Counter Machines with Infrequent Reversals

Alain Finkel
Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF, Gif-sur-Yvette, France

Shankara Narayanan Krishna
IIT Bombay, India

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

Rupak Majumdar
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

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

Abstract

Bounding the number of reversals in a counter machine is one of the most prominent restrictions to achieve decidability of the reachability problem. Given this success, we explore whether this notion can be relaxed while retaining decidability.

To this end, we introduce the notion of an f-reversal-bounded counter machine for a monotone function $f\colon \mathbb{N} \to \mathbb{N}$. In such a machine, every run of length n makes at most $f(n)$ reversals. Our first main result is a dichotomy theorem: We show that for every monotone function f, one of the following holds: Either (i) f grows so slowly that every f-reversal bounded counter machine is already k-reversal bounded for some constant k or (ii) f belongs to $\Omega(\log(n))$ and reachability in f-reversal bounded counter machines is undecidable. This shows that classical reversal bounding already captures the decidable cases of f-reversal bounding for any monotone function f. The key technical ingredient is an analysis of the growth of small solutions of iterated compositions of Presburger-definable constraints. In our second contribution, we investigate whether imposing f-reversal boundedness improves the complexity of the reachability problem in vector addition systems with states (VASS). Here, we obtain an analogous dichotomy: We show that either (i) f grows so slowly that every f-reversal-bounded VASS is already k-reversal-bounded for some constant k or (ii) f belongs to $\Omega(n)$ and the reachability problem for f-reversal-bounded VASS remains Ackermann-complete. This result is proven using run amalgamation in VASS.

Overall, our results imply that classical restriction of reversal boundedness is a robust one.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases Counter machines, reversal-bounded, reachability, decidability, complexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2023.42

Funding Alain Finkel was funded by the Institut Universitaire de France and the Agence Nationale de la Recherche, grant BraVAS (ANR-17-CE40-0028). Khushraj Madnani and Rupak Majumdar were supported in part by the Deutsche Forschungsgemeinschaft project 389792660 TRR 248–CPEC.

1 Introduction

The undecidability of the reachability problem in general multicounter machines is well-known [37]. Given this, there has been a rich landscape of restricted counter models, which have been studied with the aim of obtaining decidable reachability, while retaining as much expressiveness as possible. One of the most prominent restrictions studied is that of *reversal-bounded counter machines* [24]. As the name suggests, reversal-bounded counter machines bound the number of times a counter can change from an *incrementing phase* to a *decrementing phase*, or vice-versa, during a run. In an incrementing phase, the counter

43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023).
Editors: Patricia Bouyer and Srikanth Srinivasan; Article No. 42; pp. 42:1–42:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is only incremented; likewise, in a decrementing phase, it is only decremented. As already shown by Ibarra in 1978 [24], for a constant bound, reachability is decidable and, further, reachability relations are semilinear. The latter observation allows decision procedures for Presburger arithmetic to be used for the algorithmic analysis of reversal-bounded systems.

Partly motivated by these available tools, reversal-bounded counter machines have been studied intensively. They have been studied in terms of deciding basic properties [19], model-checking various logics [3,14], expressiveness [26,29], deciding language-theoretic properties [2], regular abstractions [39], extensions that allow arbitrary reversals below a fixed counter value [15] or a free counter [13] or a pushdown [21] (see also [25] for a survey). This research has revealed a wide range of applications. For example, they have been used for model-checking recursive programs with numeric data types [21]. As another example, an equivalent variant of reversal-bounded counter machines is the model of Parikh automata [28], originally introduced to decide monadic second order logic with certain cardinality constraints [28], but subsequently studied as a computational model in itself [4–9, 17, 18].

Another decidable restriction of counter machines is that of vector addition systems with states (VASS). These allow arbitrary reversals, but have no zero tests. VASS are a standard model for analyzing concurrent systems. Unfortunately, the reachability problem in VASS is Ackermann-complete [11, 32]. However, imposing reversal-boundedness on VASS reduces the complexity down to NP: Reachability for reversal-bounded machines is NP-complete [19]. This has a number of implications. For example, one can efficiently analyze flat counter systems [36] by turning them into reversal-bounded systems. Flat counter systems, in turn, are the basis for flat acceleration techniques [1]. Furthermore, being (not flat but) flattable is equivalent to having semilinear reachability sets in VASS [31].

Thus, reversal bounding has turned out to be a fruitful restriction that ensures decidability in counter machines, and that reduces the computational complexity significantly in the case of VASS.

Infrequent reversals. Given this success of reversal-bounding counter machines, it seems natural to explore whether there are weaker conditions one can impose on reversals that still retain some decidability, say for the reachability problem. In this work, we explore the idea of requiring reversals to be *infrequent*, meaning the number of reversals is small *in relation to the length of the run*, as opposed to being a fixed constant independent of the length of the run. More specifically, for a monotone function $f\colon \mathbb{N} \to \mathbb{N}$, we call a counter machine f*-reversal-bounded* if in every run of length n, the machines makes at most $f(n)$ reversals. In other words, every run of length n decomposes into at most $f(n)$ phases, where each phase has a particular direction for each counter: Inside a phase, each counter can either (i) not be decremented or (ii) not be incremented. Clearly, this generalizes the classical notion of reversal-boundedness, which takes f to be a constant function. With this notion, we first investigate the following question:

> *For which monotone functions* $f\colon \mathbb{N} \to \mathbb{N}$ *do* f*-reversal-bounded counter machines have a decidable reachability problem?*

Moreover, we study whether such a relaxed notion can serve to reduce the complexity of reachability in VASS:

> *For which monotone functions* $f\colon \mathbb{N} \to \mathbb{N}$ *do* f*-reversal-bounded VASS have a low-complexity reachability problem?*

Contribution. For both questions above, we provide complete, but negative answers. First, we show that there are functions f that guarantee decidable reachability (e.g., $n \mapsto \log(\log(n))$), but they must grow so slowly that for every counter machine that is f-reversal-bounded, there exists a number $k \in \mathbb{N}$ such that the machine is already k-reversal-bounded. The main technical contribution used in proving this was the analysis of growth of minimal solution of k-fold composition of Presburger definable (i.e. semilinear) relations. On the other hand, if the function f grow fast enough (e.g., $n \mapsto \log(n)$), the reachability problem is undecidable.

An analogous situation holds for the complexity of VASS reachability: There are functions f for which f-reversal-bounded VASS have lower complexity than Ackermann (e.g., $n \mapsto \log(n)$), but these grow so slowly that for every VASS that is f-reversal-bounded, there exists some $k \in \mathbb{N}$ so that the VASS is k-reversal-bounded. On the other hand, if f grows sufficiently fast, the reachability problem is again Ackermann-complete.

In short, we show that there is no monotone function that properly relaxes the reversal-boundedness condition and still either (i) guarantees decidable reachability in counter machines or (ii) lowers the complexity of reachability in VASS.

Our results. Let us make our results precise. For every monotone function $f \colon \mathbb{N} \to \mathbb{N}$, we consider two decision problems. First, $\mathsf{Reach}(f)$ is the following:

> The problem $\mathsf{Reach}(f)$:
> **Given** An f-reversal-bounded counter machine CM and a configuration c.
> **Question** Can CM reach c from its initial configuration?

Second, the problem $\mathsf{Reach}_{\mathsf{VASS}}(f)$ restricts $\mathsf{Reach}(f)$ to f-reversal-bounded VASS:

> The problem $\mathsf{Reach}_{\mathsf{VASS}}(f)$:
> **Given** An f-reversal-bounded VASS $\mathcal{V}$ and a configuration c.
> **Question** Can $\mathcal{V}$ reach c from its initial configuration?

Now our two questions above become:

1. For which monotone functions $f \colon \mathbb{N} \to \mathbb{N}$ is $\mathsf{Reach}(f)$ decidable?
2. For which monotone functions $f \colon \mathbb{N} \to \mathbb{N}$ does $\mathsf{Reach}_{\mathsf{VASS}}(f)$ have lower complexity?

We say that a monotone function $f \colon \mathbb{N} \to \mathbb{N}$ is *essentially bounded (for counter machines)* if for every counter machine CM that is f-reversal-bounded, there exists a number $k \in \mathbb{N}$ such that CM is k-reversal-bounded. Our first main result is a dichotomy for monotone functions:

▶ **Theorem 1.1.** *Let $f \colon \mathbb{N} \to \mathbb{N}$ be a monotone function. Then exactly one of the following holds: Either (i) f is essentially bounded or (ii) f belongs to $\Omega(\log n)$, and* $\mathsf{Reach}(f)$ *is undecidable.*

To state our second main result, we say that f is *essentially bounded for VASS* if for every VASS $\mathcal{V}$ that is f-reversal-bounded, there exists a number $k \in \mathbb{N}$ such that $\mathcal{V}$ is k-reversal bounded. Our second main result is the following dichotomy:

▶ **Theorem 1.2.** *Let $f \colon \mathbb{N} \to \mathbb{N}$ be a monotone function. Then exactly one of the following holds: Either (i) f is essentially bounded for VASS or (ii) f belongs to $\Omega(n)$, and* $\mathsf{Reach}(f)$ *is Ackermann-complete.*

In other words, by relaxing the definition of "bounded reversal" to "infrequent reversal", we either get counter machines (or VASS) with undecidable (or Ackermann-hard) reachability problem or get a counter machines (or VASS) that are already reversal bounded. Hence, the class of machines within the purview of the classical reversal bounded restriction is already robust. That is, the notion of infrequent reversals does not give us any new decidable (resp., computationally easier) class of counter machines (resp., VASSes).

