



Coverability in 2-VASS with One Unary Counter is in NP [★]

Filip Mazowiecki¹ , Henry Sinclair-Banks² , and Karol Węgrzycki³ 

¹ University of Warsaw, Warsaw, Poland
f.mazowiecki@mimuw.edu.pl

² Centre for Discrete Mathematics and its Applications (DIMAP) & Department of Computer Science, University of Warwick, Coventry, UK
h.sinclair-banks@warwick.ac.uk

³ Saarland University and Max Planck Institute for Informatics, Saarbrücken, Germany
wegrzycki@cs.uni-saarland.de

Abstract. Coverability in Petri nets finds applications in verification of safety properties of reactive systems. We study coverability in the equivalent model: Vector Addition Systems with States (VASS).

A k -VASS can be seen as k counters and a finite automaton whose transitions are labelled with k integers. Counter values are updated by adding the respective transition labels. A configuration in this system consists of a state and k counter values. Importantly, the counters are never allowed to take negative values. The coverability problem asks whether one can traverse the k -VASS from the initial configuration to a configuration with at least the counter values of the target.

In a well-established line of work on k -VASS, coverability in 2-VASS is already PSPACE-hard when the integer updates are encoded in binary. This lower bound limits the practicality of applications, so it is natural to focus on restrictions. In this paper we initiate the study of 2-VASS with one unary counter. Here, one counter receives binary encoded updates and the other receives unary encoded updates. Our main result is that coverability in 2-VASS with one unary counter is in NP. This improves upon the inherited state-of-the-art PSPACE upper bound. Our main technical contribution is that one only needs to consider runs in a certain compressed linear form.

Keywords: Vector Addition Systems · Coverability Problem · Linear Path Schemes

1 Introduction

Vector Addition Systems with States (VASS) are a well-studied class of infinite-state systems (see the survey [37]). These are finite automata with counters that

[★] Filip Mazowiecki is supported by the ERC grant INFSYS, agreement no. 950398. Henry Sinclair-Banks is supported by EPSRC Standard Research Studentship (DTP), grant EP/T5179X/1. Karol Węgrzycki is supported by the ERC grant TI-PEA agreement no. 850979.

can be updated, but are never allowed to take negative values. Thus, a configuration consists of a state and a vector over the natural numbers. The central decision problems are the reachability and coverability problems. The reachability problem asks whether from a given start configuration one can reach the target configuration. The coverability problem is the same except that the target configuration need not be reached exactly, counter values are allowed to be greater. Both problems are not only mathematically elegant, but they have interesting theoretical applications [7] and implementations [6]. Coverability is provably a simpler problem that is better suited for applications; reachability tools are mostly applied to coverability benchmarks [14]. Yet coverability has applications in the verification of safety conditions in reactive systems [17,21]. Such systems may require additional data structures to be accurately represented, like counters for example. Safety conditions often boil down to whether a particular state can be reached as opposed to a particular configuration [8].

Coverability and reachability have been studied for decades. The equivalent model of Petri nets was introduced already in the sixties [34]. For general VASS, Lipton proved in 1976 an EXPSPACE lower bound that applies to both coverability and reachability [31]. Two years later, Rackoff proved a matching EXPSPACE upper bound for coverability [35]. Later in 1981, Mayr proved that reachability is decidable [32] without providing an upper bound for the algorithm. The construction was simplified by Kosaraju [24] and Lambert [25], and a recent series of papers by Leroux and Schmitz ended in 2019 by proving an Ackermann upper bound [27]. A matching Ackermann lower bound was published in 2021 by two independent groups [12,26].

Plenty of attention has been given to VASS with fixed dimension, that is when the number of counters k is invariable, denoted k -VASS. For fixed dimension VASS it matters much whether the counter updates are encoded in unary or binary. Already, Rackoff gives NL and PSPACE upper bounds for coverability in unary encoded and binary encoded k -VASS, respectively [35]. The coverability problem where there are no counters is just directed graph reachability that is NL-complete [3]. Thus, coverability in unary encoded k -VASS is NL-complete, for every fixed k . Coverability in binary encoded 1-VASS is in NC^2 [2], it can therefore be decided in deterministic polynomial time. If there are two or more binary counters, coverability is PSPACE-hard [5] via a reduction from reachability in bounded one-counter automata that is PSPACE-complete [18]. Therefore, coverability in binary encoded k -VASS is PSPACE-complete for every $k \geq 2$. See Figure 1 for the complexities of coverability in VASS with a fixed number of unary and binary encoded counters. This is all in striking contrast to the reachability problem in fixed dimension VASS, since reachability in 8-VASS is already known to be nonelementary [13].

There is a prominent line of work on 2-VASS with various encodings. The seminal paper in 1979 of Hopcroft and Pansiot [23] shows reachability in 2-VASS is decidable, proving that the reachability set is effectively semi-linear. Moreover, in the same paper the authors show, by an example, that the 3-VASS reachability set need not be semi-linear. Later, this was improved as it was shown that for

		Number of unary counters		
		0	1	≥ 2
Number of binary counters	0	NL-complete [3]	NL-complete [38]	NL-complete [35]
	1	in $\text{NC}^2 \subseteq \text{P}$ [2]	in NP [<i>this paper</i>]	Open
	≥ 2	PSPACE-complete [5]	PSPACE-complete	PSPACE-complete [35]

Fig. 1. The complexities of coverability in VASS with a fixed number of unary and binary encoded counters. All NL lower bounds arise from the zero counters case, here coverability is directed graph reachability and that is well known to be NL-complete [3]. In the case of one binary counter, regardless of the number of unary counters, we are aware only of this trivial NL lower bound. Furthermore, with one binary counter and at least two unary counters, we are not aware of a non-trivial upper bound (denoted “Open” in the table). When there are at least two binary counters and any number of unary counters, coverability is PSPACE-complete. The lower bound holds for 2-VASS with two binary counters [5] and the upper bound is given by Rackoff for any fixed dimension [35]. Recall that coverability in general VASS, where the number of counters is not fixed, is EXPSPACE-complete [35].

2-VASS the reachability relation is effectively semi-linear [28]. This proof shows that every 2-VASS can be characterised by a *flat model*, i.e. where the underlying finite automaton does not contain nested cycles. A more careful analysis of that paper, resulted in a PSPACE upper bound result for reachability in binary encoded 2-VASS [5]. Since coverability in binary encoded 2-VASS is PSPACE-hard [5], the authors were able to conclude that both coverability and reachability are PSPACE-complete. Just as coverability demonstrated the difference encoding makes to complexity, so does reachability; later it was proved that reachability in unary encoded 2-VASS is NL-complete [16].

Our Results and Techniques. We consider the coverability problem for 2-VASS with one unary counter. Here, updates of one counter are encoded in binary and the updates of the other are encoded in unary, see Figure 2 for an example. Notice that the unary counter need not be limited to polynomially bounded values. Otherwise, the value of the unary counter could be encoded into the states for an instance of coverability in binary encoded 1-VASS. Furthermore, we do not impose any restrictions on the initial and the target configurations, i.e. both coordinates of these vectors are encoded in binary. Our main result is that coverability in 2-VASS with one unary counter is in NP.

Coverability in binary encoded k -VASS is PSPACE-complete, for $k \geq 2$. The lower bound limits the practicality of applications. Therefore, it is sensible to consider restricted variations and quantify their complexity. We remark that coverability in fixed dimension VASS had widely-open complexity if there was exactly one binary counter and at least one unary counter. See Figure 1 for a summary of the known results.

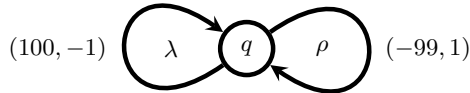


Fig. 2. Example 2-VASS with one unary counter V . Consider the instance of coverability consisting of V , the initial configuration $q(0, 1)$, and the target configuration $q(0, 10)$. Consider the path $\pi = \lambda\rho\lambda\rho\cdots\lambda\rho\rho\cdots\rho$ which induces a run in V from the initial configuration $q(0, 1)$. There are 990 repetitions of the pair of cycles $\lambda\rho$ to witness the configuration $q(990, 1)$. The cycles alternate so both counters remain non-negative throughout the run. This is followed by 10 iterations of the cycle ρ so the configuration $q(0, 11)$ is witnessed, achieving coverability of the target configuration $q(0, 10)$.

The natural starting point is the characterisation of runs via *linear path schemes* [4]. Intuitively, the authors prove that if coverability or reachability holds then there is a witnessing path of a specific shape. Namely, all paths can be characterised by a bounded language defined by a regular expression of the form $\tau_0\gamma_1^*\tau_1\cdots\tau_{k-1}\gamma_k^*\tau_k$. Here τ_0, \dots, τ_k are paths that connect disjoint cycles $\gamma_1, \dots, \gamma_k$. Since the language is bounded, checking if there is a path for a given expression essentially amounts to an instance of integer linear programming. In particular, the authors argue that both k and $|\tau_0| + |\gamma_1| + |\tau_1| + \dots + |\tau_{k-1}| + |\gamma_k| + |\tau_k|$ are pseudo-polynomially bounded [4]. However, a polynomial bound would immediately yield an NP upper bound as such a regular expression can be guessed. Given that coverability in 2-VASS with two binary counters is PSPACE-hard [5], we cannot simply directly apply the known results when dealing with 2-VASS with one binary and one unary counter. In Section 3, we provide a detailed discussion and a difficult yet motivating example in Figure 3.

