf{v}') = p_0(\mathbf{v}'_0) \ldots p_k(\mathbf{v}'_k) = t(\mathbf{w})$ and $s(\mathbf{v}) = p_0(\mathbf{v}_0) \ldots p_k(\mathbf{v}_k) = t(\mathbf{w}'')$ for the corresponding runs in the classical and monus semantics, respectively. Note that $\mathbf{v}' \geq \mathbf{v}$ implies $\mathbf{v}'_i \geq \mathbf{v}_i$ for all $0 \leq i \leq k$. By definition of $\mathbf{v}'$ it suffices to consider counters j that hit zero, *i.e.* $\mathbf{v}'_i[j] = 0$ for some $0 \leq i \leq k$. Since $\mathbf{v}'_i \geq \mathbf{v}_i$ we get $\mathbf{v}'_i[j] = 0 = \mathbf{v}_i[j]$. Hence, from i onward both runs agree on the value in counter j. Thus $\mathbf{w} = \mathbf{w}''$. ◀

The reachability problem. We begin with the Ackermann-completeness proof.

▶ **Theorem 3.7.** *Reachability in monus semantics is Ackermann-complete.*

For the upper bound we show how to reduce reachability in monus semantics to reachability in classical semantics. Let $\mathcal{V} = (d, Q, \Delta)$, $s(\mathbf{v})$, and $t(\mathbf{w})$ be the input of the reachability problem in monus semantics. We rely on Proposition 3.6. Intuitively, we have to guess a subset $Z \subseteq \{1, \ldots, d\}$ and a permutation $\sigma \colon [1, k] \to Z$ (where $k = |Z|$). Then we check whether there exists a run as described in Proposition 3.6 with $z_i = \sigma(i)$ for $i \in [1, k]$. To detect the latter run, we construct the VASS $\mathcal{V}_\sigma = (d + k, Q', T')$ as follows. It simulates $\mathcal{V}$, but it has k extra counters to freeze the values of the counter in Z at the points where the coordinates $\sigma(k), \ldots, \sigma(1)$ hit 0 as mentioned in Proposition 3.6.

To remember which counters have already been frozen the set of control states is $Q' = \{q_i \mid q \in Q,\ i \in [0, k]\}$. Intuitively, the index $i \in [0, k]$ stores the information how many counters are frozen. The index i can only increment. Note that guessing the permutation σ allows us to assume that we know the order in which the counters are frozen.

Since we deal with vectors in dimension d and $d + k$ we introduce some helpful notation. We write $\mathbf{e}_j \in \mathbb{Z}^d$ for the unity vector with $\mathbf{e}_j[j] = 1$ and with 0 on other coordinates. Given a vector $\mathbf{z} \in \mathbb{Z}^d$ we define $\mathsf{copy}(\mathbf{z}) \in \mathbb{Z}^{d+k}$ as $\mathsf{copy}(\mathbf{z})[j] = \mathbf{z}[j]$ for $1 \leq j \leq d$ and $\mathsf{copy}(\mathbf{z})[j] = \mathbf{z}[\sigma(j - d)]$ for $d < j \leq d + k$. Intuitively, it simply copies the behaviors of the corresponding counters. We generalise this notation to allow to also remove the effect on some coordinates (*i.e.* "freeze" them). Given $\mathbf{z} \in \mathbb{Z}^d$ and $0 \leq i \leq k$ we define $\mathsf{copy}_i(\mathbf{z}) \in \mathbb{Z}^{d+k}$ as $\mathsf{copy}_i(\mathbf{z})[j] = \mathsf{copy}(\mathbf{z})[j]$ for $1 \leq j \leq d + k - i$ and $\mathsf{copy}_i(\mathbf{z})[j] = 0$ for $d + k - i < j \leq d + k$. In particular $\mathsf{copy}_0(\mathbf{z}) = \mathsf{copy}(\mathbf{z})$ and $\mathsf{copy}_i(\mathbf{z})$ is 0 in the last i counters.

It remains to define the set of transitions T'. In the beginning there are transitions in T' that can arbitrarily increment each counter that belongs to Z and its extra copy: $(s_0, \mathsf{copy}(\mathbf{e}_j), s_0) \in T'$ for every $j \in Z$. Moreover, the counter in the control state can spontaneously be incremented: $(p_i, \mathbf{0}, p_{i+1})$ for every $p \in Q$ and $0 \leq i < k$. For every transition $(p, \mathbf{z}, q) \in T$ and $0 \leq i \leq k$ we define $(p_i, \mathsf{copy}_i(\mathbf{z}), q_i) \in T'$.

The following claim is straightforward by Proposition 3.6:

▷ Claim 3.8. We have $s(\mathbf{v}) \overset{*}{\Rightarrow}_{\mathcal{V}} t(\mathbf{w})$ if and only if there exists a subset $Z \subseteq \{1, \ldots, d\}$ and bijection $\sigma\colon [1,k] \to Z$ such that $s_0(\mathsf{copy}_0(\mathbf{v})) \xrightarrow{*}_{\mathcal{V}_\sigma} t_k(\mathsf{copy}_k(\mathbf{w}))$.