2 Preliminaries

Notations. We denote the set of all natural numbers (resp., integers) with $\mathbb{N}$ (resp., $\mathbb{Z}$). Given any set X, we write X^d to denote the set of all vectors of dimension d whose elements are in X. We write $\mathbf{0}^d$ for the vector all of whose entries are 0. We omit the superscript d from $\mathbf{0}^d$, when the dimension is clear from the context. Given $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{Z}^d$, we define $\boldsymbol{v}_1 \leq \boldsymbol{v}_2$ iff for all $i \in \{1, \ldots, d\}$, $\boldsymbol{v}_1[i] \leq \boldsymbol{v}_2[i]$. Moreover, $\boldsymbol{v}_1 + \boldsymbol{v}_2$ (resp., $\boldsymbol{v}_1 - \boldsymbol{v}_2$) is the vector $\boldsymbol{v}$ such that for all $i \in \{1, \ldots, d\}$, we have $\boldsymbol{v}[i] = \boldsymbol{v}_1[i] + \boldsymbol{v}_2[i]$ (resp., $\boldsymbol{v}[i] = \boldsymbol{v}_1[i] - \boldsymbol{v}_2[i]$). For a function $g\colon \mathbb{N} \to \mathbb{N}$, we write $\Omega(g)$ for the class of functions $f\colon \mathbb{N} \to \mathbb{N}$ for which there exist constants c, n_0 such that for all $n \geq n_0$, we have $f(n) \geq c \cdot g(n)$. As is customary, we write $\Omega(n)$ (resp. $\Omega(\log n)$) for the class $\Omega(g)$ with $g\colon n \mapsto n$ (resp. $g\colon n \mapsto \log(n)$).

Counter Machines and Vector Addition Systems with States. A *Counter Machine* is a 4-tuple $\mathsf{CM} = (d, Q, \Delta, q_0)$ where $d \in \mathbb{N}$ is the dimension, Q is a finite set of control states, and $\Delta \subseteq Q \times \mathsf{INST} \times Q$ is a finite set of transitions where $\mathsf{INST} = \mathbb{Z}^d \cup \{\mathsf{C}_j \stackrel{?}{=} 0 \mid j \in \{1, \ldots d\}\}$ is the set of all instructions, and $q_0 \in Q$ is the initial state.

An instruction is either some vector $\boldsymbol{z} \in \mathbb{Z}^d$, or a test of the form "$\mathsf{C}_k \stackrel{?}{=} 0$" where $k \in \{1, 2, \ldots d\}$. Instructions of the form $\mathsf{C}_k \stackrel{?}{=} 0$ are called zero tests.

A *configuration* of CM is a pair $(q, \boldsymbol{v})$ where $q \in Q$ and $\boldsymbol{v} \in \mathbb{N}^d$. The *initial configuration* is $(q_0, \mathbf{0})$. Any transition $t = (q, inst, q')$ induces a successor (partial) function $\mathsf{Succ}_t : Q \times \mathbb{N}^d \to Q \times \mathbb{N}^d$ defined as follows. If $\boldsymbol{z} \in \mathbb{Z}^d$ then $\mathsf{Succ}_t((q, \boldsymbol{v})) = (q', \boldsymbol{v}')$ where $\boldsymbol{v}' = \boldsymbol{v} + \boldsymbol{z}$. If $\boldsymbol{z}$ is a zero test instruction of the form "$\mathsf{C}_k \stackrel{?}{=} 0$" then $\mathsf{Succ}_t((q, \boldsymbol{v})) = (q', \boldsymbol{v})$ if $\boldsymbol{v}[k] = 0$ and is not defined otherwise. This successor function can be lifted to Δ to get a step relation $\to_{\mathsf{CM}}$, such that, for any pair of configurations $\mathcal{C}, \mathcal{C}'$, we have $\mathcal{C} \to_{\mathsf{CM}} \mathcal{C}'$ iff there exists a transition $t \in \Delta$ such that $\mathsf{Succ}_t(\mathcal{C}) = \mathcal{C}'$. We sometimes use the term counters and coordinates interchangeably in the context of vectors.

An (initialized) *run* of CM is a sequence of configurations $(q_0, \boldsymbol{v}_0)(q_1, \boldsymbol{v}_1) \ldots (q_n, \boldsymbol{v}_n)$ such that $\boldsymbol{v}_0 = \mathbf{0}$ and for every $0 < j \leq n$, $(q_{j-1}, \boldsymbol{v}_{j-1}) \to_{\mathsf{CM}} (q_j, \boldsymbol{v}_j)$ holds. If there exists such a run we say that $(q_n, \boldsymbol{v}_n)$ is *reachable* from $(q_0, \boldsymbol{v}_0)$ and denote it as $(q_0, \boldsymbol{v}_0) \xrightarrow{*}_{\mathsf{CM}} (q_n, \boldsymbol{v}_n)$. We drop the subscript from $\to_{\mathsf{CM}}$ and $\xrightarrow{*}_{\mathsf{CM}}$ when the counter machine is clear from context.

Vector Addition System with States (*VASS*) is a subclass of counter machines where the transitions are restricted to not have zero tests.

The *reachability problem* for counter machines asks:
Given Given a configuration $(q, \boldsymbol{w})$ of a counter machine CM.
Question Does $(q_0, \mathbf{0}) \xrightarrow{*}_{\mathsf{CM}} (q, \boldsymbol{w})$ hold?

Reversal Boundedness. Informally, a counter is said to be in an increasing phase in a part of a run, if the value of that counter does not decrease in any step of the given part. Likewise the counter is in a decreasing phase in a part of a run, if its value does not increase in any step of the given part. A reversal is a step in a given run, where one or more counters switch from increasing to decreasing phase, or vice versa. Moreover, for $r \in \mathbb{N}$, a counter machine CM is r-reversal bounded if for any run of CM, all of its runs have at most r reversals.

Let us make this more formal. Let $\mathsf{CM} = (d, Q, \Delta)$ be any counter machine. A *mode vector* is a vector $\boldsymbol{m} \in \mathbb{Z}^d$ where every entry is -1 or 1. A mode vector describes the direction in which a counter can change: If $\boldsymbol{m}(i) = 1$, then this means counter i cannot be decremented. Similarly, $\boldsymbol{m}(i) = -1$ means it cannot be incremented. A step $(q, \boldsymbol{v}) \to (q', \boldsymbol{w})$ that adheres to this is called *consistent* with $\boldsymbol{m}$. A run is *consistent* with $\boldsymbol{m}$ if all its steps in the run are consistent with $\boldsymbol{m}$.

To define phases, consider a run $\rho = (q_0, \boldsymbol{v}_0)(q_1, \boldsymbol{v}_1)\dots(q_n, \boldsymbol{v}_n)$ of CM. We say that a segment $\rho'{=}(q_i, \boldsymbol{v}_i)(q_{i+1}, \boldsymbol{v}_{i+1})\dots(q_j, \boldsymbol{v}_j)$ of ρ is a *phase* of ρ, iff, ρ' is consistent with some mode vector $\boldsymbol{m}$. Intuitively, a phase is a part of the run where each counter $1 \le i \le d$ is either increasing in all steps, or is decreasing in all steps. ρ' is said to be the *maximal* phase of ρ iff it no longer remains a phase on extending it. That is, for $\rho' = (q_i, \boldsymbol{v}_i)(q_{i+1}, \boldsymbol{v}_{i+1})\dots(q_j, \boldsymbol{v}_j)$ and $j < n$, $\rho'.(q_{j+1}, \boldsymbol{v}_{j+1})$ is no longer a phase; likewise, for $\rho' = (q_i, \boldsymbol{v}_i)(q_{i+1}, \boldsymbol{v}_{i+1})\dots(q_j, \boldsymbol{v}_j)$ and $i > 1$, $(q_{i-1}, \boldsymbol{v}_{i-1}).\rho'$ is also no more a phase of ρ.

A run ρ contains r *reversals* iff ρ can be decomposed into r maximal phases. That is, $\rho = \rho_1.\rho_2.\dots.\rho_r$ such that for each $i \in \{1, \dots r\}$, ρ_i is a maximal phase of ρ.

A counter machine CM is said to be *r-reversal bounded* iff all its runs contain at most r reversals.