To overcome this problem, we show that coverability can be witnessed by paths in *compressed linear form*. We relax the condition of the bounded language, by allowing to nest linear forms, provided that the exponents are fixed. Intuitively, an expression of the form $(\tau\gamma^*\tau')^*$ is still forbidden, but we allow for $(\tau\gamma^e\tau')^*$, where e is fixed but can be exponentially large (encoded using polynomially many bits). Such a form easily provides an NP upper bound.

We rely on two crucial observations to prove that we can focus on paths in compressed linear form. First, notice that the $*$ operation in a linear path scheme corresponds to iterating some cycle in the VASS. Since $\gamma_1, \dots, \gamma_k$ need to be short, one naturally focuses on short cycles. The issue is that there are exponentially many cycles of polynomial size. In Section 4 we prove that for coverability there are only polynomially many ‘optimal’ cycles. In Section 5 we deal with the problem when some cycle γ occurs many times in a linear path scheme witnessing coverability, resulting in a polynomial bound on k , the width of the linear path scheme. Then we prove that, either we can merge some γ_i and γ_j thus reducing the width, or that there is a cycle that has positive effect on one counter and non-negative effect on the other counter. Intuitively, in the latter

case, we can reduce the problem to coverability in 1-VASS by pumping such a cycle that forces one counter to take an arbitrarily large value. Moreover, such a cycle is witnessed by a linear path scheme. Since we need to pump this cycle, we require compressed linear forms to describe the repetitions of the cycle.

We highlight that both our crucial observations rely on that we work with coverability, not reachability. We further highlight that we address these crucial observations through our technical contributions that often depend on the fact there is one unary counter.

Further Related Work. Asymmetric treatment of the counters has been already considered for VASS. Recall that Minsky machines can be seen as VASS with the additional ability of zero-testing. For this model coverability is undecidable [33], even with two counters. This raised natural questions of what happens where only one of the counters is able to be reset or tested for zero. This, and more generally, reachability in VASS with hierarchical zero-tests are known to be decidable [36]. There is a further investigation into VASS with one zero-test [20]. Recently, work has appeared containing detailed analysis about 2-VASS where counters have different powers [19,29]. Finally, one of the most famous open problems in the community is whether reachability is decidable for 1-VASS with a pushdown stack. For these systems, coverability is known to be decidable [30]. The best known lower bound is that coverability, thus reachability also, is PSPACE-hard [15]. Our model, 2-VASS with one unary counter, can be seen as 1-VASS with a singleton alphabet pushdown stack.

The complexity of reachability in binary encoded 3-VASS remains an intriguing open problem. It is PSPACE-hard, like in dimension two, and the only known upper bound is primitive recursive, but not even elementary [27]. Recent works on reachability in fixed dimension VASS [11,9,13] provide new examples and a better understanding of the VASS model. Interestingly, many techniques applied to fixed dimension VASS are very closely related to recent progress on the nonelementary and Ackermann lower bounds for general VASS [10,12,26]. We finally and additionally motivate coverability in VASS with one binary counter and (at least) one unary counter as an avenue for finding new techniques to approach VASS problems with.

2 Preliminaries

Given an integer $z \in \mathbb{Z}$ we denote $\text{bitsize}(z) = \log_2(|z| + 1) + 1$. For a vector $\mathbf{v} := (v_1, v_2)$ we use $(\mathbf{v})_1 := v_1$ and $(\mathbf{v})_2 := v_2$ to be the projections to the first and second coordinates, respectively. We use $|\mathbf{v}|_{\max} := \max\{|v_1|, |v_2|\} + 1$ to denote the size of vector \mathbf{v} . We write $\mathbf{v} \leq \mathbf{w}$ if the inequalities hold on each coordinate. We write $\mathbf{v} < \mathbf{w}$ if at least one of the inequalities is strict.

A 2-VASS with one unary counter $V = (Q, T)$ consists of a finite set of control states Q and a set of transitions $T \subseteq Q \times \mathbb{Z} \times \{-1, 0, 1\} \times Q$. We shall refer to the first counter as the *binary counter* and the second counter as the *unary counter*. The size of V is $|V| = |Q| + \sum_{(p,b,u,q) \in T} \text{bitsize}(b)$. With

$|V|_{\max} := |Q| + |T| \cdot |T|_{\max}$ we denote the total ‘pseudo-polynomial size’ of the automaton, where $|T|_{\max}$ denotes the maximum absolute value that occurs in the transitions. Note that in a standard 2-VASS both counters are in binary, i.e. the domain of updates for the second counter is also \mathbb{Z} .

A *path* π in V is a, possibly empty, sequence of transitions $\pi = (t_i)_{i=1}^m$ such that $t_i = (q_{i-1}, b_i, u_i, q_i) \in T$. A path is *simple* if q_0, \dots, q_m are distinct. A path is a *cycle* if $q_0 = q_m$ and $m > 0$ (thus empty cycles are forbidden). We call it a q_0 -cycle to emphasise the first and last state of the cycle. A cycle is *simple* if q_1, \dots, q_m are distinct. A cycle is *short* if $m \leq |Q|$. The *length* of a path is the number of transitions in the path, denoted $\text{len}(\pi) = m$. We write $\pi[i..j]$ to denote the path that is the subsequence of transitions (t_i, \dots, t_j) in π .

A *configuration* $(p, \mathbf{u}) \in Q \times \mathbb{N}^2$, denoted $p(\mathbf{u})$, is a state paired with the current binary and unary counter values. A *run* is a sequence of configurations $(q_i(\mathbf{v}_i))_{i=0}^m$ such that $(q_{i-1}, (\mathbf{v}_i)_1 - (\mathbf{v}_{i-1})_1, (\mathbf{v}_i)_2 - (\mathbf{v}_{i-1})_2, q_i) \in T$. A run can equivalently be defined by the sequence of configurations induced by following a path π starting from an initial configuration $q_0(\mathbf{v}_0)$. We denote this run $q_0(\mathbf{v}_0) \xrightarrow{\pi} q_m(\mathbf{v}_m)$. We also write $q_0(\mathbf{v}_0) \xrightarrow{*} q_m(\mathbf{v}_m)$ to indicate the existence of a run between two configurations.

In this paper we study the *coverability* problem for VASS.

VASS COVERABILITY

INPUT: A VASS $V = (Q, T)$ and two configurations $p(\mathbf{u})$ and $q(\mathbf{v})$.

QUESTION: Does $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v}')$ hold, for some $\mathbf{v}' \geq \mathbf{v}$?

Do note that the initial configuration $p(\mathbf{u})$ and the target configuration $q(\mathbf{v})$ have both the binary and unary components encoded as binary integers. The *reachability problem* for VASS—which we will not study in this paper—requires $\mathbf{v}' = \mathbf{v}$.

Consider a path $\pi = (t_i)_{i=1}^m$, where $t_i = (q_{i-1}, b_i, u_i, q_i)$. The *effect* of π is the sum of the counter updates, i.e. the vector $\text{eff}(\pi) := \sum_{i=1}^m (b_i, u_i)$. We often focus on the two projections: the *binary effect* $\text{eff}_b(\pi) := \sum_{i=1}^m b_i$, and the *unary effect* $\text{eff}_u(\pi) := \sum_{i=1}^m u_i$.

We say that a cycle γ is *monotone* if $\text{eff}(\gamma) \geq \mathbf{0}$ or $\text{eff}(\gamma) \leq \mathbf{0}$. Otherwise, we say that γ is *non-monotone*. Note the two variants of a non-monotone cycle: a *positive-negative cycle* $\text{eff}_b(\gamma) > 0$ and $\text{eff}_u(\gamma) < 0$, and a *negative-positive cycle* $\text{eff}_b(\gamma) < 0$ and $\text{eff}_u(\gamma) > 0$.