This implies that we can decide monus reachability by guessing a subset $Z \subseteq [1,d]$, guessing a bijection $\sigma\colon [1,k] \to Z$, and deciding reachability in $\mathcal{V}_\sigma$. This yields the upper bound.

For the lower bound we reduce classical reachability to monus reachability. Let $\mathcal{V} = (d, Q, \Delta)$, $s(\mathbf{0})$ and $t(\mathbf{0})$ be the input of the reachability problem in classical semantics (without loss of generality the input vectors can be $\mathbf{0}$). We construct the VASS $\mathcal{V}' = (d+2, Q', T')$ as follows. The states are $Q' = Q \cup \{t'\}$, where t' is a fresh copy of t.

Again to deal with vectors in different dimension we introduce the following notation. Given $\mathbf{z} \in \mathbb{Z}^d$ we write $\Delta(\mathbf{z}) \in \mathbb{Z}$ for $\Delta(\mathbf{z}) = \sum_{j=1}^{d} \mathbf{z}[j]$, *i.e.* the sum of all components. Based on this we define $\mathsf{extend}(\mathbf{z}) \in \mathbb{Z}^{d+2}$ as: $\mathsf{extend}(\mathbf{z}) = (z, \Delta(z), 0)$ if $\Delta(\mathbf{z}) \geq 0$, and $\mathsf{extend}(\mathbf{z}) = (z, 0, -\Delta(z))$ otherwise.

We define T' as follows. For every $(p, \mathbf{z}, q) \in T$: $(p, \mathsf{extend}(\mathbf{z}), q) \in T'$. Thus, in the $(d+1)$-th counter, we collect the sum of all non-negative entry sums of the added vectors. Analogously, in the $(d+2)$-th counter, we collect the sum of all negative entry sums (with a flipped sign). We also add the transition $(t, \mathbf{0}, t') \in T'$, and a "count down" loop: $(t'(\mathbf{0}, -1, -1), t')$, where $(\mathbf{0}, -1, -1)$ is 0 in the first d components and -1 otherwise. The following claim completes the proof of Ackermann-hardness.

▷ Claim 3.9. We have $s(\mathbf{0}, 1, 1) \overset{*}{\Rightarrow} t'(\mathbf{0}, 1, 1)$ in $\mathcal{V}'$ if and only if $s(\mathbf{0}) \xrightarrow{*} t(\mathbf{0})$ in $\mathcal{V}$.

Proof. ($\Longleftarrow$) This is obvious, because every run in classical semantics yields a run in monus semantics between the same configurations.

($\Longrightarrow$) Suppose there is a monus run from $s(\mathbf{0}, 1, 1)$ to $t'(\mathbf{0}, 1, 1)$. Then for some $m \in \mathbb{N}$, there is a transition sequence ρ leading in monus semantics from $s(\mathbf{0}, 1, 1)$ to $t(\mathbf{0}, m, m)$. Now let us execute ρ in $\mathbb{Z}$-semantics. This execution will arrive at some configuration $t(\mathbf{v}, m, m)$ (note that the last two counters are never decreased, except for the final loop). We shall prove that (i) $\mathbf{v} = \mathbf{0}$ and (ii) this execution never drops below zero. First, according to Proposition 3.3, the resulting counter values in monus semantics are always at least the values from $\mathbb{Z}$-semantics. This implies $\mathbf{v} \leq \mathbf{0}$. Next observe that since the right-most components have the same value m, the total sum of all entry sums of added vectors (in the first d entries) must be zero. Thus, $\Delta(\mathbf{v}) = 0$. Together with $\mathbf{v} \leq \mathbf{0}$, this implies $\mathbf{v} = \mathbf{0}$, which shows (i). Second, if the execution in $\mathbb{Z}$-semantics ever drops below zero in some counter i, then by Proposition 3.3 and the fact that in $\mathbb{Z}$-semantics we reach $\mathbf{v} = \mathbf{0}$, this would imply that ρ in monus semantics ends up in a strictly positive value in counter i, which is not true. This shows (ii). Hence, we have shown that the run in $\mathbb{Z}$-semantics is actually a run in classical VASS semantics. Therefore, $s(\mathbf{0}) \xrightarrow{*} t(\mathbf{0})$ in $\mathcal{V}$. ◁

Characterizing zero-reachability. Monus zero-reachability has a simple characterization in terms of classical coverability. Here, $\mathcal{V}^{\mathsf{rev}}$ is obtained by reversing all transitions in $\mathcal{V}$ and their effects. Formally, there is a transition $(p, \mathbf{z}, q)$ in $\mathcal{V}^{\mathsf{rev}}$ iff there is a transition $(q, -\mathbf{z}, p)$ in $\mathcal{V}$.

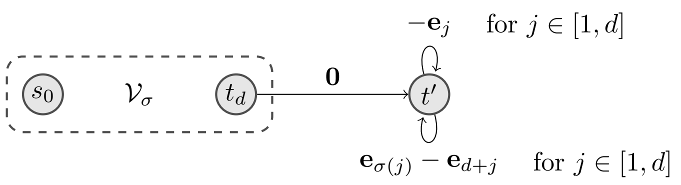

$-\mathbf{e}_j$ for $j \in [1,d]$
s_0
$\mathcal{V}_\sigma$
t_d
$\mathbf{0}$
t'
$\mathbf{e}_{\sigma(j)} - \mathbf{e}_{d+j}$ for $j \in [1,d]$

Figure 3 Construction of $\mathcal{V}'_\sigma$ in reduction from monus coverability to reachability in $\mathbb{Z}$-semantics.