3 Counter machines: Decidable case

Our proof of Theorem 1.1 consists of showing two propositions:

▶ **Proposition 3.1.** *If there exists an f-reversal-bounded counter machine that is not reversal-bounded, then f belongs to $\Omega(\log(n))$.*

▶ **Proposition 3.2.** *If f belongs to $\Omega(\log(n))$, then $\mathsf{Reach}(f)$ is undecidable.*

Together, these clearly imply Theorem 1.1: Proposition 3.2 implies that if f belongs to $\Omega(\log(n))$, then $\mathsf{Reach}(f)$ is undecidable. Moreover, Proposition 3.1 tells us that if f does not belong to $\Omega(\log(n))$, then any f-reversal-bounded counter machine must already be reversal-bounded. In this section, we prove Proposition 3.1. In Section 4, we will then prove Proposition 3.2.

We start with defining semilinear sets and Presburger arithmetic.

Semilinear Sets. A set $S \subseteq \mathbb{N}^d$ is called *linear* iff there exists finitely many vectors $\boldsymbol{v}_0, \boldsymbol{v}_1, \dots \boldsymbol{v}_k \in \mathbb{N}^d$ such that

$$S = \{\boldsymbol{v}_0 + \sum_{i=1}^{k} c_i \cdot \boldsymbol{v}_i | c_1, c_2, \dots c_n \in \mathbb{N}\}$$

A *semilinear* set is a finite union of linear sets. Linear and semilinear relations are defined similarly.

Presburger Arithmetic. Presburger Arithmetic is defined to be the first-order theory of natural numbers endowed with the addition operation $(+)$, comparison $(<)$, and equality $(=)$ predicates. All the predicates have the usual meaning over the natural numbers. For the sake of simplicity, we allow constants in $\mathbb{N}$, as well as the multiplication of a constant by a variable as terms. Note that these terms are expressible using basic Presburger formulae. The following theorems connect Presburger constraints, semilinearity and reversal boundedness.

▶ **Theorem 3.3** (Ginsburg & Spanier [16]). *A relation is definable in Presburger arithmetic if and only if it is semilinear.*

▶ **Theorem 3.4** (Ibarra [24]). *Given any k-reversal bounded counter machine* CM *(for some k), its reachability relation is semilinear.*

Iterations of semilinear sets. Proposition 3.1 says that if a counter machine can make an unbounded number of reversals, but is f-reversal-bounded, then for every n, f must allow for a run of length n with at least $\Omega(\log(n))$ reversals. To this end, we will show that for any k, there exists a run with k reversals and of length at most c^k, for some constant c.

Using standard techniques for reversal-bounded counter machines, it is not difficult to show that there exists such a run of length $2^{k^{O(1)}}$: It is easy to construct an existential Presburger formula Φ_k of size $O(k)$ such that $\Phi_k(n)$ represents a run of length n. Then it follows (for example, from [20, Theorem 2]) that there exists a run of length at most $2^{k^{O(1)}}$. However, in order to prove our complete dichotomy, we need to refine these techniques to prove an upper bound of c^k for a constant c.

For a relation $R \subseteq \mathbb{N}^d \times \mathbb{N}^d$, we define R^k to be the k-fold composition of R, meaning $R^1 := R$ and $R^{k+1} := R^k \circ R$ for $k \geq 1$. For a vector $\boldsymbol{z} \in \mathbb{Z}^d$, $\boldsymbol{z} = (z_1, \ldots, z_d)$, we define its norm as $\|\boldsymbol{z}\| = \max\{|z_i| \mid i \in \{1, \ldots, d\}$. The key step in getting a c^k upper bound is the following lemma.

▶ **Lemma 3.5.** *For any semilinear $R \subseteq \mathbb{Z}^d \times \mathbb{Z}^d$ and $S \subseteq \mathbb{Z}^d \times \mathbb{Z}^d$, there is a constant $c \in \mathbb{N}$ such that for every $k \in \mathbb{N}$, if $R^k \cap S \neq \emptyset$, then there is a $\boldsymbol{z} \in R^k \cap S$ with $\|\boldsymbol{z}\| \leq c^k$.*

In the proof of Lemma 3.5, we will rely on the following bound on solution sizes of systems of linear inequalities due to [38, Corollary 1]. Here, for a matrix $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$, with entries a_{ij} ($i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$), we follow [38] in defining the norm

$$\|\boldsymbol{A}\|_{1,\infty} = \max_i \left\{ \sum_{j=1}^{n} |a_{ij}| \right\}.$$

▶ **Theorem 3.6** (Pottier 1991). *Let $\boldsymbol{A} \in \mathbb{Z}^{m\times n}$ and $\boldsymbol{a} \in \mathbb{Z}^m$. If there exists an $\boldsymbol{x}$ with $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{a}$, then there exists one with $\|\boldsymbol{x}\| \leq (2 + \|\boldsymbol{A}\|_{1,\infty} + \|\boldsymbol{a}\|)^m$.*

Proof of Lemma 3.5. We begin by expressing R and S using a system of linear Diophantine inequalities. Since they are semilinear, they can be expressed using existential Presburger formulae $\exists \boldsymbol{u}\colon \varphi_R(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y})$ and $\exists \boldsymbol{u}\colon \varphi_S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u})$, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of d variables each, and $\boldsymbol{u}$ is a vector of existentially quantified variables. Note that the quantified variables $\boldsymbol{u}$ are in different positions in φ_R and φ_S; this will be convenient later when constructing matrices. Now φ_R and φ_S are Boolean combinations of linear Diophantine inequalities. By bringing these into DNF, we can write

$$\varphi_R(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}) \iff \bigvee_{i=1}^{r} \boldsymbol{A}_i(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}) \leq \boldsymbol{a}_i$$

$$\varphi_S(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \iff \bigvee_{i=1}^{s} \boldsymbol{B}_i(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}) \leq \boldsymbol{b}_i$$

for some matrices $\boldsymbol{A}_i, \boldsymbol{B}_i \in \mathbb{Z}^{\ell\times m}$ and $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{Z}^{\ell}$. Here m is the combined number of variables across all of $\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{y}$ and ℓ is the (maximal) number of inequalities needed for φ_R and φ_S.

$$\begin{pmatrix} \begin{bmatrix} \bar{\boldsymbol{A}}_1 \end{bmatrix} & & & & \\ & \begin{bmatrix} \bar{\boldsymbol{A}}_2 \end{bmatrix} & & & \\ & & \ddots & & \\ & & & \begin{bmatrix} \bar{\boldsymbol{A}}_k \end{bmatrix} & \\ \begin{bmatrix} \boldsymbol{C} \end{bmatrix} & & & & \begin{bmatrix} \boldsymbol{D} \end{bmatrix} \end{pmatrix} \cdot \boldsymbol{w} \leq \begin{pmatrix} \begin{bmatrix} \bar{\boldsymbol{a}}_1 \end{bmatrix} \\ \begin{bmatrix} \bar{\boldsymbol{a}}_2 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \bar{\boldsymbol{a}}_k \end{bmatrix} \\ \begin{bmatrix} \boldsymbol{b} \end{bmatrix} \end{pmatrix}$$

Figure 1 Inequality in proof of Lemma 3.5. All other elements not in matrices $\boldsymbol{A_1}, \ldots \boldsymbol{A_k}, \boldsymbol{C}, \boldsymbol{D}$ are 0.

Suppose there is a vector $\boldsymbol{z} \in R^k \cap S$. Then a system of linear inequalities of the form sketched in Figure 1 has a solution. Here, for each $i \in \{1, \ldots, k\}$, there is some $j \in \{1, \ldots, r\}$ such that $\bar{\boldsymbol{A}}_i = \boldsymbol{A}_j$ and $\bar{\boldsymbol{a}}_i = \boldsymbol{a}_j$. Moreover, the overlap between $\bar{\boldsymbol{A}}_i$ and $\bar{\boldsymbol{A}}_{i+1}$ is exactly the right-most d columns of $\bar{\boldsymbol{A}}_i$ and the left-most d columns of $\bar{\boldsymbol{A}}_{i+1}$. Moreover, there is some $j \in \{1, \ldots, s\}$ such that the matrix $\boldsymbol{C}$ consists of the first d columns of $\boldsymbol{B}_j$ and $\boldsymbol{D}$ consists of the last $m - d$ columns of $\boldsymbol{B}_j$ (and the overlap to $\bar{\boldsymbol{A}}_k$ is exactly in the right-most d columns of $\bar{\boldsymbol{A}}_k$).