Let γ be a cycle. Given $e \in \mathbb{N}$ we write γ^e for the path obtained by e repetitions of γ . We refer to e as the *exponent*. A linear path scheme is a regular expression of the form $\tau_0 \gamma_1^* \tau_1 \cdots \tau_{k-1} \gamma_k^* \tau_k$, where the paths $\tau_0, \tau_1, \dots, \tau_k$ connect disjoint cycles $\gamma_1, \dots, \gamma_k$. Note that a collection of cycles is disjoint if no two cycles have a common state. Given $\ell = (\tau_0, \gamma_1, \tau_1, \dots, \tau_{k-1}, \gamma_k, \tau_k)$, we say the a path π is in linear form ℓ if $\pi = \pi_\ell = \tau_0 \gamma_1^{e_1} \tau_1 \cdots \tau_{k-1} \gamma_k^{e_k} \tau_k$ for some exponents e_1, \dots, e_k . Note that in this definition every path has a linear form, e.g. $\tau_0 = \pi$ is valid. To leverage the definition, we will ask whether paths are in a linear form of certain size. The size of a linear form ℓ is $\sum_{i=0}^k \text{len}(\tau_i) + \sum_{i=1}^k \text{len}(\gamma_i)$. The size of π_ℓ is $\sum_{i=0}^k \text{len}(\tau_i) + \sum_{i=1}^k \text{len}(\gamma_i) + \sum_{i=1}^k \text{bitsize}(e_i)$, i.e. includes the exponents. We refer to k as the *width* of the linear form.

3 Coverability in 2-VASS with One Unary Counter

In this section we briefly discuss why the state-of-the-art techniques are not enough to prove that coverability in 2-VASS with one unary counter is in NP. Blondin et al. [4] show that for a given 2-VASS V there exists a set of linear path schemes S such that if $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$ in V , then there exists a path π in a linear path scheme $\rho \in S$ such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$. For every linear path scheme $\rho \in S$ the width of ρ , and therefore the width of every path, is bounded above by $\text{poly}(|Q|, |T|_{\max})$ [4, Theorem 3.1]. Such a path π is not necessarily a polynomial size witness, as the width depends on $|T|_{\max}$ polynomially. We provide an example of a 2-VASS with one unary counter where the width of every linear form ℓ for a path is exponential in the input size. This demonstrates that the combinatorial structure of linear path schemes is not self-sufficient to show that there always exists a polynomial size witness of coverability.

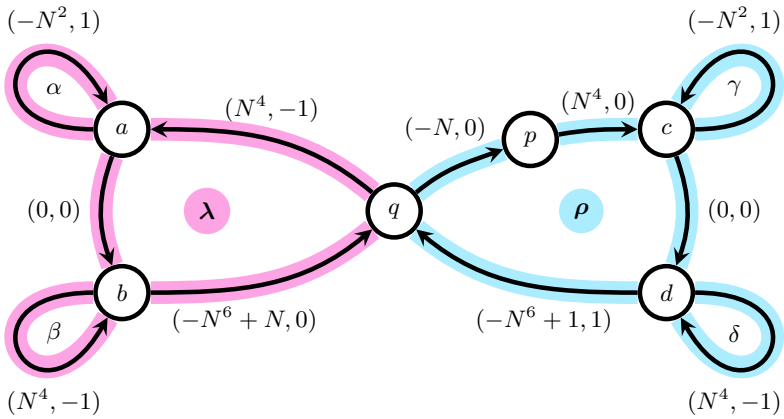


Fig. 3. Example 2-VASS with one unary counter V , where $N = 2^n$, where n is an input parameter (thus making N exponentially large). Consider the coverability instance with the initial configuration $q(0, 1)$, and the target configuration $q(N, 1)$. Let $\lambda = t_{qa}\alpha^{N^2}t_{ab}\beta^{N^2}t_{bq}$ and $\rho = t_{qp}t_{pc}\gamma^{N^2}t_{cd}\delta^{N^2}t_{dq}$, where t_{xy} is the transition from state x to state y . Observe that $\text{eff}(\lambda) = (N, -1)$ and $\text{eff}(\rho) = (-N+1, 1)$, thus $\text{eff}(\lambda\rho) = (1, 0)$. It is easy to then see that $q(0, 1) \xrightarrow{(\lambda\rho)^N} q(N, 1)$. Intuitively the cycles λ and ρ alternate so both counters remain non-negative throughout the run. In the appendix, we prove that there does not exist a linear form of polynomial size for a path that induces a coverability run.

Paths in Compressed Linear Form. Nevertheless, there is a natural way to succinctly describe the path presented in Figure 3. Let $\sigma = \lambda\rho$, and note that

$$\sigma^N = \left(t_{qa} \alpha^{N^2} t_{ab} \beta^{N^2} t_{bq} t_{qp} t_{pc} \gamma^{N^2} t_{cd} \delta^{N^2} t_{dq} \right)^N.$$

All paths and cycles are ‘small’, and the bitsize of N and N^2 are polynomial in n , so σ itself is a path in linear form. We introduce the following generalisation of linear form paths that encapsulates the idea behind paths of this kind of arrangement.

Definition 1 (Compressed linear form path). *A path π is in compressed linear form if $\pi = \rho_0 \sigma_1^{f_1} \rho_1 \cdots \rho_{k-1} \sigma_k^{f_k} \rho_k$ for some connected paths in linear form $\rho_0, \rho_1, \dots, \rho_k$; cycles in linear form $\sigma_1, \dots, \sigma_k$; and exponents f_1, \dots, f_k . The size of a compressed linear form path is the sum of the sizes of all ρ_i and σ_i (including the bitsize of their exponents) plus the bitsize of the exponents f_i .*

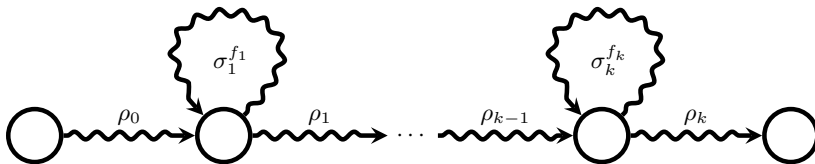


Fig. 4. A compressed linear form path.

The following theorem is our main contribution.

Theorem 1. *Let V be a 2-VASS with one unary counter and fix two configurations $p(\mathbf{u})$ and $q(\mathbf{v})$. If $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$, then there exists a path in compressed linear form π such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v}')$ and $\mathbf{v}' \geq \mathbf{v}$. The size of the compressed linear form path is polynomial in $|V| + \text{bitsize}(\mathbf{u}) + \text{bitsize}(\mathbf{v})$.*

Corollary 1. *Coverability in 2-VASS with one unary counter is in NP.*

Proof. By Theorem 1 it suffices to consider paths in compressed linear form of polynomial size, that can be guessed in NP. It suffices to observe that a coverability instance on a given compressed linear form amounts to an instance of integer linear programming. Intuitively, this is because the nested cycles are fixed. Thus to check whether a run drops below zero it suffices to check before applying a cycle and after applying it for the last time (see e.g. [5, Section V, Lemma 14]). □

We highlight that it is rather unexpected that only one extra ‘level’ of linear form paths is enough to obtain polynomial size witnesses of coverability in a 2-VASS with one unary counter, since the problem is PSPACE-complete for general

2-VASS. Roughly speaking, the example given in Figure 3 observes the most complex behaviour possible and this instance of coverability is witnessed by a compressed linear form path. More specifically, compressed linear form paths containing only one linear form cycle suffice as witnesses for coverability in 2-VASS with one unary counter. Therefore, all witnesses can be represented by a compressed linear form path $\rho\sigma^N\tau$ where ρ and τ are linear form paths to and from the single linear form cycle σ which is iterated N times.

The rest of the paper is dedicated to proving Theorem 1. We heavily exploit both distinguishing features of the problem: the fact that one counter receives unary encoded updates (as opposed to both counters in binary) and the fact that we aim to assert coverability (as opposed to reachability). Our approach is as presented in the introduction. In 4 we observe that we can polynomially bound the total number of distinct short cycles. We formalise this and show that there are only polynomially many ‘irreplaceable’ short cycles. In 5 we provide a ‘reshuffling procedure’. If some short cycle γ repeats exponentially many times we aim to modify the path π by moving the cycles γ close to each other. Then either every short cycle γ will appear only in polynomially many ‘bundles’ γ^e , or we find a cycle σ such that $\text{eff}(\sigma) > \mathbf{0}$. In the latter case, by pumping σ we are essentially left with one counter. Finally, in Section 6 we conclude the proof of Theorem 1.

4 Replacing Short Cycles

In this section, we show that there are only polynomially many short cycles that need occur in a run witnessing coverability. Fix a path $\pi = (q_{i-1}, b_i, u_i, q_i)_{i=1}^k$. Let $0 \leq i_b, i_u \leq k$ be the first indices such that $g_b = \sum_{i=1}^{i_b} b_i$ and $g_u = \sum_{i=1}^{i_u} u_i$ are at their lowest, respectively. Note that $g_b, g_u \leq 0$ since by convention if we consider $i_b, i_u = 0$ then the sum evaluates to 0. We call and denote these two numbers the *binary guard* $\text{grd}_b(\pi) = g_b$ and the *unary guard* $\text{grd}_u(\pi) = g_u$. The following claim immediately follows from these definitions.

Claim 1. Both $\text{grd}_b(\pi[i_b + 1..k]) = 0$ and $\text{grd}_u(\pi[i_u + 1..k]) = 0$.