▶ **Lemma 3.10.** *For any $\mathbf{v}$, we have $s(\mathbf{v}) \overset{*}{\Rightarrow}_{\mathcal{V}} t(\mathbf{0})$ iff $t(\mathbf{0}) \overset{*}{\rightarrow}_{\mathcal{V}^{\text{rev}}} s(\mathbf{v}')$ for some $\mathbf{v}' \geq \mathbf{v}$.*

Proof. By Proposition 3.6, $s(\mathbf{v}) \overset{*}{\Rightarrow} t(\mathbf{0})$ yields a $\mathbf{v}' \geq \mathbf{v}$ with $s(\mathbf{v}') \overset{*}{\rightarrow} t(\mathbf{0})$. Conversely, if $s(\mathbf{v}') \overset{*}{\rightarrow} t(\mathbf{0})$, then we can pick $Z = [1,d]$ in Proposition 3.6 to obtain $s(\mathbf{v}) \overset{*}{\Rightarrow} t(\mathbf{0})$. ◀

This together with the known complexity of classical coverability [33, 35] immediately implies:

▶ **Proposition 3.11.** *The monus zero-reachability problem is* EXPSPACE*-complete.*

Characterizing coverability. Our third characterization describes coverability in monus semantics in terms of reachability in $\mathbb{Z}$-semantics:

▶ **Proposition 3.12.** *Let $\mathcal{V} = (d, Q, \Delta)$ be a VASS and let $s(\mathbf{v})$ and $t(\mathbf{w})$ be configurations. Then $s(\mathbf{v}) \overset{*}{\Rightarrow} t(\mathbf{w}'')$ for some $\mathbf{w}'' \geq \mathbf{w}$ if and only if there is a permutation σ of $\{1, \ldots, d\}$ and $\mathbb{Z}$-configurations $p_d(\mathbf{v}_d), \ldots, p_1(\mathbf{v}_1)$, $t(\mathbf{w}')$ so that*
1. *$s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{*} p_d(\mathbf{v}_d) \xrightarrow[\mathbb{Z}]{*} p_{d-1}(\mathbf{v}_{d-1}) \xrightarrow[\mathbb{Z}]{*} \cdots \xrightarrow[\mathbb{Z}]{*} p_1(\mathbf{v}_1) \xrightarrow[\mathbb{Z}]{*} t(\mathbf{w}')$,*
2. *for each $j \in \{1, \ldots, d\}$, we have $\mathbf{w}'[j] + |\min(\mathbf{v}_{\sigma^{-1}(j)}[j], 0)| \geq \mathbf{w}[j]$.*

Proof. ($\implies$) Let ρ be any path such that $s(\mathbf{v}) \overset{\rho}{\Rightarrow} t(\mathbf{w}'')$ and $\mathbf{w}'' \geq \mathbf{w}$. Then, by Proposition 3.3 $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{\rho} t(\mathbf{w}'' + \mathbf{m})$, where $\mathbf{m}$ is the vector of minimum values in the $\mathbb{Z}$ run. The required permutation σ represents the order $\sigma(d), \ldots, \sigma(1)$ in which these coordinates reach their corresponding minimum values. Hence, $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{\rho} t(\mathbf{w}'' + \mathbf{m})$ is the same as $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{*} p_d(\mathbf{v}_d) \xrightarrow[\mathbb{Z}]{*} p_{d-1}(\mathbf{v}_{d-1}) \xrightarrow[\mathbb{Z}]{*} \cdots \xrightarrow[\mathbb{Z}]{*} p_1(\mathbf{v}_1) \xrightarrow[\mathbb{Z}]{*} t(\mathbf{w}')$, such that $\mathbf{v}_d[\sigma(d)] = \mathbf{m}[\sigma(d)], \ldots, \mathbf{v}_1[\sigma(1)] = \mathbf{m}[\sigma(1)]$, and $\mathbf{w}''[j] = \mathbf{w}'[j] - \mathbf{m}[j] = \mathbf{w}'[j] + |\mathbf{m}[j]| = \mathbf{w}'[j] + |\min(\mathbf{v}_{\sigma^{-1}(j)}[j], 0)|$ for all $1 \leq j \leq d$. As $\mathbf{w}'' \geq \mathbf{w}$, $\mathbf{w}'[j] + |\min(\mathbf{v}_{\sigma^{-1}(j)}[j], 0)| \geq \mathbf{w}[j]$ for all $1 \leq j \leq d$.