Let $\boldsymbol{A}$ be the matrix in Figure 1 and $\boldsymbol{a}$ be the vector on the right-hand side of Figure 1. Observe that $\boldsymbol{A}$ has $(k+1) \cdot \ell$ rows. By its choice, the inequality $\boldsymbol{A}\boldsymbol{w} \leq \boldsymbol{a}$ has a solution. Thus, by Theorem 3.6, we have a solution $\boldsymbol{w'}$ with $\|\boldsymbol{w'}\| \leq (2 + \|\boldsymbol{A}\|_{1,\infty} + \|\boldsymbol{a}\|)^{(k+1)\cdot\ell}$. However, since each row sum of $\boldsymbol{A}$ is a row sum of either (i) a matrix $\boldsymbol{A}_j$ for some $j \in \{1, \ldots, r\}$ or (ii) a matrix $\boldsymbol{B}_j$ for some $j \in \{1, \ldots, s\}$, we know that

$$\|\boldsymbol{A}\|_{1,\infty} \leq \max\left(\{\|\boldsymbol{A}_j\|_{1,\infty} \mid j \in \{1, \ldots, r\}\} \cup \{\|\boldsymbol{B}_j\|_{1,\infty} \mid j \in \{1, \ldots, s\}\}\right) \tag{1}$$

and moreover

$$\|\boldsymbol{a}\| \leq \max\left(\{\|\boldsymbol{a}_j\| \mid j \in \{1, \ldots, r\}\} \cup \{\boldsymbol{b}_j \mid j \in \{1, \ldots, s\}\}\right). \tag{2}$$

We pick M to be an upper bound of the right-hand sides of (1) and (2). Then we have

$$\|\boldsymbol{w'}\| \leq (2 + \|\boldsymbol{A}\|_{1,\infty} + \|\boldsymbol{a}\|)^{(k+1)\ell} \leq (2 + 2M)^{(k+1)\ell} \leq (2 + 2M)^{2k\ell}.$$

Thus, setting $c := (2 + 2M)^{2\ell}$ gives us the desired bound: By projecting $\boldsymbol{w'}$ to appropriate components, we obtain a vector $\boldsymbol{z'} \in R^k \cap S$ with $\|\boldsymbol{z'}\| \leq \|\boldsymbol{w'}\| \leq c^k$. ◀

Reachability relations. Our next step is to apply the well-known fact that the reachability relation of runs along a single phase of a counter machine is Presburger-definable. Recall that a *phase* is a run in which no counter reverses. More precisely, our next lemma makes a slightly different (but equally simple) claim: The reachability relation along runs that consist of a single phase followed by a single transition (that leaves that phase) is Presburger-definable.

Given a counter machine CM with d counters, we define the relation $R_{\mathsf{CM}} \subseteq \mathbb{N}^{1+2d} \times \mathbb{N}^{1+2d}$ as follows. We have $(i, \boldsymbol{m}, \boldsymbol{x}, j, \boldsymbol{m'}, \boldsymbol{y}) \in R_{\mathsf{CM}}$ if and only if there exists a run ρ consisting of transitions $t_1 \ldots t_{n+1}$ such that there exists a $\boldsymbol{x} \in \mathbb{N}^d$ and a state $q_k \in Q$ such that

1. the run $t_1 \cdots t_n$ is a run from $(q_i, \boldsymbol{x})$ to $(q_k, \boldsymbol{x}')$ that is consistent with $\boldsymbol{m}$ and
2. t_{n+1} is not consistent with $\boldsymbol{m}$, but t_{n+1} is consistent with $\boldsymbol{m'}$
3. t_{n+1} leads from $(q_k, \boldsymbol{x}')$ to $(q_j, \boldsymbol{y})$.

Thus, R_{CM} is the reachability relation for a run consistent with $\boldsymbol{m}$, plus one transition that is consistent with $\boldsymbol{m'}$, and such that the last step reverses some counter. Moreover, we also encode the mode vector in the components of R_{CM}. The following is entirely standard, but we include a proof for completeness.

▶ **Lemma 3.7.** *For every counter machine* CM*, the relation* R_{CM} *is Presburger-definable.*

Proof. It suffices to prove that for any mode vectors $\boldsymbol{m}$ and $\boldsymbol{m'}$ in $\mathbb{N}^d$ and any $i, j \in \mathbb{N}$, the set $R' \subseteq \mathbb{N}^d \times \mathbb{N}^d$ of all $(\boldsymbol{x}, \boldsymbol{y})$ with $(i, \boldsymbol{m}, \boldsymbol{x}, j, \boldsymbol{m'}, \boldsymbol{y}) \in R_{\mathsf{CM}}$ is Presburger-definable. This is because there are only finitely many choices for $i, j, \boldsymbol{m}, \boldsymbol{m'}$. However, for given $i, j, \boldsymbol{m}, \boldsymbol{m'}$, it is easy to construct a reversal-bounded counter machine CM' with d counters and states s, t such that in CM', we have $s(\boldsymbol{x})$ can reach $t(\boldsymbol{y})$ if and only if $(\boldsymbol{x}, \boldsymbol{y})$ is Presburger-definable. To this end, CM' simulates transitions consistent with the mode $\boldsymbol{m}$ in CM, and then it simulates one more transition consistent with mode $\boldsymbol{m'}$ (which is not consistent with $\boldsymbol{m}$). Since the reachability relation in every reversal-bounded counter machine is Presburger-definable [24], R' is Presburger-definable and the result follows. ◀

With Lemmas 3.5 and 3.7 in hand, we are now ready to prove Proposition 3.1.

Proof of Proposition 3.1. Suppose CM is an f-reversal-bounded counter machine that is not reversal bounded. Without loss of generality, we assume that CM has one counter that is incremented in every step; this counter thus always holds the length of the run.

We consider the following function: $L\colon \mathbb{N} \to \mathbb{N} \cup \{\infty\}$, where for each k, $L(k)$ is the length of the shortest run that contains (at least) k reversals. If there is no such run, we set $L(k) := \infty$. Since CM is not reversal-bounded, we have $L(k) \in \mathbb{N}$ for every $k \in \mathbb{N}$.

Note that any run of CM starts in the mode $\mathbf{1}^d$, because initially, no counter can be decremented. Moreover, all counters are zero initially, and we start in q_0. This motivates the following definition. Let $S \subseteq \mathbb{Z}^{1+2d} \times \mathbb{Z}^{1+2d}$ be the set $S = \{(0, \mathbf{1}^d, \mathbf{0}^d)\} \times \mathbb{Z}^{1+2d}$ of vectors. In other words, S contains those vectors where the first entry is 0, the next d entries contain 1, and the next d entries contain 0 (and the last $1+2d$ entries are unrestricted). Then clearly, a configuration $(q_j, \boldsymbol{y})$ can be reached using exactly k reversals if and only if

$$(0, \mathbf{1}^d, \mathbf{0}^d, j, \boldsymbol{m'}, \boldsymbol{y}) \in R_{\mathsf{CM}}^k$$

for some mode vector $\boldsymbol{m'}$. Thus, by our assumption that CM is not reversal-bounded, we have $R_{\mathsf{CM}}^k \cap S \neq \emptyset$ for every $k \in \mathbb{N}$. Now let c be the constant provided by Lemma 3.5 for R_{CM} and S. Then for every $k \in \mathbb{N}$, there is a vector $\boldsymbol{z}_k \in R_{\mathsf{CM}}^k \cap S$ with $\|\boldsymbol{z}_k\| \leq c^k$. Since the length of each run is encoded in a counter, the length of the corresponding run is thus $\leq \|\boldsymbol{z}_k\| \leq c^k$. In particular:

$$L(k) \leq c^k \ \forall k \in \mathbb{N}$$

Now observe that for every $k \in \mathbb{N}$, we have $f(L(k)) \geq k$, because CM is f-reversal-bounded and there exists a run of length $L(k)$ with at least k reversals. Since f is monotone and $L(k) \leq c^k$, we have $k \leq f(L(k)) \leq f(c^k)$ for every $k \in \mathbb{N}$. Again by monotonicity of f, this implies that for every n, we have $f(n) \geq \log(n)/\log(c)$. Thus f belongs to $\Omega(\log(n))$. ◀

4 Counter machines: Undecidable case

In this section, we prove Proposition 3.2. That is, given any monotone function $f : \mathbb{N} \to \mathbb{N}$ such that f belongs to $\Omega(\log(n))$, we show that $\mathsf{Reach}(f)$ is undecidable. To this end, we show that any counter machine can be simulated by an f-reversal bounded counter machine.

Making f concrete. Instead of working with an arbitrary function f in $\Omega(\log(n))$, it will be convenient to work with a concrete function of the form $c \cdot \log$. Let us now see why it suffices to show undecidability of $\mathsf{Reach}(c \cdot \log)$. The fact that f belongs to $\Omega(\log n)$ means that there exist $c, k \in \mathbb{N}$ such that $f(n) \geq c \cdot \log(n)$ for all $n \geq k$. This implies that $\mathsf{Reach}(c \cdot \log)$ reduces to $\mathsf{Reach}(f)$: Given a $c \cdot \log$-reversal-bounded machine, we modify it to begin with k fresh steps that do not reverse any counter. Then clearly, the new machine is f-reversal-bounded. Hence, to show $\mathsf{Reach}(f)$ is undecidable it suffices to show that $\mathsf{Reach}(c \cdot \log)$ is undecidable.