Much like the *nadir* of a cycle in a one-counter net, defined in [1], we define the *binary-nadir state* as q_{i_b} , i.e. the first state in which the binary counter first attains the lowest value when executing π . We call the *binary-nadir decomposition* $\pi = \pi_1^b \pi_2^b$, for $\pi_1^b = \pi[1..i_b]$ and $\pi_2^b = \pi[i_b + 1..k]$, as intimated in Claim 1. Notice that this decomposition necessitates the binary guard of the path π is equal to the binary effect of the prefix π_1^b , $\text{grd}_b(\pi) = \text{eff}_b(\pi_1^b) = \text{grd}_b(\pi_1^b)$. Furthermore, the suffix of the binary-nadir decomposition has zero binary guard $\text{grd}_b(\pi_2^b) = 0$. We primarily utilise binary-nadir states and binary-nadir decompositions, hence the omission of matching unary-nadir states and unary-nadir-decompositions.

Definition 2 (Replaceable cycles). *Let γ be a q -cycle and let p be the binary-nadir state of γ . We say that γ is replaceable if there exists a q -cycle γ' with the same binary-nadir state p , such that*

- (a) $\text{eff}_b(\gamma') \geq \text{eff}_b(\gamma)$ and $\text{eff}_u(\gamma') \geq \text{eff}_u(\gamma)$,
 (b) $\text{grd}_b(\gamma') \geq \text{grd}_b(\gamma)$ and $\text{grd}_u(\gamma') \geq \text{grd}_u(\gamma)$, and
 (c) $\text{len}(\gamma') \leq \text{len}(\gamma)$.

Additionally, at least one inequality is strict and we write $\gamma \prec \gamma'$.

We say a cycle is *irreplaceable* if it is not replaceable. We also say that an irreplaceable q -cycle γ with the binary-nadir state p is *characterised* by the five values: $\text{eff}_b(\gamma)$, $\text{eff}_u(\gamma)$, $\text{grd}_b(\gamma)$, $\text{grd}_u(\gamma)$, and $\text{len}(\gamma)$.

Lemma 1 (Replacing cycles). *Let $\pi = \pi_1\gamma\pi_2$, where γ is a q -cycle. Suppose $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$ then the following hold.*

- If γ is replaceable, then there exists an irreplaceable q -cycle $\gamma \prec \gamma'$ such that $p(\mathbf{u}) \xrightarrow{\pi_1\gamma'\pi_2} q(\mathbf{v}')$.
- If γ is irreplaceable, then for every irreplaceable q -cycle γ' that has the same characterisation as γ , $p(\mathbf{u}) \xrightarrow{\pi_1\gamma'\pi_2} q(\mathbf{v}')$.

In both cases $\mathbf{v}' \geq \mathbf{v}$ and $\text{len}(\pi) \geq \text{len}(\pi_1\gamma'\pi_2)$.

For convenience, we define the polynomial $R(|Q|) := |Q|^4(|Q|+1)(2|Q|+1)^2$.

Lemma 2. *There exists at most $R(|Q|)$ many irreplaceable short cycles with different characterisations.*

Proof. We fix two states q and p and consider only q -cycles γ with the binary-nadir state p . Thus in the final argument one must multiply everything by $|Q|^2$. Since we consider short cycles, the unary effect and the unary guard are small, i.e. $-|Q| \leq \text{eff}_u(\gamma) \leq |Q|$ and $-|Q| \leq \text{grd}_u(\gamma) \leq 0$.

Towards a contradiction, suppose there exists more than $|Q|^2(|Q|+1)(2|Q|+1)^2$ many such irreplaceable q -cycles with different characterisations. By the pigeonhole principle there must exist two cycles, denoted in binary-nadir decomposition $\gamma = \gamma_1\gamma_2$ and $\gamma' = \gamma'_1\gamma'_2$, that have the same values $\text{eff}_u(\gamma_1) = \text{eff}_u(\gamma'_1)$, $\text{eff}_u(\gamma_2) = \text{eff}_u(\gamma'_2)$, $\text{grd}_u(\gamma) = \text{grd}_u(\gamma')$, $\text{len}(\gamma_1) = \text{len}(\gamma'_1)$, and $\text{len}(\gamma_2) = \text{len}(\gamma'_2)$.

We know that the irreplaceable q -cycles γ and γ' have different characterisations, so it must be the case that their binary effects differ $\text{eff}_b(\gamma) \neq \text{eff}_b(\gamma')$. Otherwise, the cycle with the lesser binary guard is replaceable, because the unary effect, unary guard, and length do not differ. Without loss of generality, suppose $\text{eff}_b(\gamma) > \text{eff}_b(\gamma')$, then $\text{grd}_b(\gamma) < \text{grd}_b(\gamma')$. Otherwise, γ' would be replaceable as $\gamma \prec \gamma'$.

Now consider the q -cycle $\sigma = \gamma'_1\gamma_2$, also with the binary-nadir state p . We will show that $\gamma \prec \sigma$ contradicting the fact that γ is an irreplaceable q -cycle. First, observe that σ has greater binary effect than γ as

$$\text{eff}_b(\sigma) = \text{eff}_b(\gamma'_1) + \text{eff}_b(\gamma_2) > \text{eff}_b(\gamma_1) + \text{eff}_b(\gamma_2) = \text{eff}_b(\gamma),$$

where the inequality holds because $\text{grd}_b(\gamma) < \text{grd}_b(\gamma')$. Second, σ and γ have equal unary effect because $\text{eff}_u(\gamma'_1) = \text{eff}_u(\gamma_1)$. Third, we show that σ has a

greater binary guard than γ . Since γ_2 is the suffix of the binary-nadir decomposition of γ , it must be true that $\text{grd}_b(\gamma_2) = 0$. By Claim 1 $\text{grd}_b(\sigma) = \text{grd}_b(\gamma'_1)$. Combining these facts, $\text{grd}_b(\sigma) = \text{grd}_b(\gamma') > \text{grd}_b(\gamma)$. Fourth, σ has at least the unary guard of γ because, in particular, the unary guard of the prefix of a path is at most the unary guard of the entire path.

$$\begin{aligned} \text{grd}_u(\sigma) &= \min\{\text{grd}_u(\gamma'_1), \text{eff}_u(\gamma'_1) + \text{grd}_u(\gamma_2)\} \\ &\geq \min\{\text{grd}_u(\gamma'), \text{eff}_u(\gamma'_1) + \text{grd}_u(\gamma_2)\} \\ &= \min\{\text{grd}_u(\gamma), \text{eff}_u(\gamma_1) + \text{grd}_u(\gamma_2)\} = \text{grd}_u(\gamma). \end{aligned}$$

Fifth and finally, σ and γ have equal length because $\text{len}(\gamma'_1) = \text{len}(\gamma_1)$. We have at least one strict inequality. Thus, we have reached the desired contradiction. \square

5 Reshuffling Linear Form Paths

5.1 Reshuffling Procedure

There can be many linear forms for a path π . We will try to find an ‘optimal’ one, so we introduce a cost function to quantify linear forms. Recall that a linear form ℓ is a sequence of paths $\tau_0, \tau_1, \dots, \tau_k$ and a sequence of cycles $\gamma_1, \dots, \gamma_k$. If π is in the linear form $\ell = (\tau_0, \gamma_1, \tau_1, \dots, \tau_{k-1}, \gamma_k, \tau_k)$ then we write $\pi_\ell = \tau_0 \gamma_1^{e_1} \tau_1 \dots \tau_{k-1} \gamma_k^{e_k} \tau_k$, where $\pi = \pi_\ell$ (the index is here to stress the exact linear form). For this section, we will consider linear forms only containing short cycles γ , they will play a key role in the following arguments.

We define a cost function that assigns, to a linear form ℓ , the following pair of naturals $C(\ell) := \left(\sum_{i=0}^k \text{len}(\tau_i), k\right)$. For convenience, we define the polynomial $P(|Q|) := 2(|Q|^2 + 1)(|Q|^2 + 2) \cdot R(|Q|)$, where R is the polynomial defined for Lemma 2. We say that a linear form ℓ is *narrow* if $C(\ell) \leq (|Q|(P(|Q|) + 1), P(|Q|))$, otherwise we say that ℓ is *wide*. We say that the triple (π', σ, π'') is a monotone cycle decomposition of a path π if σ is a monotone cycle, $\pi = \pi' \sigma \pi''$, and $\text{len}(\sigma) < \text{len}(\pi)$.