($\impliedby$) This is a direct consequence of Proposition 3.3. It implies that given any permutation σ on $\{1, \ldots, d\}$ and any run $s(\mathbf{v}) \xrightarrow[\mathbb{Z}]{*} p_d(\mathbf{v}_d) \xrightarrow[\mathbb{Z}]{*} p_{d-1}(\mathbf{v}_{d-1}) \xrightarrow[\mathbb{Z}]{*} \cdots \xrightarrow[\mathbb{Z}]{*} p_1(\mathbf{v}_1) \xrightarrow[\mathbb{Z}]{*} t(\mathbf{w}')$ such that $\mathbf{w}'[j] - \min(\mathbf{v}_{\sigma^{-1}(j)}[j], 0) \geq \mathbf{w}[j]$, there is a run from configuration $s(\mathbf{v})$ and reaching a configuration $t(\mathbf{w}'')$ where $\mathbf{w}''[j] = \mathbf{w}'[j] - \mathbf{m}[j] \geq \mathbf{w}'[j] - \min(\mathbf{v}_{\sigma^{-1}(j)}[j], 0) \geq \mathbf{w}[j]$ for all $1 \leq j \leq d$. ◀

We conclude the following.

▶ **Proposition 3.13.** *Monus coverability is* NP*-complete.*

Proof. First we show NP-hardness. In [28, Prop. 5.11], it is shown that it is NP-hard to decide whether a regular language over some alphabet Σ, given as an NFA, contains a word in which every letter appears exactly once. Given such an NFA $\mathcal{A}$ over $\Sigma = \{a_1, \ldots, a_d\}$, we construct a d-VASS $\mathcal{V}$. The VASS $\mathcal{V}$ simulates $\mathcal{A}$ such that when $\mathcal{A}$ reads a_i, $\mathcal{V}$ increments counter i. Moreover, $\mathcal{V}$ maintains a number $k \in \{0, \ldots, d\}$ in its state, which always holds the number of letters read so far. Thus, $\mathcal{V}$ has states q_k, where q is a state of $\mathcal{A}$ and $k \in \{1, \ldots, d\}$. Moreover, let s and t be the initial and final state of $\mathcal{A}$, respectively. Then in $\mathcal{V}$, one can cover $t_d(1, \ldots, 1)$ from $s_0(\mathbf{0})$ in monus semantics if and only if $\mathcal{A}$ accepts some word as above.

We turn to the NP upper bound. Suppose we are given a d-VASS $\mathcal{V} = (d, Q, \Delta)$ and configurations $s(\mathbf{u}), t(\mathbf{v})$. We employ Proposition 3.12. First non-deterministically guess a permutation σ of $[1, d]$. We now construct a $2d$-VASS $\mathcal{V}'_\sigma$ and two configurations c'_1, c'_2 such that in $\mathcal{V}'_\sigma$, we have $c'_1 \xrightarrow[\mathbb{Z}]{*} c'_2$ if and only if there is a run as in Proposition 3.12 with this σ. Since reachability in $\mathbb{Z}$-semantics is NP-complete [25], this yields the upper bound.

Our VASS $\mathcal{V}'_\sigma$ is a slight extension of the VASS $\mathcal{V}_\sigma$ from Theorem 3.7, see Figure 3. Recall that for a permutation $\sigma\colon [1, k] \to Z$, $\mathcal{V}_\sigma$ keeps k extra counters that freeze the values of the counters in Z, in the order $\sigma(k), \sigma(k-1), \ldots, \sigma(1)$. We use this construction, but for our permutation σ of $[1, d]$. Thus, $\mathcal{V}_\sigma$ simulates a run of $\mathcal{V}$ and then freezes the counters $\sigma(d), \ldots, \sigma(1)$ in the extra d counters, in this order. The steps that freeze counters define the vectors $\mathbf{v}_d, \ldots, \mathbf{v}_1$ in Proposition 3.12. Note that for each $\mathbf{v}_i$, only $\mathbf{v}_i[\sigma(i)]$ is important.

To verify the second condition in Proposition 3.12, we introduce an extra state t' and extra transitions as depicted in Figure 3. After executing $\mathcal{V}_\sigma$, $\mathcal{V}'_\sigma$ then has two types of loops: One to move tokens from the counters $d + j$ to counters $\sigma(j)$ (for each $j \in [1, d]$), and one to reduce tokens in counters $1, \ldots, d$. Thus there exists σ such that $s_0(\mathsf{copy}_0(\mathbf{u})) \xrightarrow[\mathbb{Z}]{*} t'(\mathsf{copy}_d(\mathbf{v}))$ in $\mathcal{V}'_\sigma$ if and only if $s(\mathbf{u}) \overset{*}{\Rightarrow} t(\mathbf{v}'')$ for some $\mathbf{v}'' \geq \mathbf{v}$ in $\mathcal{V}$. This proves the NP upper bound. ◀

4 Two-dimensional VASS

In this section we prove the results of Table 1 related to 2-VASS, both for unary and binary encoding. Note that for all three considered problems, reachability, zero reachability, and coverability, we always have an NL lower bound, inherited from state reachability in finite automata. The latter is well-known to be NL-hard, and a VASS without counters (in all considered semantics) is a finite state automaton.