Making the constant concrete. It will be even more convenient to show undecidability in the case $c = 11$. Our next lemma argues that this suffices.

▶ **Lemma 4.1.** *For every monotone function $g \colon \mathbb{N} \to \mathbb{N}$ and every constant $c \in \mathbb{N}$,* $\mathsf{Reach}(g)$ *is decidable if and only if* $\mathsf{Reach}(c \cdot g)$ *is decidable*

Proof. Since every g-reversal-bounded counter machine is also a $c \cdot g$-reversal-bounded machine, $\mathsf{Reach}(g)$ trivially reduces to $\mathsf{Reach}(c \cdot g)$. For the converse, we will prove that $\mathsf{Reach}(c \cdot g)$ reduces to $\mathsf{Reach}(g)$. Given any $c \cdot g$-reversal bounded counter machine $\mathsf{CM} = (d, Q, \Delta, q_0)$ construct $\mathsf{CM}_c = (d, Q_c, \Delta_c, (q_0, 0))$ such that it adds c dummy transitions between any two consecutive steps of CM. Formally, $Q_c = Q \times \{0, \dots c\}$, and Δ_c is the smallest set containing transitions, $((q, i), \mathbf{0}^d, (q, i + 1))$ (where $i \in \{0, \dots, c - 1\}$), and $((q, c), \mathit{inst}, (q', 0))$ where $(q, \mathit{inst}, q') \in \Delta$. Then, $c \cdot g$-reversal-boundedness of the input machine clearly implies g-reversal-boundedness of the resulting machine. Moreover, $(q, \boldsymbol{w})$ is reachable from $(q_0, \mathbf{0})$ in CM if and only if $((q, 0), \boldsymbol{w})$ is reachable from $((q_0, 0), \mathbf{0})$. ◀

Now Lemma 4.1 indeed implies that, decidability of $\mathsf{Reach}(11 \cdot \log)$ is equivalent to decidability of $\mathsf{Reach}(c \cdot \log)$. Hence, we just need to show that $\mathsf{Reach}(11 \cdot \log)$ is undecidable.

Proving Reach(11 · log) is undecidable – Main Step. In this step, we show that any counter machine $\mathsf{CM} = (d, Q, \Delta, q_0)$ can be simulated by an $11 \cdot \log$-reversal bounded counter machine $\mathsf{CM}' = (d', Q', \Delta', q_0')$. Like Lemma 4.1, we will add dummy steps between any two steps simulating CM. But unlike Lemma 4.1, we need to add more and more dummy steps after simulating every step of CM. More precisely, between the simulation of the N^{th} and $(N + 1)^{th}$ step of CM, we need to add exponentially many (in N) dummy steps, using reversals which are polynomial in N. To do this, we construct a gadget for every state $q \in Q$, which does the above, before simulating any outgoing transition from q using 4 additional auxiliary counters. We now give the formal construction followed by the proof of correctness.

Formal construction of CM′. Given $\mathsf{CM} = (d, Q, \Delta, q_0)$, we construct a counter machine $\mathsf{CM}' = (Q', d', \Delta', (q_0, 0))$ where $Q' = Q \times \{0, \dots 10\}$, $d' = d + 4$, and Δ' is defined a little later. For the sake of simplicity, we assume $d = 2$. Hence, counters $\mathsf{C}_3, \mathsf{C}_4, \mathsf{C}_5, \mathsf{C}_6$ are the auxiliary counters used to add the required dummy steps.

Transitions simulating steps of CM. For any state $q \in Q$, transitions exiting q in CM are simulated by transitions exiting $(q, 10) \in Q'$. More precisely, $(q, [a_1, a_2], q') \in \Delta$ iff $((q, 10), [a_1, a_2, 0, 0, 0], (q', 0)) \in \Delta'$.

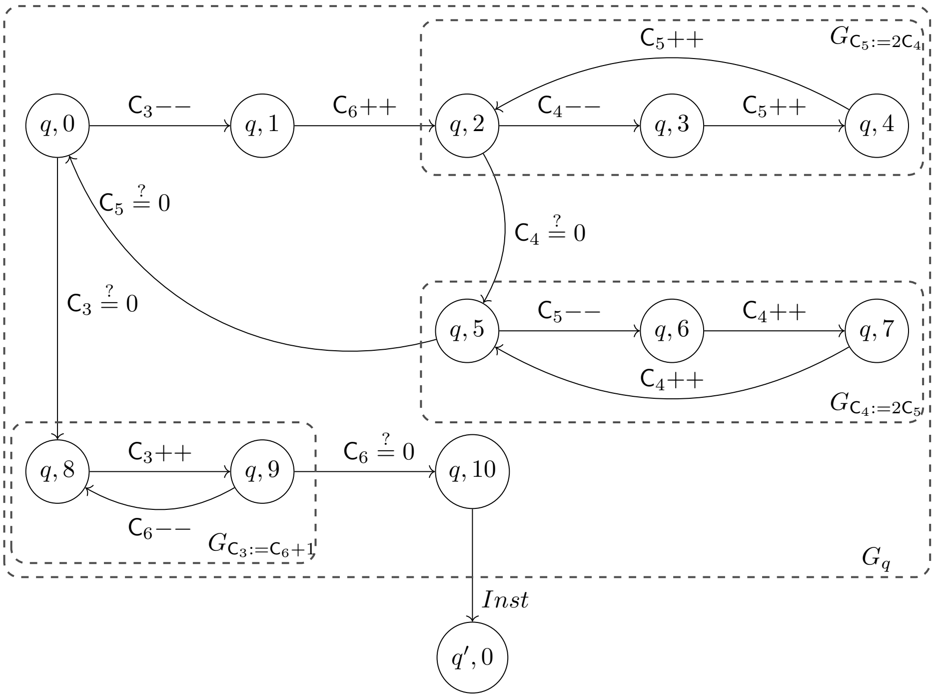

Figure 2 Diagram showing simulation of transition $q \xrightarrow{Inst} q'$. G_q is the gadget for state q. For the sake of readability, for all $1 \leq i \leq 6$, we write $\mathsf{C}_i --$ ($\mathsf{C}_i ++$) for vector $\boldsymbol{z}$ such that for all $1 \leq j \leq i$, $\boldsymbol{z}[j] = -1$ ($\boldsymbol{z}[j] = +1$) if $j = i$ and $\boldsymbol{z}[j] = 0$, otherwise.

Transitions adding dummy steps. The rest of the transitions are those appearing within the gadgets. For every state $q \in Q$, we construct a gadget G_q, as shown in Figure 2, such that before imitating any outgoing transition from state q, G_q induces the required number of dummy steps.

Enforcing long runs with less reversals, G-Invariant Property. We say that our gadget satisfies the *G-Invariant* property iff on entering the gadget with counter values $[a, b, s, t, 0, 0]$ we exit the gadget with the values of these counters as $[a, b, s + 1, 4^s \cdot t, 0, 0]$. Moreover, the number of reversals made within this gadget (including the transition outgoing from the gadget) is at most $4s + 5$.

Observe the gadget in Figure 2. The gadget adds dummy transitions before it simulates any transition of CM outgoing from p as follows: It performs an identical transition from $(q, 10)$, namely, $(q, 10) \xrightarrow{Inst} (q', 0)$. The transitions from $(q, 0)$ to $(q, 10)$ enforces the runs to be long enough to make sure that CM' is a $c \cdot$ log-reversal bounded counter machine. The analysis of CM' being 11 log-reversal bounded is shown a little later. More specifically, the run from $(q, 0)$ to $(q, 10)$ satisfies the G-Invariant property. Let the initial values of these auxiliary counters be $\mathsf{C}_3 = s, \mathsf{C}_4 = t, \mathsf{C}_5 = 0, \mathsf{C}_6 = 0$. Notice the following:

- When the control enters $(q, 2)$ it remains in the part of the gadget $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ and exits this part by entering into $(q, 5)$ with the new value of C_4 as 0 and C_5 incremented by double of what the value of C_4 was when the control first entered the $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ part. By symmetry, part $G_{\mathsf{C}_4:=2\mathsf{C}_5}$ increments C_4 by the double of the old value of C_5.
- For every decrement of C_3, C_6 is incremented, and the control enters $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ followed by entering $G_{\mathsf{C}_4:=2\mathsf{C}_5}$. After this the control again comes back to $(q, 0)$.
- Hence, for each decrement of C_3, $\mathsf{C}_4 := 4\mathsf{C}_4$, C_6 is incremented and $\mathsf{C}_5 := 0$.