Lemma 3 (Reshuffling). *Let π be a path such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$. Then there exists a path ρ such that $p(\mathbf{u}) \xrightarrow{\rho} q(\mathbf{w})$ where $\mathbf{w} \geq \mathbf{v}$, $\text{len}(\rho) \leq \text{len}(\pi)$, and either*

- (i) *there exists a narrow linear form for ρ , or*
- (ii) *there exists a monotone cycle decomposition of ρ .*

Proof. We start with a series of preparations. In the early part of this proof, we provide simple observations to ascertain some auspicious properties of our path. In the later part of this proof, we present the ‘reshuffling procedure’ and conclude with one of the cases in the statement of this lemma. In this proof we will compare linear forms using the lexicographic order \prec_{lex} , that is known to be a linear-order and a well-order. Formally,

$$\begin{aligned} C(\ell') \prec_{lex} C(\ell) &\iff (C(\ell'))_1 < (C(\ell))_1 \text{ or,} \\ &\quad (C(\ell'))_1 = (C(\ell))_1 \text{ and } (C(\ell'))_2 < (C(\ell))_2. \end{aligned}$$

We start with a path π' such that $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{v}')$ where $\mathbf{v}' \geq \mathbf{v}$, $\text{len}(\pi') \leq \text{len}(\pi)$, and π' has a linear form ℓ' that has the least cost among all linear forms for all like-paths. That means there does not exist another path π'' such that $p(\mathbf{u}) \xrightarrow{\pi''} q(\mathbf{v}'')$ where $\mathbf{v}'' \geq \mathbf{v}$, $\text{len}(\pi'') \leq \text{len}(\pi)$, and π'' has a linear form ℓ'' such that $C(\ell'') \prec_{lex} C(\ell')$.

For the first observation, suppose there exists $0 \leq i \leq k$ such that $\text{len}(\tau_i) > |Q|$. Then the path τ_i can be written as $\tau_i = \tau' \gamma \tau''$, where γ is a short cycle. We can define the linear form ℓ'' by modifying ℓ' where τ_i is swapped for $\tau' \gamma \tau''$. Although this increments the number of cycles k , we decrease the total length of the paths as $\text{len}(\tau') + \text{len}(\tau'') < \text{len}(\tau_i)$ (recall that empty cycles are forbidden). Thus $C(\ell'') \prec_{lex} C(\ell')$ contradicting the assumption that ℓ has minimum cost. Therefore, we assume that $\text{len}(\tau_i) \leq |Q|$ for all $0 \leq i \leq k$.

For the second observation, we define $U := \{0 \leq i \leq m : (\mathbf{v}_i)_2 < |Q|\}$ to be the set of indices of configurations in the run that have unary counter value less than $|Q|$. Observe that if $|U| > |Q|^2 + 1$ then there are two indices $0 < i < j \leq m$ such that the two corresponding configurations in the run have matching states $q_i = q_j$ and equal unary counter values $(\mathbf{v}_i)_2 = (\mathbf{v}_j)_2$. Then, regardless of sign of its binary effect, $\pi'[i..j]$ is a monotone cycle. Here, case (ii) immediately holds by decomposing π' itself using the monotone cycle $\pi'[i..j]$, given that $i > 0$ and $j \leq m$ implies $\text{len}(\pi'[i..j]) = j - i < m = \text{len}(\pi')$. Therefore, we assume $|U| \leq |Q|^2 + 1$. We continue with the aim of satisfying the conditions of case (ii) by finding a monotone cycle decomposition.

Let $d = |\{\gamma_1, \dots, \gamma_k\}|$ be the number of distinct cycles in the linear form ℓ' . By Lemma 1 and Lemma 2, we can assume that $d \leq R(|Q|)$. Otherwise, we can exchange replaceable q -cycles for irreplaceable q -cycles using the first point in Lemma 1. It is possible that for a particular characterisation, we can observe more than one irreplaceable q -cycle. Then using the second point in Lemma 1, we can arbitrarily select one of these irreplaceable q -cycles with equal characterisations to exchange all others with. By applying these cycle replacements to π' , we obtain a different path ρ . Definition 2 ensures that we do so without decreasing the effect (a), without allowing the counters to take a negative value (b), and without increasing the length of the path (c). Therefore $p(\mathbf{u}) \xrightarrow{\rho} q(\mathbf{w})$ and $\mathbf{w} \geq \mathbf{v}' \geq \mathbf{v}$, and $\text{len}(\rho) \leq \text{len}(\pi') \leq \text{len}(\pi)$. We remark since cycles have been exchanged one-for-one, then ρ takes a linear form ℓ with the same path segments as ℓ' . Therefore, it is clear that neither the number of cycles k , nor the sum of the lengths of the paths between cycles, have changed. We also know that ℓ is a linear form for ρ with minimum cost $C(\ell) = C(\ell')$, as per the initialisation in this proof.

Suppose $\rho = \rho_\ell = \tau_0 \gamma_1^{e_1} \tau_1 \cdots \tau_{k-1} \gamma_k^{e_k} \tau_k$. Let $(q_j(\mathbf{v}_j))_{j=0}^m$ be the run obtained by following the path ρ_ℓ from the initial configuration $q_0(\mathbf{v}_0) = p(\mathbf{u})$ to the final configuration $q_m(\mathbf{v}_m) = q(\mathbf{w})$. We may assume that ℓ is wide. Otherwise, case (i) is immediately satisfied. We also know that $\text{len}(\rho_\ell) \geq \max\{(C(\ell))_1, (C(\ell))_2\} > P(|Q|)$. We may also assume that each cycle $\gamma_1, \dots, \gamma_k$ is non-monotone, i.e. it is positive-negative or negative-positive. Otherwise, case (ii) immediately holds by decomposing ρ itself using some monotone cycle γ_i , given that $\text{len}(\gamma_i) \leq |Q| <$

$P(|Q|) < \text{len}(\rho_\ell)$. Notice this is valid since each $e_i > 0$ by the minimality of $C(\ell)$, otherwise you can write $\dots \tau_{i-1} \gamma_i^0 \tau_i \dots$ with one less cycle, decreasing $(C(\ell))_2$.

From the first observation, we get $\sum_{i=0}^k \text{len}(\tau_i) \leq (k + 1)|Q|$. Given that ℓ is wide, either $|Q|(P(|Q|) + 1) < (C(\ell'))_1 = \sum_{i=0}^k \text{len}(\tau_i) \leq (k + 1)|Q|$ that implies $P(|Q|) < k$, or $P(|Q|) < (C(\ell'))_2 = k$. Regardless, $P(|Q|) < k$ holds. Recall that $|U| \leq |Q|^2 + 1$ from the second observation. Since there are relatively ‘few’ configurations indexed by U , there must exist a relatively ‘distant’ pair of consecutive configurations indexed by U . More formally, there are i and j such that $0 \leq i < j \leq k$ and $j - i \geq 2(|Q|^2 + 2)R(|Q|)$ and all configurations that occur in the run over the path segment $\tau_i \gamma_{i+1}^{e_{i+1}} \dots \gamma_j^{e_j} \tau_j$ have unary counter value at least $|Q|$. Notice that $j - i$ is the number of cycles in this path segment. Since $j - i \geq 2(|Q|^2 + 2)R(|Q|)$ and by pigeonhole principle on the number of irreplaceable cycles, there is a common irreplaceable cycle γ repeated at least $x = 2(|Q|^2 + 2)$ many times. We will focus on the first x such occurrences of this cycle. Let s_1, \dots, s_x be the indices of this cycle γ , i.e. $\gamma = \gamma_{s_1} = \dots = \gamma_{s_x}$. To highlight these cycles, we decompose this path segment into

$$\tau_i \gamma_{i+1}^{e_{i+1}} \dots \gamma_j^{e_j} \tau_j = \Lambda_0 \gamma^{f_1} \Lambda_1 \dots \Lambda_{x-1} \gamma^{f_x} \Lambda_x,$$

where $f_j := e_{s_j}$ and Λ_j are the concatenated paths (and cycles) in between iterations of γ , see Figure 5. To reiterate, we know that all configurations that occur in the run over this path segment have at least $|Q|$ unary counter value and γ is a short cycle.

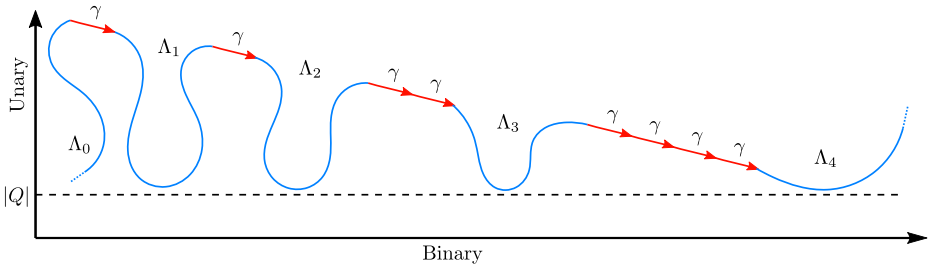


Fig. 5. The decomposition of the path segment into $\Lambda_0 \gamma^{f_1} \Lambda_1 \dots \Lambda_{x-1} \gamma^{f_x} \Lambda_x$, as above. Notice that the unary counter is always at least $|Q|$ as no configurations indexed by U are present.