When dealing with binary/unary updates one needs to be careful with the input size. In all problems suppose a VASS $\mathcal{V} = (d, Q, T)$ is in the input. If we are interested in the unary encoding its size is defined as $d + |Q| + \sum_{(p,\mathbf{z},q)\in T} \|\mathbf{z}\|$, where $\|\mathbf{z}\|$ is the absolute value of the maximal coordinate in $\mathbf{z}$. In the binary encoding one needs to change $\|\mathbf{z}\|$ to $\lceil \log(\|\mathbf{z}\| + 1) \rceil$. From this point onwards, we use the term *succinct* VASS for VASS where updates are encoded in binary.

We consider each of the three problems separately.

Reachability. Here we only prove the PSPACE upper bound for monus reachability in binary encoded 2-VASS, which implies the same upper bound for unary encoding. The PSPACE lower bound for binary encoding is inherited from zero reachability, see Proposition 4.3 below.

▶ **Proposition 4.1.** *In succinct 2-VASS, reachability with monus semantics is in* PSPACE.

According to Proposition 3.6, reachability with monus semantics is equivalent to existence of a run under classical semantics, where said run is subject to some additional constraints. Recall that *Presburger arithmetic* is the first-order theory of $(\mathbb{N}, +, <, 0, 1)$. We observe that all the additional constraints of Proposition 3.6 can be expressed by quantifier-free Presburger formulas. This leads us to the so-called *constrained runs problem* for succinct 2-VASS, which was recently shown to be in PSPACE [3], following the fact that classical reachability itself is PSPACE-complete for succinct 2-VASS [5].

Formally, the *constrained runs problem* for succinct 2-VASS is the following:

Given A succinct 2-VASS $\mathcal{V}$, a number $m \in \mathbb{N}$, states $q_1, \ldots, q_m$ in $\mathcal{V}$, a quantifier-free Presburger formula $\psi(x_1, y_1, \ldots, x_m, y_m)$, and numbers $s, t \in [1, m]$ with $s \leq t$.

Question Does there exist a run $q_0(0,0) \xrightarrow{*} q_1(x_1, y_1) \xrightarrow{*} \cdots \xrightarrow{*} q_m(x_m, y_m)$ that visits a final state between $q_s(x_s, y_s)$ and $q_t(x_t, y_t)$ and satisfies $\psi(x_1, y_1, \ldots, x_m, y_m)$?

▶ **Lemma 4.2** ([3, Prop. 6.5]). *The constrained runs problem for succinct 2-VASS is in* PSPACE.

We can now prove Proposition 4.1 by reducing to the constrained runs problem: Let $\mathcal{V}$ be a 2-VASS with configurations $s(\mathbf{v})$ and $t(\mathbf{w})$. According to Proposition 3.6, existence of a run $s(\mathbf{v}) \overset{*}{\Rightarrow} t(\mathbf{w})$ is equivalent to existence of states p_1, p_2 and a set $Z \subseteq [1,2]$ such that a run $s(\mathbf{v}') \xrightarrow{*} t(\mathbf{w})$ with $\mathbf{v}' \geq \mathbf{v}$ that is subject to additional requirements enforced by conditions (2) and (3) of the Proposition 3.6. Our PSPACE algorithm enumerates all possibilities of p_1, p_2 and Z, constructing an instance of the constrained run problem each time, and checking for a constrained run in PSPACE using Lemma 4.2. If such a run exists in at least one of the instances, the algorithm accepts, otherwise it rejects. To construct each instance the algorithm first modifies $\mathcal{V}$ to ensure that a starting configuration $s(\mathbf{v}')$ is reachable for any $\mathbf{v}' \geq \mathbf{v}$. To this end a new initial state q_0 is added, with two loops that increment one of the counters each, and a transition that goes to s by adding $\mathbf{v}$. Then the additional requirements of Proposition 3.6 are encoded in quantifier-free Presburger arithmetic, as required by the constrained run problem. Clearly the constructed algorithm runs in PSPACE and decides $s(\mathbf{v}) \overset{*}{\Rightarrow} t(\mathbf{w})$. For more details refer the full version.

Zero reachability.

▶ **Proposition 4.3.** *Monus zero reachability in 2-VASS is* PSPACE*-complete under binary encoding and* NL*-complete under unary encoding.*

Proof. This is a simple consequence of monus zero reachability being interreducible with classical coverability: Classical coverability in 2-VASS under binary encoding is PSPACE-complete under binary encoding (in [5, Corollary 3.3], this is deduced from [36, p. 108] and [17, Corollary 10] and NL-complete under unary encoding [36, p. 108].

Let $\mathcal{V}$ be a 2-VASS with configurations $s(\mathbf{v})$ and $t(\mathbf{0})$. Then according to Proposition 3.6, we know that $t(\mathbf{0})$ is monus reachable from $s(\mathbf{v})$ if and only if in $\mathcal{V}^{\mathsf{rev}}$ the configuration $s(\mathbf{v})$ is coverable from $t(\mathbf{0})$ with classical semantics. On the other hand, given configurations $s(\mathbf{v})$ and $t(\mathbf{w})$ of a 2-VASS $\mathcal{V}$, we add a new state s' and transition $(s', \mathbf{v}, s)$ to construct the 2-VASS $\mathcal{V}'$. Then classical coverability of $t(\mathbf{w})$ from $s(\mathbf{v})$ in $\mathcal{V}$ is equivalent to the same from $s'(\mathbf{0})$ in $\mathcal{V}'$. Now applying Proposition 3.6 in reverse, the latter is further equivalent to monus reachability of $s'(\mathbf{0})$ from $t(\mathbf{w})$ in $\mathcal{V}'^{\mathsf{rev}}$. ◀

Coverability. By Proposition 3.13, monus coverability is in NP in arbitrary dimension. Thus, it remains to show the NP lower bound.