- The above continues till $\mathsf{C}_3 = 0$ (i.e. s times) after which $\mathsf{C}_6 = s$, $\mathsf{C}_3 = 0$, $\mathsf{C}_5 = 0$, $\mathsf{C}_4 = 4^s \cdot t$ and the control enters $(q, 8)$.
- From $(q, 8)$, for every increment of C_3, C_6 is decremented, and this continues until $\mathsf{C}_6 = 0$ at $(q, 9)$ after which $\mathsf{C}_3 = s + 1$ and the control enters $(q, 10)$. This is followed by simulating instructions from q as in the original machine CM.
- Hence the values of the auxiliary counters at $(q, 10)$ will be $\mathsf{C}_3 = s+1, \mathsf{C}_4 = 4^s \cdot t, \mathsf{C}_5 = 0, \mathsf{C}_6 = 0$, implying the satisfaction of the G-Invariant property for the given gadget.
- Finally notice the number of reversals made within this gadget. There are 2 reversals between entering and exiting the part $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ of the gadget (the very first time C_5 is incremented and the very first time C_4 is decremented). Similarly, there are 2 reversals between entering and exiting the part $G_{\mathsf{C}_4:=2\mathsf{C}_5}$ and $G_{\mathsf{C}_3:=\mathsf{C}_6+1}$. The total number of times the control passes through the parts $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ and $G_{\mathsf{C}_5:=2\mathsf{C}_4}$ is s each. Similarly, the control passes exactly once from the part $G_{\mathsf{C}_3:=\mathsf{C}_6+1}$ causing $4s + 2$ reversals. Moreover, there is exactly one reversal when transitions $(q, 0) \to (q, 1)$ and $(q, 1) \to (q, 2)$ are taken the very first time within the gadget. Finally, there can be at most one reversal while simulating $Inst$. Hence, there are at the most $4s + 5$ reversals.

This technique of moving tokens between different auxiliary counters and doubling was also used in [23] to show that VASS with 3 or more dimensions can have non-semilinear reachable sets. As they were interested in reach sets, they did not require all the runs to be large. On the other hand, we need long runs to show undecidability and with infrequent reversals. Hence, unlike [23], we need zero tests.

It is interesting to note that a VASS gadget satisfying the G-Invariant property cannot exist. If it did, it would imply that there is a $\log(n)$-reversal bounded VASS that does not have a constant number of reversals, which contradicts our Theorem 1.2.

Proving CM' is $11 \cdot \log$-reversal bounded. Assume that we start with $[a, b, 1, 1, 0, 0]$ as our initial configuration. Consider any run $\rho = (q_0, \boldsymbol{v}_0) \dots (q_n, \boldsymbol{v}_n)$ of CM', where $\boldsymbol{v}_0 = \mathbf{0}$.

Lower bounding the length of the run. Suppose ρ enters the gadget during the $(N+1)^{th}$ time (i.e., it has simulated N instructions of the CM and is about to simulate the $(N+1)^{th}$ instruction). Let $(q_0, \boldsymbol{v}_0) \dots (q_\ell, \boldsymbol{v}_\ell)$ be the prefix of ρ such that, at the ℓ^{th} step, the N^{th} instruction of the CM was simulated. As our gadget preserves the G-Invariant property, we can inductively show that the value of C_4 in $\boldsymbol{v}_\ell$ i.e. $\boldsymbol{v}_\ell[4]$ is

$$t_N = 4 \cdot 4^2 \cdot 4^3 \cdot \dots \cdot 4^N = 2^{N^2+N}.$$

Since in the initial configuration, C_4 equals 1 and in each step of CM', C_4 is increased by at most 1, this means $\ell_N \geq t_N$.[1]

Upper bounding the number of reversals. The number of reversals k in ρ is at most the maximum possible number of reversals k_{N+1} at the time when the control exits the $N + 1^{th}$ gadget. That is,

$$k_{N+1} = \sum_{i=1}^{N+1} (4i + 5) = 4 \cdot \frac{(N+1)(N+2)}{2} + 5(N+1)$$
$$= 2N^2 + 6N + 4 + 5N + 5 \leq 11N^2 + 11N \quad (3)$$

for any $N \geq 1$.[2] See Figure 3 for intuition.

---

[1] Note that the value of ℓ_N is much larger than t_N for higher values of N. In fact the number of dummy steps added within the N^{th} entry of the gadget itself is at least t_N. This is because C_4 becomes 0 at least once, within the gadget. Hence, to again reach the value t_N, it needs to execute at least t_N steps. Hence, the number of steps performed within the gadget itself is at least t_N.

[2] For $N \leq 1$, any counter machine will have zero reversals. Hence, the value of the function bounding the frequency of the reversal is important only for $N \geq 1$

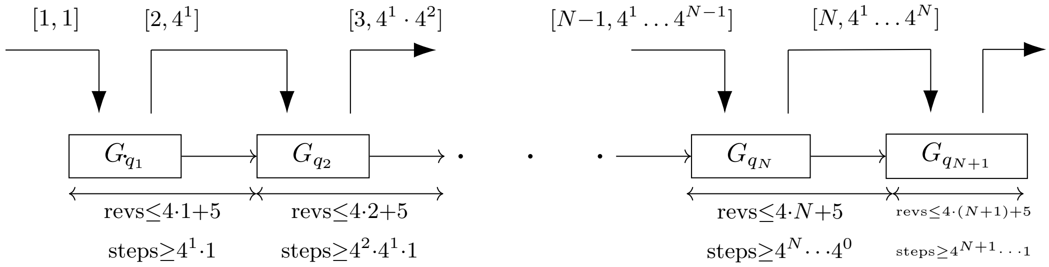

Figure 3 Figure showing the evaluation of counter values of auxiliary counters C_3 and C_4, which are used to estimate a lower bound on the length of the runs. The lower bounds on the steps, and the upper bound on the number of reversals performed within a gadget is mentioned below the corresponding gadget.

Hence, $k \leq k_{N+1} \leq 11 \cdot \log(t_N) \leq 11 \cdot \log(\ell) \leq 11 \cdot \log(n)$. Hence, CM' is $11 \cdot \log$-reversal bounded.

We have thus shown that $\mathsf{Reach}(11 \cdot \log)$ is undecidable and therefore, by Lemma 4.1, $\mathsf{Reach}(\log)$ is undecidable.

5 Vector addition systems

In this section, we prove Theorem 1.2. Let $\mathcal{V} = (d, Q, \Delta, q_0)$ be a d-dimensional vector addition system. A *step* is a triple $((p, \boldsymbol{u}), \boldsymbol{a}, (q, \boldsymbol{v})) \in (Q \times \mathbb{N}^d) \times \Delta \times (Q \times \mathbb{N}^d)$ with $\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{a}$. A *run* is a sequence

$$((q_0, \boldsymbol{u}_1), \boldsymbol{a}_1, (q_1, \boldsymbol{v}_1))((q_1, \boldsymbol{u}_2), \boldsymbol{a}_2, (q_2, \boldsymbol{v}_2)) \cdots ((q_{n-1}, \boldsymbol{u}_n), \boldsymbol{a}_n, (q_n, \boldsymbol{v}_n))$$

of steps such that $\boldsymbol{v}_i = \boldsymbol{u}_{i+1}$ for each $1 \leq i < n$ and $\boldsymbol{u}_1 = \boldsymbol{0}$.

We now define the notion of Well-Quasi Ordering (WQO), which is useful for the proof.

Well-Quasi Ordering and Higman's Lemma. We fix a set X and a relation $\leq$ over X. $\leq$ is said to be a *preorder/quasi order* if it is reflexive ($\forall u \in X : u \leq u$), and transitive ($\forall u, v, w \in X : u \leq v$ and $v \leq w$ implies $u \leq w$). A preorder is a *well-quasi order* (WQO) iff for every infinite sequence $u_1 u_2 \ldots$ over X, there exists a pair $i < j$ such that $u_i \leq u_j$. Let $\preceq$ be an order relation over sequences of X (that is, over X^*) defined as follows. We write $u_1 u_2 \ldots u_m \preceq u'_1 u'_2 \ldots u'_n$ if there exists a strictly increasing function from $\kappa : \{1, \ldots m\} \mapsto \{1, \ldots n\}$ such that for all $i \in \{1, \ldots, m\}$, $u_i \leq u'_{\kappa(i)}$. With this notation, we can phrase Higman's lemma as follows.

▶ **Lemma 5.1** (Higman's Lemma [22]). *If $\leq$ is a WQO over X, then $\preceq$ is a WQO over X^*.*

Run embeddings. The proof of Theorem 1.2 will employ the concept of run embeddings, which was introduced by Jančar [27] and Leroux [30]. Towards its definition, we first define an ordering on steps. Given steps $s = ((p, \boldsymbol{u}), \boldsymbol{a}, (q, \boldsymbol{v}))$ and $s' = ((p', \boldsymbol{u}'), \boldsymbol{a}', (q', \boldsymbol{v}'))$, we write $s \leq s'$ if $p' = p$, $q' = q$, $\boldsymbol{a}' = \boldsymbol{a}$, and $\boldsymbol{u} \leq \boldsymbol{u}'$. Here, $\boldsymbol{u} \leq \boldsymbol{u}'$ means that $\boldsymbol{u}(i) \leq \boldsymbol{u}'(i)$ for every $i \in \{1, \ldots, d\}$. Note that $s \leq s'$ implies $\boldsymbol{v} \leq \boldsymbol{v}'$.