Reshuffling Procedure. In the rest of the proof we will modify the path segment (above) of the path ρ_ℓ with a procedure that we call *reshuffling*. At the end of this procedure we will find a monotone cycle and satisfy case (ii) of this lemma. We either find this cycle directly, or we obtain a linear form ℓ'' such that $C(\ell'') \prec_{lex} C(\ell)$ contradicting the assumption that ℓ has minimal cost.

Note that $x = 2(|Q|^2 + 2)$ is even, and for every pair of consecutive cycles γ_{2j-1} and γ_{2j} (for $1 < 2j \leq x$), consider the subsegment $\gamma^{f_{2j-1}} \Lambda_{2j-1} \gamma^{f_{2j}}$. There

are two scenarios depending on the variant of the non-monotone cycle γ . In the scenario where γ is positive-negative, we move an iteration of γ from right to left obtaining $\gamma^{f_{2j-1}+1}\Lambda_{2j-1}\gamma^{f_{2j}-1}$. In the scenario where γ is negative-positive, we move an iteration of γ in the opposite direction obtaining $\gamma^{f_{2j-1}-1}\Lambda_{2j-1}\gamma^{f_{2j}+1}$.

We repeat this procedure until one of two conditions are met. The first is when there are no iterations of γ on one side, so either f_{2j-1} or f_{2j} becomes 0. The second is when there appears a configuration, in the run over the path subsegment after reshuffling, with unary counter value less than $|Q|$. See Figure 6 for a pictorial presentation of reshuffling in the scenario where γ is positive-negative.

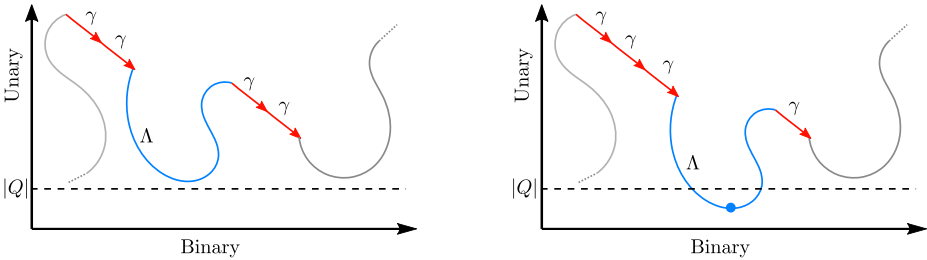


Fig. 6. Reshuffling around a path Λ (blue) where γ (red) is positive-negative. Before reshuffling, the path subsegment $\cdots\gamma\Lambda\gamma\cdots$ all configurations have unary counter value at least $|Q|$ in the run (left). After reshuffling, the path subsegment $\cdots\gamma\gamma\Lambda\cdots$, there is a configuration with unary counter value less than $|Q|$ in the run (right).

We claim that after each reshuffling step, the corresponding run remains executable, so we must check that both counters remain non-negative. Notice that by only moving a cycle, the total effect of the path subsegment remains the same. Therefore, if the run was executable before reshuffling, we can safely assume that the prefix before the path subsegment and the suffix after the path subsegment are still executable. For that reason, consider the counter values of configurations occurring in the run over the reshuffled path subsegment. We focus on a single step of the reshuffling procedure that concerns the subsegment $\gamma^{f_{2j-1}}\Lambda_{2j-1}\gamma^{f_{2j}}$.

Suppose γ is a positive-negative cycle. Then the reshuffling procedure moves γ from right to left. We claim that since $f_{2j-1} > 0$ and $\Lambda_0\gamma^{f_1}\Lambda_1\cdots\Lambda_{2j-1}\gamma^{f_{2j-1}}$ is executable, the subsegment $\Lambda_0\gamma^{f_1}\Lambda_1\cdots\Lambda_{2j-1}\gamma^{f_{2j-1}+1}$ is executable from the initial configuration. This is because one prerequisite of the reshuffling procedure is that all configurations occurring in the run over the path subsegment have at least $|Q|$ unary counter value. Moreover, the cycle γ has length at most $|Q|$ so $\text{grd}_u(\gamma) \geq -|Q|$ means the unary counter value remains non-negative. As for the binary counter value, since a single execution of γ increases the binary counter and an iteration of γ was already executed before reshuffling, $\Lambda_0\gamma^{f_1}\Lambda_1\cdots\Lambda_{2j-1}\gamma^{f_{2j-1}+1}$ is executable. In the same way, from the initial con-

figuration, $\Lambda_0 \gamma^{f_1} \Lambda_1 \dots \Lambda_{2j-1} \gamma^{f_{2j-1}+1} \Lambda_{2j} \gamma^{f_{2j}-1}$ is executable. This is because $\text{eff}_u(\gamma) \geq -|Q|$, and again, all configurations occurring in the run over the path subsegment have at least $|Q|$ unary counter value, and also because of the monotonicity on the binary counter.

The argument when γ is a negative-positive cycle is analogous. This concludes the correctness analysis of the reshuffling procedure.

Finishing Reshuffling. We analyse what happens when reshuffling is finished. Suppose that there exists a pair $2j - 1$ and $2j$ such that the reshuffling finishes under the first condition where all iterations of γ have been moved to one side of Λ_{2j-1} . In this case we obtain a new linear form ℓ'' for ρ , where one collection of the cycle γ has been removed (decrementing k). So $(C(\ell''))_2 = k - 1 < (C(\ell))_2$ and the two adjacent path segments can be combined without changing the summed length of paths so $(C(\ell''))_1 = (C(\ell))_1$. Therefore, $C(\ell'') \prec_{lex} C(\ell)$ contradicting the assumption ℓ has the minimal cost.

Otherwise, for every $1 \leq j \leq x/2$ the reshuffling of pair $2j - 1$ and $2j$ finishes under condition the second condition. So there is a configuration with unary counter value less than $|Q|$ in the run induced from the path ρ for each pair $2j - 1$ and $2j$ (see Figure 7). Recall that $\frac{x}{2} = |Q|^2 + 2$, that is the number of pairs. Akin to the first observation (in the beginning of this proof), we use the pigeonhole principle on the number of such configurations to obtain two configurations with matching states and equal unary counter values. The path segment inducing the part of the run between these two configurations is a monotone cycle, regardless of the binary effect. Again, it must be true that the length of this cycle is less than the length of the whole path, so we obtain a monotone cycle decomposition of ρ . Thus case (ii) of the lemma holds. \square

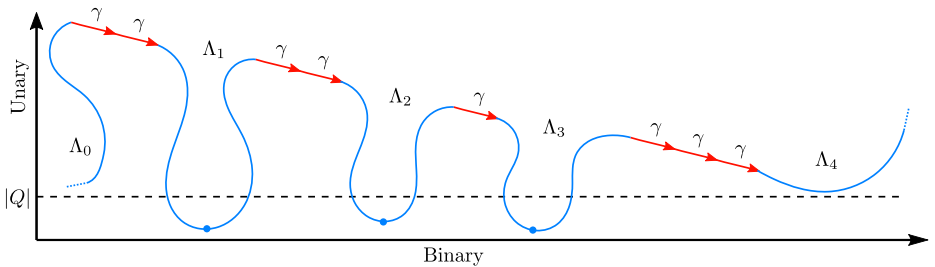


Fig. 7. After reshuffling is finished under condition the second condition, we can find a zero unary effect cycle using the (sufficiently many) configurations with unary counter less $|Q|$.

5.2 Applying Reshuffling

Lemma 3 does not necessarily return a narrow linear form for a path π witnessing coverability. Instead it may return a monotone cycle decomposition (ρ, σ, τ) of π .

Our next goal is to show that there exists polynomial size certificates for ρ and σ (Lemma 4), and then to show that there exists a polynomial size certificate for τ (Lemma 5). Like linear forms, there can be many monotone cycle decompositions for a path. Following, we will use the cost function assigning monotone cycle decompositions to pairs of natural numbers $D((\rho, \sigma, \tau)) := (\text{len}(\rho\sigma), \text{len}(\sigma))$. Note that we can compare two decompositions using their cost, even if they are for two different paths.

Lemma 4. *Suppose $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$ yet there is no narrow linear form ℓ for any path π such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{w})$ and $\mathbf{w} \geq \mathbf{v}$, then there exists a path π' such that*

- (a) $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{w}')$ where $\mathbf{w}' \geq \mathbf{v}$,
- (b) there is a monotone cycle decomposition (ρ, σ, τ) of π' where $\text{eff}(\sigma) > \mathbf{0}$, and
- (c) there are narrow linear forms for both ρ and σ .

Proof. We will again use the lexicographical order \prec_{lex} to compare the cost of monotone cycle decompositions. Let π be a path of minimum length such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{w})$ where $\mathbf{w} \geq \mathbf{v}$. Let $c = (\rho, \sigma, \tau)$ be the monotone cycle decomposition of π that minimizes the cost $D(c)$ under the \prec_{lex} order. Such a decomposition must exist, otherwise applying Lemma 3 would return a narrow linear form ℓ' for ρ such that $p(\mathbf{u}) \xrightarrow{\ell'} q(\mathbf{w}')$ and $\mathbf{w}' \geq \mathbf{w} \geq \mathbf{v}$, contradicting an assumption of this lemma. Observe that $\text{eff}(\sigma) > \mathbf{0}$, otherwise one can remove σ and consider the shorter path $\rho\tau$, contradicting the minimal length of π . Next, we argue that ρ and σ do not have monotone cycle decompositions, we then leverage Lemma 3 to obtain the narrow linear forms required.