▶ **Proposition 4.4.** *Monus coverability in succinct 2-VASS is* NP*-hard.*

Proof. We reduce from the *subset sum* problem, which is well-known to be NP-hard. Here, we are given binary encoded numbers $a_1, \ldots, a_n, a \in \mathbb{N}$ and are asked whether there is a vector $(x_1, \ldots, x_n) \in \{0,1\}^n$ such that $x_1 a_1 + \cdots + x_n a_n = a$. Given such an instance, we construct the 2-VASS in Figure 4. It is clear that we can cover $t(1,1)$ from $s(0,0)$ iff the subset-sum instance is positive: Covering 1 in the first counter means our sum is at least a, whereas covering 1 in the second counter means our sum is at most a. ◀

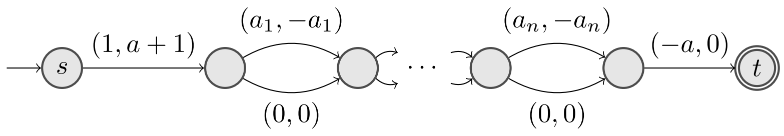


Figure 4 2-VASS to show NP-hardness of coverability in dimension two.

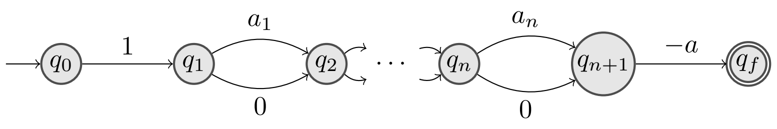


Figure 5 1-VASS to show NP-hardness of monus reachability in dimension one with binary encoded counter updates.

▶ **Proposition 4.5.** *Monus coverability in unary-encoded* 2*-VASS is in* NL*.*

Proof. This follows using the same construction as for Proposition 3.13: Given a 2-VASS, there are only two permutations σ of $\{1, 2\}$. Thus, we can try both permutations σ and construct the VASS $\mathcal{V}'_\sigma$ in logspace. Then, $\mathcal{V}_\sigma$ has dimension $2d$. Thus, we reduce monus coverability in 2-VASS to reachability in $\mathbb{Z}$-semantics in 4-VASS. Since reachability with $\mathbb{Z}$-semantics in each fixed dimension can be decided in NL [24], this provides an NL upper bound. ◀

5 One-dimensional VASS

Reachability. We begin with the proofs regarding reachability.

▶ **Proposition 5.1.** *Monus reachability in* 1*-VASS is in* NL *under unary encoding and in* NP *under binary encoding.*

The proof of Proposition 5.1 relies on the following simple consequence of Proposition 3.6:

▶ **Lemma 5.2.** *Let $\mathcal{V}$ be a* 1*-VASS. Then $s(m) \overset{*}{\Rightarrow}_{\mathcal{V}} t(n)$ if and only if (i) $s(m) \xrightarrow{*}_{\mathcal{V}} t(n)$ or (ii) there exist a state q and number $m' \geq m$ with $s(m') \xrightarrow{*}_{\mathcal{V}} q(0)$ and $q(0) \xrightarrow{*}_{\mathcal{V}} t(n)$.*

For Proposition 5.1, we reduce to reachability in one-counter automata. A *one-counter automaton (OCA)* is a 1-VASS with zero-tests, i.e. special transitions that test the counter for zero instead of adding a number. For encoding purposes, zero tests take up as much space as a transition adding 0 to the counter. In our reduction, the update encoding is preserved: If the input 1-VASS has unary encoding, then the OCA has unary updates as well. If the input 1-VASS has binary updates, then the OCA will too. Then, we can use the fact that in OCA with unary updates, reachability is in NL [38] and for binary updates, it is in NP [26].

The OCA first guesses whether to simulate a run of type (i) or of type (ii) in Lemma 5.2. Then for type (i), it just simulates a classical 1-VASS. For type (ii), it first non-deterministically increments the counter, and then simulates a run of the 1-VASS. However, on the way, it keeps a flag signaling whether the counter has hit 0 at some point (which it can maintain using zero tests). Thus, when simulating runs of type (ii), the OCA only accepts if zero has been hit. For a detailed description, refer to the full version.

▶ **Proposition 5.3.** *Monus reachability in* 1*-VASS is* NP*-hard under binary encoding.*

As in Proposition 4.4, we reduce from subset sum. Given $a_1, \dots, a_n, a$ in binary, we construct the 1-VASS in Figure 5. Then $q_0(0) \overset{*}{\Rightarrow} q_f(1)$ iff this is a positive instance. Refer to the full version.

Zero reachability and coverability.