The ordering on steps now induces an embedding ordering on runs. Suppose $\rho = s_1 \cdots s_m$ and $\rho' = s'_1 \cdots s'_n$ are runs. An *embedding of ρ in ρ'* is a strictly monotone map $\sigma \colon \{1, \ldots, m\} \to \{1, \ldots, n\}$ such that (i) $\sigma(1) = 1$ and (ii) $s_i \leq s'_{\sigma(i)}$ for every $i \in \{1, \ldots, m\}$.

Run amalgamation. A key property of the run embedding is the *amalgamation property* as observed by Leroux and Schmitz [34]. Aside from our paper, amalgamation of runs has been used in several other works for constructing runs in VASS and related models [2,10,12,33]. If $\boldsymbol{c} = (p, \boldsymbol{u}) \in Q \times \mathbb{N}^d$ is a configuration of a VASS and $\boldsymbol{w} \in \mathbb{N}^d$ is a vector, then by $\boldsymbol{c} + \boldsymbol{w}$ we denote the configuration $(p, \boldsymbol{u} + \boldsymbol{w})$. If σ is an embedding of ρ in ρ', then we can define a new run as follows. First, we write ρ as $\boldsymbol{c}_0 \xrightarrow{t_1} \boldsymbol{c}_1 \xrightarrow{t_2} \cdots \xrightarrow{t_n} \boldsymbol{c}_n$ with configurations $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_n$ and transitions $t_1, \ldots, t_n$. Then since σ is an embedding, we can write ρ' as

$$\boldsymbol{c}_0 + \boldsymbol{w}_0 \xrightarrow{\tau_0} \boldsymbol{c}_0 + \boldsymbol{w}_1 \xrightarrow{t_1} \boldsymbol{c}_1 + \boldsymbol{w}_1 \xrightarrow{\tau_1} \boldsymbol{c}_1 + \boldsymbol{w}_2 \xrightarrow{t_2} \boldsymbol{c}_2 + \boldsymbol{w}_2 \xrightarrow{\tau_2} \cdots \xrightarrow{\tau_{n-1}} \boldsymbol{c}_{n-1} + \boldsymbol{w}_k \xrightarrow{t_n} \boldsymbol{c}_n + \boldsymbol{w}_n \xrightarrow{\tau_n} \boldsymbol{c}_n + \boldsymbol{w}_{n+1}$$

where $\tau_0, \ldots, \tau_n$ are transition sequences and $\boldsymbol{w}_0, \ldots, \boldsymbol{w}_{n+1}$ are vectors in $\mathbb{N}^d$ with $\boldsymbol{w}_0 = \boldsymbol{0}$. Now we define a new run ρ'^{σ} as

$$\begin{aligned}
&\boldsymbol{c}_0 + \boldsymbol{w}_0 \xrightarrow{\tau_0} \boldsymbol{c}_0 + \boldsymbol{w}_1 \xrightarrow{\tau_0} \boldsymbol{c}_0 + \boldsymbol{w}_1 + \boldsymbol{w}_1 \xrightarrow{t_1} \boldsymbol{c}_1 + \boldsymbol{w}_1 + \boldsymbol{w}_1 \\
&\quad \xrightarrow{\tau_1} \boldsymbol{c}_1 + \boldsymbol{w}_1 + \boldsymbol{w}_2 \xrightarrow{\tau_1} \boldsymbol{c}_1 + \boldsymbol{w}_2 + \boldsymbol{w}_2 \xrightarrow{t_2} \boldsymbol{c}_2 + \boldsymbol{w}_2 + \boldsymbol{w}_2 \\
&\quad \vdots \\
&\quad \xrightarrow{\tau_{n-1}} \boldsymbol{c}_{n-1} + \boldsymbol{w}_{n-1} + \boldsymbol{w}_n \xrightarrow{\tau_{n-1}} \boldsymbol{c}_{n-1} + \boldsymbol{w}_n + \boldsymbol{w}_n \xrightarrow{t_n} \boldsymbol{c}_n + \boldsymbol{w}_n + \boldsymbol{w}_n \\
&\quad \xrightarrow{\tau_n} \boldsymbol{c}_n + \boldsymbol{w}_{n+1} + \boldsymbol{w}_n \xrightarrow{\tau_n} \boldsymbol{c}_n + \boldsymbol{w}_{n+1} + \boldsymbol{w}_{n+1}.
\end{aligned}$$

Thus, the action sequence of this run is $\tau_0\tau_0 t_1 \tau_1\tau_1 \cdots t_n \tau_n \tau_n$. Of course, this process can be repeated. If we do this m times, we obtain a run with the action sequence $\tau_0^m t_1 \tau_1^m \cdots t_n \tau_n^m$.

If there exists an embedding of a run ρ into ρ', then we write $\rho \trianglelefteq \rho'$. Since the ordering $\leq$ on steps is a well-quasi ordering and $\trianglelefteq$ is just the embedding relation induced by $\leq$, the following is a direct consequence of Higman's Lemma (Lemma 5.1):

▶ **Lemma 5.2.** *On the set of runs of $\mathcal{V}$, the ordering $\trianglelefteq$ is a well-quasi ordering.*

Reversal increasing embeddings. If ρ and ρ' are as above and σ is an embedding of ρ into ρ', then we say that σ is *reversal increasing* if for some $i \in [0, n]$, the action sequence τ_i contains at least one reversal. We use this notion to prove the first step of Theorem 1.2.

▶ **Lemma 5.3.** *Let $f \colon \mathbb{N} \to \mathbb{N}$ be a monotone function that is not essentially bounded for VASS. Then there is a constant $c \in \mathbb{N}$ such that $f(n) \geq (n - c)/c$ for every $n \in \mathbb{N}$, $n \geq c$.*

Proof. Let $f \colon \mathbb{N} \to \mathbb{N}$ be any monotone function which is not essentially bounded for VASS. Then there exists a VASS $\mathcal{V}$ that is f-reversal bounded, but not reversal bounded.

We first claim that then there exist two runs ρ and ρ' such that ρ embeds into ρ' via a reversal increasing embedding. Towards a contradiction, suppose there is no reversal increasing embedding between runs. Since $\trianglelefteq$ is a well-quasi ordering, the set of runs has a finite set $\{\rho_1, \ldots, \rho_r\}$ of minimal runs. Let $R \in \mathbb{N}$ be the maximal number of reversals within the runs $\rho_1, \ldots, \rho_r$. Since every run of $\mathcal{V}$ embeds at least one of the minimal runs $\rho_1, \ldots, \rho_r$ and every embedding is not reversal increasing, every run of $\mathcal{V}$ has at most r reversals. Thus, f is essentially bounded, against our assumption. This proves our claim. Thus, we have runs ρ and ρ' such that ρ embeds into ρ' via some reversal increasing embedding σ.

Let us now show that f belongs to $\Omega(n)$. Suppose ρ and ρ' are our constructed runs such that ρ has action sequence $t_1 \cdots t_N$. Moreover, for each $m \in \mathbb{N}$, let $\tau_0^m t_1 \tau_1^m \cdots t_N \tau_N^m$ be the action sequences resulting from amalgamating ρ and ρ' exactly m times. Since σ is reversal increasing, observe that $\tau_0^m t_1 \tau_1^m \cdots t_N \tau_N^m$ has at least m reversals. Moreover, if we set $e := |\tau_0| + \cdots + |\tau_N|$, then the length of $\tau_0^m t_1 \tau_1^m \cdots t_N \tau_N^m$ is $N + m \cdot e$.

We have thus constructed, for every $m \in \mathbb{N}$, a run of length $N + m \cdot e$ with at least m reversals. We now claim that with $c := e + N$, we have indeed $f(n) \geq (n - c)/c$ for every $n \in \mathbb{N}$, $n \geq c$. Our constructed runs yield $f(N + me) \geq m$ for every $m \geq 1$. Therefore, if $n > N$ and $n \equiv N \pmod{e}$, then $f(n) \geq (n - N)/e$. Now for every $n \in \mathbb{N}$, $n \geq c$, we can pick $i \in [1, e]$ such that $n - i \equiv N \pmod{e}$ and thus $f(n - i) \geq (n - i - N)/e$. Since f is monotone, this implies $f(n) \geq f(n - i) \geq (n - i - N)/e \geq (n - c)/c$, as claimed. ◀

Proof of Theorem 1.2. Suppose $f : \mathbb{N} \to \mathbb{N}$ is a monotone function that is not essentially bounded for VASS. According to Lemma 5.3, there is a constant $c \in \mathbb{N}$ such that $f(n) \geq (n - c)/c$ for every $n \in \mathbb{N}$, $n \geq c$. This already implies that f belongs to $\Omega(n)$. It remains to argue that $\mathsf{Reach}_{\mathsf{VASS}}(f)$ is Ackermann-hard. This is simple: Given a VASS $\mathcal{V}_0$, we turn it into a VASS $\mathcal{V}_1$ that begins by taking c empty steps, and afterwards, it simulates each step of $\mathcal{V}_0$ by first taking c empty steps (i.e. steps that do not change any counters). Then $\mathcal{V}_1$ is clearly f-reversal-bounded, because in any run of length n, there are at most $(n - c)/c$ steps that add a non-zero vector. Since reachability in VASS is Ackermann-complete [11, 32, 35], this shows that reachability in f-reversal-bounded VASS is also Ackermann-complete. ◀

References

1 Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Philippe Schnoebelen. Flat acceleration in symbolic model checking. In Doron A. Peled and Yih-Kuen Tsay, editors, *Automated Technology for Verification and Analysis, Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005, Proceedings*, volume 3707 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2005. doi:10.1007/11562948_35.