Path ρ cannot be decomposed further. Towards a contradiction, assume that there is a monotone cycle decomposition $c' = (\rho', \sigma', \tau')$ of ρ . Observe that the following monotone cycle decomposition $c' = (\rho', \sigma', \tau'\sigma\tau)$ of π has lower cost $D(c') \prec_{lex} D(c)$ as $(D(c'))_1 = \text{len}(\rho') + \text{len}(\sigma') < \text{len}(\rho) + \text{len}(\sigma) = (D(c))_1$. This contradicts the assumption that (ρ, σ, τ) has minimum cost.

Suppose $p(\mathbf{u}) \xrightarrow{\rho} p'(\mathbf{x})$. Since there is no monotone cycle decomposition, applying Lemma 3 to ρ returns a path ρ' with a narrow linear form such that $p(\mathbf{u}) \xrightarrow{\rho'} p'(\mathbf{x}')$ where $\mathbf{x}' \geq \mathbf{x}$ and $\text{len}(\rho') \leq \text{len}(\rho)$.

Cycle σ cannot be decomposed further. Towards a contradiction, assume that there is a monotone cycle decomposition (ρ', σ', τ') of σ . Observe that the following monotone cycle decomposition $c' = (\rho\rho', \sigma', \tau'\tau)$ of π has lower cost $D(c') \prec_{lex} D(c)$ as $(D(c'))_1 = \text{len}(\rho) + \text{len}(\rho') + \text{len}(\sigma') \leq \text{len}(\rho) + \text{len}(\sigma) = (D(c))_1$ and $(D(c'))_2 = \text{len}(\sigma') < \text{len}(\sigma) = (D(c))_2$. This contradicts the assumption that (ρ, σ, τ) has minimum cost.

Suppose $p'(\mathbf{x}) \xrightarrow{\sigma} p'(\mathbf{y})$. Since there is no monotone cycle decomposition, applying Lemma 3 to σ returns a path σ' with a narrow linear form such that $p'(\mathbf{x}) \xrightarrow{\sigma'} p'(\mathbf{y}')$ where $\mathbf{y}' \geq \mathbf{y}$ and $\text{len}(\sigma') \leq \text{len}(\sigma)$. In particular, it is also true that $\text{eff}(\sigma') \geq \text{eff}(\sigma) > \mathbf{0}$.

Replacing ρ' for ρ and σ' for σ in π yields a path π' . Clearly if $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{w})$ where $\mathbf{w} \geq \mathbf{v}$, then $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{w}')$ where $\mathbf{w}' \geq \mathbf{w} \geq \mathbf{v}$. Finally, (ρ', σ', τ) is monotone cycle decomposition of π' such that $\text{eff}(\sigma') > \mathbf{0}$ and ρ' and σ' have narrow linear forms, as required. \square

We now aim to obtain a narrow linear form for τ . Note that Lemma 4 gives us a monotone cycle σ with positive effect on at least one counter, i.e. $\text{eff}(\sigma) > \mathbf{0}$. By pumping σ we can force one of the counters to take an arbitrarily large value (following, the vector \mathbf{x} reflects this large value for Lemma 5). Then, loosely speaking, the problem reduces to coverability in 1-VASS. However, proving the existence of a polynomial size compressed linear form path in Theorem 1 requires more care. Note that Lemma 5 is stated for 2-VASS (not necessarily with one unary counter). First we need to recall the following bound on counter values observed throughout runs. Recall that $|V|_{\max} := |Q| + |T| \cdot |T|_{\max}$ is the pseudo-polynomial size of the input.

Theorem 2 (Corollary from Theorem 3.2 in [4]). *Consider a 2-VASS (with both counters in binary) $V = (Q, T)$ and let $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$, then there exists a run $p(\mathbf{u}) = q_0(\mathbf{v}_0), q_1(\mathbf{v}_1), \dots, q_m(\mathbf{v}_m) = q(\mathbf{v})$ such that $|\mathbf{v}_0|_{\max}, |\mathbf{v}_1|_{\max}, \dots, |\mathbf{v}_m|_{\max} \leq (|V|_{\max} + |\mathbf{u}|_{\max} + |\mathbf{v}|_{\max})^{\mathcal{O}(1)}$.*

In the following lemma, that is proved in the appendix, given a 2-VASS V , the initial configuration $p(\mathbf{u})$, and target configuration $q(\mathbf{v})$, we write B in place of $(|V|_{\max} + |\mathbf{u}|_{\max} + |\mathbf{v}|_{\max})^{\mathcal{O}(1)}$ from Theorem 2 and we fix $\mathbf{x} = (4B|Q|^2|V|_{\max}^2, 0)$.

Lemma 5. *Consider a 2-VASS (with both counters in binary) $V = (Q, T)$ and let $p(\mathbf{u}) \xrightarrow{*} q(\mathbf{v})$, then there exists a narrow linear form path π' such that $p(\mathbf{u} + \mathbf{x}) \xrightarrow{\pi'} q(\mathbf{v}')$ for some $\mathbf{v}' \geq \mathbf{v}$.*

6 Proof of Theorem 1

Before proving Theorem 1, we employ the fact that for a general 2-VASS, not necessarily with one unary counter, the exponents of cycles in linear forms can be pseudo-polynomially bounded.

Lemma 6 (Corollary from Lemma 18 in [5]). *Let π be path in a 2-VASS with a linear form $\pi = \tau_0 \gamma_1^{f_1} \tau_1 \dots \gamma_k^{f_k} \tau_k$ such that $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$. Then there exist a path $\pi' = \tau_0 \gamma_1^{e_1} \tau_1 \dots \tau_{k-1} \gamma_k^{e_k} \tau_k$ such that $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{v}')$ where $\mathbf{v}' \geq \mathbf{v}$ and $\text{bitsize}(e_1), \dots, \text{bitsize}(e_k)$ are all bounded by a polynomial in $|V| + \text{bitsize}(\mathbf{u}) + \text{bitsize}(\mathbf{v})$.*

Proof of Theorem 1. Let $p(\mathbf{u}) \xrightarrow{\pi} q(\mathbf{v})$ for some path π . If there is a narrow linear form ℓ for π then by Lemma 6 we obtain $\pi' = \tau_0 \gamma_1^{e_1} \tau_1 \dots \tau_{k-1} \gamma_k^{e_k} \tau_k$ such that $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{v}')$ where $\mathbf{v}' \geq \mathbf{v}$ and $\text{bitsize}(e_1), \dots, \text{bitsize}(e_k)$ are all bounded

above by a polynomial in $|V| + \text{bitsize}(\mathbf{u}) + \text{bitsize}(\mathbf{v})$. Since ℓ is a narrow linear form, we know that $k \leq P(|Q|)$ so $\sum_{i=1}^k \text{len}(\gamma_i) \leq k|Q| \leq |Q|P(|Q|)$ and we also know that $\sum_{i=0}^k \text{len}(\tau_i) \leq |Q|(P(|Q|) + 1)$. Together, this implies the linear form path π' is of polynomial size.

It remains to consider the case when there is no narrow linear form ℓ for π . By Lemma 4 (via Lemma 3) there exists a path π' such that $p(\mathbf{u}) \xrightarrow{\pi'} q(\mathbf{v}')$ and $\mathbf{v}' \geq \mathbf{v}$. Moreover, there is a monotone cycle decomposition (ρ, σ, τ) of π' such that $\text{eff}(\sigma) > \mathbf{0}$ and there are narrow linear forms for both ρ and σ .

Assume that $(\text{eff}(\sigma))_1 > 0$. This is without loss of generality because if $(\text{eff}(\sigma))_1 = 0$ then one can flip the coordinates in V , \mathbf{u} and \mathbf{v} (for the remainder of the proof it will not matter that one counter is unary). Let $p'(\mathbf{m})$ be the configuration such that $p(\mathbf{u}) \xrightarrow{\rho} p'(\mathbf{m}) \xrightarrow{\sigma\tau} q(\mathbf{v}')$. Observe that since $\text{eff}(\sigma) > \mathbf{0}$ for every $i \in \mathbb{N}$ the path $\rho\sigma^i$ induces the run $p(\mathbf{u}) \xrightarrow{\rho\sigma^i} p'(\mathbf{m} + i \cdot \text{eff}(\sigma))$. Consider $x = (\mathbf{x})_1 = 4B|Q|^2|V|_{\max}^2$ (for Lemma 5), clearly x is large enough so that $p(\mathbf{u}) \xrightarrow{\rho\sigma^x} p'(\mathbf{m}')$ and $\mathbf{m}' \geq \mathbf{m} + \mathbf{x}$. By Lemma 5 there exists a narrow linear form for a path τ' such that $p'(\mathbf{m}') \xrightarrow{\tau'} q(\mathbf{v}'')$ and $\mathbf{v}'' \geq \mathbf{v}'$.