▶ **Proposition 5.4.** *Monus zero-reachability in* 1*-VASS is in* NL *under unary encoding and in* NC^2 *under binary encoding.*

Since monus zero-reachability reduces to classical coverability (Lemma 3.10), this follows from existing 1-VASS results: Coverability in 1-VASS is in NL under unary encoding [38] and NC^2 under binary encoding [2].

▶ **Proposition 5.5.** *Monus coverability in* 1*-VASS is in* NL *under unary encoding and in* NC^2 *under binary encoding.*

The first statement follows from Proposition 5.1 and the fact that monus coverability reduces to monus reachability by simply adding a new final state where we can count down. For the NC^2 bound, we use the following consequence of Lemma 3.10 (see the full version).

▶ **Lemma 5.6.** *Let $\mathcal{V}$ be a* 1*-VASS with configurations $s(m)$ and $t(n)$. Then $t(n)$ is monus coverable from $s(m)$ in $\mathcal{V}$ if and only if $t(n)$ is coverable from $s(m)$ in $\mathcal{V}$ under classical semantics or there is a state q of $\mathcal{V}$ such that $t(n)$ is coverable from $q(0)$ in $\mathcal{V}$ under classical semantics and $s(m)$ is coverable from $q(0)$ in $\mathcal{V}^{\mathsf{rev}}$ under classical semantics.*

Proof of Proposition 5.5. It remains to prove the NC^2 upper bound, for which we check the requirements of Lemma 5.6. Let k be the number of states of the input 1-VASS. Observe that Lemma 5.6 yields a logical disjunction over $k+1$ disjuncts, where one disjunct consists of a single coverability check and the remaining k each consist of a logical conjunction over two coverability checks. Classical coverability of binary encoded 1-VASS is in NC^2 [2], and by the definition of this complexity class, we can combine $2k+1$ such checks according to the aforementioned logical relationship and still yield an NC^2-algorithm. Note that this is only possible because k is linear in the size of the input. ◀

References

1 Parosh Aziz Abdulla, Karlis Cerans, Bengt Jonsson, and Yih-Kuen Tsay. General decidability theorems for infinite-state systems. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 313–321. IEEE Computer Society, 1996. doi:10.1109/LICS.1996.561359.

2 Shaull Almagor, Nathann Cohen, Guillermo A. Pérez, Mahsa Shirmohammadi, and James Worrell. Coverability in 1-VASS with Disequality Tests. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 38:1–38:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.38.

3 Pascal Baumann, Roland Meyer, and Georg Zetzsche. Regular Separability in Büchi VASS. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023)*, volume 254 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.STACS.2023.9.

4 Michael Blondin. The ABCs of Petri net reachability relaxations. *ACM SIGLOG News*, 7(3), 2020. doi:10.1145/3436980.3436984.

5 Michael Blondin, Matthias Englert, Alain Finkel, Stefan Göller, Christoph Haase, Ranko Lazic, Pierre McKenzie, and Patrick Totzke. The Reachability Problem for Two-Dimensional Vector Addition Systems with States. *J. ACM*, 68(5):34:1–34:43, 2021. doi:10.1145/3464794.

6 Michael Blondin, Alain Finkel, Christoph Haase, and Serge Haddad. Approaching the coverability problem continuously. In *Proc. 22nd International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 480–496, 2016. doi:10.1007/978-3-662-49674-9_28.

7 Michael Blondin, Christoph Haase, and Philip Offtermatt. Directed reachability for infinite-state systems. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 – April 1, 2021, Proceedings, Part II*, volume 12652 of *Lecture Notes in Computer Science*, pages 3–23. Springer, 2021. doi:10.1007/978-3-030-72013-1_1.

8 Wojciech Czerwinski, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. Reachability in fixed dimension vector addition systems with states. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory, CONCUR 2020, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPIcs*, pages 48:1–48:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CONCUR.2020.48.

9 Wojciech Czerwinski and Lukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1229–1240. IEEE, 2021. doi:10.1109/FOCS52979.2021.00120.

10 Wojciech Czerwinski and Lukasz Orlikowski. Lower bounds for the reachability problem in fixed dimensional vasses. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2–5, 2022*, pages 40:1–40:12. ACM, 2022. doi:10.1145/3531130.3533357.

11 Stéphane Demri, Marcin Jurdzinski, Oded Lachish, and Ranko Lazic. The covering and boundedness problems for branching vector addition systems. *J. Comput. Syst. Sci.*, 79(1):23–38, 2013. doi:10.1016/j.jcss.2012.04.002.

12 Alex Dixon and Ranko Lazic. Kreach: A tool for reachability in petri nets. In Armin Biere and David Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems – 26th International Conference, TACAS 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Part I*, volume 12078 of *Lecture Notes in Computer Science*, pages 405–412. Springer, 2020. doi:10.1007/978-3-030-45190-5_22.

13 Catherine Dufourd, Alain Finkel, and Philippe Schnoebelen. Reset nets between decidability and undecidability. In Kim Guldstrand Larsen, Sven Skyum, and Glynn Winskel, editors, *Automata, Languages and Programming, 25th International Colloquium, ICALP'98, Aalborg, Denmark, July 13-17, 1998, Proceedings*, volume 1443 of *Lecture Notes in Computer Science*, pages 103–115. Springer, 1998. doi:10.1007/BFb0055044.