2 Pascal Baumann, Flavio D'Alessandro, Moses Ganardi, Oscar H. Ibarra, Ian McQuillan, Lia Schütze, and Georg Zetzsche. Unboundedness problems for machines with reversal-bounded counters. In Orna Kupferman and Pawel Sobocinski, editors, *Foundations of Software Science and Computation Structures - 26th International Conference, FoSSaCS 2023, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2023, Paris, France, April 22-27, 2023, Proceedings*, volume 13992 of *Lecture Notes in Computer Science*, pages 240–264. Springer, 2023. doi:10.1007/978-3-031-30829-1_12.

3 Marcello M. Bersani and Stéphane Demri. The complexity of reversal-bounded model-checking. In Cesare Tinelli and Viorica Sofronie-Stokkermans, editors, *Frontiers of Combining Systems, 8th International Symposium, FroCoS 2011, Saarbrücken, Germany, October 5-7, 2011. Proceedings*, volume 6989 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2011. doi:10.1007/978-3-642-24364-6_6.

4 Alin Bostan, Arnaud Carayol, Florent Koechlin, and Cyril Nicaud. Weakly-unambiguous Parikh automata and their link to holonomic series. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 114:1–114:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.114.

5 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Affine Parikh automata. *RAIRO Theor. Informatics Appl.*, 46(4):511–545, 2012. doi:10.1051/ita/2012013.

6 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Bounded Parikh automata. *Int. J. Found. Comput. Sci.*, 23(8):1691–1710, 2012. doi:10.1142/S0129054112400709.

7 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Unambiguous constrained automata. *Int. J. Found. Comput. Sci.*, 24(7):1099–1116, 2013. doi:10.1142/S0129054113400339.

8 Michaël Cadilhac, Andreas Krebs, and Pierre McKenzie. The algebraic theory of Parikh automata. *Theory Comput. Syst.*, 62(5):1241–1268, 2018. doi:10.1007/s00224-017-9817-2.

9 Lorenzo Clemente, Wojciech Czerwinski, Slawomir Lasota, and Charles Paperman. Regular separability of Parikh automata. 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 117:1–117:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.ICALP.2017.117.

10 Lorenzo Clemente, Wojciech Czerwinski, Slawomir Lasota, and Charles Paperman. Separability of reachability sets of vector addition systems. In Heribert Vollmer and Brigitte Vallée, editors, *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, volume 66 of *LIPIcs*, pages 24:1–24:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.STACS.2017.24.

11 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.

12 Wojciech Czerwinski and Georg Zetzsche. An approach to regular separability in vector addition systems. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 341–354. ACM, 2020. doi:10.1145/3373718.3394776.

13 Zhe Dang, Oscar H. Ibarra, and Pierluigi San Pietro. Liveness verification of reversal-bounded multicounter machines with a free counter. In Ramesh Hariharan, Madhavan Mukund, and V. Vinay, editors, *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science, 21st Conference, Bangalore, India, December 13-15, 2001, Proceedings*, volume 2245 of *Lecture Notes in Computer Science*, pages 132–143. Springer, 2001. doi:10.1007/3-540-45294-X_12.

14 Stéphane Demri and Arnaud Sangnier. When model-checking freeze LTL over counter machines becomes decidable. In C.-H. Luke Ong, editor, *Foundations of Software Science and Computational Structures, 13th International Conference, FOSSACS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6014 of *Lecture Notes in Computer Science*, pages 176–190. Springer, 2010. doi:10.1007/978-3-642-12032-9_13.

15 Alain Finkel and Arnaud Sangnier. Reversal-bounded counter machines revisited. In Edward Ochmanski and Jerzy Tyszkiewicz, editors, *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings*, volume 5162 of *Lecture Notes in Computer Science*, pages 323–334. Springer, 2008. doi:10.1007/978-3-540-85238-4_26.

16 Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

17 Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Remarks on Parikh-recognizable omega-languages, 2023. arXiv:2307.07238.

18 Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. Parikh automata over infinite words. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India*, volume 250 of *LIPIcs*, pages 40:1–40:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.40.

19 Eitan M. Gurari and Oscar H. Ibarra. The complexity of decision problems for finite-turn multicounter machines. In Shimon Even and Oded Kariv, editors, *Automata, Languages and Programming, 8th Colloquium, Acre (Akko), Israel, July 13-17, 1981, Proceedings*, volume 115 of *Lecture Notes in Computer Science*, pages 495–505. Springer, 1981. doi:10.1007/3-540-10843-2_39.

20 Christoph Haase. Subclasses of Presburger arithmetic and the weak EXP hierarchy. In Thomas A. Henzinger and Dale Miller, editors, *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14–18, 2014*, pages 47:1–47:10. ACM, 2014. doi:10.1145/2603088.2603092.

21 Matthew Hague and Anthony Widjaja Lin. Model checking recursive programs with numeric data types. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification – 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 743–759. Springer, 2011. doi:10.1007/978-3-642-22110-1_60.

22 Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(1):326–336, 1952.

23 John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979. doi:10.1016/0304-3975(79)90041-0.

24 Oscar H Ibarra. Reversal-bounded multicounter machines and their decision problems. *Journal of the ACM (JACM)*, 25(1):116–133, 1978.

25 Oscar H. Ibarra, Jianwen Su, Zhe Dang, Tevfik Bultan, and Richard A. Kemmerer. Conter machines: Decidable properties and applications to verification problems. In Mogens Nielsen and Branislav Rovan, editors, *Mathematical Foundations of Computer Science 2000, 25th International Symposium, MFCS 2000, Bratislava, Slovakia, August 28 – September 1, 2000, Proceedings*, volume 1893 of *Lecture Notes in Computer Science*, pages 426–435. Springer, 2000. doi:10.1007/3-540-44612-5_38.

26 Matthias Jantzen and Alexy Kurganskyy. Refining the hierarchy of blind multicounter languages and twist-closed trios. *Inf. Comput.*, 185(2):159–181, 2003. doi:10.1016/S0890-5401(03)00087-7.

27 Petr Jančar. Decidability of a temporal logic problem for Petri nets. *Theor. Comput. Sci.*, 74(1):71–93, 1990. doi:10.1016/0304-3975(90)90006-4.

28 Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 – July 4, 2003. Proceedings*, volume 2719 of *Lecture Notes in Computer Science*, pages 681–696. Springer, 2003. doi:10.1007/3-540-45061-0_54.

29 Michel Latteux. Cônes rationnels commutatifs. *J. Comput. Syst. Sci.*, 18(3):307–333, 1979. doi:10.1016/0022-0000(79)90039-4.

30 Jérôme Leroux. Vector addition system reachability problem: a short self-contained proof. In Thomas Ball and Mooly Sagiv, editors, *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2011, Austin, TX, USA, January 26-28, 2011*, pages 307–316. ACM, 2011. doi:10.1145/1926385.1926421.

31 Jérôme Leroux. Presburger vector addition systems. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 23–32. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.7.

32 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.

33 Jérôme Leroux, M. Praveen, Philippe Schnoebelen, and Grégoire Sutre. On functions weakly computable by pushdown Petri nets and related systems. *Log. Methods Comput. Sci.*, 15(4), 2019. doi:10.23638/LMCS-15(4:15)2019.

34 Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 56–67. IEEE Computer Society, 2015. doi:10.1109/LICS.2015.16.

35 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.

36 Jérôme Leroux and Grégoire Sutre. Flat counter automata almost everywhere! In Doron A. Peled and Yih-Kuen Tsay, editors, *Automated Technology for Verification and Analysis, Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005, Proceedings*, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005. doi:10.1007/11562948_36.

37 Marvin L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics*, 74(3):437–455, 1961. URL: http://www.jstor.org/stable/1970290.

38 Loic Pottier. Minimal solutions of linear Diophantine systems: Bounds and algorithms. In Ronald V. Book, editor, *Rewriting Techniques and Applications, 4th International Conference, RTA-91, Como, Italy, April 10-12, 1991, Proceedings*, volume 488 of *Lecture Notes in Computer Science*, pages 162–173. Springer, 1991. doi:10.1007/3-540-53904-2_94.

39 Georg Zetzsche. The complexity of downward closure comparisons. 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 123:1–123:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.123.