We conclude by considering the compressed linear form path $\rho\sigma^x\tau'$ such that $p(\mathbf{u}) \xrightarrow{\rho\sigma^x\tau'} q(\mathbf{v}'')$ and $\mathbf{v}'' \geq \mathbf{v}' \geq \mathbf{v}$. Since ρ , σ , and τ' have narrow linear forms, we can also bound the exponents using Lemma 6 as in the beginning of this proof. Finally, $\text{bitsize}(x)$ is polynomial in $|V| + \text{bitsize}(\mathbf{u}) + \text{bitsize}(\mathbf{v})$ much like the exponents of the cycles in the linear forms. Therefore, the size of the compressed linear form $\rho\sigma^x\tau'$ is polynomial in $|V| + \text{bitsize}(\mathbf{u}) + \text{bitsize}(\mathbf{v})$. \square

7 Conclusion and Future Work

In this paper we proved that coverability in 2-VASS with one unary counter is in NP, a drop in complexity from PSPACE for general 2-VASS. We achieve this by using our new techniques. Most notably, we polynomially bounded the number of short cycles that need to be used (Section 4). Then, we attempt to find a polynomial linear form path by replacing short cycles and reshuffling the path (Section 5).

A natural extension is to consider whether coverability in 3-VASS with one binary counter and two unary counters is also in NP. More generally, there is the problem of determining the complexity of coverability in k -VASS with one binary counter and $k - 1$ unary counters. The technique for polynomially bounding the number of short cycles that need be used can easily be generalised to these higher dimension VASS with only one binary counter. However, it is not clear how to modify and use our reshuffling technique. Another open problem is whether reachability in 2-VASS with one unary counter is also in NP. Note that completeness would immediately follow from the fact that reachability in binary encoded 1-VASS is NP-hard [22].

References

1. Shaull Almagor, Udi Boker, Piotr Hofman, and Patrick Totzke. Parametrized Universality Problems for One-Counter Nets. 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 47:1–47:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:[10.4230/LIPICs.CONCUR.2020.47](https://doi.org/10.4230/LIPICs.CONCUR.2020.47).
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](https://doi.org/10.4230/LIPICs.CONCUR.2020.38).
3. Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.
4. 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](https://doi.org/10.1145/3464794).
5. Michael Blondin, Alain Finkel, Stefan Göller, Christoph Haase, and Pierre McKenzie. Reachability in Two-Dimensional Vector Addition Systems with States Is PSPACE-Complete. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 32–43. IEEE Computer Society, 2015. doi:[10.1109/LICS.2015.14](https://doi.org/10.1109/LICS.2015.14).
6. 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](https://doi.org/10.1007/978-3-030-72013-1_1).
7. Mikołaj Bojańczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data words. *ACM Trans. Comput. Log.*, 12(4):27:1–27:26, 2011. doi:[10.1145/1970398.1970403](https://doi.org/10.1145/1970398.1970403).
8. Hubert Comon and Yan Jurski. Multiple counters automata, safety analysis and presburger arithmetic. In Alan J. Hu and Moshe Y. Vardi, editors, *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer, 1998. doi:[10.1007/BFb0028751](https://doi.org/10.1007/BFb0028751).
9. Wojciech Czerwiński, Sławomir 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](https://doi.org/10.4230/LIPICs.CONCUR.2020.48).
10. Wojciech Czerwiński, Sławomir Lasota, Ranko Lazic, Jérôme Leroux, and Filip Mazowiecki. The Reachability Problem for Petri Nets Is Not Elementary. *J. ACM*, 68(1):7:1–7:28, 2021. doi:[10.1145/3422822](https://doi.org/10.1145/3422822).

11. Wojciech Czerwiński, Sławomir Lasota, Christof Löding, and Radosław Piórkowski. New Pumping Technique for 2-Dimensional VASS. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPICs*, pages 62:1–62:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.MFCS.2019.62.
12. Wojciech Czerwiński and Łukasz 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.
13. Wojciech Czerwiński and Łukasz 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.
14. 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.
15. Matthias Englert, Piotr Hofman, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Juliusz Straszyński. A lower bound for the coverability problem in acyclic pushdown VAS. *Inf. Process. Lett.*, 167:106079, 2021. doi:10.1016/j.ipl.2020.106079.
16. Matthias Englert, Ranko Lazić, and Patrick Totzke. Reachability in Two-Dimensional Unary Vector Addition Systems with States is NL-complete. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 477–484. ACM, 2016. doi:10.1145/2933575.2933577.
17. Javier Esparza, Ruslán Ledesma-Garza, Rupak Majumdar, Philipp J. Meyer, and Filip Niksic. An SMT-Based Approach to Coverability Analysis. In Armin Biere and Roderick Bloem, editors, *Computer Aided Verification - 26th International Conference, CAV 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 18-22, 2014. Proceedings*, volume 8559 of *Lecture Notes in Computer Science*, pages 603–619. Springer, 2014. doi:10.1007/978-3-319-08867-9_40.
18. 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.
19. Alain Finkel, Jérôme Leroux, and Grégoire Sutre. Reachability for Two-Counter Machines with One Test and One Reset. In Sumit Ganguly and Paritosh K. Pandya, editors, *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018*,

- Ahmedabad, India*, volume 122 of *LIPICs*, pages 31:1–31:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.FSTTCS.2018.31.
20. Alain Finkel and Arnaud Sangnier. Mixing coverability and reachability to analyze VASS with one zero-test. In Jan van Leeuwen, Anca Muscholl, David Peleg, Jaroslav Pokorný, and Bernhard Rumpe, editors, *SOFSEM 2010: Theory and Practice of Computer Science, 36th Conference on Current Trends in Theory and Practice of Computer Science, Spindleruv Mlýn, Czech Republic, January 23-29, 2010. Proceedings*, volume 5901 of *Lecture Notes in Computer Science*, pages 394–406. Springer, 2010. doi:10.1007/978-3-642-11266-9_33.
 21. Pierre Ganty and Rupak Majumdar. Algorithmic verification of asynchronous programs. *ACM Trans. Program. Lang. Syst.*, 34(1):6:1–6:48, 2012. doi:10.1145/2160910.2160915.
 22. 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.
 23. John E. Hopcroft and Jean-Jacques Pansiot. On the Reachability Problem for 5-Dimensional Vector Addition Systems. *Theor. Comput. Sci.*, 8:135–159, 1979. doi:10.1016/0304-3975(79)90041-0.
 24. S. Rao Kosaraju. Decidability of Reachability in Vector Addition Systems (Preliminary Version). In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 267–281. ACM, 1982. doi:10.1145/800070.802201.
 25. Jean-Luc Lambert. A Structure to Decide Reachability in Petri Nets. *Theor. Comput. Sci.*, 99(1):79–104, 1992. doi:10.1016/0304-3975(92)90173-D.
 26. 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.
 27. 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.
 28. Jérôme Leroux and Grégoire Sutre. On Flatness for 2-Dimensional Vector Addition Systems with States. In Philippa Gardner and Nobuko Yoshida, editors, *CONCUR 2004 - Concurrency Theory, 15th International Conference, London, UK, August 31 - September 3, 2004, Proceedings*, volume 3170 of *Lecture Notes in Computer Science*, pages 402–416. Springer, 2004. doi:10.1007/978-3-540-28644-8_26.
 29. 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, September 1-4, 2020, Vienna, Austria (Virtual Conference)*, volume 171 of *LIPICs*, pages 37:1–37:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.CONCUR.2020.37.
 30. 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.

31. Richard Lipton. The Reachability Problem Requires Exponential Space. *Department of Computer Science. Yale University*, 62, 1976.
32. Ernst W. Mayr. An Algorithm for the General Petri Net Reachability Problem. *SIAM J. Comput.*, 13(3):441–460, 1984. doi:10.1137/0213029.
33. Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., 1967.
34. C. Petri. Kommunikation mit Automaten, Ph. D. dissertation. *University of Bonn*, 1962.
35. Charles Rackoff. The Covering and Boundedness Problems for Vector Addition Systems. *Theor. Comput. Sci.*, 6:223–231, 1978. doi:10.1016/0304-3975(78)90036-1.
36. Klaus Reinhardt. Reachability in Petri Nets with Inhibitor Arcs. *Electron. Notes Theor. Comput. Sci.*, 223:239–264, 2008. doi:10.1016/j.entcs.2008.12.042.
37. Sylvain Schmitz. The Complexity of Reachability in Vector Addition Systems. *ACM SIGLOG News*, 3(1):4–21, 2016. URL: <https://dl.acm.org/citation.cfm?id=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.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