14 Matthias Englert, Piotr Hofman, Slawomir Lasota, Ranko Lazic, Jérôme Leroux, and Juliusz Straszynski. A lower bound for the coverability problem in acyclic pushdown VAS. *Inf. Process. Lett.*, 167:106079, 2021. doi:10.1016/j.ipl.2020.106079.

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

16 John Fearnley and Marcin Jurdziński. Reachability in Two-Clock Timed Automata Is PSPACE-Complete. In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming*, pages 212–223, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

17 John Fearnley and Marcin Jurdzinski. Reachability in two-clock timed automata is pspace-complete. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming – 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part II*, volume 7966 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2013. doi:10.1007/978-3-642-39212-2_21.

18 Diego Figueira, Ranko Lazic, Jérôme Leroux, Filip Mazowiecki, and Grégoire Sutre. Polynomial-space completeness of reachability for succinct branching VASS in dimension one. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 119:1–119:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.119.

19 Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001. doi:10.1016/S0304-3975(00)00102-X.

20 Estíbaliz Fraca and Serge Haddad. Complexity analysis of continuous Petri nets. *Fundamenta Informaticae*, 137(1):1–28, 2015. doi:10.3233/FI-2015-1168.

21 Moses Ganardi, Rupak Majumdar, Andreas Pavlogiannis, Lia Schütze, and Georg Zetzsche. Reachability in bidirected pushdown VASS. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 124:1–124:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.124.

22 Steven M. German and A. Prasad Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992. doi:10.1145/146637.146681.

23 Stefan Göller, Christoph Haase, Ranko Lazic, and Patrick Totzke. A polynomial-time algorithm for reachability in branching VASS in dimension one. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 105:1–105:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.105.

24 Eitan M. Gurari and Oscar H. Ibarra. The complexity of decision problems for finite-turn multicounter machines. *J. Comput. Syst. Sci.*, 22(2):220–229, 1981. doi:10.1016/0022-0000(81)90028-3.

25 Christoph Haase and Simon Halfon. Integer vector addition systems with states. In Joël Ouaknine, Igor Potapov, and James Worrell, editors, *Reachability Problems – 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings*, volume 8762 of *Lecture Notes in Computer Science*, pages 112–124. Springer, 2014. doi:10.1007/978-3-319-11439-2_9.

26 Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in succinct and parametric one-counter automata. In Mario Bravetti and Gianluigi Zavattaro, editors, *CONCUR 2009 – Concurrency Theory, 20th International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings*, volume 5710 of *Lecture Notes in Computer Science*, pages 369–383. Springer, 2009. doi:10.1007/978-3-642-04081-8_25.

27 John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979.

28 Eryk Kopczynski. Complexity of problems of commutative grammars. *Log. Methods Comput. Sci.*, 11(1), 2015. doi:10.2168/LMCS-11(1:9)2015.

29 Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1241–1252. IEEE, 2021. doi:10.1109/FOCS52979.2021.00121.

30 Jérôme Leroux and Sylvain Schmitz. Reachability in vector addition systems is primitive-recursive in fixed dimension. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*, pages 1–13. IEEE, 2019. doi:10.1109/LICS.2019.8785796.

31 Jérôme Leroux and Grégoire Sutre. Reachability in Two-Dimensional Vector Addition Systems with States: One Test Is for Free. In Igor Konnov and Laura Kovács, editors, *31st International Conference on Concurrency Theory (CONCUR 2020)*, volume 171 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2020.37.

32 Jérôme Leroux, Grégoire Sutre, and Patrick Totzke. On the coverability problem for pushdown vector addition systems in one dimension. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming – 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 324–336. Springer, 2015. doi:10.1007/978-3-662-47666-6_26.

33 Richard Lipton. The reachability problem is exponential-space hard. *Yale University, Department of Computer Science, Report*, 62, 1976.

34 Filip Mazowiecki, Henry Sinclair-Banks, and Karol Węgrzycki. Coverability in 2-vass with one unary counter is in np. In Orna Kupferman and Pawel Sobocinski, editors, *Foundations of Software Science and Computation Structures*, pages 196–217, Cham, 2023. Springer Nature Switzerland.

35 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

36 Louis E Rosier and Hsu-Chun Yen. A multiparameter analysis of the boundedness problem for vector addition systems. *Journal of Computer and System Sciences*, 32(1):105–135, 1986.

37 Sylvain Schmitz. The complexity of reachability in vector addition systems. *ACM SIGLOG News*, 3(1):4–21, 2016. doi:10.1145/2893582.2893585.

38 Leslie G. Valiant and Mike Paterson. Deterministic one-counter automata. *J. Comput. Syst. Sci.*, 10(3):340–350, 1975. doi:10.1016/S0022-0000(75)80005-5.

39 Wil M. P. van der Aalst. Verification of workflow nets. In *Proc. 18th International Conference on Application and Theory of Petri Nets (ICATPN)*, volume 1248, pages 407–426, 1997. doi:10.1007/3-540-63139-9_48.